Wen Yu
Haibo He
Nian Zhang (Eds.)

LNCS 5551

Advances in Neural Networks – ISNN 2009

6th International Symposium on Neural Networks, ISNN 2009
Wuhan, China, May 2009
Proceedings, Part I

Springer

Lecture Notes in Computer Science 5551

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Wen Yu Haibo He Nian Zhang (Eds.)

Advances in Neural Networks – ISNN 2009

6th International Symposium
on Neural Networks, ISNN 2009
Wuhan, China, May 26-29, 2009
Proceedings, Part I

Springer

Volume Editors

Wen Yu
Centro de Investigación y de Estudios Avanzados
del Instituto Politécnico Nacional (CINVESTAV-IPN)
Departamento de Control Automático
A.P. 14-740, Av. IPN 2508, 07360 México D.F., Mexico
E-mail: yuw@ctrl.cinvestav.mx

Haibo He
Stevens Institute of Technology
Department of Electrical and Computer Engineering
Castle Point on Hudson, Hoboken, NJ 07030, USA
E-mail: hhe@stevens.edu

Nian Zhang
South Dakota School of Mines & Technology
Department of Electrical and Computer Engineering
501 East St. Joseph Street, Rapid City, SD 57701, USA
E-mail: nian.zhang@sdsmt.edu

Library of Congress Control Number: Applied for

CR Subject Classification (1998): F.1, F.2, D.1, G.2, I.2, C.2, I.4-5, J.1-4

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

ISSN 0302-9743
ISBN-10 3-642-01506-9 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-01506-9 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12666126 06/3180 5 4 3 2 1 0

Preface

This book and its companion volumes, LNCS vols. 5551, 5552 and 5553, constitute the proceedings of the 6th International Symposium on Neural Networks (ISNN 2009), held during May 26–29, 2009 in Wuhan, China. Over the past few years, ISNN has matured into a well-established premier international symposium on neural networks and related fields, with a successful sequence of ISNN symposia held in Dalian (2004), Chongqing (2005), Chengdu (2006), Nanjing (2007), and Beijing (2008). Following the tradition of the ISNN series, ISNN 2009 provided a high-level international forum for scientists, engineers, and educators to present state-of-the-art research in neural networks and related fields, and also to discuss with international colleagues on the major opportunities and challenges for future neural network research.

Over the past decades, the neural network community has witnessed tremendous efforts and developments in all aspects of neural network research, including theoretical foundations, architectures and network organizations, modeling and simulation, empirical study, as well as a wide range of applications across different domains. The recent developments of science and technology, including neuroscience, computer science, cognitive science, nano-technologies and engineering design, among others, have provided significant new understandings and technological solutions to move the neural network research toward the development of complex, large-scale, and networked brain-like intelligent systems. This long-term goal can only be achieved with the continuous efforts of the community to seriously investigate different issues of the neural networks and related fields. To this end, ISNN 2009 provided a great platform for the community to share their latest research results, discuss critical future research directions, stimulate innovative research ideas, as well as facilitate international multidisciplinary collaborations.

ISNN 2009 received 1235 submissions from about 2459 authors in 29 countries and regions (Australia, Brazil, Canada, China, Democratic People's Republic of Korea, Finland, Germany, Hong Kong, Hungary, India, Islamic Republic of Iran, Japan, Jordan, Macao, Malaysia, Mexico, Norway, Qatar, Republic of Korea, Singapore, Spain, Taiwan, Thailand, Tunisia, UK, USA, Venezuela, Vietnam, and Yemen) across six continents (Asia, Europe, North America, South America, Africa, and Oceania). Based on the rigorous peer reviews by the Program Committee members and the reviewers, 409 high-quality papers were selected for publication in the LNCS proceedings, with an acceptance rate of 33.1%. These papers cover major topics of the theoretical research, empirical study, and applications of neural networks. In addition to the contributed papers, the ISNN 2009 technical program included five plenary speeches by Anthony Kuh (University of Hawaii at Manoa, USA), Jose C. Principe (University of Florida, USA), Leszek Rutkowski (Technical University of Czestochowa, Poland), Fei-Yue Wang (Institute of Automation, Chinese Academy of Sciences, China) and Cheng Wu (Tsinghua University, China). Furthermore, ISNN 2009 also featured five special sessions focusing on emerging topics in neural network research.

As organizers of ISNN 2009, we would like to express our sincere thanks to the Huazhong University of Science and Technology, The Chinese University of Hong Kong, and the National Natural Science Foundation of China for their sponsorship, to the IEEE Wuhan Section, the IEEE Computational Intelligence Society, the International Neural Network Society, the Asia Pacific Neural Network Assembly, and the European Neural Network Society for their technical co-sponsorship, and to the Systems Engineering Society of Hubei Province and the IEEE Hong Kong Joint Chapter on Robotics and Automation and Control Systems for their logistic support.

We would also like to sincerely thank the General Chair and General Co-chairs for their overall organization of the symposium, members of the Advisory Committee and Steering Committee for their guidance in every aspect of the entire conference, and the members of the Organizing Committee, Special Sessions Committee, Publication Committee, Publicity Committee, Finance Committee, Registration Committee, and Local Arrangements Committee for all their great effort and time in organizing such an event. We would also like to take this opportunity to express our deepest gratitude to the members of the International Program Committee and all reviewers for their professional review of the papers; their expertise guaranteed the high qualify of technical program of ISNN 2009!

Furthermore, we would also like to thank Springer for publishing the proceedings in the prestigious series of *Lecture Notes in Computer Science*. Moreover, we would like to express our heartfelt appreciations to the plenary and panel speakers for their vision and discussion of the latest research developments in the field as well as critical future research directions, opportunities, and challenges.

Finally, we would like to thank all the speakers, authors, and participants for their great contribution and support that made ISNN 2009 a great success.

May 2009

Wen Yu
Haibo He
Nian Zhang

Organization

General Chair

Shuzi Yang, China

General Co-chairs

Youlun Xiong, China
Yongchuan Zhang, China

Advisory Committee Chairs

Shoujue Wang, China
Paul J. Werbos, USA

Advisory Committee Members

Shun-ichi Amari, Japan
Zheng Bao, China
Tianyou Chai, China
Guanrong Chen, China
Shijie Cheng, China
Ruwei Dai, China
Jay Farrell, USA
Chunbo Feng, China
Russell Eberhart, USA
David Fogel, USA
Walter J. Freeman, USA
Kunihiko Fukushima, Japan
Marco Gilli, Italy
Aike Guo, China
Xingui He, China
Zhenya He, China
Petros Loannou, USA
Janusz Kacprzyk, Poland
Nikola Kasabov, New Zealand
Okyay Kaynak, Turkey
Frank L. Lewis, USA
Deyi Li, China
Yanda Li, China
Chin-Teng Lin, Taiwan

Robert J. Marks II, USA
Erkki Oja, Finland
Nikhil R. Pal, India
Marios M. Polycarpou, USA
Leszek Rutkowsk, Poland
Jennie Si, USA
Youxian Sun, China
Joos Vandewalle, Belgium
DeLiang Wang, USA
Fei-Yue Wang, USA
Donald C. Wunsch II, USA
Lei Xu, China
Xin Yao, UK
Gary G. Yen, USA
Bo Zhang, China
Nanning Zheng, China
Jacek M. Zurada, USA

Steering Committee Chairs

Jun Wang, Hong Kong
Derong Liu, China

Steering Committee Members

Jinde Cao, China
Shumin Fei, China
Chengan Guo, China
Min Han, China
Zeng-Guang Hou, China
Xiaofeng Liao, China
Bao-Liang Lu, China
Fuchun Sun, China
Zhang Yi, China
Fuliang Yin, China
Hujun Yin, UK
Huaguang Zhang, China
Jianwei Zhang, Germany

Organizing Committee Chairs

Hongwei Wang, China
Jianzhong Zhou, China
Yi Shen, China

Program Committee Chairs

Wen Yu, Mexico
Haibo He, USA
Nian Zhang, USA

Special Sessions Chairs

Sanqing Hu, USA
Youshen Xia, China
Yunong Zhang, China

Publications Chairs

Xiaolin Hu, China
Minghui Jiang, China
Qingshan Liu, China

Publicity Chairs

Tingwen Huang, Qatar
Paul S. Pang, New Zealand
Changyin Sun, China

Finance Chair

Xiaoping Wang, China

Registration Chairs

Charlie C. L. Wang, China
Zhenyuan Liu, China
Weifeng Zhu, China

Local Arrangements Chairs

Zhigang Zeng, China
Chao Qi, China
Liu Hong, China

Program Committee Members

José Alfredo, Brazil
Sabri Arik, Turkey
Xindi Cai, USA
Yu Cao, USA
Matthew Casey, UK
Emre Celebi, USA
Jonathan Chan, Thailand
Sheng Chen, UK
Yangquan Chen, USA
Ji-Xiang Du, China
Hai-Bin Duan, China
Andries Engelbrecht, South Africa
Péter érdi, USA
Jufeng Feng, China
Chaojin Fu, China
Wai Keung Fung, Canada
Erol Gelenbe, UK
Xinping Guan, China
Chengan Guo, China
Ping Guo, China
Qing-Long Han, Australia
Hanlin He, China
Daniel Ho, Hong Kong
Zhongsheng Hou, China
Huosheng Hu, UK
Jinglu Hu, Japan
Junhao Hu, China
Marc van Hulle, Belgium
Danchi Jiang, Australia
Haijun Jiang, China
Shunshoku Kanae, Japan
Rhee Man Kil, Republic of Korea
Sungshin Kim, Korea
Arto Klami, Finland
Rakhesh Singh Kshetrimayum, India
Hon Keung Kwan, Canada
Chuandong Li, China
Kang Li, UK
Li Li, China
Michael Li, Australia
Ping Li, Hong Kong
Shutao Li, China
Xiaoli Li, UK
Xiaoou Li, Mexico
Yangmin Li, Macao
Hualou Liang, USA
Jinling Liang, China
Wudai Liao, China
Alan Liew, Australia
Ju Liu, China
Li Liu, USA
Meiqin Liu, China
Wenxin Liu, USA
Yan Liu, USA
Jianquan Lu, Hong Kong
Jinhu Lu, China
Wenlian Lu, China
Jinwen Ma, China
Ikuko Nishkawa, Japan
Seiichi Ozawa, Japan
Jaakko Peltonen, Finland
Juan Reyes, Mexico
Jose de Jesus Rubio, Mexico
Eng. Sattar B. Sadkhan, Iraq
Gerald Schaefer, UK
Michael Small, Hong Kong
Qiankun Song, China
Humberto Sossa, Mexico
Bingyu Sun, China
Norikazu Takahashi, Japan
Manchun Tan, China
Ying Tan, China
Christos Tjortjis, UK
Michel Verleysen, Belgium
Bing Wang, UK
Dan Wang, China
Dianhui Wang, Australia
Meiqing Wang, China
Rubin Wang, China
Xin Wang, China
Zhongsheng Wang, China
Jinyu Wen, China
Wei Wu, China
Degui Xiao, China
Rui Xu, USA
Yingjie Yang, UK
Kun Yuan, China
Xiaoqin Zeng, China
Jie Zhang, UK
Liqing Zhang, China

Publications Committee Members

Guici Chen
Huangqiong Chen
Shengle Fang
Lizhu Feng
Junhao Hu
Feng Jiang
Bin Li
Yanling Li
Mingzhao Li
Lei Liu
Xiaoyang Liu
Cheng Wang
Xiaohong Wang
Zhikun Wang
Shiping Wen
Ailong Wu
Yongbo Xia
Li Xiao
Weina Yang
Zhanying Yang
Tianfeng Ye
Hongyan Yin
Lingfa Zeng
Yongchang Zhang
Yongqing Zhao
Song Zhu

Technical Committee Members

Helena Aidos
Antti Ajanki,
Tholkappia AraSu
Hyeon Bae
Tao Ban
Li Bin
Binghuang Cai
Lingru Cai
Xindi Cai
Qiao Cai
Chao Cao
Hua Cao
Jinde Cao
Kai Cao
Wenbiao Cao
Yuan Cao
George Cavalcanti
Lei Chang
Mingchun Chang
Zhai Chao
Cheng Chen
Gang Chen
Guici Chen
Ke Chen
Jiao Chen
Lei Chen
Ming Chen
Rongzhang Chen
Shan Chen
Sheng Chen
Siyue Chen
TianYu Chen
Wei Chen
Xi Chen
Xiaochi Chen
Xiaofeng Chen
XinYu Chen
Xiong Chen
Xuedong Chen
Yongjie Chen
Zongzheng Chen
Hao Cheng
Jian Cheng
Long Cheng
Zunshui Cheng
Rong Chu
Bianca di Angeli C.S. Costa
Jose Alfredo Ferreira Costa
Dadian Dai
Jianming Dai
Jayanta Kumar Debnath
Spiros Denaxas
Chengnuo Deng
Gang Deng
Jianfeng Deng
Kangfa Deng

Zhipo Deng
Xiaohua Ding
Xiuzhen Ding
Zhiqiang Dong
Jinran Du
Hongwu Duan
Lijuan Duan
Xiaopeng Duan
Yasunori Endo
Andries Engelbrecht
Tolga Ensari
Zhengping Fan
Fang Fang
Haitao Fang
Yuanda Fang
June Feng
Lizhu Feng
Yunqing Feng
Avgoustinos Filippoupolitis
Liang Fu
Ruhai Fu
Fang Gao
Lei Gao
Ruiling Gao
Daoyuan Gong
Xiangguo Gong
Fanji Gu
Haibo Gu
Xingsheng Gu
Lihe Guan
Jun Guo
Songtao Guo
Xu Guo
Fengqing Han
Pei Han
Qi Han
Weiwei Han
Yishan Han
Yunpeng Han
Hanlin He
Jinghui He
Rui He
Shan He
Tonejun He
Tongjun He
Wangli He
Huosheng Hu
Li Hong
Liu Hong
Ruibing Hou
Cheng Hu
Jin Hu
Junhao Hu
Hao Hu
Hui Hu
Ruibin Hu
Sanqing Hu
Xiaolin Hu
Xiaoyan Hu
Chi Huang
Darong Huang
Diqiu Huang
Dongliang Huang
Gan Huang
Huayong Huang
Jian Huang
Li Huang
Qifeng Huang
Tingwen Huang
Zhangcan Huang
Zhenkun Huang
Zhilin Huang
Rey-Chue Hwang
Sae Hwang
Hui Ji
Tianyao Ji
Han Jia
Danchi Jiang
Shaobo Jiang
Wei Jiang
Wang Jiao
Xianfa Jiao
Yiannis Kanellopoulos
Wenjing Kang
Anthony Karageorgos
Masanori KaWakita
Haibin Ke
Seong-Joo Kim
Peng Kong
Zhanghui Kuang
Lingcong Le
Jong Min Lee
Liu Lei
Siyu Leng

Bing Li
Changping Li
Chuandong Li
Hui Li
Jian Li
Jianmin Li
Jianxiang Li
Kelin Li
Kezan Li
Lei Li
Li Li
Liping Li
Lulu Li
Ming Li
Na Li
Ping Li
Qi Li
Song Li
Weiqun Li
Wenlong Li
Wentian Li
Shaokang Li
Shiying Li
Tian Li
Wei Li
Wu Li
Xiang Li
Xiaoli Li
Xiaoou Li
Xin Li
Xinghai Li
Xiumin Li
Yanlin Li
Yanling Li
Yong Li
Yongfei Li
Yongmin Li
Yuechao Li
Zhan Li
Zhe Li
Jinling Liang
Wudai Liao
Wei Lin
Zhihao Lin
Yunqing Ling
Alex Liu
Bo Liu
Da Liu
Dehua Li
Dayuan Liu
Dongbing Liu
Desheng Liu
F. C. Liu
Huaping Liu
Jia Liu
Kangqi Liu
Li Liu
Ming Liu
Qian Liu
Qingshan Liu
Shangjin Liu
Shenquan Liu
Shi Liu
Weiqi Liu
Xiaoyang Liu
Xiuquan Liu
Xiwei Liu
XinRong Liu
Yan Liu
Yang Liu
Yawei Liu
Yingju Liu
Yuxi Liu
Zhenyuan Liu
Zijian Liu
Yimin Long
Georgios Loukas
Jinhu Lu
Jianquan Lu
Wen Lu
Wenlian Lu
Wenqian Lu
Tongting Lu
Qiuming Luo
Xucheng Luo
Chaohua Ma
Jie Ma
Liefeng Ma
Long Ma
Yang Ma
Zhiwei Ma
Xiaoou Mao
Xuehui Mei
Xiangpei Meng

Xiangyu Meng
Zhaohui Meng
Guo Min
Rui Min
Yuanneng Mou
Junichi Murata
Puyan Nie
Xiushan Nie
Gulay Oke
Ming Ouyang
Yao Ouyang
Seiichi Ozawa
Neyir Ozcan
Joni Pajarinen
Hongwei Pan
Linqiang Pan
Yunpeng Pan
Tianqi Pang
Kyungseo Park
Xiaohan Peng
Zaiyun Peng
Gao Pingan
Liquan Qiu
Jianlong Qiu
Tapani Raiko
Congjun Rao
Fengli Ren
Jose L. Rosseilo
Gongqin Ruan
Quan Rui
Sattar B. Sadkhan
Renato Jose Sassi Sassi
Sibel Senan
Sijia Shao
Bo Shen
Enhua Shen
Huayu Shen
Meili Shen
Zifei Shen
Dianyang Shi
Jinrui Shi
Lisha Shi
Noritaka Shigei
Atsushi Shimada
Jiaqi Song
Wen Song
Yexin Song
Zhen Song
Zhu Song
Gustavo Fontoura de Souza
Kuo-Ho Su
Ruiqi Su
Cheng Sun
Dian Sun
Junfeng Sun
Lisha Sun
Weipeng Sun
Yonghui Sun
Zhaowan Sun
Zhendong Sun
Manchun Tan
Xuehong Tan
Yanxing Tan
Zhiguo Tan
Bing Tang
Hao Tang
Yili Tang
Gang Tian
Jing Tian
Yuguang Tian
Stelios Timotheou
Shozo Tokinaga
Jun Tong
Joaquin Torres Sospedra
Hiroshi Wakuya
Jin Wan
B.H. Wang
Cheng Wang
Fan Wang
Fen Wang
Gang Wang
Gaoxia Wang
Guanjun Wang
Han Wang
Heding Wang
Hongcui Wang
Huayong Wang
Hui Wang
Huiwei Wang
Jiahai Wang
Jian Wang
Jin Wang
Juzhi Wang
Kai Wang

Lan Wang
Lili Wang
Lu Wang
Qilin Wang
Qingyun Wang
Suqin Wang
Tian Wang
Tianxiong Wang
Tonghua Wang
Wei Wang
Wenjie Wang
Xiao Wang
Xiaoping Wang
Xiong Wang
Xudong Wang
Yang Wang
Yanwei Wang
Yao Wang
Yiping Wang
Yiyu Wang
Yue Wang
Zhanshan Wang
Zhengxia Wang
Zhibo Wang
Zhongsheng Wang
Zhihui Wang
Zidong Wang
Zhuo Wang
Guoliang Wei
Li Wei
Na Wei
Shuang Wei
Wenbiao Wei
Yongchang Wei
Xiaohua Wen
Xuexin Wen
Junmei Weng
Yixiang Wu
You Wu
Huaiqin Wu
Zhihai Wu
Bin Xia
Weiguo Xia
Yonghui Xia
Youshen Xia
Zhigu Xia
Zhiguo Xia
Xun Xiang
Chengcheng Xiao
Donghua Xiao
Jiangwen Xiao
Yongkang Xiao
Yonkang Xiao
Yong Xie
Xiaofei Xie
Peng Xin
Chen Xiong
Jinghui Xiong
Wenjun Xiong
Anbang Xu
Chen Xu
Hesong Xu
Jianbing Xu
Jin Xu
Lou Xu
Man Xu
Xiufen Yu
Yan Xu
Yang Xu
Yuanlan Xu
Zhaodong Xu
Shujing Yan
Dong Yang
Fan Yang
Gaobo Yang
Lei Yang
Sihai Yang
Tianqi Yang
Xiaolin Yang
Xing Yang
Xue Yang
Yang Yang
Yongqing Yang
Yiwen Yang
Hongshan Yao
John Yao
Xianfeng Ye
Chenfu Yi
Aihua Yin
Lewen Yin
Qian Yin
Yu Ying

Xu Yong
Yuan You
Shuai You
Chenglong Yu
Liang Yu
Lin Yu
Liqiang Yu
Qing Yu
Yingzhong Yu
Zheyi Yu
Jinhui Yuan
Peijiang Yuan
Eylem Yucel
Si Yue
Jianfang Zeng
Lingjun Zeng
Ming Zeng
Yi Zeng
Zeyu Zhang
Zhigang Zeng
Cheng Zhang
Da Zhang
Hanling Zhang
Haopeng Zhang
Kaifeng Zhang
Jiacai Zhang
Jiajia Zhang
Jiangjun Zhang
Jifan Zhang
Jinjian Zhang
Liming Zhang
Long Zhang
Qi Zhang
Rui Zhang
Wei Zhang
Xiaochun Zhang
Xiong Zhang
Xudong Zhang
Xuguang Zhang
Yang Zhang
Yangzhou Zhang
Yinxue Zhang
Yunong Zhang
Zhaoxiong Zhang
YuanYuan
Bin Zhao
Jin Zhao
Le Zhao
Leina Zhao
Qibin Zhao
Xiaquan Zhao
Zhenjiang Zhao
Yue Zhen
Changwei Zheng
Huan Zheng
Lina Zheng
Meijun Zheng
Quanchao Zheng
Shitao Zheng
Ying Zheng
Xun Zheng
Lingfei Zhi
Ming Zhong
Benhai Zhou
Jianxiong Zhou
Jiao Zhou
Jin Zhou
Jinnong Zhou
Junming Zhou
Lin Zhou
Rong Zhou
Song Zhou
Xiang Zhou
Xiuling Zhou
Yiduo Zhou
Yinlei Zhou
Yuan Zhou
Zhenqiao Zhou
Ze Zhou
Zhouliu Zhou
Haibo Zhu
Ji Zhu
Jiajun Zhu
Tanyuan Zhu
Zhenqian Zhu
Song Zhu
Xunlin Zhu
Zhiqiang Zuo

Table of Contents – Part I

Theoretical Analysis

Stability

Time-Delay Neural Networks

Machine Learning

Neural Modeling

Decision Making Systems

Table of Contents – Part II

Fuzzy Systems and Fuzzy Neural Networks

Support Vector Machines and Kernel Methods

Genetic Algorithms

Clustering and Classification

Pattern Recognition

Intelligent Control

Table of Contents – Part III

Optimization

Robotics

Image Processing

Signal Processing

Biomedical Applications

Fault Diagnosis

Telecommunication, Sensor Network and Transportation Systems

Applications I

Applications II

Optimal Inversion of Open Boundary Conditions Using BPNN Data-Driven Model Combined with Tidal Model

Mingchang Li[1], Guangyu Zhang[1], Bin Zhou[1], Shuxiu Liang[2], and Zhaochen Sun[2]

[1] Laboratory of Environmental Protection in Water Transport Engineering, Tianjin Research Institute of Water Transport Engineering Tianjin 300456, China
[2] State Key Laboratory of Coastal and Offshore Engineering, DUT, Dalian 116024, China
Lmcsq1997@163.com

Abstract. One of major difficulties with numerical tidal models is accurate inversion of open boundary conditions. A data-driven model based on artificial neural network is developed to retrieve open boundary values. All training data are calculated by numerical tidal model, so the tidal physics are not disturbed. The basic idea is to find out the relationship between open boundary values and the values of interior tidal stations. Case testes are carried out with a real ocean bay named Liaodong Bay, part of the Bohai Sea, China. Four major tidal constituents, M_2, S_2, $O_1$and K_1, are considered in coupled inversion method. Case studies show that the coupled inversion for open boundary conditions can make a more satisfactory inversion for a practical problem.

Keywords: Data-driven model, Open boundary conditions, Optimal inversion, Tidal current.

1 Introduction

Ocean is the source of material and energy for production and consumption. Being exploited widely, water quality and ecological system of ocean has been affected tremendously, especially of estuary and nears-shore region along big cities. So the sea area use demonstration, marine environmental impact assessment and total amount control of pollutant is required for marine environmental protection. However, accurate numerical modeling by tidal and ecosystem model is the fundamental step of these works.

Numerical tidal models have been applied widely to study tidal hydrodynamics for recent decades (Abbott, 1997; Davies, 1997). In order to make the numerical results come close to real ocean conditions, model parameters, initial and boundary conditions have to be estimated. The process is called model calibration. One of the most significant difficulties is the inversion of open boundary conditions, especially for costal waters. Usually, the initial boundary values can be estimated based on the data of tidal stations nearby, or provided by a large-scale model (Egbert and Bennet et al, 1994). Try and error (Gerristen, 1995) is a widely used technique during model calibration. However, model calibration process might cost much time. Data assimilation methods have been employed for model calibration with the abundance of satellite

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 1–10, 2009.

data. Adjoint technique is the most widely used one among them(Bennett,1982; Hall,1982; Cacuci,1985; Panchang,1989; Larder,1993; Seiler,1993; Zhu,1997; Vogeler,1999; Zhang,1999; Han,2000,2001; Lu,2002; Heemink,2002; Zhang,2003; Ayoub,2001,2006; Ferron,2003; Gebbie,2004; Ma and Jing,2005). Optimal inversion for parameters, initial and boundary conditions can be obtained by this technique. However, both tidal model equations and adjoint equations need to be calculated. Adjoint equations are as complicated as tidal model equations, so much time is needed. For practical engineering, it is lack in tidal data. When adjoint technique is chosen, much uncertainty might exist in the calibrated model. The application of data-assimilation method is limited.

This paper aims to develop a more practical technique for optimal inversion of open boundary conditions. In the technique, only open boundary is adjusted, whereas initial values and model parameters are assumed to be correct. Optimal boundary conditions are estimated by data-driven model which is based on artificial neural network. POM (Blumberg and Mellor 1987) is employed to simulate the tidal hydrodynamics for interested area.

The structure of the paper is as follows. In section 2, the basic idea and theory of data-driven, Back-Propagation Neural Network (BPNN) and tidal numerical model are introduced briefly. The detailed steps about how to inverse open boundary are described in section 3. In section 4, the method in section 3 is verified with Liaodong bay. Four major tidal constituents, M_2, S_2, $O_1$and K_1, are considered. In section 5, conclusions are made.

2 Numerical Model

2.1 Data-Driven Model

The so-called data-driven models, is different from knowledge-driven models (physically-based modeling). These kinds of models are based on a limited knowledge of the modelling process and rely purely on the data describing input and output characteristics. They make abstractions and generalizations of the process, so play often a complementary role to physically-based models. Data-driven model can use results from artificial neural networks (ANN), expert systems, fuzzy logic concepts, rule-induction and machine learning systems (Solomatine, 2002). The fundamental expression is as follows:

$$(y_1,\cdots,y_i,\cdots y_m) = F(x_1,\cdots,x_i,\cdots x_n) \tag{1}$$

Where $x_1,\cdots,x_i,\cdots x_n$ and $(y_1,\cdots,y_i,\cdots y_m)$ are the input and output variables respectively ; F is the objective function which need to be dug by model. In present paper, the results of ANN are used for the fitting of F .

2.2 Back-Propagation Neural Network

The BPNN proposed by Rumelhart et al. (1986) is the most commonly used among the entire artificial neural network models. The BPN uses the gradient steepest descent method to determine the weight of connective neurons. The key point is the error back-propagation technique. In the learning process of the BPN, the

interconnection weights are adjusted from back layers to front layers to minimize the output error. The merit of the BPN is that it can approach any nonlinear continuous function after being trained (Hormik, 1991). The detailed information can be referred in Li and Liang et al (2007). There are only two differences in present study. One is the leaning rate and it is set to be 0.05. The other one is the introduction of appended momentum and its value is 0.5.

2.3 Tidal Physical Numerical Model

Tidal physical model is fundamental since it tries to explain the underlying physical processes. There are many tidal hydrodynamic models which can retrieve real ocean conditions with great precision. POM developed by Blumberg and Mellor (1987) is one of them. POM has been used widely not only for ocean circulation modeling but for costal waters studies. There are several modes can be tuned in the model and barotropic 2-D model is chosen in the paper.

3 Optimal Inversion Method

Tens or even hundreds integral computation has to be repeated in the process of trial and error for the inversion of open boundary conditions. The results, however, are only approximate resolution. In present paper, a new technique is developed which combines data-driven model with tidal physical model automatically. In the technique, tidal physical model repeats a series of designed computations. Then, a data set which contain the corresponding relationship between open boundary values [$x_1,\cdots,x_i,\cdots x_n$ in equation (1)] and the values of interior tidal /current stations [$(y_1,\cdots,y_i,\cdots y_m)$ in equation (1)] are stored. The task of data-driven model is to find out the relationship [F in equation (1)] between $(x_1,\cdots,x_i,\cdots x_n)$ and $(y_1,\cdots,y_i,\cdots y_m)$. After measurement data are transported into the model, optimal boundary values will be inversed. The detailed technique is as follows:

Step 1: Choice of control variables

There are two kinds of open boundary for barotropic tidal model, water level or flux. Water level is usually used since it is easier and more accurate to measure. Realistic water level can be decomposed into the sum of tidal constitutes. Each of them has two constants- amplitude and phase. Realistic water level might involve more than hundreds of tidal constituents. They act each other, especially in coastal shallow waters. If all of them are included, the computation cost is excessive and uncertainty increases (Friedrichs, 2007). So major tidal constituents have to be analyzed, which aim to select control variables.

Step 2: Cases computation by physical tidal model

In tidal numerical models, the governing equations have to discretized into computation domain. Along one open boundary line, values of variables are interpolated linearly by two or more control nodes. In present paper, initial guess values for all the control nodes are assumed and their corresponding ranges are set. If the number of control nodes is m, and n values are taken for one control variable, there are as many as $\prod_{i=m}\left[C_n^1\right]_i = n^m$

designed cases. All the designed cases are computed by tidal model one by one. The results of tidal elevation and current are stored and output for data-driven model.

Step 3: open boundary inversion by data-driven model

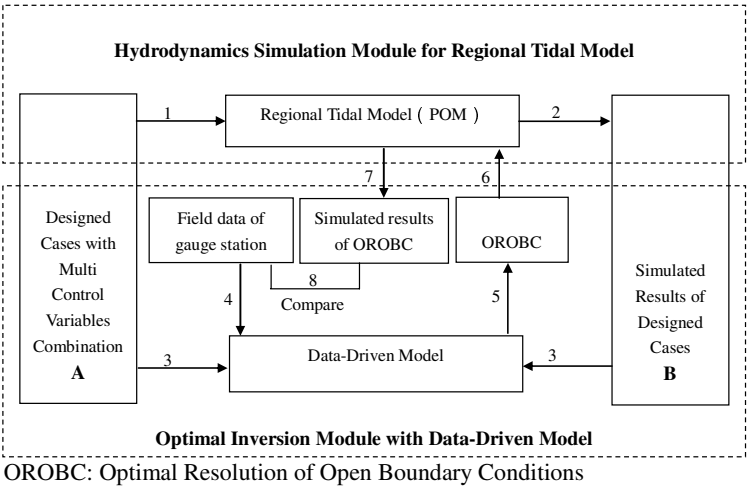
Harmonic constants for interior stations are analyzed using the results of designed cases.

Input the harmonic constants of interior stations and their corresponding boundary values into data-driven model. After training, the relationships of interior stations and open boundary are generalized.

Input the harmonic constants of interior stations analyzed from measurement data into the above relationship and the optimal solution are solved.

Step 4: verification of optimal solution

Input the optimal solution into the physical numerical model and repeat the computation. The relative error between measurement and results of numerical computation are calculated.



Note: OROBC: Optimal Resolution of Open Boundary Conditions

Fig. 1. Diagram of optimal inversion of Boundary Conditions Using Data-driven Model Combined with Tidal Model

In figure 1, the process of open boundary inversion is described. 1-8 is the sequence of it. In the whole process, there are two modules—hydrodynamic and optimal inversion. The computation of designed cases and the final verification are finished by hydrodynamic module. Optimal inversion module is responsible for the analysis of hydrodynamic results and generalization of relationship between open boundary and interior stations. A and B is the connection of the two modules.

4 Case Studies

Case testes are carried out with a real ocean bay named Liaodong Bay, north part of Bohai Sea, China. The total area is about 18,300 km2. It is very shallow and its averaged water depth is less than 20m. The sea bottom is very flat and its mean slope is

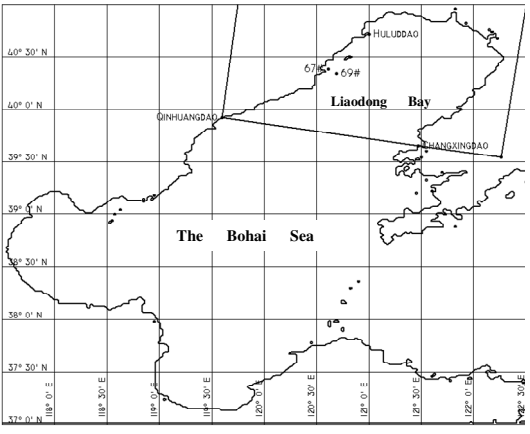


Fig. 2. Computation domain and measurement stations in the Liaodong bay

1/2000-1/2500. The tide hydrodynamic is controlled by the tidal waves of the Bohai Sea and Yellow Sea, so the tidal current is complicated. Figure 2 shows the location of computated domain and measurement points. There is only one open boundary line controlled by Qinghuangdao and Changxingdao. Measurement data of tidal elevation from Huludao and tidal current from 69# are used.

4.1 Choices of Control Variables

In the interested ocean bay, the four major tidal constituents can account for more than 90% of total tidal elevation. However, amplitude of shallow water constituents may be as large as one of them. Two cases are designed to test the effects of shallow water constituents. In case 1, only the four major tidal constituents, M_2, S_2, $O_1$and K_1, are considered. In case 2, F_4 and F_6 are also included.

In figure 3, the time series of tidal elevation for Huludao by case1 and case 2 are compared. The same comparisons are made for 69# in fig. 4. Correlation coefficient is 0.9999 and 0.9995 and the absolute error is only 0.02m and 0.02cm/s respectively. Therefore, the effects of F_4 and F_6 can be ignored and the M_2, S_2, $O_1$and K_1 are chosen as control variables. Each control variable includes two sub-variables—phase and amplitude.

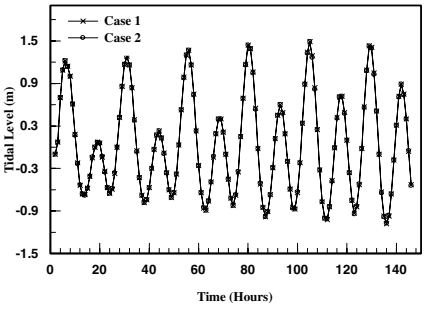


Fig. 3. Comparison of tidal elevation for Huludao

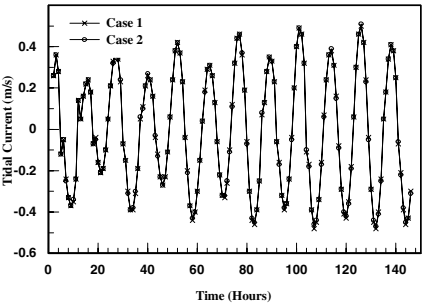


Fig. 4. Comparison of tidal current for 69#

Table 1. Values for amplitude (m) of control nodes along open boundary

Tidal constituents	Qinghuang dao	Changxing dao	Tidal constituents	Qinghuang dao	Changxing dao
M_2	0.08	0.25	S_2	0.06	0.1
	0.12	0.39		0.09	0.15
	0.16	0.52		0.12	0.2
	0.25	0.77		0.18	0.3
K_1	0.21	0.17	O_1	0.24	0.13
	0.32	0.26		0.36	0.19
	0.43	0.35		0.48	0.26
	0.64	0.53		0.72	0.4

Table 2. Values for phase (°)of control nodes along open boundary

Tidal constituents	Qinghuang dao	Changxing dao	Tidal constituents	Qinghuang dao	Changxing dao
M_2	-140	50	S_2	-60	100
	-155	65		-75	115
	-170	80		-89	130
K_1	120	30	O_1	10	55
	135	45		25	70
	150	60		40	85

An interesting phenomenon is founded by analyzing the result of designed cases, that is the tidal amplitude is not affected by the change of tidal phase in its assumed range. Therefore, the amplitude and phase for each control variables are inversed independently. When amplitude is inversed, its phase is set to be initial guess value and vice verse. When this rule is applied, less time is consumed.

In table 1 and table 2, the values of amplitude and phase of control variables are listed. For each control variable, four values are taken in its range for amplitude and three values for phase. The total designed cases are 16 and 9 for amplitude and phase.

4.2 Coupled Inversion and Its Results

The so called separated inversion is to inverse tidal constituents one by one. For example, when M_2 is to be inversed, only M_2 is control variables and tidal elevation along open boundary is calculated based on harmonic constant of M_2 only.

Different from Separated inversion, the four major constituents are input as control variables simultaneously in coupled inversion. When to inverse amplitude, all the values for three control variables are sequenced from big to small. When M_2 is set to be 0.12, S_2 is set to be 0.09, O_1 is set to be 0.36 and K_1 is set to be 0.32 for Qinghuangdao respectively. Therefore, when M_2 is set to be 0.25, S_2, O_1, K_1 should be 0.1, 0.13 and 0.17 respectively for Changxingdao. The same rule works on other combination. According to this rule, there are 16 designed cases for amplitude inversion.

Table 3 shows the results of the coupled inversion for M_2, S_2, O_1 and K_1 by optimal inversion method.

Table 3. Results of coupled estimation

Open boundary nodes	Items	Tidal constituents			
		M_2	S_2	K_1	O_1
Qinghuangdao	Amplitude (m)	0.1507	0.1000	0.4184	0.4441
	Phase (º)	-155.20	-74.65	135.45	27.17
Changxingdao	Amplitude (m)	0.4247	0.1948	0.3449	0.2428
	Phase (º)	65.54	112.67	45.16	70.19

4.3 Verification of Optimal Solution

Table 3 is optimal open boundary values inversed by coupled inversion. Input them to tidal model and repeat computation to verify the accuracy of the optimal solution. The results are shown in table 4. The maximal error for amplitude is -0.04m from M2 and for phase is 4.76° from O1, both in 69#. The error indicates coupled inversion can inverse realistic open boundary values in better accuracy.

Table 4. Comparison of amplitudes (m) between measurement and simulation for interior stations

Interior stations	Items	Tidal constituents			
		M_2	S_2	K_1	O_1
Huludao	Measurement	0.9197	0.2579	0.3470	0.2707
	Simulation	0.8921	0.2923	0.3244	0.2523
	Error	0.0276	-0.0344	0.0226	0.0184
69#	Measurement	0.510	0.150	0.090	0.068
	Simulation	0.550	0.162	0.081	0.061
	Error	-0.040	-0.012	0.009	0.007

Table 5. Comparison of phases (°) between measurement and simulation for interior stations

Interior stations	Items	Tidal constituents			
		M_2	S_2	K_1	O_1
Huludao	Measurement	116.88	170.88	78.54	58.34
	Simulation	118.40	172.40	78.80	59.80
	Error	-1.52	-1.52	-0.26	-1.46
69#	Measurement	6.4	52.1	-84.11	-51.44
	Simulation	7.8	54.9	-83.5	-56.2
	Error	-1.4	-2.8	-0.61	4.76

5 Conclusion

In this paper, a new method is developed to inverse open boundary values. In the method, data–driven model and physical tidal model are coupled automatically. Physical numerical model repeat a number of computation for designed cases, the results of tidal elevation are stored and output for data-driven model. Data-driven model generalizes the relationship between open boundary and interior stations. After measurement data is imported, optimal solution are obtained.

In realistic case study, measurement data of tidal elevation are used to constrain the model better in a limited data. The results show the coupled inversion is suitable for realistic open boundary inversion.

Compared with adjoint method, present method has two superiorities. One is simplicity. There is no need to deduce and solve complicated adjoint equations. Data-driven model based on ANN is easy to be developed. The other one is its flexibility. In adjoint method, different adjoint equations are needed to be deduced according to different numerical models. If the basic equations are changed, e.g., different closure models, the adjoint equations need to be altered accordingly. In present method, the data-driven model can be kept unchanged when different physical tidal models are used. Adjoint technique has to repeat computation for both tide equations and adjoint equations. Even in good initial guess, much time is consumed compared with present method.

References

1. Abbott, M.B.: Range of Tidal Flow Modeling. Journal of Hydraulic Engineering 123, 255–277 (1997)
2. Davies, A.M., Jones, J.E., Xing, J.: Review of Recent Developments in Tidal Hydrodynamic Modeling. Journal of Hydraulic Engineering 4, 278–292 (1997)
3. Egbert, G.D., Bennett, A.F., Foreman, M.G.G.: TOPEX/POSEIDON Tides Estimated Using a Global Inverse Model. Journal of Geophysical Research 99, 24821–24852 (1994)
4. Gerritsen, H., Vries, H., Philippart, M.: The Dutch Continental Shelf Model. Quantitative Skill Assessment for Coastal Ocean Models. Coastal Estuarine Studies 47, 425–468 (1995)

5. Bennett, A.F., Mcintosh, P.C.: Open Ocean Modeling as an Inverse Problem: Tidal Theory. Journal of Physical Oceanography 12, 1004–1018 (1982)
6. Hall, M.C.G., Cacuci, D.G., Schlesinger, M.E.: Sensitivity Analysis of a Radiative-Convective Model by Adjoint Method. Journal of Atmospheric Science 39, 2038–2050 (1982)
7. Cacuci, D.G.: The Forward and Adjoint Methods of Sensitivity Analysis. Uncertainty Analysis 282 (1988)
8. Panchang, V.G., O'Brien, J.J.: On the Determination of Hydraulic Model Parameter Using tThe Adjoint Stste Formulation. Modeling Marine System 1, 5–18 (1989)
9. Larder: Optimal Control of Open Boundary Conditions for a Numerical Tidal Model. Computer Methods in Applied Mechanics and Engineering 102, 367–387 (1993)
10. Seiler: Estimation of Open Boundary Conditions with the Adjoint Method. Journal of Geophysical Research 98, 22855–22870 (1993)
11. Zhu, J., Zeng, Q., et al.: Estimation of Coastal Ocean Model Open Boundary Conditions from Nearshore Tide Gauge Station Using Adjoint Method. Science In China (Series D) 27(5), 462–468 (1997)
12. Vogeler, A., Schroeter, J.: Fitting a Regional Ocean Model with Adjustable Open Boundaries to TOPEX/POSEIDON data. Journal of Geophysical Research 104, 20789–20799 (1999)
13. Zhang, K.Q., Marotzke, J.: The Importance of Open-Boundary Estimation for an Indian Ocean GCM-data Synthesis. Journal of Marine Research 57, 305–334 (1999)
14. Han, G., He, B., Ma, J., et al.: Optimizing Open Boundary Conditions of Nonlinear Tidal Model Using Adjoint Method: The Establishment of Adjoint Model and Twin-Experiment. ACTA Oceanologica SINICA 22(6), 27–33 (2000)
15. Han, G., Fang, G., Ma, J., et al.: Optimizing Open Boundary Conditions of Nonlinear Tidal Model Using Adjoint Method. Assimilation Experiment for Tide in the Huanghai Sea and the East China Sea. ACTA Oceanologica SINICA 23(2), 25–31 (2001)
16. Lv, X., Fang, G.: Inversion of the Tides on the Open Boundary of the BOHAI SEA by Adjoint Method. Oceanologia ET Limnologia Sinica 33, 113–120 (2002)
17. Heemink, A.W., Mouthaan, E.E.A., Roest, M.R.T., Vollebregt, E.A.H., Robaczewska, K.B., Verlaam, M.: Inverse 3D Shallow Water Flow Modeling of the Continental Shelf. Continental Shelf Research 22, 465–484 (2002)
18. Zhang, A.I., Wei, E., Parker, B.: Optimal Estimation of Tidal Open Boundary Conditions Using Predicted Tides and Adjoint Data Assimilation Technique. Continental Shelf Research 23, 1055–1070 (2003)
19. Ayoub, N.: Estimation of Boundary Values in a North Atlantic Circulation Model Using an Adjoint Method. Ocean Modelling 12, 319–347 (2006)
20. Ayoub, N., Stammer, D., Wunsch, C.: Estimating the North Atlantic Circulation with Nesting and Open-boundary Conditions Using an Adjoint Model. ECCO Report, No.10, Scripps Institution of Oceanography (2001), `http://www.ecco-group.org`
21. Foreman, M.G.G., Sutherland, G., Cummins, P.F.: M_2 Tidal Dissipation Around Vancouver Island: an Inverse Approach. Continental Shelf Research 24, 2167–2185 (2004)
22. Gebbie, G.A.: Subduction in an Eddy-resolving State Estimate of the Northeast Atlantic Ocean. MIT/WHOI PhD Thesis 198 (2004)
23. Ma, Z., Jing, A.: Data Assimilation Method Applied in Marine Science–Its Significance, System Configuration and Development Situation. Coastal Engineering 24, 83–99 (2005)
24. Blumberg, A.F., Mellor, G.L.: A Description of a Three-dimensional Coastal Ocean Circulation Model. In: Heaps, N.S. (ed.) Coastal and Estuarine Sciences 4: Three-Dimensional Coastal Ocean Models, Amer. Geophys. Union, pp. 1–16 (1987)

25. Solomatine, D.P.: Data-Driven Modelling: Paradigm, Methods, Experiences. In: Proc. 5th Int. Conference on Hydroinformatics, Cardiff, UK, pp. 757–763 (2002)
26. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning Representations by Back Propagating Errors. Nature 323, 533–536 (1986)
27. Hornik, K.: Approximation Capabilities of Multilayer Feedforward Networks. Neural Networks 2, 359–366 (1991)
28. Li, M.C., Liang, S.: Application of Artificial Neural Networks to Tide Forecasting. Journal of Dalian University of Technology 47, 101–105 (2007)
29. Friedrichs, M.A.M.: A Data Assimilative Marine Ecosystem Model of the Central Equatorial Pacific: Numerical Twin Experiments. Journal of Marine Research 59, 859–894 (2001)

Time-Varying Matrix Square Roots Solving via Zhang Neural Network and Gradient Neural Network: Modeling, Verification and Comparison

Yunong Zhang[1], Yiwen Yang[2], and Ning Tan[2]

[1]School of Information Science and Technology
[2]School of Software
Sun Yat-Sen University, Guangzhou 510275, China
ynzhang@ieee.org
http://www.ee.sysu.edu.cn/teacher/detail.asp?sn=129

Abstract. A special kind of recurrent neural networks (RNN) with implicit dynamics has recently been proposed by Zhang *et al*, which could be generalized to solve online various time-varying problems. In comparison with conventional gradient neural networks (GNN), such RNN (or termed specifically as Zhang neural networks, ZNN) models are elegantly designed by defining matrix-valued indefinite error functions. In this paper, we generalize and investigate the ZNN and GNN models for online solution of time-varying matrix square roots. In addition, software modeling techniques are investigated to model and simulate both neural-network systems. Computer-modeling results verify that superior convergence and efficacy could be achieved by such ZNN models in this time-varying problem solving, as compared to the GNN models.

Keywords: Time-varying matrix square roots, Recurrent neural networks, Zhang neural networks, Gradient neural networks.

1 Introduction

To our knowledge, the conventional gradient or gradient-based neural networks (GNN) could be viewed as a useful and important method for time-invariant problems solving [1,2]. However, many time-varying problems intrinsically exist in mathematics, science and engineering areas [2,3,4,5,6], such as the time-varying matrix square roots (TVMSR) problem depicted as below:

$$X^2(t) - A(t) = 0, \quad t \in [0, +\infty), \tag{1}$$

where, being a smoothly time-varying positive-definite matrix, $A(t) \in R^{n\times n}$ and its time derivative $\dot{A}(t)$ are both assumed known numerically (or at least measurable accurately). In addition, let $X(t)$ denote the time-varying square root of $A(t)$, which is to be solved for. In this paper, we investigate the TVMSR problem to find matrix $X(t) \in R^{n\times n}$ satisfying the time-varying nonlinear matrix equation $X^2(t) = A(t)$ for any $t \geqslant 0$. For such a time-varying equation, GNN models

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 11–20, 2009.

and methods may not work well, since they could only approximately track the theoretical solution $A^{1/2}(t)$ with relatively large residual errors [4,5,6]. In contrast, for such time-varying problems, a special kind of RNN models has recently been proposed by Zhang *et al* [2,3,4,5,6] (formally, since March 2001) for their real-time solution. In this paper, to solve the TVMSR problem, a Zhang neural network (ZNN) model is generalized and designed by using a matrix-valued indefinite error-function, instead of using scalar-valued norm-based lower-bounded energy functions which are usually associated with GNN models and methods.

The remainder of this paper is organized as follows. In Section 2, we present and compare the ZNN and GNN models/methods for online solution of time-varying matrix square roots. In Section 3, simple but effective software-modeling techniques are investigated for such RNN models. Illustrative verification results are presented in Section 4. Finally, we concludes this paper with Section 5.

2 Neural-Network Solvers

In the ensuing subsections, the ZNN and GNN models for solving the time-varying matrix square roots problem are developed comparatively.

2.1 ZNN Model

Firstly, to solve time-vary matrix square root $A^{1/2}(t)$ by Zhang *et al*'s neural-dynamic method [2,3,4,5,6], we can define the following matrix-valued error function:

$$E(t) = X^2(t) - A(t) \in R^{n\times n},$$

where, if the error function $E(t)$ equals zero, $X(t)$ achieves the time-varying theoretical solution $A^{1/2}(t)$ of the time-varying matrix equation depicted in (1).

Secondly, in order to make every entry $e_{ij}(t) \in R$ $(i, j = 1, 2, \cdots, n)$ of $E(t) \in R^{n\times n}$ converge to zero, a general form of the time derivative of $E(t)$, denoted by $\dot{E}(t)$, can be chosen as (i.e., the ZNN design formula [2,3,4,5,6]):

$$\frac{dE(t)}{dt} = -\Gamma\mathcal{F}\big(E(t)\big), \tag{2}$$

where design parameter $\Gamma \in R^{n\times n}$ is a positive-definite matrix used to scale the convergence rate of the neural network, and for simplicity, we can use $\gamma > 0 \in R$ in place of Γ. In addition, the activation-function array $\mathcal{F}(\cdot) : R^{n\times n} \to R^{n\times n}$ is a matrix-valued entry-to-entry mapping, in which each scalar-valued processing-unit $f(\cdot)$ could be a monotonically-increasing odd activation function. In this paper, two types of $f(\cdot)$ are investigated as examples for the RNN construction:

1)linear activation function $f(e_{ij}) = e_{ij}$; and,

2)power-sigmoid activation function

$$f(e_{ij}) = \begin{cases} e_{ij}^p, & \text{if } |e_{ij}| \geqslant 1 \\ \frac{1+\exp(-\xi)}{1-\exp(-\xi)} \cdot \frac{1-\exp(-\xi e_{ij})}{1+\exp(-\xi e_{ij})}, & \text{otherwise} \end{cases} \tag{3}$$

with suitable design parameters $\xi \geqslant 2$ and $p \geqslant 3$.

Thirdly, expanding ZNN design formula (2) leads to the following implicit dynamic equation of ZNN model for online matrix square roots finding [in other words, it solves the nonlinear time-varying equation (1)]:

$$X(t)\dot{X}(t) + \dot{X}(t)X(t) = -\gamma\mathcal{F}\big(X^2(t) - A(t)\big) + \dot{A}(t), \tag{4}$$

where $X(t)$, starting from an initial condition $X(0) \in R^{n\times n}$, is the activation state matrix corresponding to theoretical time-varying matrix square root $X^*(t) := A^{1/2}(t)$ of $A(t)$.

In order to build up the MATLAB Simulink verification model [7] of the above ZNN dynamic equation (4), we may need to transform it into the following explicit-dynamic equation by simply adding $\dot{X}(t)$ on both sides (i.e., via our so-called derivative-feedback or velocity-feedback technique [5,6]):

$$\dot{X}(t) = \dot{X}(t)\big(I - X(t)\big) - X(t)\dot{X}(t) - \gamma\mathcal{F}\big(X^2(t) - A(t)\big) + \dot{A}(t). \tag{5}$$

Moreover, for ZNN (4) solving for the time-varying matrix square root of $A(t)$, the following preliminaries [8,9,10] and proposition [4,5,6,9] can be given.

Square-root existence condition. *If smoothly time-varying matrix $A(t) \in R^{n\times n}$ is positive-definite (in general sense [10]) at any time instant $t \in [0, +\infty)$, then there exists a time-varying matrix square root $X(t) \in R^{n\times n}$ for matrix $A(t)$.*

Proposition. *Consider a smoothly time-varying matrix $A(t) \in R^{n\times n}$ in nonlinear equation (1), which satisfies the square-root existence condition. If a monotonically-increasing odd activation-function-array $\mathcal{F}(\cdot)$ is used, then*

- state-matrix $X(t) \in R^{n\times n}$ of ZNN (4), starting from randomly-generated positive-definite diagonal initial-state-matrix $X(0)$, could converge to theoretical positive-definite time-varying matrix square root $X^*(t)$ of $A(t)$; and,
- state-matrix $X(t) \in R^{n\times n}$ of ZNN (4), starting from randomly-generated negative-definite diagonal initial-state-matrix $X(0) \in R^{n\times n}$, could converge to theoretical negative-definite time-varying matrix square root $X^*(t)$ of $A(t)$.

In addition, if a linear-activation-function array $\mathcal{F}(\cdot)$ is used, exponential convergence with rate γ could be achieved for ZNN (4). As compared to the linear-array situation, superior (and/or much superior) convergence can be achieved for ZNN (4) by using an array $\mathcal{F}(\cdot)$ made of power-sigmoid activation functions (3).

2.2 GNN Model and Comparisons

For the purpose of comparison, we can design a GNN model for constant matrix square roots solving and then apply it to the solution of the time-varying problem (by assuming a so-called short-time immobility/invariableness). In view of [1], the following linear GNN model can be obtained for handling the static form of equation (1), which is designed based on the scalar-valued norm-based lower-bounded energy function $\|X^2(t) - A\|_F^2/2$:

$$\dot{X}(t) = -\gamma X^T(t)\big(X^2(t) - A\big) - \gamma\big(X^2(t) - A\big)X^T(t). \tag{6}$$

In addition, by using the nonlinear-activation technique [1,2,4,6,11], we could have the following generalized nonlinear GNN model of (6):

$$\dot{X}(t) = -\gamma X^T(t)\mathcal{F}\big(X^2(t) - A\big) - \gamma\mathcal{F}\big(X^2(t) - A\big)X^T(t).$$

It is also worth mentioning that, just like almost all numerical algorithms and neural-dynamic computational schemes, this conventional GNN method is designed intrinsically for static problems solving [e.g., the square roots finding with constant matrix $A \in R^{n\times n}$ in (1)] and could only have exact convergence results for the situation of static problems solving.

Moreover, before ending this section, we would like to present the following important remarks about the comparison between ZNN (4) and GNN (6).

1) ZNN model (4) is designed based on the elimination of every entry of the matrix-valued indefinite error function $E(t) = X^2(t) - A(t)$ (which could theoretically be positive, negative, bounded, or even unbounded). In contrast, GNN model (6) is designed based on the elimination of the scalar-valued norm-based energy function $\|X^2(t) - A\|_F^2/2$ which could only be positive or at least lower-bounded.

2) ZNN model (4) is depicted in an implicit dynamics, i.e., $\dot{X}(t)X(t) + X(t)\dot{X}(t) = \cdots$, which might coincide well with systems in nature and in practice (e.g., in analogue electronic circuits and mechanical systems [3] owing to Kirchhoff's and Newton's laws, respectively). In contrast, GNN model (6) is depicted in an explicit dynamics, i.e., $\dot{X}(t) = \cdots$, which is usually associated with conventional Hopfield-type and/or gradient-based artificial-neural-network models. Note that explicit dynamics can be viewed as a special case of implicit dynamics (i.e., with the mass matrix being an identity matrix), and that implicit dynamics can be transformed to explicit dynamics readily [e.g., via (5)].

3) ZNN model (4) could systematically and methodologically exploit the time-derivative information of problem-matrix $\dot{A}(t)$ during its real-time solving process. ZNN model (4) could thus exponentially and superiorly converge to the exact time-varying theoretical solution of (1). In contrast, GNN model (6) has not exploited such important information, thus less effective on solving the time-varying problem. More specifically, when applied to the time-varying problem solving, GNN model (6) could only generate approximate solutions to (1) with much larger steady-state computational errors.

4) In essence, by making good use of the time-derivative information [e.g., $\dot{A}(t)$ in (4)], the ZNN model and method actually belong to a prediction approach, which could be more effective on the system convergence to a "moving" theoretical solution. On the other hand, GNN model (6) and its design method belong to the conventional tracking approach, which follows from the change of the problem in a posterior (or to say, after-the-fact) passive manner, and thus theoretically can not catch the theoretical solution on the move.

5) Moreover, we could find that the connection from Newton iteration to ZNN models [11]. That is, Newton iteration for solving static problems appears to be a special case of the discrete-time ZNN models (by considering the use of linear activation functions as well as setting their step-size to be 1). In addition, the

derivation of ZNN models [such as (4)] might only need bachelors' mathematical knowledge if we look at the scalar case of ZNN design formula (2). In contrast, the derivation of GNN models [such as (6)] appears to require much deeper mathematical knowledge of postgraduates' or even PhD's level.

3 Software-Modeling Techniques

While Sections 1 and 2 present ZNN (4) and GNN (6) together with their theoretical analysis results, the following modeling techniques [5,6,7] are investigated in this section based on the MATLAB Simulink [7] environment.

3.1 Blocks Involved

There is a comprehensive block library in Simulink, which includes various kinds of blocks (e.g., sinks, sources, linear/nonlinear components, and connectors). The following blocks are used to construct the models of ZNN (4) and GNN (6).

1) The *Gain* block could be used to scale the neural-network convergence rate (e.g., as the scaling parameter γ).

2) The *Product* block provides two types of multiplication, either element-wise or matrix-wise. In this work, we use matrix-wise product by setting the option "Multiplication" to be "Matrix(*)".

3) The *Constant* block generates a constant scalar or constant matrix as specified by its parameter "Constant value". For example, a 3-dimensional identity matrix is generated by setting "Constant value" to be "[1 0 0; 0 1 0; 0 0 1]".

4) The *Subsystem* block is used to construct the sigmoid or power-sigmoid activation-function array, making the whole system simpler and more readable.

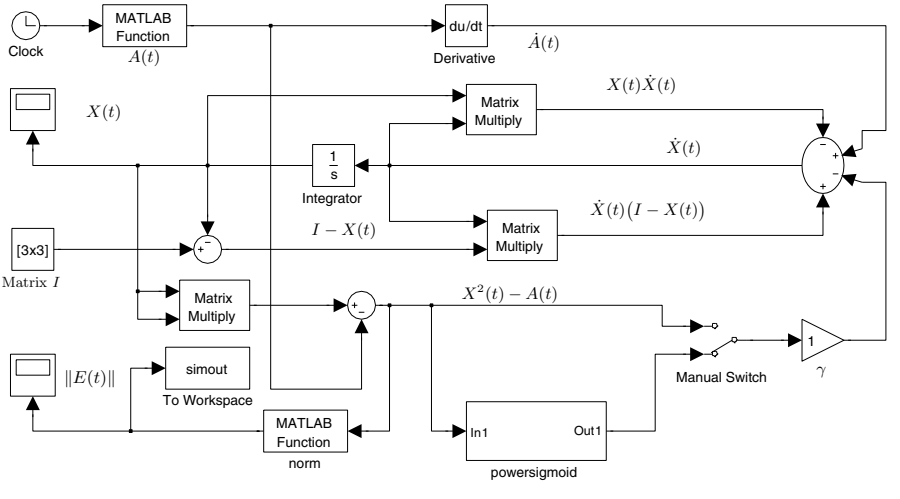


Fig. 1. Overall ZNN Simulink model which solves for time-varying matrix square roots

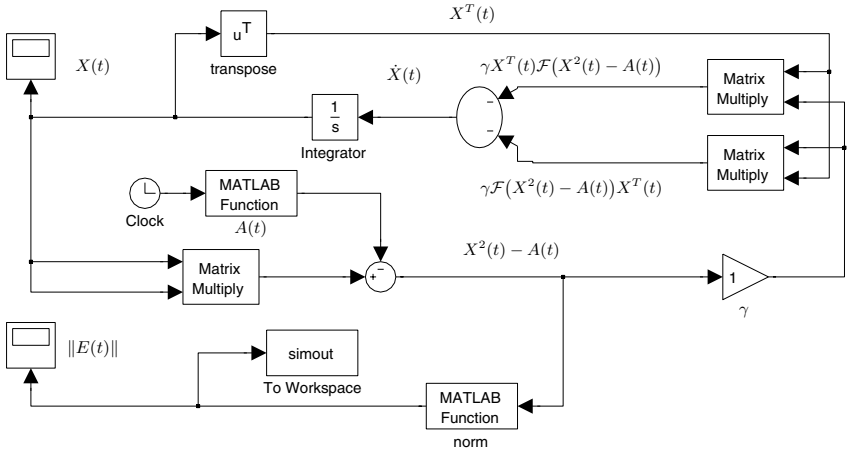


Fig. 2. Overall GNN Simulink model applied to time-varying square roots solving

5) The *MATLAB Fcn* block can be used to generate matrix $A(t)$ with the *Clock* block's output as its input or can be used to compute the matrix norm.

6) The *Math Function* block can perform various common mathematical operations, and, in our context, generates the transpose of a matrix and so on.

7) The *To Workspace* block, with its option "Save format" set to be "Array", is used to save the modeling results and data to the workspace.

8) The *Integrator* block makes continuous-time integration on the input signals over a period of time. In this work, we set its "Initial condition" as "$diag(2 * rand(3,1))$" in order to generate a diagonal positive-definite initial state matrix $X(0)$ with its diagonal elements randomly distributed in [0,2].

3.2 Generating Activation-Function Arrays

The modeling investigated in this paper includes two types of activation-function arrays mentioned in Section 2. For the linear-activation-function array, as the array output is the same as its input, we can simply use a connecting line or use the *purelin* block under the catalog of "Neural Network Blockset" to represent it. However, for the power-sigmoid-activation-function array, as it is composed of power and sigmoid activation functions, we can construct the two functions (or function-arrays) as the underpinning subsystems. On one hand, for the power function, we use the *Math Function* block by choosing "pow" in its function list and setting its parameter p as 3. On the other hand, the sigmoid function can be constructed by using some basic blocks with their detailed construction presented in our previous works [5,6]. Now, to combine the power and sigmoid subsystems, we can use a *Switch* block, where "u2>=Threshold" is chosen for option "Criteria for passing first input", and the value of "Threshold"is 1.

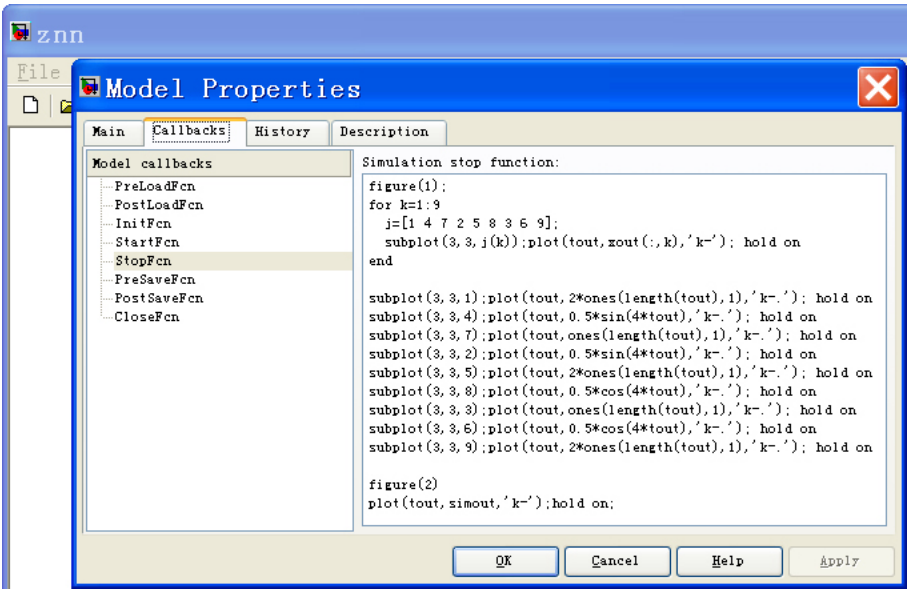


Fig. 3. Necessary "StopFcn" code of "Callbacks" in dialog box "Model Properties"

3.3 Configuration Parameters and Others Setting

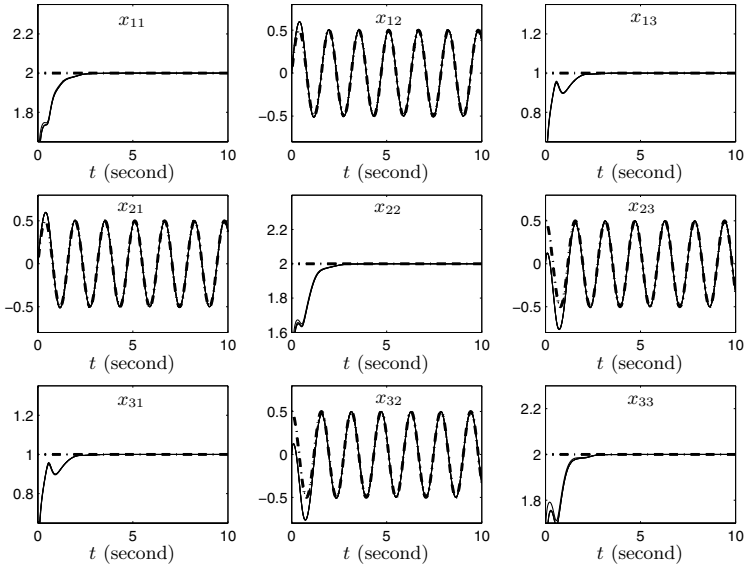
After the overall RNN models are built up and depicted in Figs. 1 and 2, we have to modify some of the default modeling-environment options. For example, firstly, let us open the dialog box entitled "Configuration Parameters". Secondly, let us set the modeling/simulation options as follows: 1) Solver: "ode23t" (which is much different from our previous work); 2) Max step size: "0.2"; 3) Min step size: "auto"; 4) Absolute tolerance:"auto"; 5) Relative tolerance:"1e-6"(i.e., 10^{-6}); and 6) Algebraic loop: "none". In addition, the check box in front of "States" as of the option "Data Import/Export" should be selected, which is for the purpose of better displaying the RNN-modeling results and is associated with the following "StopFcn" code (as of "Callbacks" in the dialog box entitled "Model Properties" which is started from the "File" pull-down menu).

4 Modeling and Verification Results

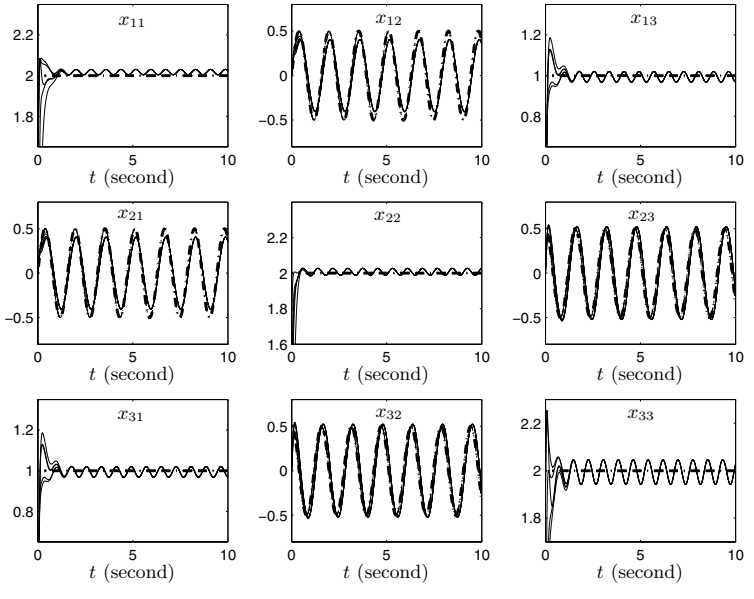
To verify the performance, efficacy and superiority of ZNN (4) in comparison with GNN (6), we consider the following time-varying matrix $A(t)$ with its time-varying theoretical square root $X^*(t)$ given below for comparison purposes:

$$A(t) = \begin{bmatrix} 5+0.25s^2 & 2s+0.5c & 4+0.25s\times c \\ 2s+0.5c & 4.25 & 2c+0.5s \\ 4+0.25s\times c & 2c+0.5s & 5+0.25c^2 \end{bmatrix}, \quad X^*(t) = \begin{bmatrix} 2 & 0.5s & 1 \\ 0.5s & 2 & 0.5c \\ 1 & 0.5c & 2 \end{bmatrix},$$

where s and c denote $\sin(4t)$ and $\cos(4t)$, respectively. The ZNN and GNN models depicted in Figs. 1 and 2 are now applied to solving the TVMSR problem (1).

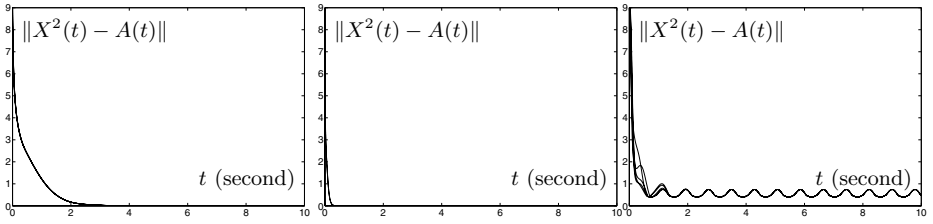


(a) Neural state matrix $X(t)$ of ZNN (4) using a power-sigmoid processing array



(b) Neural state matrix $X(t)$ of GNN (6) using a linear processing array

Fig. 4. Online solution of time-varying matrix square root $A^{1/2}(t)$ by ZNN (4) and GNN (6) with $\gamma = 1$, where the theoretical solution is denoted in dash-dotted curves



(a) ZNN (4) with $\gamma = 1$ (b) ZNN (4) with $\gamma = 10$ (c) GNN (6) with $\gamma = 1$

Fig. 5. Residual-error profile of RNN models during time-varying square roots finding

4.1 Convergence Verification

As illustrated in Fig. 4(a), starting from randomly-generated positive-definite diagonal initial-state $X(0) \in [0, 2]^{3\times 3}$, the neural state matrix $X(t)$ of ZNN (4) with design parameter $\gamma = 1$ could converge rapidly to the theoretically time-varying matrix square root $X^*(t)$ in an error-free manner. In contrast, in Fig. 4(b), GNN (6) does not track the theoretical solution $X^*(t)$ well, instead with quite large solution errors.

4.2 Residual-Error Verification

To monitor and show the solution process of ZNN model (4) and GNN model (6), residual error $\|X^2(t)-A(t)\|$ could also be exploited (which, for many engineering applications, might be the only and preferable choice). The left two sub-graphs of Fig. 5 show that, starting from randomly-generated positive-definite diagonal initial-state $X(0) \in [0, 2]^{3\times 3}$, the residual error of ZNN (4) converges to zero exactly, and that superior performance can be achieved by increasing the value of γ from 1 to 10. In comparison, as shown in Fig. 5(c), the residual error of GNN model (6) is relatively much larger (never vanishing to zero) and oscillating.

5 Conclusions

A special neural-network model (namely, ZNN) is proposed and investigated for the online time-varying matrix square roots finding. The important software-modeling techniques have been introduced and discussed for such neural-network construction. Computer-verification results have demonstrated further that superior convergence and efficacy could be achieved by such ZNN models for online time-varying matrix square roots finding, as compared to the well-known GNN approach and models.

Acknowledgments. This work is partially supported by the Program for New Century Excellent Talents in University (NCET-07-0887). Besides, before joining the Sun Yat-Sen University (SYSU) in 2006, Yunong Zhang had been with

National University of Ireland at Maynooth (Hamilton Institute), University of Strathclyde, National University of Singapore, Chinese University of Hong Kong, South China University of Technology, and Huazhong University of Science and Technology, since 1992. The learning and research experience had been related to or had inspired this research.

References

1. Zhang, Y.: Revisit the Analog Computer and Gradient-Based Neural System for Matrix Inversion. In: Proceedings of IEEE International Symposium on Intelligent Control, Cyprus, pp. 1411–1416 (2005)
2. Zhang, Y., Ge, S.S.: A General Recurrent Neural Network Model for Time-Varying Matrix Inversion. In: Proceedings of the 42nd IEEE Conference on Decision and Control, Hawaii, pp. 6169–6174 (2003)
3. Zhang, Y., Jiang, D., Wang, J.: A Recurrent Neural Network for Solving Sylvester Equation with Time-Varying Coefficients. IEEE Transactions on Neural Networks 13(5), 1053–1063 (2002)
4. Zhang, Y., Peng, H.: Zhang Neural Network for Linear Time-Varying Equation Solving and its Robotic Application. In: Proceedings of the 6th International Conference on Machine Learning and Cybernetics, Hong Kong, pp. 3543–3548 (2007)
5. Zhang, Y., Guo, X., Ma, W.: Modeling and Simulation of Zhang Neural Network for Online Linear Time-Varying Equations Solving Based on Matlab Simulink. In: Proceedings of the 7th International Conference on Machine Learning and Cybernetics, Kunming, pp. 805–810 (2008)
6. Ma, W., Zhang, Y., Wang, J.: Matlab Simulink Modeling and Simulation of Zhang Neural Networks for Online Time-Varying Sylvester Equation Solving. In: Proceedings of International Joint Conference on Neural Networks, Hong Kong, pp. 286–290 (2008)
7. The MathWorks Inc.: Simulink 7 Getting Started Guide, Natick, MA (2008)
8. Higham, N.J.: Stable Iterations for the Matrix Square Root. Numerical Algorithms 15(2), 227–242 (1997)
9. Hasan, M.A., Hasan, A.A., Rahman, S.: Fixed Point Iterations for Computing Square Roots and the Matrix Sign Function of Complex Matrices. In: Proceedings of the 39th IEEE Conference on Decision and Control, Sydney, pp. 4253–4258 (2000)
10. Zhang, Y.: On the LVI-Based Primal-Dual Neural Network for Solving Online Linear and Quadratic Programming Problems. In: Proceedings of American Control Conference, Portland, pp. 1351–1356 (2005)
11. Zhang, Y., Ma, W., Yi, C.: The Link between Newton Iteration for Matrix Inversion and Zhang Neural Network (ZNN). In: Proceedings of IEEE International Conference on Industrial Technology, Singapore, pp. 1–6 (2008)

Analysis of Time-Varying EEG Based on Wavelet Packet Entropy

Minfen Shen[1], Jialiang Chen[1], and Patch J. Beadle[2]

[1]College of Engineering, Shantou University, Guangdong 515063, China
[2]School of System Engineering, The University of Portsmouth, Portsmouth, U. K.
mfshen@stu.edu.cn

Abstract. To investigate the time-varying characteristics of the multi-channels electroencephalogram (EEG) signals with 4 rhythms, a useful approach is developed to obtain the EEG's rhythms based on the multi-resolution decomposition of wavelet transformation. Four specified rhythms can be decomposed from EEG signal in terms of wavelet packet analysis. A novel method for time-varying brain electrical activity mapping (BEAM) is also proposed using the time-varying rhythm for visualizing the dynamic EEG topography to help studying the changes of brain activities for one rhythm. Further more, in order to detect the changes of the nonlinear features of the EEG signal, wavelet packet entropy is proposed for this purpose. Both relative wavelet packet energy and wavelet packet entropy are regarded as the quantitative parameter for computing the complexity of the EEG rhythm. Some simulations and experiments using real EEG signals are carried out to show the effectiveness of the presented procedure for clinical use.

Keywords: Time-varying EEG; Wavelet decomposition; Wavelet packet entropy; Rhythm.

1 Introduction

EEG signals are the activity of ensembles of generators producing oscillation in several rhythms' activities with very complex mechanics [1]. In clinical applications, four basic rhythms from EEG have been associated with various states of brain under different brain functions and cognitions. EEG analysis has become an important way for investigating the state of the human brain function and reorganization process [2]. Four basic rhythms decomposed from the EEG signal are regarded as delta rhythm (1-4Hz), theta rhythm (4-8Hz), alpha rhythm (8-13Hz) and beta rhythm (13-30Hz) which are all defined with frequency band.

As we known, Fourier transform enable us to measure different rhythms and estimate the frequency components with each rhythm. However, the spectral decomposition with Fourier transform cannot detect the time-varying EEG's rhythms. In many clinical applications, we need to study the dynamic changes of the EEG and its rhythms. Short time Fourier transform (STFT) can provide us a useful time-varying frequency analysis method for non-stationary signals. But it has been noted that the

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 21–28, 2009.

short time Fourier transform depends critically on the choice of the window. With a narrow window, the frequency resolution will be poor; and if the window is too wide, the time localization will be less precise. The assumption of stationary of the EEG signal is another drawback of the method. Wavelet transform brings a solution to this problem. As a multi-resolution analysis method, wavelet transform can provide us a more accurate temporal localization and a good way for detecting a seizure [3]. In clinical, a more accurate frequency band of the EEG rhythm is required. To detect the specified rhythm, wavelet packet transform is used to reflect the changes of the desired band related with the rhythm. Wavelet packet transform is one of the useful method for the analyzing the non-stationary process [4]. It can generate spectral resolution fine enough to meet the problem requirement. Entropy derived from information theory can characterize the degree of randomness of time sequence and to quantify the difference between two probability distributions [5, 6]. Spectral entropy is a nature approach to quantify the degree of order of a signal [7, 8]. The spectral entropy is a measure of how concentrated or widespread the Fourier power spectrum of a signal is.

Another aim of this paper is to demonstrate the application of wavelet packet entropy measure to analysis of the segment of spontaneous EEG. This application may turn especially useful for studying EEG synchronization in conditions with certain limitation for long duration records of EEG signals.

2 Method of Wavelet Analysis

2.1 Wavelet Packet

One of drawbacks for the wavelet transform is that the frequency resolution is poor in the high frequency region. The wavelet transform may not provide a spectral resolution fine enough to meet the problem requirement in clinical application. To deal this problem, wavelet packet transform can be used as a generalization of a wavelet in that each octave frequency band of wavelet spectrum is further subdivided into finer frequency band by using the tow-scale relations repeatedly.

The wavelet packet function can be defined as

$$\psi_{j+1}^{2i-1}(t) = \sqrt{2}\sum_{k=-\infty}^{\infty} h(k)\psi_j^i(2t-k) \tag{1}$$

$$\psi_{j+1}^{2i}(t) = \sqrt{2}\sum_{k=-\infty}^{\infty} g(k)\psi_j^i(2t-k) \tag{2}$$

The $\psi(t)$ is called as the mother wavelet function. The $h(k)$, $g(k)$ are quartered mirror filters associated with the scaling function and the mother wavelet function.

The recursive relations between the j level and the j+1 level are:

$$f_{j+1}^{2i-1}(t) = \sum_{k=-\infty}^{\infty} h(k) f_j^i(2t-k) \tag{3}$$

$$f_{j+1}^{2i-1}(t) = \sum_{k=-\infty}^{\infty} g(k) f_j^i(2t-k) \tag{4}$$

The wavelet coefficients c_j^k can be obtain as

$$c_j^k = \int_{-\infty}^{\infty} f(t)\psi_j^i(t)dt \tag{5}$$

Thus, each wavelet packet subspace can be viewed as the output of a filter turned to a particular basis. A signal can be decomposed into a series of wavelet packet components as specified. We can select a set of wavelet packets for a given level of resolution for matching the desired rhythm. Different combination of wavelet packet should be chosen for the specific rhythm required.

2.2 Energy via Wavelet Packet Component

Since wavelet packet node energy is more robust in representing a signal than using the wavelet packet coefficients directly, we define the signal energy as[9]

$$E_f = \int_{-\infty}^{\infty} f^2(t)dt \tag{6}$$

Wavelet packet component energy $E_{f_j^i}$ can be defined as the energy stored in the component signal

$$E_{f_j^i} = \int_{-\infty}^{\infty} f_j^i(t)^2 dt \tag{7}$$

Total signal energy can be decomposed into a summation of wavelet packet component energy that corresponds to different frequency bands. Total energy can be obtained by

$$E_{tot} = E_f = \sum_{i=1}^{2j} E_{f_j^i} \tag{8}$$

A reasonable tree structure must be designed for analyzing the specific frequency band. For example, the signal would be covered by $f_1^2(t)$, $f_2^1(t)$ and $f_2^2(t)$ or by $f_1^2(t)$, $f_2^2(t)$, $f_3^1(t)$ and $f_3^2(t)$. Define that the energy of each sub-band as E_l. Then, the normalized value which represent the relative wavelet packet energy

$$P_l = \frac{E_l}{E_{tot}} \tag{9}$$

Equation (9) represents the energy distribution in each wavelet packet. It is clear that P_l is sensitive to the energy changes. It represents the energy relation among each wavelet packet.

2.3 Wavelet Packet Entropy

As we known, Shannon entropy gives a useful criterion for analyzing the complexity and the probability. It is a dynamic quantity distribution of the amount of disorder in system, which can view as a measure of uncertainty regarding the information content of a system. With the definition of entropy given by Shannon, the wavelet packet entropy can be defined as

$$S_{wp} = -\sum p_l \ln[p_l] \tag{10}$$

If a signal is very ordered (suppose a single frequency signal), all the energy will be in one frequency band. The energy of all other frequency band will be nearly zero. As a result, the relative wavelet packet energy will be 1,0,0….,which will lead to zero or very low value in the wavelet packet entropy. In another aspect, a very disordered signal (suppose a random signal) with energy distribution in every frequency band. The relative wavelet packet energy will be almost the same and lead to a maximum value in wavelet packet entropy [10].

Entropy is a description of uncertainty in the signal duration. It is not useful for analyzing non-stationary signal. To study temporal evolution, the signal is divided in to nonoverlapping temporal windows. Define the length of the window M, and the signal is divided into k segments. The total length of the signal is $N = KM$. Wavelet packet entropy is performed in each time window. The mean wavelet packet energy at frequency band l for the time window k is given by

$$E_l^{(k)} = \frac{1}{N_l} \sum_{t=(k-1)M+1}^{kM} f_j^i(t)^2 , \quad \text{with k=1,2…} \tag{11}$$

where N_l is the number of points at the frequency band l for the time window k . The total mean energy at the window is

$$E_{tot}^{(k)} = \sum_l E_l^{(k)} \tag{12}$$

$$P_l^{(k)} = E_l^{(k)} \Big/ E_{tot}^{(k)} \tag{13}$$

The time-dependent wavelet packet entropy will be given by

$$S_{wp}^{(k)} = -\sum p_l^{(k)} \ln[p_l^{(k)}] \tag{14}$$

Mean wavelet packet energy at frequency band l

$$\langle E_l \rangle = \frac{1}{K} \sum_{k=1}^{K} E_l^{(k)} \tag{15}$$

and the total mean of the wavelet packet energy average is

$$\langle E_{tot} \rangle = \sum_l \langle E_l \rangle \tag{16}$$

The mean energy probability distribution for the whole signal is

$$q_l = \frac{\langle E_l \rangle}{\langle E_{tot} \rangle} \tag{17}$$

The mean wavelet packet entropy is given as

$$\langle S_{wp} \rangle = -\sum q_l \ln[q_l] \tag{18}$$

3 Experiment

3.1 Data Recording

ERP activity was recorded from 14 scalp electrode sites using disc electrodes embedded in a nylon mesh cap placed on the subjects' head. The electrode locations, according to the international 10-20 system[11], consisted of Fp1, Fp2, F3, F4, C3, C4, P3, P4, O1, O2, F7, F8, T5, T6. The reference electrode was placed on the tip of the nose. Original EEG signals were recorded and stored in a personal computer. The sampling rate of the system was 100 Hz, and the EEG signals was amplified by SYNAMPS amplifiers (Neuroscan, Inc.) filtered on-line with a low-frequency half amplitude cutoff at 0.01 Hz and a high-frequency half amplitude cutoff at 50 Hz. The EEG signals were recorded under various brain functions and with eyes open or closed. Rejection of EEG segments affected by blink, muscular or other kinds of artifact activity was performed off-line by an experienced EEG expert visual inspection of the recordings.

3.2 Data Analysis

From the EEG signals analysis of [12], it was well established that the four kinds of brain rhythms played an important role in the brain function analysis. The wavelet packet transform allowed us to decompose the signals accurately in the specific frequency bands which could not be achieved by the wavelet transform. In order to obtain an accurate separation of the four kinds of rhythm of the EEG, a six-level wavelet packet decomposition was performed. Four kinds of brain rhythms were obtained: delta rhythm (0.78-13.28Hz), theta rhythm (3.91-7.8Hz), alpha rhythm (7.8-13.28Hz) and beta rhythm (13.28-30.47Hz).

To investigate the effect of the wavelet packet decomposition, a four seconds EEG signal was applied. The EEG was recorded when the subject closed the eyes. 400 points of EEG signal were decomposed by a six-level Discrete Meyer wavelet. Four kinds of rhythm were obtained by the wavelet packet tree. From figure 1, we could see that the rhythms were well extracted by the wavelet packet decomposition.

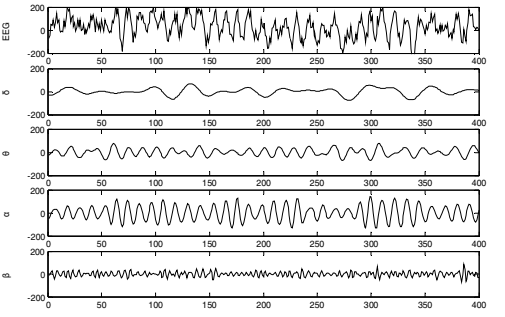


Fig. 1. A segment of original EEG signal in channel P3 with eyes closed and the decomposition result of the wavelet packet decomposition

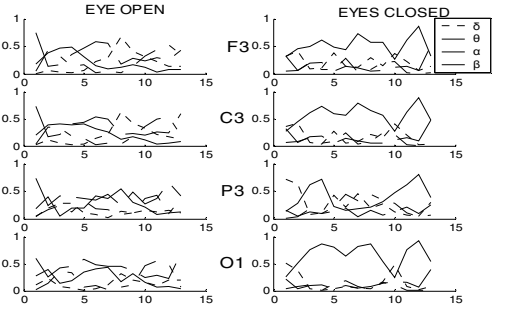


Fig. 2. Component energy of each rhythm in the open eyes status and closed eyes status respectively

4 Result and Discussion

It has been proved that the alpha rhythm in the spontaneous EEG will enhance when the subject close his eyes, which represent different brain state compared to the open eyes status. Two segments of EEG signal were chosen to be tested. The first segment is a 2 seconds EEG signal with eyes open. The other EEG signal is a 2 seconds period with the subject's eyes closed. From figure 2, we can obviously see that when the subject opens his eyes, the four kinds of rhythms were comparable to each other. Nevertheless the alpha rhythm was enhanced and became the domain rhythm when the subject closed the eyes. From figure 3, we can see that the wavelet packet entropy is lower in the closed eyes status, which represent that the brain activity is more order than the eyes open status.

In order to reflect the rhythms' time-varying characteristic of EEG signals in the whole cerebral cortex, we calculate wavelet packet entropy of each channel, and then, using 2D interpolation method to get the topographic map of mean wavelet packet entropy of EEG signals. Figure 4 is topographic maps of mean wavelet packet entropy of 14 channels EEG signals in the open eyes and closed eyes status. From this figure, we can clearly observe the difference between the two statuses.

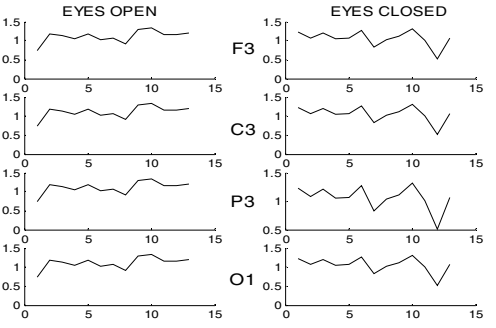


Fig. 3. Wavelet packet entropy in the open eyes and closed eyes status respectively

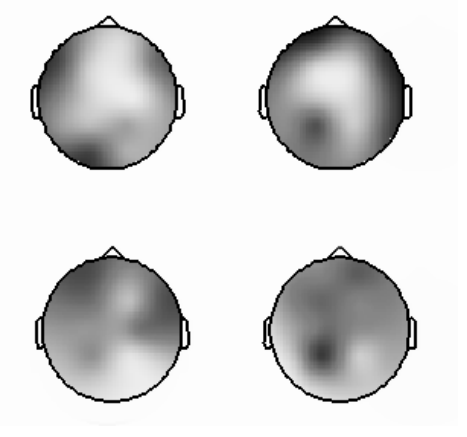


Fig. 4. Topographic map of mean wavelet packet entropy in the open eyes and closed eyes status respectively (upper: open eyes, lower: closed eyes)

5 Conclusion

This paper presents an effective method, which is based on the multi-resolution decomposition with wavelet packet design, to analyze the clinical EEG signals. The experimental results show that the wavelet packet decomposition can effectively distinguish each rhythm of EEG signal, and both the wavelet packet energy and wavelet packet entropy can be used to effectively measure the complexity of the EEG signal. Due to the better matching in time-frequency characteristics of EEG signal, our method, compared with the Wavelet method, is more flexible and accurate for the designing of specific filter banks and the detecting of different EEG rhythms. We can also see that our method can be used as a new way for analyzing other kinds of medical signals in practice.

Acknowledgements. This research is supported by the Natural Science Foundation of China (Grant No.60571066) and Natural Science Foundation of Guangdong.

References

1. Pardey, J., Roberts, S., Tarassenko, L.: A Review of Parametric Modeling Techniques for EEG Analysis. Med. Eng. Phys. 18(1), 2–11 (1996)
2. Shen, M.F., Sun, L.S., Chan, F.H.Y.: The Classification of Transient Time-Varying EEG Signals Via Wavelet Packets Decomposition. In: 8th International Conference on Neural Information Processing (2001)
3. Gamero, L.G., Plastino, A., Torres, M.E.: Wavelet Analysis and Nonlinear Dynamics in a Nonextensive Setting. Physica A 246, 487–509 (1997)
4. Pesquet, J., Krim, H., Carfantan, H.: Timeinvariant Orthonormal Wavelet Representations. IEEE Trans. on Signal Processing 44(8), 1964–1970 (1996)
5. Tong, S., Bezerianos, A., Paul, J., Zhu, Y., Thakor, N.: Nonextensive Entropy Measure of EEG Following Brain injury From Cardiac Arrest Source. Physica A: Statistical Mechanics and its Applications 305, 619–628 (2002)
6. AlNashash, H.A., Paul, J.S., Thakor, N.V.: Wavelet Entropy Method for EEG Analysis: Application to Global Brian Injury. In: Proceeding of the 1st International IEEE EMBS Conference on Neural Engineering (2003)
7. Quiroga, R.Q., Rosso, O.A., Basar, E., Schurmann, M.: Wavelet Entropy in Event-related Potentials: a new Method Shows Odering of EEG Oscillations. Biological Cybernetics 84, 291–299 (2001)
8. Lemire, D., Pharand, C., Rajaonah, J., Dube, B., Leblanc, A.R.: Wavelet Time Entropy, T Wave Morphology and Myocardial Ischemia. IEEE Transactions On Biomedical Engineering 47, 967–970 (2000)
9. Quinquis, A.: Few Practical Applications of Wavelet Packets. Digital Signal Processing 8, 49–60 (1998)
10. Thakor, N., Bezerianos, A., Tong, S.: Time-dependent Entropy Estimation of EEG Rhythm Changes Following Brain Ischemia. Annals of Biomedical Engineering 31, 221–232 (2003)
11. Hughes, J.R.: EEG in Clinical Practice. Butterworth-Heinemann (1994)
12. Delorme, A., Makeig, S.: EEG Changes Accompanying Learned Regulation of 12-Hz EEG Activity. IEEE Transactions on Neural Systems and Rehabilitation Engineering 11, 133–137 (2003)

A New Practical Method on Hydrological Calculation

Lihua Feng and Xingcai Zhang

Department of Geography, Zhejiang Normal University, Jinhua, 321004, China
fenglh@zjnu.cn

Abstract. Artificial Neural Networks (ANN) deal with information through interactions among neurons (or nodes), approximating the mapping between inputs and outputs based on non-linear functional composition. They have the advantages of self-learning, self-organizing, and self-adapting. It is practical to use ANN technology to carry out hydrologic calculations. To this end, this note has fundamentally set up a system of calculation and analysis based on ANN technology, given an example of application with good results. It shows that ANN technology is a relatively effective way of solving problems in hydrologic calculation.

Keywords: Artificial neural networks, BP algorithm, Hydrologic calculation.

1 Introduction

In hydrologic calculation, it is common to set up mathematical models or draw related graphs based on existing data. Hence, it involves issues of pattern recognition [1]. There is, however, no satisfactory mathematical model $y = f(x)$ that would fix hydrologic elements [2,3]. Since the newly developed technology of Artificial Neural Networks (ANN for short) has advantages of self-learning, self-organizing, and self-adapting [4], there are many successful applications of it on pattern recognition [5,6]. Therefore, based on the principle and method of ANN [7], we study some related issues of hydrologic calculation in this note.

2 Method

An ANN is a complex network that consists of many simple neural cells [8]. It is roughly modeled on the human brain. It has a parallel distribution information processing device and can approximate the mapping between input and output by compositions of nonlinear functions [9]. It does not require any design of mathematical models. It can learn solely based on experience; process various fuzzy, nonlinear, noisy data through neuron simulation, memory, and association; and process calculation analysis using the method of self-adapting pattern recognition [10].

ANN algorithms include Hebbian, Delta, Kohonen, and BP [4]. The BP algorithm (Error Back Propagation) was presented in 1985 by Rumelhart and his PDP team. It realized Minsky's thought on multilayer neural networks. A typical multilayer-feed-forward neural network consists of a number of neurons that are connected together,

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 29–35, 2009.

usually arranged in layers. Its first layer is the input layer. Its final layer is the output layer. All other layers are hidden layers, which contain the neurons that do the real work.

A neural network that uses the error back propagation algorithm is said to be a BP network, whose learning process consists of the feed-forward and feed-backward. Each sample signal in the feed-forward process is applied by the Sigmoid function $f(x) = 1/(1+e^{-x})$ before it is passed to next layer. The situation of neurons on each layer can only affect the situation of neurons on the next layer. If the output layer does not produce the desired value, then the errors will be fed back from the outputs to the inputs through the network, and the weights of nodes in each layer will be changed along the way. The algorithm repeats in this way until the error values are sufficiently small.

Let m be the number of layers, y_j^m denote the output from the node j in the layer m, $y_j^0 = x_j$ denote the input at node j, W_{ij}^m be the weight of connection between node I and node j, and θ_j^m be the threshold at the node j in the layer m. The BP network is training as follows:

(1) Initialize each weight and threshold to a random value in $(-1, \quad 1)$.

(2) Select a pair of data (x^k, T^k) from the training data and substitute inputs into the input layer such that

$$y_i^0 = x_i^k \qquad (\text{for } i) \tag{1}$$

Where k denotes the number of iterations.

(3) Pass the signal forward by using the formula:

$$y_j^m = F(s_j^m) = F(\sum_i W_{ij}^m y_i^{m-1} + \theta_j^m) \tag{2}$$

The calculation processes the output at each node j from the first layer through the last layer until it completes. Where $F(s)$ is the Sigmoid function.

(4) Calculate the error for each node j in the output layer as follows:

$$\delta_j^m = y_j^m(1-y_j^m)(T_j^k - y_j^m) \tag{3}$$

Where the error is obtained by the difference of the actual output value and the desired target value.

(5) Calculate the error for each node j in each hidden layer as follows:

$$\delta_j^{m-1} = F'(s_j^{m-1})\sum_i W_{ij}^m \delta_i^m \tag{4}$$

The error is obtained by feeding back errors layer by layer, where $m = m, m-1, \cdots, 1$.

(6) Change the weights and thresholds backward layer by layer:

$$W_{ij}^m(t+1) = W_{ij}^m(t) + \eta\delta_j^m y_i^{m-1} + \alpha[W_{ij}^m(t) - W_{ij}^m(t-1)] \tag{5}$$

$$\theta_j^m(t+1) = \theta_j^m(t) + \eta\delta_j^m + \alpha[\theta_j^m(t) - \theta_j^m(t-1)] \tag{6}$$

Where t is the number of iterations; η is the learning rate ($\eta \in (0,1)$); α is the momentum value ($\alpha \in (0,1)$.

(7) Go to step (2), start the next iteration, repeat (2) through (7) until the network error

$$E = \sum_k \sum_j (T_j^k - y_j^m)^2 / 2 \tag{7}$$

is sufficiently small as expected.

Once the network completes its training, its weights and thresholds are determined. Thus, we can start a calculation analysis.

3 Result

We demonstrate an application of the ANN technology in hydrologic calculation in this section by examining the peak stage [11] at the Shi-Gou station in Sui-Jiang, China as shown in Table 1.

Let H_S be the peak stage recorded at the Shi-Gou station. Then H_S can be expressed as

$$H_S = f(H_G, H_T, P)$$

Where H_G denotes the peak stage recorded at the Gu-Shui station, which is the upper reaches of the Shi-Gou station; H_T denotes the peak stage at the Shi-Gou station recorded at the same time as H_G ; P is the precipitation of space interval. Since H_G, H_T , and P are the inputs while H_S is the output, there are three nodes in the input layer and one node in the output layer. It follows from Kolmogorov's law that there are eight nodes in the hidden layer. Hence, the ANN in our hydrologic calculation has the topological structure (3, 8, 1).

In order to speed up the convergence, let us normalize the original data x_i as follows

$$x_i^{'} = (x_i - x_{\min}) / (x_{\max} - x_{\min}) \tag{8}$$

Where $x_{\max}, x_{\min}$ denote the maximal value and the minimal value of the flood series, respectively. Thus, each $x_i^{'} \in [0,1]$.

We can input $x_i^{'}$ into the input layer of the BP algorithm and select training data to start the training and learning process. We choose the learning rate η= 0.85 and the momentum value α= 0.60. In order to test the BP algorithm after each training and learning, we take the first thirteen flood series as the training samples, and the last three flood series as the testing samples. After one hundred thousand times of training and learning from the training samples, the network error E = 0.003, which is less

than expected error; thus, the BP algorithm is convergent. It is clear as shown in Table 1 that the imitation is very good since the average error $\bar{e}$ of the series is only about 0.07 meters and the maximal error $e_{\max}$ of the series is only about 0.19 meters.

Since the trained network has imitated and memorized the functional relationship between input and output, it can be used to determine the flood stage. It is obvious as shown in Table 1 that the result of tests for the three flood series is good since the prediction errors are less than 1%.

Table 1. Peak stage at the Shi-Gou stations and result of its calculation (unit: m)

	Order	H_G	H_T	P(mm)	H_S	Output	Fit value	Error
	1	36.90	14.84	102	16.92	0.9804	16.83	-0.09
	2	29.12	12.44	23	12.72	0.1125	12.83	0.11
	3	29.93	11.50	52	13.06	0.1446	12.98	-0.08
Training	4	31.40	12.98	63	14.08	0.4119	14.21	0.13
	5	31.51	12.76	88	14.29	0.4277	14.28	-0.01
	6	31.80	14.39	67	14.72	0.5174	14.70	-0.02
	7	30.05	13.10	70	13.60	0.2823	13.61	0.01
	8	29.82	12.58	52	13.44	0.2116	13.29	-0.15
	9	28.64	12.59	122	12.85	0.1165	12.85	0.00
sample	10	28.89	10.89	37	12.37	0.0200	12.40	0.03
	11	28.54	11.71	64	12.31	0.0420	12.50	0.19
	12	29.78	12.16	31	13.11	0.1629	13.06	-0.05
	13	28.94	11.77	22	12.56	0.0543	12.56	0.00
Testing	14	30.26	12.34	44	13.36	0.2416	13.42	0.06
	15	30.22	13.15	72	13.70	0.3136	13.76	0.06
sample	16	32.19	14.68	136	15.44	0.6597	15.35	-0.09

4 Discussion

4.1 ANN's Application in Hydrologic Calculations

As mentioned at the beginning, it is hard to find a function f that would express the relationship between a dependent variable y and an independent variable x of hydrologic elements such that $y = f(x)$, even in the simplest case like the relation between the discharge Q and the stage H. On the other hand, we have seen that the greatest advantage of an ANN is that it does not need a mathematical model. It can imitate and memorize any complex relationship between inputs and outputs by training and learning upon historical data, and carry out the calculation analysis by association. Therefore, many issues (including forecast) in hydrologic calculation can be analyzed by using ANN technology. The main issues are as follows:

(1) Calculation of discharge-stage. Where the input variable is the stage, the output variable is the discharge.

(2) Forecast of the propagation time of flood peak with multiple factors. Where factors may include the simultaneous discharge of lower reaches, the difference of discharges of upper reaches, the precipitation of space interval, the backwater of lower reaches, and the discharge of multiple tributaries, these factors are input variables. The output variable is the propagation time of flood peak.

(3) Forecast of the precipitation-runoff with multiple factors. Where the input variables include the rainfall at each single station P_i $(i = 1, 2, \cdots, n)$, the average of areal rainfall, the earlier stage affecting rainfall P_a, the water storage W_0 of drainage basin prior rain, the duration of rainfall *T*, the intensity of rainfall, the evaporation in the interval of precipitation, and the initial discharge Q_0, the output variable is the runoff.

(4) Flood routing through reservoir. Where the input variables include the precipitation within a time interval at each single station P_i $(i = 1, 2, \cdots, n)$, the average of areal rainfall within a time interval, the average of discharge into reservoir within a time interval, and the initial dam stage of a time interval, the output variable is either the dam stage at the end of a time interval or the average of discharge out of reservoir within a time interval.

(5) Calculation of the largest flood of different drainage basin areas. Where the input variable is the area of a drainage basin, the output variable is the largest peak discharge of an actual survey.

(6) Calculation of hydrologic data extension of a design station. Where input variables are the annual rainfall, the annual runoff, the modulus of annual flow, and so on, of a reference station, the output variable is the annual runoff of a design station.

In addition, there are many other issues, such as the forecast of low flow with multiple factors, the forecast of melted snow runoff, the ice-condition forecast, the forecast of tidal river stages, the medium and long term hydrologic forecast, the relation between the point rainfall and the areal rainfall, the flood peak-volume relation, the discharge-sediment relation, the unit sediment-section sediment relation, the natural annual runoff restoration and so on, that can also be analyzed by using the ANN technology. To that end, we collected a large number of data and carried out calculations. It turns out that the results are generally satisfactory as long as we select proper parameters. Therefore, we believe that the ANN technology has a bright future of applications in hydrologic calculations [12].

4.2 Selection of Parameters

(1) The selection of the number of nodes for the hidden layer.
In 1989, Robert Hecht and Nielson proved that a continuous function on any closed interval can be approximated by a BP network with one hidden layer, thus, a BP network with three layers can carry out any mapping from *n*-dimensional space to *m*-dimensional space. Thus, we have adopted a three-layer BP network with single hidden layer in our calculation. It follows from the Kolmogorov's theorem that the number of nodes in the hidden layer is at least $2n + 1$, where *n* is the number of nodes in the input layer. Since $n = 3$, the number of nodes in the hidden layer is at least 7. Considering the accuracy, we determined that the number of nodes for the hidden layer is 8

(2) The selection of the momentum value

In order to improve the network training speed, a momentum value $a \in (0,1)$ is added in the formula that is used to modify the weights. The momentum takes into account the extent to which a particular weight was changed on the previous iteration. When the value of *a* is equal to 0, the change of weights is obtained via the method of gradient descent. When the value of *a* is equal to 1, the change of weights is set to

equal to the change on the previous iteration such that the part of the change generated by the method of gradient descent is ignored. Therefore, when the value of a is increased, it will help to find the set of weights that provides the best performance of the network. In our case, we choose the value of 0.60 for the momentum a.

(3) The selection of the learning rate

It is very important to choose a proper value of the learning rate η during the course of training the network. The value of the learning rate η is a positive number below 1, and which should not be too high. When the value of η is too high, it may result in an unstable state. On the other hand, if the value of η is too small, it may take long time to complete the course of training the network. The bigger of the value of η, the faster of the modification of the weights will be. Therefore, we may choose a large value of η provided that it will not cause any instability of the performance. In our case, we choose the value of 0.85 for the learning rate η.

5 Conclusion

Although it is often impossible to find a specific function for the relation between a dependent variable and an independent variable in hydrologic elements, the ANN technology can provide us an alternative solution for such a hard issue. It deals with information through interactions among neurons (or nodes), and approximates the mapping between inputs and outputs based on the non-linear functional composition. It has the advantages of self-learning, self-organizing, and self-adapting. Therefore, it is practical to use the ANN technology to carry out hydrologic calculations. Our calculation results have confirmed that.

The error in our example (see Table 1) is only between 0.44% and 0.58%, which is certainly good enough for a hydrologic forecast. However, it will be very difficult to achieve the same accuracy if we use the traditional approach [11]. Furthermore, the ANN technology allows us to have multi-variables in both input and output layers. This is very important for hydrologic calculations since the stage, discharge, and other hydrological variables are often functions of many influential variables.

This note, aiming at the issues in hydrologic calculation, has preliminarily set up a system of calculation and analysis based on ANN technology. We have developed applied functional software along with our research. This is a new attempt in hydrologic calculation. If we combine it with other algorithms, there is no doubt that we will be able to improve the accuracy and level of the hydrologic calculation.

Acknowledgments. This work was supported by National Natural Science Foundation of China (No. 40771044).

References

1. Acharya, U.R., Bhat, P.S.: Classification of Heart Rate Data Using Artificial Neural Network and Fuzzy Equivalence Relation. Pattern Recognition 36, 61–68 (2003)
2. Mukhopadhyay, A.: Application of Visual, Statistical and Artificial Neural Network Methods in the Differentiation of Water from the Exploited Aquifers in Kuwait. Hydrogeology Journal 11, 343–356 (2003)

3. Wilby, R.L., Abrahart, R.J., Dawson, C.W.: Detection of Conceptual Model Rainfall–Runoff Processes Inside an Artificial Neural Network. Hydrological Sciences Journal 48, 163–181 (2003)
4. Ertay, T., Çekyay, B.: Integrated Clustering Modeling with Backpropagation Neutral Network for Efficient Customer Relationship Management. In: Ruan, D., et al. (eds.) Intelligent Data Mining: Techniques and Applications, pp. 355–373. Springer, Heidelberg (2006)
5. Konstantin, L., Norman, G.L.: Application of an Artificial Neural Network Simulation for Top-Of-Atmosphere Radiative Flux Estimation from CERES. Journal of Atmospheric and Oceanic Technology 20, 1749–1757 (2003)
6. Marina, C., Alfredo, S., Paolo, A.: Artificial Neural Network Approach to Flood Forecasting in the River Arno. Hydrological Sciences Journal 48, 381–398 (2003)
7. Zhou, J.C., Zhou, Q.S., Han, P.Y.: Artificial Neural Network–The Realization of the Sixth Generation Computer, pp. 47–51. Scientific Popularization Publisher (1993)
8. Lippmann, R.P.: An Introduction to Computing With Neural Nets. IEEE ASSP Magazine 4, 4–22 (1987)
9. Chat, S., Abdullah, K.: Estimation of All-Terminal Network Reliability Using an Artificial Neural Network. Computers and Operations Research 29, 849–868 (2002)
10. Brion, G.M., Lingireddy, S.: Artificial Neural Network Modeling: A Summary of Successful Applications Relative to Microbial Water Quality. Water Science and Technology 47, 235–240 (2003)
11. Li, H.L., Li, L.L., Yan, J.F.: Hydrologic Forecast, pp. 12–13. China Waterpower Press (1979)
12. Reed, S., Schaake, J., Zhang, Z.: A Distributed Hydrologic Model and Threshold Frequency-Based Method for Flash Flood Forecasting at Ungauged Locations. Journal of Hydrology 337, 402–420 (2007)

Bernoulli Neural Network with Weights Directly Determined and with the Number of Hidden-Layer Neurons Automatically Determined

Yunong Zhang and Gongqin Ruan

School of Information Science and Technology, Sun Yat-sen University
Guangzhou 510275, China
zhynong@mail.sysu.edu.cn, ynzhang@ieee.org

Abstract. Conventional back-propagation (BP) neural networks have some inherent weaknesses such as slow convergence and local-minima existence. Based on the polynomial interpolation and approximation theory, a special type of feedforward neural-network is constructed in this paper with hidden-layer neurons activated by Bernoulli polynomials. Different from conventional BP and gradient-based training algorithms, a weights-direct-determination (WDD) method is proposed for the Bernoulli neural network (BNN) as well, which determines the neural-network weights directly (just in one general step), without a lengthy iterative BP-training procedure. Moreover, by analyzing the relationship between BNN performance and its different number of hidden-layer neurons, a structure-automatic-determination (SAD) algorithm is further proposed, which could obtain the optimal number of hidden-layer neurons in a constructed Bernoulli neural network in the sense of achieving the highest learning-accuracy for a specific data problem or target function/system. Computer-simulations further substantiate the efficacy of such a Bernoulli neural network and its deterministic algorithms.

Keywords: Bernoulli polynomials, Feedforward neural networks, Weights direct determination, Structure automatic determination, Hidden layer.

1 Introduction

Artificial neural networks have become a useful tool in dealing with various scientific and engineering problems such as system design and control [1,2,3,4], image processing [5], and robot inverse-kinematics [4,6] due to their remarkable advantages such as parallelism, distributed storage and computation, adaptive-learning ability. Among the most important neural-network models, BP neural networks have been widely investigated and also applied to many practical fields [7,8]. But the inherent weaknesses still exist in BP-type neural networks and their algorithms (including those improved variants), such as, slow convergence of iterative training, local-minima existence of solution, and uncertainties in the optimal number of hidden-layer neurons [8].

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 36–45, 2009.

Different from algorithmic improvements about the BP iterative-training procedure, by our successful experience [1,9,10], adopting linearly-independent or orthogonal activation functions to construct the neural network might be a much better choice. In this paper, based on the polynomial interpolation and approximation theory [11,12], we further try enhancing the neural-network performance by proposing a so-called Bernoulli neural network. This special neural network has a three-layer structure as well, but with hidden-layer neurons activated by Bernoulli polynomials. More importantly, in order to avoid the usually-lengthy iterative-training procedure (which is mostly based on gradient and BP methods), a pseudoinverse-based weights-direct-determination method is derived for the Bernoulli neural network, which can determine the theoretically optimal weights directly (just in one step).

Furthermore, as we all know possibly, selecting the number of hidden-layer neurons is an important and difficult issue because it may affect the overall performance of neural networks very much. Specifically speaking, fewer hidden neurons may not achieve a satisfactory performance, whereas too many hidden-layer neurons may lead to a higher complexity of circuit-implementation, software computation and even an over-fitting phenomenon [8,10]. By observing the relationship between BNN performance and hidden-layer-neurons' number, a structure-automatic-determination method/algorithm is proposed further in this paper for Bernoulli neural network. Specifically, it exploits the weighs-direct-determination method at its every trial and generates finally the optimal number of hidden-layer neurons. Via the WDD and SAD algorithms, the highest accuracy of learning can be achieved by the Bernoulli neural network for a specific data system or target function approximation. Computer-simulation results substantiate the superiority of the Bernoulli neural network and its algorithms.

2 Neural-Network Model and Theoretical Basis

In this section, we propose the Bernoulli neural network (or termed, Bernoulli polynomial neural network) firstly, and then present some important definitions and theorems which guarantee its approximation ability.

2.1 Model Structure

Fig. 1 shows the proposed model of Bernoulli neural network, which, in the hidden layer, has n neurons activated by a group of degree-increasing Bernoulli polynomials $\phi_j(x), j = 0, 1, \cdots, n-1$. The BNN input-layer or output-layer each has one neuron activated by simple linear function $f(x) = x$. Moreover, the weights between input-layer and hidden-layer neurons are all fixed to be 1, whereas the weights between hidden-layer and output-layer neurons, denoted as $w_j, j = 0, 1, \cdots, n-1$, are to be decided or adjusted. In addition, all neuronal thresholds are fixed to be nil. These settings could simplify the neural-network structure design, circuit implementation and computational complexity. More importantly, even so, the approximation ability of the Bernoulli neural network can still be theoretically guaranteed.

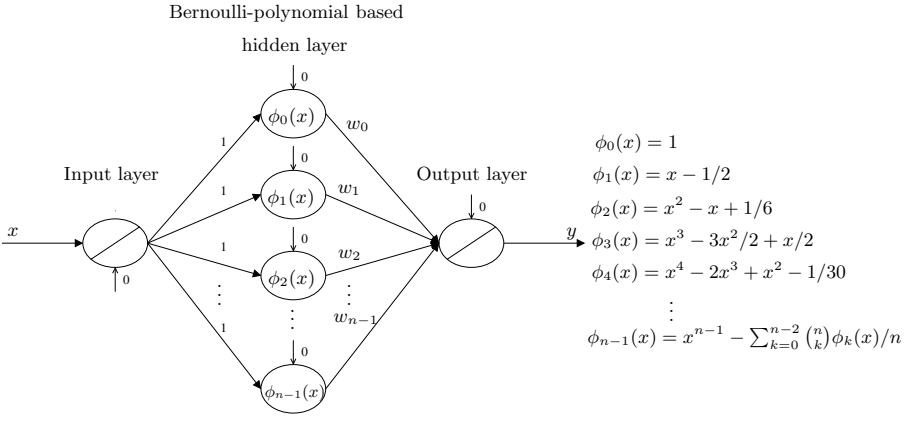


Fig. 1. A new neural network with hidden neurons activated by Bernoulli polynomials

2.2 Theoretical Basis

About the Bernoulli neural network proposed in Subsection 2.1, people may wonder whether it works. Now we show its theoretical basis. As we know [13,14], Bernoulli polynomials are quite important in various expansion and approximation formulas with many applications in combinatorics and number theory.

Firstly, as noticed, there are a variety of different expressions for Bernoulli polynomials. In this paper, we present a simple recurrence relation of Bernoulli polynomials for the facility of such a neural-network construction. In [14], the Bernoulli polynomials are defined via a matrix-determinant approach, and then, by simplifying it, the stable recurrence expression of Bernoulli polynomials can be shown as follows with $\phi_0(x) = 1$:

$$\phi_j(x) = x^j - \frac{1}{j+1} \sum_{k=0}^{j-1} \left(\binom{j+1}{k} \phi_k(x) \right), \text{ with } \binom{j+1}{k} := \frac{(j+1)!}{k!(j+1-k)!}, \quad (1)$$

where $\phi_j(x)$ denotes a Bernoulli polynomial of degree $j = 1, 2, 3, \cdots, n, \cdots$, with its first few analytic expressions given in Fig. 1 as well as in the Appendix.

Similar to the fundamental idea appearing in our previous work [1,9,10], the final approximation mathematical essence of such a Bernoulli-neural-network construction can be viewed as a procedure of constructing a generalized polynomial function so as to interpolate or approximate an unknown target function/system using a given set of sample data. When approximating an (unknown) objective function $\varphi(x)$, from Fig. 1, the relation between input- and output-neurons of Bernoulli neural network can be exactly expressed as

$$y = \phi(x) = w_0\phi_0(x) + w_1\phi_1(x) + \cdots + w_{n-1}\phi_{n-1}(x), \quad (2)$$

of which the learning and approximating abilities can be guaranteed by the following definition and theorem [note that $\varphi(x)$ is also termed target function].

Definition 1 [9,10,15,16]. *Assume that* $\varphi(x), \phi_j(x) \in C[a,b], j = 0, 1, \cdots, n-1$ *(in words, target function* $\varphi(x)$ *and the* j*th polynomial function* $\phi_j(x)$ *of polynomial-function sequence* $\{\phi_j(x)\}_{j=0}^{n-1}$ *are continuous over the closed interval* $[a,b]$*), and that* $\{\phi_j(x)\}_{j=0}^{n-1}$ *is a set of linearly independent polynomial-functions. For given weighting-function* $\rho(x)$ *on interval* $[a,b]$*, appropriate coefficients* $w_0, w_1, \cdots, w_{n-1}$ *can be chosen for the generalized polynomial* $\phi(x) = \sum_{j=0}^{n-1} w_j \phi_j(x)$ *so as to minimize* $\int_a^b (\varphi(x) - \phi(x))^2 \rho(x)\, dx$*. Then,* $\phi(x)$ *is termed the least-square approximation of* $\varphi(x)$ *with respect to* $\rho(x)$ *over interval* $[a,b]$*.*

Theorem 1 [9,10,15,16]. *For* $\varphi(x) \in C[a,b]$*, its least-square approximation function* $\phi(x)$ *could exist uniquely, of which the coefficients (in other words, the neural-network weights shown in Fig. 1),* $w_0, w_1, \cdots, w_{n-1}$*, are solvable.*

3 BNN Weights-Direct-Determination

From the description of the above Bernoulli-neural-network structure, it can be viewed as a special type of BP neural networks, which can still adopt BP algorithms as its iterative-training rule. However, if a BP algorithm is employed, it may take much time (or even infinite time) to converge to a solution under a user-specified accuracy of learning (e.g., 10^{-18} in the ensuing simulation-study of ours). As mentioned above, for the Bernoulli neural network, its weights can be determined directly (i.e., via the so-called weights-direct-determination method), instead of a lengthy BP iterative-training procedure. Now we show it as follows.

By taking $(x_i, \gamma_i := \varphi(x_i))|_{i=1}^m$ as the given training-sample pairs, we can define the batch-processing error (BPE) function E for BNN as below:

$$E = \frac{1}{2} \sum_{i=1}^{m} \left(\gamma_i - \sum_{p=0}^{n-1} w_p \phi_p(x_i) \right)^2. \tag{3}$$

Then we have the following theorem about the BNN weights-direct-determination method (with proof similar to [9] but omitted in view of space limitation).

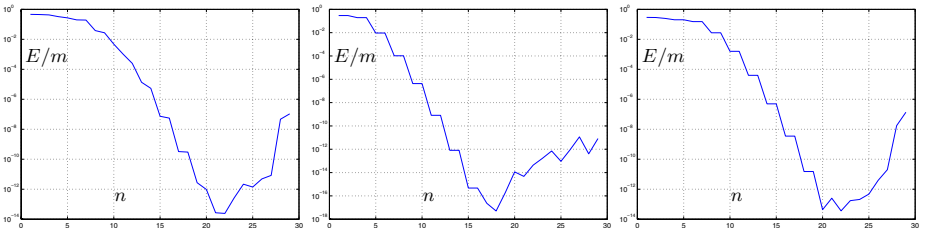
Theorem 2. *Let superscript* ^{T} *denote the transpose of a matrix or vector. As for the Bernoulli neural network depicted in Fig. 1, let us define its weights vector* w*, target-output vector* γ*, and input-activation-matrix* X *respectively as*

$$w = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{n-1} \end{bmatrix}, \gamma = \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_m \end{bmatrix}, X = \begin{bmatrix} \phi_0(x_1) & \phi_1(x_1) & \dots & \phi_{n-1}(x_1) \\ \phi_0(x_2) & \phi_1(x_2) & \dots & \phi_{n-1}(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(x_m) & \phi_1(x_m) & \dots & \phi_{n-1}(x_m) \end{bmatrix} \in R^{m \times n}.$$

Then, with superscript $^+$ *denoting the pseudoinverse of a matrix, the steady-state weights vector of the Bernoulli neural network can be determined directly as*

$$w = X^+ \gamma := (X^T X)^{-1} X^T \gamma, \tag{4}$$

which is optimal in the sense of minimizing the batch-processing error E *in (3).*



(a) As of target function (5) (b) As of target function (6) (c) As of target function (7)

Fig. 2. Approximation error of BNN versus its number of hidden-layer neurons

BNN Structure-Automatic-Determination Algorithm

Input: Training-sample pairs $(x_i, \gamma_i)|_{i=1}^{m}$

Output: The optimal number of hidden neurons, n

Initialize the loop index $q = 2$

Initialize MinE=E to be a very large value (e.g., 10^{10})

Initialize the optimal number $n = 1$

Construct Bernoulli polynomial $\phi_0(x) = 1$

While $E \leqslant$MinE or $n == q - 2$,

- Construct Bernoulli polynomial $\phi_{q-1}(x)$
- Calculate the neural-network weights using the weights-direct-determination method (4)
- Calculate the resultant BNN learning error E
- If $E <$MinE,

  MinE $\leftarrow E$

  $n \leftarrow q$

  End
- $q \leftarrow q + 1$

End

Fig. 3. Algorithmic description of structure-automatic-determination (SAD) method which decides the optimal number of hidden-layer neurons in Bernoulli neural network

4 BNN Structure-Automatic-Determination

It might be well known that the number of hidden-layer neurons has an important influence on the performance of neural networks, but currently there is still no deterministic theory which handles this issue. As an attempt to develop a suitable algorithm for determining the optimal number of hidden-layer neurons in Bernoulli neural network, we firstly observe and analyze the relationship between BNN performance and the hidden-neurons' number via the approximation examples of a variety of target functions. For example, Fig. 2 shows the BNN-approximation error (specifically, E/m) of the following three tested target-functions versus the number of BNN hidden-layer neurons:

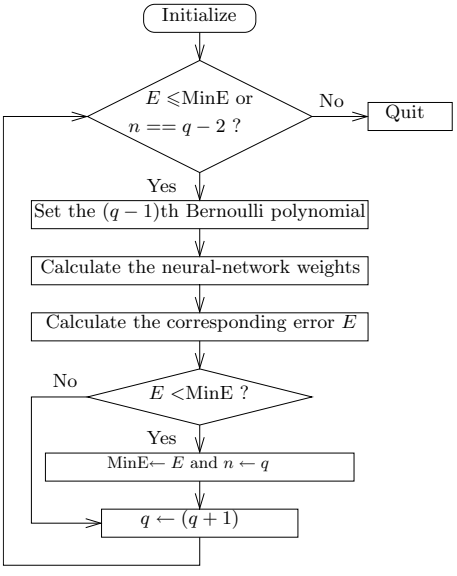


Fig. 4. Program flowchart of BNN structure-automatic-determination method

$$\varphi(x) = e^x \cos(3\pi x), \tag{5}$$

$$\varphi(x) = sin(6x)/cos(x) - x, \tag{6}$$

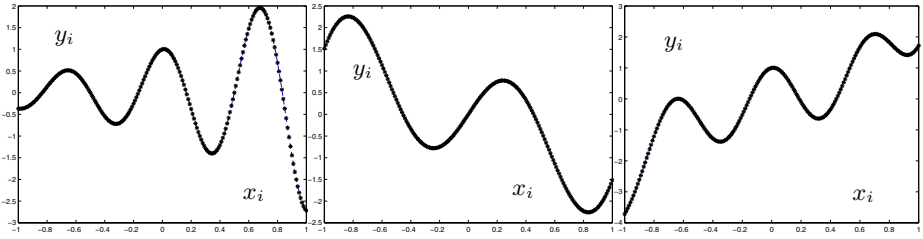
$$\varphi(x) = xe^{x^2} + cos(3\pi x), \tag{7}$$

which are out of the twelve target functions we have tested (but no other tests are presented in this paper due to space limitation and results' similarity). From Fig. 2 and other testing results, we observe that the BNN performance has a relatively consistent trend with respect to the number of hidden-neurons (which increases from 2 to 30), no matter what target function is tested. That is, at the beginning, the BPE defined in (3) would generally decrease over the increase of the hidden-neurons' number. In addition, the BPE-decreasing procedure will not terminate until it reaches the optimal number of hidden-layer neurons (at this point BPE is the smallest). After that, as the number n increases, the BPE will go up generally. So, we could search the optimal number of hidden-layer neurons by increasing n one by one, in addition to using the aforementioned weights-direct-determination method. The structure-automatic-determination algorithm could thus be developed and presented in Figs. 3 and 4.

In addition, it is worth mentioning that during the search procedure through increasing the number of hidden-layer neurons, the BPE error E calculated at the current loop (with a larger value of the number of hidden-layer neurons, q) might become larger than (or equal to) that at the previous loop (e.g., with a smaller value of the number of hidden-layer neurons, $q-1$). The key point of the proposed SAD algorithm is to deal with such a situation. From our simulation results (e.g., Fig. 2), it is observed that, for (almost) all tested target-functions,

Table 1. Learning and approximation results of the proposed Bernoulli neural network

Function #	Optimal number n	Runtime (second)	Average error of training, E/m	Average error of testing, E/m
Target (5)	23	0.062	2.44×10^{-14}	2.40×10^{-14}
Target (6)	19	0.042	5.08×10^{-18}	5.02×10^{-18}
Target (7)	23	0.060	3.62×10^{-14}	3.38×10^{-14}



(a) As of target function (5) (b) As of target function (6) (c) As of target function (7)

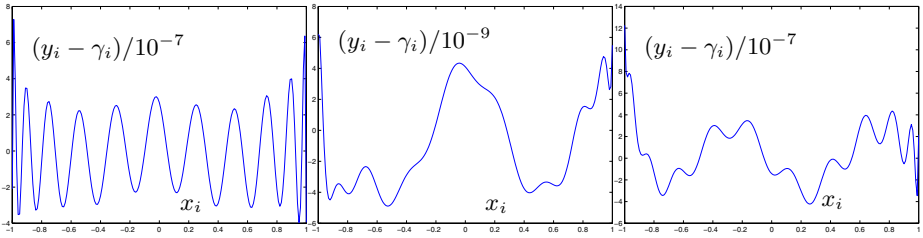
Fig. 5. Approximation results of Bernoulli neural net with WDD and SAD algorithms

if $q-1$ is not the true optimal-number of hidden-layer neurons, the BPE error E with $q+1$ hidden-neurons will become less than that with $q-1$ hidden-neurons (although the BPE error E with q hidden-neurons is larger than that with $q-1$ hidden-neurons). Moreover, as from our simulative observation, if the errors E with $q+1$ and q hidden-neurons are both larger than that with $q-1$ hidden-neurons, we can say quite definitely that $q-1$ is the optimal number n of hidden-layer neurons of the Bernoulli neural network for that specific approximation task $(x_i, \gamma_i)|_{i=1}^{m}$. As q starts from 2 and increases, such a structure-automatic-determination algorithm depicted in Figs. 3 and 4 continues to search the optimal number of hidden-layer neurons until the above objective is achieved.

5 Computer-Simulation Verification

For the purpose of testing, verifying and illustrating the efficacy of this novel Bernoulli neural network and its algorithms, the mentioned three target-functions in Section 4 [i.e., in (5)-(7)] are employed here for the simulation study. Let us take and sample uniformly over the interval $[-1.0, 1.0]$ with gap-size 0.01 to generate the related data-sets $\{(x_i, \gamma_i), i = 1, 2, \cdots, 201\}$ (i.e., $m = 201$). Then, the Bernoulli neural network is simulated with its WDD and SAD algorithms, and the numerical results are shown comparatively in Table 1, Figs. 5 and 6.

Moreover, it is worth pointing out that, with the optimal number of hidden-layer neurons as in Table 1 (e.g., in its last row), the Bernoulli neural network can achieve the smallest BPE error E for that specific function-approximation task, which implies that its performance is deterministically the best. More importantly, by using the proposed weights-direct-determination and



(a) As of target function (5) (b) As of target function (6) (c) As of target function (7)

Fig. 6. Approximation error of Bernoulli neural net with WDD and SAD algorithms

structure-automatic-determination methods (instead of using iterative BP-training algorithms based on the well-known gradient-descent method [2,5,7,8,9]), the constructed Bernoulli neural network can find the optimal number of hidden-neurons and achieve the best performance in very short time (e.g., within 0.7s; see the third column of Table 1). Besides, the average errors of training and testing are both very tiny, which are shown in the rightmost two columns of Table 1 as well as in Fig. 6 so that we see little difference between the expected target output (denoted by blue dashed curves) and the neural-network output (denoted by black asterisk curves). This might reveal the excellent approximation ability of Bernoulli neural network trained by WDD and SAD algorithms [8,9,10]. Before ending this section, it is worth mentioning that, after training, we select 200 untrained points (with boundary-points unconsidered [17]) to test the generalization ability of the Bernoulli neural network, and the results are excellent which are shown in the last column of Table 1.

6 Conclusions

To remedy the weaknesses of conventional BP neural networks, a novel type of forward neural network is constructed by using the linearly-independent Bernoulli polynomials to be the hidden-layer activation functions. Different from using the conventional gradient-based methods [2,5,7,8,9] for BP iterative-training algorithms, we propose a pseudoinverse-based weights-direct-determination (WDD) method [9,10], which could now calculate the neural-network weights directly (just in one step and with no more lengthy/time-consuming BP iterative training procedure). Moreover, based on the weights-direct-determination method, a structure-automatic-determination algorithm is further proposed for the neural network, which could determine the optimal number of hidden-layer neurons much more effectively, rapidly and deterministically. Theoretical analysis and simulation results have both substantiated the efficacy and superiority of the proposed Bernoulli neural network and algorithms.

References

1. Zhang, Y., Wang, J.: Recurrent Neural Networks for Nonlinear Output Regulation. Automatica 37(8), 1161–1173 (2001)
2. Zhang, Y., Wang, J.: Global Exponential Stability of Recurrent Neural Networks for Synthesizing Linear Feedback Control Systems via Pole Assignment. IEEE Transactions on Neural Networks 13(3), 633–644 (2002)
3. Zhang, Y., Jiang, D., Wang, J.: A Recurrent Neural Network for Solving Sylvester Equation with Time-Varying Coefficients. IEEE Transactions on Neural Networks 13(5), 1053–1063 (2002)
4. Zhang, Y., Ge, S.S.: Design and Analysis of a General Recurrent Neural Network Model for Time-Varying Matrix Inversion. IEEE Transactions on Neural Networks 16(6), 1477–1490 (2005)
5. Steriti, R.J., Fiddy, M.A.: Regularized Image Reconstruction Using SVD and a Neural Network Method for Matrix Inversion. IEEE Transactions on Signal Processing 41(10), 3074–3077 (1993)
6. Zhang, Y., Ge, S.S., Lee, T.H.: A Unified Quadratic-Programming-Based Dynamical System Approach to Joint Torque Optimization of Physically Constrained Redundant Manipulators. IEEE Transactions on Systems, Man, and Cybernetics 34(5), 2126–2132 (2004)
7. Sadeghi, B.H.M.: A BP-Neural Network Predictor Model for Plastic Injection Molding Process. Journal of Materials Processing Technology 103, 411–416 (2000)
8. Demuth, H., Beale, M., Hagan, M.: Neural Network Toolbox 5 User's Guide. MathWorks Inc., Natick (2007)
9. Zhang, Y., Li, W., Yi, C., Chen, K.: A Weights-Directly-Determined Simple Neural Network for Nonlinear System Identification. In: Proceedings of IEEE International Conference on Fuzzy Systems, pp. 455–460. IEEE Press, Los Alamitos (2008)
10. Zhang, Y., Zhong, T., Li, W., Xiao, X., Yi, C.: Growing Algorithm of Laguerre Orthogonal Basis Neural Network with Weights Directly Determined. In: Huang, D.-S., Wunsch II, D.C., Levine, D.S., Jo, K.-H. (eds.) ICIC 2008. LNCS (LNAI), vol. 5227, pp. 60–67. Springer, Heidelberg (2008)
11. Mathews, J.H., Fink, K.D.: Numerical Methods Using MATLAB. Pearson Education Inc., Beijing (2004)
12. Mo, G., Liu, K.: Function Approximation Method. Science Press, Beijing (2003)
13. Costabile, F.A., Dell'Accio, F.: Expansion over a Rectangle of Real Functions in Bernoulli Polynomials and Application. BIT Numerical Mathematics 41(3), 451–464 (2001)
14. Costabile, F.A., Dell'Accio, F., Gualtieri, M.I.: A New Approach to Bernoulli Polynomials. Rendiconti di Matematica 26(7), 1–12 (2006)
15. Lin, C.S.: Numerical Analysis. Science Press, Beijing (2007)
16. Kincaid, D., Cheney, W.: Numerical Analysis: Mathematics of Scientific Computing. China Machine Press, Beijing (2003)
17. Leithead, W.E., Zhang, Y.: $O(N^2)$-Operation Approximation of Covariance Matrix Inverse in Gaussian Process Regression Based on Quasi-Newton BFGS Methods. Communications in Statistics - Simulation and Computation 36(2), 367–380 (2007)

Appendix: The 5th through 20th Bernoulli Polynomials

$$\phi_5(x) = x^5 - \frac{5}{2}x^4 + \frac{5}{3}x^3 - \frac{1}{6}x,$$

$$\phi_6(x) = x^6 - 3x^5 + \frac{5}{2}x^4 - \frac{1}{2}x^2 + \frac{1}{42},$$

$$\phi_7(x) = x^7 - \frac{7}{2}x^6 + \frac{7}{2}x^5 - \frac{7}{6}x^3 + \frac{1}{6}x,$$

$$\phi_8(x) = x^8 - 4x^7 + \frac{14}{3}x^6 - \frac{7}{3}x^4 + \frac{2}{3}x^2 - \frac{1}{30},$$

$$\phi_9(x) = x^9 - \frac{9}{2}x^8 + 6x^7 - \frac{21}{5}x^5 + 2x^3 - \frac{3}{10}x,$$

$$\phi_{10}(x) = x^{10} - 5x^9 + \frac{15}{2}x^8 - 7x^6 + 5x^4 - \frac{3}{2}x^2 + \frac{5}{66},$$

$$\phi_{11}(x) = x^{11} - \frac{11}{2}x^{10} + \frac{55}{6}x^9 - 11x^7 + 11x^5 - \frac{11}{2}x^3 + \frac{5}{6}x,$$

$$\phi_{12}(x) = x^{12} - 6x^{11} + 11x^{10} - \frac{33}{2}x^8 + 22x^6 - \frac{33}{2}x^4 + 5x^2 - \frac{691}{2730},$$

$$\phi_{13}(x) = x^{13} - \frac{13}{2}x^{12} + 13x^{11} - \frac{143}{6}x^9 + \frac{286}{7}x^7 - \frac{429}{10}x^5 + \frac{65}{3}x^3 - \frac{691}{210}x,$$

$$\phi_{14}(x) = x^{14} - 7x^{13} + \frac{91}{6}x^{12} - \frac{1001}{30}x^{10} + \frac{143}{2}x^8 - \frac{1001}{10}x^6 + \frac{455}{6}x^4 - \frac{691}{30}x^2 + \frac{7}{6},$$

$$\phi_{15}(x) = x^{15} - \frac{15}{2}x^{14} + \frac{35}{2}x^{13} - \frac{91}{2}x^{11} + \frac{715}{6}x^9 - \frac{429}{2}x^7 + \frac{455}{2}x^5 - \frac{691}{6}x^3 + \frac{35}{2}x,$$

$$\phi_{16}(x) = x^{16} - 8x^{15} + 20x^{14} - \frac{182}{3}x^{12} + \frac{572}{3}x^{10} - 429x^8 + \frac{1820}{3}x^6 - \frac{1382}{3}x^4 + 140x^2 - \frac{3617}{510},$$

$$\phi_{17}(x) = x^{17} - \frac{17}{2}x^{16} + \frac{68}{3}x^{15} - \frac{238}{3}x^{13} + \frac{884}{3}x^{11} - \frac{2431}{3}x^9 + \frac{4420}{3}x^7 - \frac{23494}{15}x^5 + \frac{2380}{3}x^3 - \frac{3617}{30}x,$$

$$\phi_{18}(x) = x^{18} - 9x^{17} + \frac{51}{2}x^{16} - 102x^{14} + 442x^{12} - \frac{7293}{5}x^{10} + 3315x^8 - \frac{23494}{5}x^6 + 3570x^4 - \frac{10851}{10}x^2 + \frac{43867}{798},$$

$$\phi_{19}(x) = x^{19} - \frac{19}{2}x^{18} + \frac{57}{2}x^{17} - \frac{646}{5}x^{15} + 646x^{13} - \frac{12597}{5}x^{11} + \frac{20995}{3}x^9 - \frac{446386}{35}x^7 + 13566x^5 - \frac{68723}{10}x^3 + \frac{43867}{42}x,$$

$$\phi_{20}(x) = x^{20} - 10x^{19} + \frac{95}{3}x^{18} - \frac{323}{2}x^{16} + \frac{6460}{7}x^{14} - 4199x^{12} + \frac{41990}{3}x^{10} - \frac{223193}{7}x^8 + 45220x^6 - \frac{68723}{2}x^4 + \frac{219335}{21}x^2 - \frac{174611}{330}.$$

Extension of Stochastic Dynamical System Involving Generalized Brownian Functionals

Lin Yu

College of Sciences, China Three Gorges University,
Yichang, Hubei 443000, P.R.China
yulin@ctgu.edu.cn

Abstract. Applying white noise analysis theory, an extension of the stochastic dynamical system involving generalized Brownian functionals and anticipating diffusion coefficient is proposed, moreover, as it has been shown in the proof of theorem 3.1 that the Picard's iteration works rather well in the U-functional method in the extended system.

Keywords: White noise calculus; U-functional; Generalized Brownian functional; Wick product.

1 Introduction

Usually, a stochastic dynamical system may be viewed as a stochastic differential equation where some of the coefficients are subject to white noise perturbations. Perhaps the most celebrated example is the diffusion system formulated by the Ito type stochastic deferential equation as below:

$$dX_t = b(t, X_t)dt + \sigma(t, X_t)dB_t, \tag{1}$$

where $B_t = B(t, \omega)$ denotes "white noise"(i.e. Brownnian motion), which represents random fluctuations due to changes in the environment.

Quite often, however, the nature of the noise is not "white" but biased in some sense. Applying Hida's white noise analysis theory as a framework, in this article, we set up a new model which extended the system of (1.1). In our extended stochastic system, not only the noise is not need to be "white"(in fact, it may be any generalized Brownian functional), but also the diffusion coefficient is not need to be non-anticipating. As well as, we point out that in the U-functional method the Picard's iteration works rather well.

2 White Noise Spaces

Since white noise is so fundamental for our construction, it is necessary to recall some basic facts about Hida's white noise calculus and one can refer to [1-3] for some more details.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 46–51, 2009.

Let $S(\mathbb{R})$ be the Schwartz space of real-valued rapidly decreasing functions on $\mathbb{R}$, Its dual space $S'(\mathbb{R})$ consists of the tempered distributions. Then we have get the following Gel'fand triple:

$$S(\mathbb{R}) \subset L^2(\mathbb{R}) \subset S'(\mathbb{R}).$$

Let μ be the standard Gaussian measure on $S'(\mathbb{R})$, i.e. its characteristic function is given by

$$\int_{S'(\mathbb{R})} e^{i<x,\xi>} d\mu(x) = e^{-\frac{1}{2}|\xi|_2^2}, \quad \xi \in S(\mathbb{R}),$$

where $< \cdot, \cdot >$ is the pairing of $S'(\mathbb{R})$ and $S(\mathbb{R})$, and $|\cdot|_2$ is the norm of $L^2(\mathbb{R})$. We call $(S'(\mathbb{R}), \mu)$ the white noise space.

Denote $(L^2) = L^2(\mathbb{R})$, for any $p \geq 0$, define the Sobolev space as bellow:

$$(S)_p = \left\{ \varphi \in (L^2) : \|\varphi\|_{2,p} = \|\Gamma(A)^p \varphi\|_2 < \infty \right\},$$

where $A = -(\frac{d}{dx})^2 + x^2 + 1$ and $\Gamma(A)$ is the operator of second quantization of A. Let $(S)_p^*$ be the dual space of $(S)_p$, and $(S)^*$ be the inductive limit of $\{(S)_p^* : p \geq 0\}$, (S) be the projective limit of $\{(S)_p : p \geq 0\}$. We have the following continuous inclusion maps:

$$(S) \subset (S)_p \subset (L^2) \subset (S)_p^* \subset (S)^*, \quad p \geq 0.$$

We will call (S) the space of test functionals and $(S)^*$ the space of Hida distributions(or generalized Brownian functional).

For any $\varphi \in (L^2)$, the following $\mathcal{S}$-transform was introduced by Hida(1980):

$$\mathcal{S}[\varphi](\xi) = e^{-\frac{1}{2}|\xi|_2^2} \int_{S'(\mathbb{R})} e^{<x,\xi>} \varphi(x) d\mu(x), \quad \xi \in S(\mathbb{R}).$$

Noticing, for any $\xi \in S(\mathbb{R})$, $exp(< \cdot, \xi >) \in (S)$, we can extend the $\mathcal{S}$-transform to $(S)^*$. We will call this extension the U-transform. For any $\Phi \in (S)^*$, its U-functional is defined to be

$$U[\Phi](\xi) = e^{-\frac{1}{2}|\xi|_2^2} \ll \Phi, e^{<\cdot,\xi>} \gg, \quad \xi \in S(\mathbb{R}),$$

where $\ll \cdot, \cdot \gg$ is the pairing of $(S)^*$ and (S). We can define the Wick product $\Phi \diamond \Psi$ of two Hida distributions Φ and Ψ as the Hida distribution with the U-functional given by

$$U[\Phi \diamond \Psi](\xi) = U[\Phi](\xi) \cdot U[\Psi](\xi), \quad \xi \in S(\mathbb{R}).$$

For any $\varphi \in (L^2)$, the Hida derivative $\vartheta_t \varphi$ of φ is defined as

$$\vartheta_t \varphi = \mathcal{S}^{-1} \frac{\delta}{\delta \xi(t)} \mathcal{S} \varphi, \quad \varphi \in (L^2),$$

where $t \in \mathbb{R}$ and $\frac{\delta}{\delta\xi(t)}$ is the Frchet derivative. Denote ϑ_t^* the adjoint operator of ϑ_t, the following integration is called Hitsuda-Skorohod integration:

$$\int_0^t \vartheta_t^* \varphi(s) ds, \quad \varphi \in L^2(\mathbb{R}, (L^2)).$$

where $\varphi(t)$ is not need to be non-anticipating.

3 Hitsuda-Skorohod Type Model with Anticipating Diffusion

Lemma 1. [2] Suppose $\varphi(t)$ is non-anticipating and $E(\int_0^1 \varphi(t)^2 dt) < \infty$. Then

$$\int_0^1 \varphi(t) dB(t) = \int_0^1 \vartheta_t^* \varphi(t) dt. \tag{2}$$

Remark. The left hand side of (2) is Ito integral. Lemma 1 shows that the Hitsuda-Skorohod integral is an extension of Ito integral and the integrand function in the right hand side is not need to be non-anticipating.

In the sense of Hitsuda-Skorohod integral, the model (1) can be extended as the following, where the diffusion is not need to be non-anticipating.

$$\frac{dM^*(t,\omega)}{dt} = f(t, M^*(t,\omega)) + \vartheta_t^* \sigma(t, M^*(t,\omega)) \quad \text{with} \quad M^*(0,\omega) = M_0^*. \tag{3}$$

Equivalently, (3) can be written as below:

$$M^*(t,\omega) = M_0^* + \int_0^t f(s, M^*(s,\omega)) ds + \int_0^t \vartheta_s^* \sigma(s, M^*(s,\omega)) ds. \tag{4}$$

We will point out that in the U-functional method, the Picard's iteration works rather well for the model (4).

Theorem 1. In model (4), let $f, \sigma : [0,1] \times (S)^* \to (S)^*$ satisfy the following conditions:

(i) For any $x \in \mathbb{R}$ and $\xi \in S(\mathbb{R})$, we have that

$$\int_0^1 |U[f(t,x)](\xi)| dt < \infty, \qquad \int_0^1 |U[\sigma(t,x)](\xi)| dt < \infty.$$

(ii) For any $\Phi \in (S)^*$, there exist some $p \geq 1$ such that $f(t,\Phi)$, $\sigma(t,\Phi) \in (S)_p^*$, for all $t \in [0,1]$.

(iii) There exist constants $C_1, C_2 > 0$ such that for any $\Phi_1, \Phi_2 \in (S)^*$, $\xi \in S(\mathbb{R})$ and $t \in [0,1]$, we have

$$|U[f(t,\Phi_1)](\xi) - U[f(t,\Phi_2)](\xi)|^2 \leq C_1^2 |U[\Phi_1](\xi) - U[\Phi_2](\xi)|^2,$$

$$|U[\sigma(t,\Phi_1)](\xi) - U[\sigma(t,\Phi_2)](\xi)|^2 \leq C_1^2 |U[\Phi_1](\xi) - U[\Phi_2](\xi)|^2.$$

Then there exists uniquely a weak solution of the integral equation (4) in the space $(S)^*$ of generalized Brownian functionals.

Proof. At first, we prove the uniqueness. Suppose that both M_t^* and $\widetilde{M}_t^*$ in $(S)^*$ are the solutions of (4). Denote F_t and $\widetilde{F}_t$ are the U-functionals of M_t^* and $\widetilde{M}_t^*$, respectively. From the Lipschitz condition of (iii), we have that

$$\begin{aligned}
|F_t(\xi) - \widetilde{F}_t(\xi)| &\le \left| \int_0^t U[f(s, M_s^*)](\xi)ds - \int_0^t U[f(s, \widetilde{M}_s^*)](\xi)ds \right| \\
&\quad + \left| \int_0^t \xi(s)U[\sigma(s, M_s^*)](\xi)ds - \int_0^t \xi(s)U[\sigma(s, \widetilde{M}_s^*)](\xi)ds \right| \\
&\le C_1 \int_0^t |U[M_s^*](\xi) - U[\widetilde{M}_s^*](\xi)|ds \\
&\quad + C_2|\xi|_\infty \int_0^t |U[M_s^*](\xi) - U[\widetilde{M}_s^*](\xi)|ds \\
&= (C_1 + C_2|\xi|_\infty) \int_0^t |F_s(\xi) - \widetilde{F}_s(\xi)|ds.
\end{aligned}$$

where $|\xi|_\infty = \sup|\xi(t)| : 0 \le t \le 1$. By Gronwall's inequality, we have $F_t(\xi) = \widetilde{F}_t(\xi)$ for any $\xi \in S(\mathbb{R})$, This means that $M_t^* = \widetilde{M}_t^*$. Now we define

$$M_t^{*(0)} = M_0^*, \quad M_t^{*(n+1)} = M_0^* + \int_0^t f(s, M_s^{*(n)})ds + \int_0^t \vartheta_s^* \sigma(s, M_s^{*(n)})ds, \quad n \ge 0.$$

Denote $F_t^{(n)}$ the U-functional of $M_t^{*(n)}$, then

$$F_t^{(0)} = M_0^*,$$

$$F_t^{(n+1)}(\xi) = M_0^* + \int_0^t U[f(s, M_s^{*(n)})](\xi)ds + \int_0^t \xi(s)U[\sigma(s, M_s^{*(n)})](\xi)ds, \quad n \ge 0.$$

From Lipschitz condition (iii), we have

$$|F_t^{(n+1)}(\xi) - F_t^{(n)}(\xi)| \le (C_1 + C_2|\xi|_\infty) \int_0^t |F_{t_1}^{(n)}(\xi) - F_{t_1}^{(n-1)}(\xi)|dt_1.$$

So, for any $t \in [0, 1]$ and $\xi \in S(\mathcal{R})$, we have

$$\begin{aligned}
&|F_t^{(n+1)}(\xi) - F_t^{(n)}(\xi)| \\
&\le (C_1 + C_2|\xi|_\infty)^n \int_0^t \int_0^{t_1} \cdots \int_0^{t_{n-1}} |F_{t_n}^{(1)}(\xi) - F_{t_n}^{(0)}(\xi)|dt_n dt_{n-1} \cdots dt_1 \\
&\le (n!)^{-1}(C_1 + C_2|\xi|_\infty)^n t^n \left(\int_0^1 |U[f(s, M_0^*)](\xi)|ds + |\xi|_\infty \int_0^1 |U[\sigma(s, M_0^*)](\xi)|ds \right).
\end{aligned}$$

Consequently,

$$\begin{aligned}
&\sup_{0 \le t \le 1} |F_t^{(n+1)}(\xi) - F_t^{(n)}(\xi)| \\
&\le (n!)^{-1}(C_1 + C_2|\xi|_\infty)^n \left(\int_0^1 |U[f(s, M_0^*)](\xi)|ds + |\xi|_\infty \int_0^1 |U[\sigma(s, M_0^*)](\xi)|ds \right).
\end{aligned}$$

By condition (i), we get that $\{F_t^{(n)}(\xi) : n \geq 0\}$ is convergent uniformly on $[0,1]$ for any $\xi \in S(\mathbb{R})$. Set $F_t(\xi) = \lim_{n\to\infty} F_t^{(n)}(\xi)$ and note that

$$F_t^{(n)}(\xi) = e^{-\frac{1}{2}|\xi|_2^2} \ll M^{*(n)}, e^{<\cdot,\xi>} \gg .$$

So

$$\lim_{n\to\infty} \ll M_t^{*(n)}, e^{<\cdot,\xi>} \gg = F_t(\xi) e^{-\frac{1}{2}|\xi|_2^2}.$$

Since $span\{e^{<\cdot,\xi>} : \xi \in S(\mathbb{R})\}$ is dense in (S), then we have proved that $\{M_t^{*(n)} : n \geq 0\}$ is convergent uniformly in a dense subspace of (S).

On the other hand, from condition (ii), there exist constants $D_1, D_2 > 0$, such that

$$|F_t^{*(n)}(\xi)| \leq D_1 exp(D_2|\xi|_{2,p}),$$

this means that there exist constants $q > 0$ and $L > 0$, such that

$$\|M_t^{*(n)}\|_{2,-q} \leq L, \quad \forall t \in [0,1], \quad n \geq 0,$$

where $\|\cdot\|_{2,-q}$ is the norm on Sobolev space $(S)_q^*$. Then $\{M_t^{*(n)} : n \geq 0\}$ is convergent weakly in $(S)^*$ with limit M_t^*, that is

$$\lim_{n\to\infty} \ll M_t^{*(n)}, \varphi \gg = \ll M_t^*, \varphi \gg, \qquad \varphi \in S(\mathbb{R}),$$

or

$$F_t(\xi) = e^{-\frac{1}{2}|\xi|_2^2} \ll M_t^*, e^{<\cdot,\xi>} \gg \qquad \varphi \in S(\mathbb{R}).$$

This means that F_t is the U-functional of M_t^* and we have

$$F_t(\xi) = M_0^* + \int_0^t U[f(s, M_s^*)](\xi)ds + \int_0^t \xi(s) U[\sigma(s, M_s^*)](\xi)ds.$$

That is

$$M_t^* = M_0^* + \int_0^t f(s, M_s^*)ds + \int_0^t \vartheta_s^* \sigma(s, M_s^*)ds.$$

4 Model Involving Generalized White Noise Functional

Theorem 2. Let $\Psi \in (S)^*$, then $\vartheta_t^* \Psi = B'(t) \diamond \Psi$, where $B(t,\omega) = <\omega, I_{[0,t]}>$, $t \geq 0$, $\omega \in S'(\mathbb{R})$ is Brownian motion, $B'(t)$ is the Hida derivative of $B(t)$, $\diamond$ is the Wick production.

Proof. For any $\Phi_1, \Phi_2 \in (S)^*$, $\xi \in S(\mathbb{R})$, we have

$$U[\Phi_1 \diamond \Phi_2](\xi) = U[\Phi_1](\xi) \cdot U[\Phi_2](\xi).$$

Then

$$U[B'(t) \diamond \Psi](\xi) = U[B'(t)](\xi) \cdot U[\Psi](\xi).$$

Note that $U[B'(t)](\xi) = \xi(t)$, and then

$$U[B'(t) \diamond \Psi](\xi) = \xi(t) \cdot U[\Psi](\xi), \quad \xi \in S(\mathbb{R}).$$

On the other hand,

$$U[\vartheta_t^* \Psi] = \xi(t) \cdot U[\Psi](\xi), \quad \xi \in S(\mathbb{R}).$$

Consequently,

$$U[\vartheta_t^* \Psi] = U[B'(t) \diamond \Psi](\xi) \quad \xi \in S(\mathbb{R}).$$

That is $\vartheta_t^* \Psi = B'(t) \diamond \Psi$.

From Theorem 2, we have $\vartheta_t^* \sigma(s, M^*(s)) = B'(s) \diamond \sigma(s, M^*(s))$, this means that (4) is equal to

$$M^*(t) = M_0^* + \int_0^t f(s, M^*(s))ds + \int_0^t B'(s) \diamond \sigma(s, M^*(s))ds. \tag{5}$$

Note that $B'(t) \in (S)^*$, for any generalized white noise functional $\theta(t) \in (S)^*$, the integral $\int_0^t \theta(s) \diamond \sigma(s, M^*(s))ds$ makes sense. Then the model can be extended as the following form:

$$M^*(t) = M_0^* + \int_0^t f(s, M^*(s))ds + \int_0^t \theta(s) \diamond \sigma(s, M^*(s))ds. \tag{6}$$

Remark. Since $\theta(s)$ in model (6) may be any generalized white noise functional, so the white noise which results in the random perturbations is not need to be "white" in our extended system.

Similarly, we can prove the following result.

Theorem 3. In model (6), suppose that f and σ satisfy the conditions (i)-(iii) in theorem 1 and $\theta : [0, 1] \to (S)^*$ satisfy:

(A) $\sup_{0 \le t \le 1} |U[\theta(t)]|(\xi) < \infty, \quad \xi \in S(\mathbb{R})$;

(B) There exists $q \ge 1$, such that for all $0 \le t \le 1$, $\theta(t) \in (S)_q^*$.

Then there exist uniquely a weak solution of the integral equation (6) in the space $(S)^*$.

Acknowledgments. Supported by the Scientific Research Foundation of Hubei Province (D200613001).

References

1. Hida, T.: Brownian Motion. Springer, Heideberg (1980)
2. Hida, T., Kuo, H.H., Potthoff, J., Streit, L.: White Noise: An Infinite Dimensional Calculis. Kluwer Academic Publ., Dordrecht (1993)
3. Huang, Z.Y., Yan, J.A.: Introduction to Infinite Dimensional Stochastic Analysis. Science Press/Kluwer Academic Publ., Dordrecht (2000)
4. Sun, C.S., Gao, H.J.: Exact Solutions for a Variable Coefficient and Wick-Type Stochastic Long-Short Wave Resonance Equation. Commun. Nonlinear Sci. Numer. Simulat. 14, 1551–1560 (2009)

Information Extraction System Based on Hidden Markov Model

Dong-Chul Park[1], Vu Thi Lan Huong[1], Dong-Min Woo[1], Duong Ngoc Hieu[2], and Sai Thi Hien Ninh[2]

[1] Dept. of Information Engineering, Myong Ji University, Korea
[2] Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology, Vietnam

Abstract. A novel approach using Hidden Markov Model (HMM) for the task of finding prices of products on internet sites is proposed in this paper. The proposed Information Extraction System based on HMM (IESHMM) utilizes HMM for its capability to process temporal information. The proposed IESHMM first processes web pages that are returned from search engines and then extracts specific fields such as prices, descriptions, locations, images of products, and other information of interest. The proposed IESHMM is evaluated with real-world problems and compared with a conventional method. The results show that the proposed IESHMM outperforms the other method by 22.9 % and 37.2% in terms of average recall and average precision, respectively.

1 Introduction

Information extraction can be defined as a process of filling fields in a database by automatically extracting information or knowledge from unstructured or semi-structured documents. Web pages are not well structured and there is no schema to describe the contents of web pages. Models can be largely categorized into three classes for information extraction (IE): dictionary-based models[1], rule-based models[2], and statistics-based models [3]. When compared with many other techniques used in statistics-based methods, the Hidden Markov Model (HMM) has a strong theoretical foundation with a well-established training algorithm and HMM can process data quite robustly. In fact, HMMs have been successfully applied to information extraction problems including various speech recognition tasks. The HMM implemented by Leek [3] was intricately designed for the task of extracting (gene name and chromosome location) pairs from scientific abstracts in the medical domain. The success of a HMM lies in its dependence on parameter estimation algorithms that allows training of data-dependent parameters.

The process of extracting information from the result pages yielded by a search engine is termed as web information extraction. Earlier studies relied on training and human assistance or generating extraction rules for web pages. Recently, several automated or nearly automated IE methods have been proposed

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 52–59, 2009.

as follows: Product extraction from the Web (PEWEB) [6], Object mining and extraction system (OMINI) [8] and Information extraction based on Pattern Discovery (IEPAD) [7]. IEPAD parses the HTML content of the web pages into a string and then uses a Patricia(PAT) tree with some heuristics to find the candidate patterns. The best pattern can then be chosen by the human user. Thus, IEPAD is a near-automated extraction method. This method shows a limited performance due to limitations of the PAT tree. OMINI, meanwhile, builds a tag tree from an input web page, and then applies some heuristics to extract a subtree that may contain the data records of interest. A separator that is found by another set of heuristics is used to segment the sub-tree into data records. The separator contains only one HTML tag. However, it is insufficient to separate the data region to extract the data record. Rather than building a HTML tree as in the other approaches, PEWEB takes advantage of entropy measurement to identify a product region containing potential products. The product description results are the sub-tree node that has a high entropy ratio.

In this paper, we propose an Information Extraction System based on HMM, called IESHMM, to extract prices of goods on the internet. From a large number of web pages relevant to prices of goods appearing on the internet, the proposed approach can help to extract the prices of goods of interest with maximal accuracy. When the numerous prices on web pages regarding a good of interest can be transformed into structured data and stored in a database, users will be able to obtain the information accurately.

The remainder of this paper is organized as follows: A brief review of HMM is presented in Section 2. Section 3 presents the proposed IESHMM. Section 4 describes an actual implementation of IESHMM and experiments involving a practical price extraction problem and presents results including comparisons with a conventional PEWEB algorithm. Finally, Section 5 concludes the paper.

2 Hidden Markov Model

A Hidden Markov Model (HMM) is a statistical model in which the system is assumed to be a Markov process with unknown parameters and the hidden parameters are found from the observable parameters. In a HMM, the state is not directly visible, but variables influenced by the state are visible. Each state has a probability distribution. HMMs are especially useful for their applications in temporal pattern recognition including speech recognition.

A discrete output, a first order HMM, is a finite state automaton and can be represented by a 5-tuple {S, V, Π , A, B} where S is a set of values for the hidden states, $S = \{s_1...s_N\}$, N is the total number of possible states; V is a set of values for the observations $V = \{v_1...v_N\}$, M is the total number of possible observation values; Π is a set of initial probabilities for all the states, $\Pi = \{\pi_i\}$, i = 1, 2, ..., N; A is a mapping defining the probabilities of each state transition $A = \{P(q \rightarrow q')\}$;and B is a mapping defining the emission probability of each observation value on each state, $B = \{P(q \uparrow \sigma)\}$ [10].

There are three basic problems to solve for a HMM:

- **Problem 1:** Given the observation sequence, $O = O_1O_2...O_T$, and a model, $\lambda = (A, B, \Pi)$, how to efficiently compute $P(O|\lambda)$, the probability of the observation sequence, given the model.
- **Problem 2:** Given the observation sequence, $O = O_1O_2...O_T$, and a model, $\lambda = (A, B, \Pi)$, how to choose a corresponding state sequence, $Q = q_1q_2...q_T$, which is optimal to generate the observation sequence. The optimal measure can be the maximum likelihood.
- **Problem 3:** How to adjust the parameters $\lambda = (A, B, \Pi)$ to maximize the likelihood of all observation sequences.

In this paper, we mainly focus on Problem 3 for training model parameters and Problem 2 for extracting data records.

3 Information Extraction System Based on HMM

The proposed IESHMM shown in Fig. 1 consists of five components such as Page retrieval, Segmentation and Parser, Segment filter, Observation creator, and Extractor.

The input of the proposed IESHMM is the URLs of a search engine's interface, which contains the names of the product types. The output of the system is the list of extracted slots of each product: name, price, image, and URL. Usually a web page is a HTML document that contains a sequence of tag delimited data elements. The HTML segment tree is built by the nested blocks of the HTML tags in the web page. A segment is a group of elements and is used as a unit

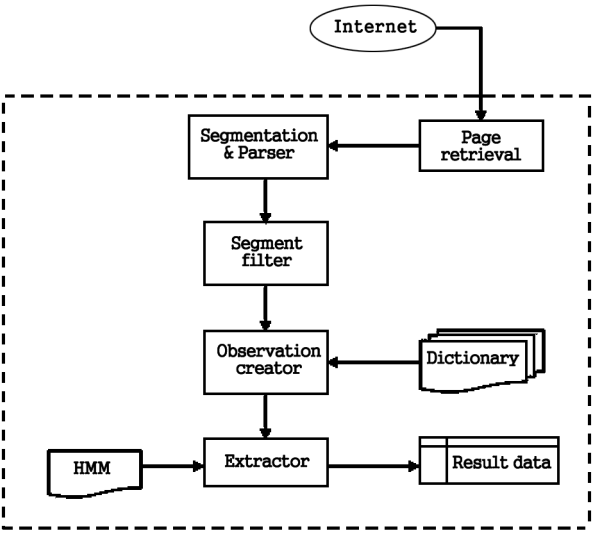


Fig. 1. Overview of the proposed information extraction system

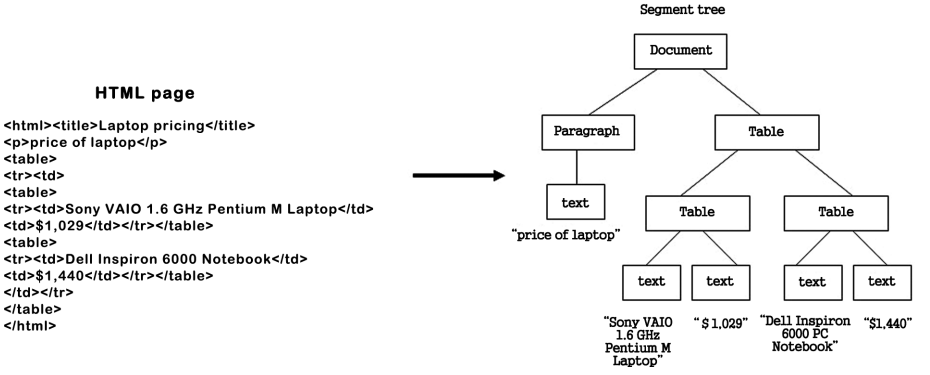


Fig. 2. A HTML document and the corresponding segment tree

in our model. Four major segments are paragraph, table, list, and heading. A HTML document is the largest segment. An example of a HTML page and the corresponding segment tree is shown in Fig. 2. The product region is identified by finding the existence of multiple similar nodes of a tag node with the following properties:

- All nodes have the same parent.
- The nodes are adjacent.
- The node contains the name of the product or the currency units.

The segment filter decides which segment belongs to the product segments. The adjacent segments containing product names, description, images, and prices are then grouped into a larger segment for the observation creator. For the purpose of IE, a segment is represented as an input observation for a trained HMM. Each type of product has a different HMM, which has the same HMM structure but different initial and transition probabilities. Extracted slots are modelled by target states. The most probable state token is then found by using the Viterbi algorithm [10]. The HMM employed in this approach is used to extract multiple slots simultaneously while the HMM adopted in [9] uses a separate HMM for each slot.

In order to train the HMM model for each product of interest (for example, laptop, USB flash drive, web camera, and computer mouse), observations that consist of 100 HTML pages obtained by the returns from a commercial search engine are used as a training data set. The maximum likelihood (ML) algorithm is generally used to label the training data and the Baum-Welch algorithm is used for partially labelled or unlabelled training data. Three parameters are required in the training process: the initial probability π, the transition probability matrix A, and the emission probability matrix B.

$$\pi_i = \frac{I(i)}{\sum_{j=1}^{N} I(j)}, \quad 1 \le i \le N. \tag{1}$$

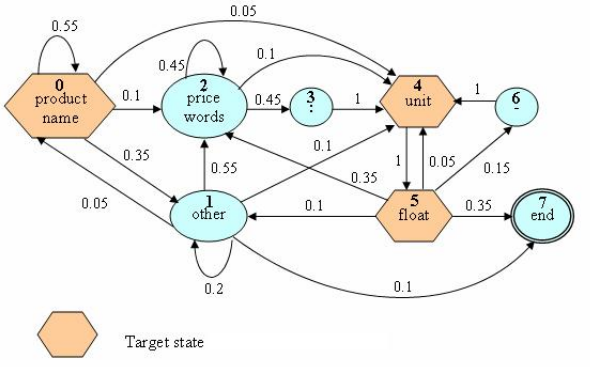


Fig. 3. A trained HMM for extracting usb flash drive problem

where $I(\cdot)$ is the initial probability that a sequence starts from a specific state and N is the number of states included in the model.

Each element a_{ij} in the transition probability matrix A is defined as follows:

$$a_{ij} = \frac{C_{i,j}}{\sum_{k=1}^{N} C_{i,k}}, \ 1 \le i, j \le N, \tag{2}$$

where $C_{i,j}$ is the number of transitions that transmit from state $S(i)$ to state $S(j)$.

The emission probability matrix element $b_j(V_k)$ is defined as follows:

$$b_j(V_k) = \frac{E_j(V_k)}{\sum_{i=1}^{M} E_j(V_i)}, \quad 1 \le j \le N, \ \ 1 \le k \le M, \tag{3}$$

where $E_j(V_k)$ is the number of times that one token V_k is emitted at specific S(j).

The trained HMM used in extracting " usb flash drive" problem is illustrated in Fig. 3. The emission probability of one observation is the sum of emission probabilities of each token in the block. That is,

$$b_j(O_t) = \sum_{k=1}^{K} b_j(O_{tk}), \tag{4}$$

where K is the number tokens in the observation.

The Viterbi algorithm is then applied to find the state sequence of the maximal probability that matches the observation block sequence.

4 Experiments and Results

In this section, we demonstrate the performance of the proposed IESHMM using real world data in comparison with a conventional system, PEWEB, which is

a state-of-art web information extraction system based on HTML tag structure analysis.

The data set for training each product contains observations of 100 URLs returned from a general-purpose search engine, Google, and the next 100 URLs are prepared for testing. Some typical web sites with sufficient information about product features and prices are listed in Table 1 while other web sites with irrelevant features are omitted. The trained HMM used in the performance evaluation includes 8 states for labeling.

4.1 Measurement for Performance Evaluation

The performances are evaluated in terms of the precision and recall [4], which are widely used to evaluate information retrieval and extraction systems. These are defined as follows:

$$\text{Precision} = \frac{CE}{EO}, \qquad \text{Recall} = \frac{CE}{CO},$$

where CE is the total number of correctly extracted observations, EO is the total number of extracted observations on the page, and CO is the total number of correct observations (target observations). Precision defines the correctness of the data records identified while recall is the percentage of the relevant data records identified from the web page.

4.2 Performance Evaluation

The proposed IESHMM is evaluated and compared with PEWEB [5] on a real-world problem. In PEWEB, a product description or a data record is considered as a block containing both the product information and the noisy information. This implies that it contains not only useful information such as : name, manufacturer, price, images, and descriptions, but also advertisements, links, navigation components, etc. In some cases, an extracted data record contains two different products. Thus, a genuine data record can not be properly produced from the extraction. That is, it may not be advisable to save a genuine data record from the extraction in separate slots in the database for later use. Unlike PEWEB, the proposed IESHMM can handle the web pages deeper, because it is able to extract the specific field of the data record: image, name, price, and URL without including noisy objects.

Table 1 illustrates results of the performance comparison between the proposed IESHMM and PEWEB. A total of 18 web sites containing different formats and product information are evaluated in this experiments. In Table 1, column 2 lists the URL of each site. Due to long URLs of most pages, some details are omitted. The listed web pages are returned from a commercial search engine, Google, and the web pages include sufficient information about product features including prices, images, and descriptions regarding the specific products such as usb flash drive, laptop, web camera, computer mouse, etc. In our experiments, some products related to computers are used as sample queries to

Table 1. Performance Comparison between IESHMM and PEWEB

No.	web sites	Product	IESHMM		PEWEB	
			Rtn	Correct	Rtn	Correct
1	www.flash-memory-store.com	21	16	16	26	20
2	www.tigerdirect.com	19	16	16	11	2
3	www.supermediastore.com	32	30	28	38	29
4	www.usbflashdrivestore.com	25	24	24	27	24
5	www.buy.com	15	14	14	9	0
6	www.ecost.com	25	22	22	25	25
7	www.overstock.com	11	10	10	14	10
8	usbflashstore.com	25	24	24	27	25
9	shopping.aol.com	16	16	16	9	7
10	www.pricespider.com	10	10	10	15	0
11	www.usanotebook.com	27	21	21	28	27
12	www.dealtime.com	21	17	17	16	5
13	www.geeks.com	28	12	12	55	28
14	www.nextag.com	11	10	10	15	0
15	www.kenzo.com	16	13	13	20	16
16	www.mysimon.com	25	21	21	0	0
17	www.pricewatch.com	15	15	15	33	15
18	computing.kelkoo.co.uk	20	17	17	5	5
Total		362	308	306	378	238
Recall(Rc) and Precision(Pr)			Rc : 84.5% Pr : 99.3%		Rc : 65.7% Pr : 62.1%	

evaluate the performance of information extraction systems. Column 3 shows the number of target products available in the corresponding URL. Columns 4 and 5 are the number of extracted products and correctly extracted products of the proposed IESHMM, respectively. The next 2 columns are the results of the PEWEB system.

As shown in Table 1, the average recall obtained by the proposed IESHMM is 84.5% while recall of 65.7% is obtained by PEWEB. The improvement of IESHMM over PEWEB in terms of the average recall is 18.8%. With respect to the extraction precision, the proposed IESHMM proves to be a more powerful tool for the information extraction. The average precision obtained by the proposed IESHMM is 99.3% while precision of 62.1% is obtained by PEWEB. The improvement of IESHMM over PEWEB in the average precision is 37.2%.

5 Conclusion

In this paper, a novel and effective information extraction method using HMM to extract product prices from internet is proposed. This method can correctly identify the data region containing a product record. When a data region consists of only one data record, most of the conventional algorithms fail to correctly identify the data region. Furthermore, most web pages fail when one or more

advertisements separate the series of data records to two or more data records. The proposed IESHMM works correctly in both of the above cases. A vital advantage of the proposed IESHMM method is that the extracted data record is put into a suitable collection format that can be easily stored in a relational database. This collection data can be further used for various knowledge discovery applications. The proposed IESHMM method overcomes the drawbacks of the conventional PEWEB in processing HTML contents. Experiments show that the proposed IESHMM outperforms the PEWEB method significantly in terms of precision and recall.

Acknowledgement

This work was supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korean government (MOST)(Grant No.: R01-2007-000-20330-0).

References

1. Riloff, E.: JonesR.: Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping. In: Proceedings of the Sixteenth National Conference on Artificial Intelligence, pp. 811–816 (1999)
2. Sonderland, S.: Learning Information Extraction Rules for Semi-Structured and Free Text. Machine Learning 34, 233–272 (1999)
3. Leek, T.R.: Information Extraction Using Hidden Markov Models. Master thesis, UC san Diego (1997)
4. Raghavan, V.V., Wang, G.S., Bollmann, P.: A Critical Investigation of Recall and Precision as Measures of Retrieval System Performance. ACM Trans. Information System 7(3), 205–229 (1989)
5. http://www.jaist.ac.jp/~hieuxuan/softwares/peweb/
6. Phan, X.H., Horiguchi, S., Ho, T.B.: PEWEB: Product Extraction from the Web Based on Entropy Estimation. In: Proc. of the 2004 IEEE/WIC/ACM International Conference on the Web Intelligence, pp. 590–593 (2004)
7. Chang, C., Lui, S.: IEPAD: Information Extraction Based on Pattern Discovery. In: Proc. of World Wide Web conference, pp. 682–688 (2001)
8. Buttler, D., Liu, L., Pu, C.: A Fully Automated Extraction System for the World Wide Web. In: Proc.of IEEE ICDCS, vol. 361 (2001)
9. Freitag, D., McCallum, A.: Information Extraction with HMMs and Shrinkage. In: Proc. of AAAI Workshop on Machine Learning for IE, pp. 31–36 (1999)
10. Rabiner, L.R.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proc. of IEEE 77(2), 57–286 (1989)

Network Intrusion Detection with Workflow Feature Definition Using BP Neural Network

Yong Wang[1,2], Dawu Gu[1], Wei Li[1], Hongjiao Li[2], and Jing Li[2]

[1] Department of Computers Science and Enginerring, Shanghai Jiao Tong University, 200240 Shanghai, China
[2] Department of Computers Science and Technolgy, Shanghai University of Electric Power, 20090 Shanghai, China
wy616@126.com

Abstract. The major problem of existing intrusion detection using neural network models is recognition of new attacks and low accuracy. The paper describes an intrusion detection method based on workflow feature definition according to KDD cup 99 types with feed forward BP neural network. The workflow can define new attacks sequence to help BP neural network recognize new attacks. The method takes network traffic data to analyze and classify the behaviors of the authorized users and recognize the possible attacks. The experiment results show that the design is effective.

Keywords: IDS, BP, Neural Network, Workflow.

1 Introduction

Network intrusion detection is an important area in computer security in recent years. It is difficult for most of the currently available network security techniques to cope with the dynamic and increasingly complex nature of the attacks on distributed computer systems. Intrusion Detection System (IDS) still faces some challenges such as low detection rates, high false alarm rates and requirement of heavy computational power. The major problem of existing models is recognition of new attacks, low accuracy and system adaptability.

To overcome these difficulties, there are lots of techniques for intrusion detection by Neural Networks. Intrusion detection system using evolutionary neural networks can predict attacks in a network environment [2]. Strategy of dynamic change learning rate value in BP neural network improves the learning rate in BP network [3]. Fuzzy adaptive resonance theory neural network and rule heuristics are used to build a model of intrusion detection in database security management [4]. Intelligent methods for IDS are hot spots in the field of network security. The genetic neural network based on the genetic algorithm is applied in IDS [5]. The method of recognition of attack class combines principal component neural networks with multilayer perceptrons [6]. Intrusion detection system based on self-organizing maps and back propagation network is used for visualizing and classifying intrusion [7]. Innovative machine of learning algorithm utilizes semi-parametric learning model and adaptive

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 60–67, 2009.

boosting techniques to reduce learning bias and generalization variance in difficult classification [8]. Feature selection is important for intrusion detection system. There are some methods of identifying intrusion and normal pertinent features and evaluating the applicability of these features [9]. Multilayer perceptions for classification of network intrusion detection data are characterized by skewed class distributions. Investigated methods include oversampling, under sampling and generating artificial data records [10].

This paper describes an intrusion detection method based on workflow feature definition with feed forward BP neural network. The workflow can define new attacks sequence to help BP neural network to recognize new attacks. The method takes network traffic data to analyze and classify the behaviors of the authorized users and recognize the possible attacks.

2 Dataset Description

2.1 Attack Types from KDD 99

Many researchers use the datasets in the KDD Cup 99 intrusion detection competition to evaluate machine learning techniques. The KDD Cup 99 intrusion detection datasets are based on the 1998 DARPA initiative. The 1998 DARPA Intrusion Detection Evaluation Program was prepared and managed by MIT Lincoln Labs. They set up an environment to acquire nine weeks of raw TCP dump data for a local-area network simulating a true Air Force environment and peppering it with multiple attacks. Additional machines are used to generate traffic and a sniffer that records all network traffic using the TCP dump format. The simulation runs for seven weeks.

A connection is a sequence of TCP packets starting and ending at some well defined times, between which data flows to and from a source IP address to a target IP address under some well defined protocol. Each connection is labeled as either normal, or as an attack, with exactly one specific attack type. The traffic is made up by normal connections and attacks that fall into one of five categories:

Probing: surveillance and other probing, preparing the attack, for example: ping sweeping, port scan, probing the FTP, SMTP, HTTP service ICMP scanning and so on.

Denial of Service (DOS): clogs up so much memory on the target system that cannot serve its users, for example: ping of death, teardrop attacks, SYN-Flood attacks, Smurf attacks, UDP-Flood attacks, Distributed DOS attacks and so on.

Remote to Local (R2L): unauthorized access from a remote machine, for example: guessing password, restarting your computer, downloading your files, control your computer and so on.

User to Root (U2R): unauthorized access to local super user (root) privileges, for example: buffer overflow attacks, rootkit Trojan and so on.

Other Type: including unknown attack types and reserved attack types. The training attack type is shown as table 1:

Table 1. Training attack types with unknown feature from KDD cup 99

DOS	R2L	U2R	Probing	Other Type
back	ftp_write	buffer_overflow	ipsweep	mobile attack
land	guess_passwd	loadmodule	nmap	new attack
neptune	imap	perl	portsweep	unknown attack
pod	multihop	rootkit	satan	reserved attack
smurf	phf			
teardrop	spy			
	warezclient			
	warezmaster			

2.2 Input Vectors with Workflow Feature Definition

The BP algorithm needs a set of examples of network behaviour. Where vector P is an input to the network.

$$a^1 = \log sig\left(IW^{1,1} p + b^1\right) \tag{1}$$

The output vector is represented by a in equation 1. The p is multiplied by the matrix weight w to form wp and added by bias b . The transfer function is $logsig$.

The length of input vector P is 64 bits. The vector P includes four parts as follows:

State: record invader attack state, for example: TCP, UDP, ICMP, SYN_RECEIVED, ESTABLISHED, packet size and so on.

Intention: recording invader intention action, maybe not happen actually and maybe will really happened, for example: download, key record, upload, inserting remote thread and so on.

Workflow: recording invader's actually attack steps. It must be happened, for example: attack_ID with prior pointer and next pointer. The workflow is multi structure with double-link node.

Error Message: recording computer's error message caused by invader to help IDS find potential invade, for example: network unreachable, host unreachable, time exceed and so on. The input vector P is shown as table 2:

Table 2. Input vectors according to attack workflow feature definition

State(16bits)	Intention (16bits)	Workflow(16bits)	Error message(16bits)
TCP/UDP/ICMP	Download intention	attack head	Network unreachable
syn_received	Key record intention	Prior attack pointer	Host unreachable
established	Upload intention	Attack_ID_1	Proto unreachable
packet size	Insert thread	next attack pointer	Source route failed
packet sequence	Register Table	Prior attack pointer	Source host isolated
port	IDT hooking	Attack_ID_2	Time exceed
address type	SSDT hooking	next attack pointer	Bad length
Other state	Other intention	Other workflow	Other error message

2.3 Attack Workflow Feature Definition Analysis with Windows Workflow

The workflow structure is a double-linked list. The structure is helpful when you want to build the attack event sequence. It's easily to find the prior attack event or next attack event while you locate one of the attack events. You can also build binary tree structure or crossing double-linked list. Different structures indicate different type attack. When the input vectors can give accurate description of attack event, the neural network can get high recognize percentage.

We use Windows Workflow Foundation (WF) to help us analysis the workflow attack feature definition. WF is part of the .NET Framework 3.5 for Visual Studio 2008. WF makes it possible, and for many workflow scenarios, even easy to create robust, manageable workflow-based applications.

WF allows you to create two different types of workflows: sequential, and state machine. Sequential workflows provide a structured series of steps in which one activity leads to another, and steps generally occur immediately one after another. State machine workflows provide a set of states; transitions between the states define the behavior. In general, state machine workflows react to events which move the workflow's current activity from one state to another.

We can choose sequential or state machine workflow project to analysis the attack workflow feature definition. Make sure the relationship between the state and feature. The accurate attack workflow feature definition can improve recognize efficiency of neural network.

2.4 Target Vectors of Feed Forward BP

Vector T is the corresponding target output. As each input is applied to the network, the network output is compared to the target.

The target T has 26 vectors according to training attack types.This element was used during training as the target output of the neural network. The target vector is 26-demension column vector. Each vector T ranges from zero to one. When the corresponding bit value is one, it indicates the target attack type would happen.

The vector T is defined as follows:

Target_DOS_back = $[1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$

Target_DOS_land = $[0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$

Target_DOS_neptune = $[0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$

The other target attack vectors can be done in the same way.

3 Training Procedure with Feed Forward BP Network in IDS

3.1 Architecture of Feed Forward BP in IDS

The architecture of Feed Forward BP is three-layer structure. It's train function is traingdx and perform function is sse.

```
BP_IDS_net=newff(minmax(P),[10,26],{'logsig','logsig'},'traingdx');
```

The feed forward BP neural network architecture is shown as Figure 1.

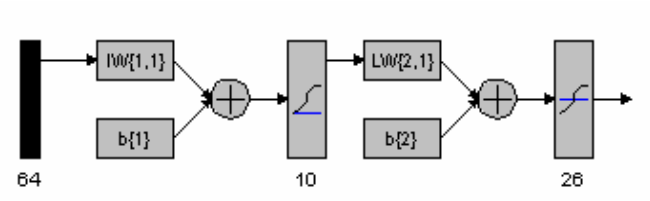


Fig. 1. Two-layer feed forward BP network architecture

3.2 Training Neural Network with Corrected Attack Dataset

The BP neural network was trained with the input data. When the generated output results don't satisfy the target output result, adjust the error from the distortion of target output.

We import KDD Cup 99 correct data set to SQLserver2005, use business intelligent development visual studio to build analysis services. After analysis of the corrected data, we build our own corrected attack set according to predefined train attack types and target vectors. The corrected attack data set comes from national research centre of anti computer intrusion and virus defence. The number of attack records is twenty thousand items.

The quality of corrected attack set influence greatly the accurate target vectors recognize. We retrain or stop training the network depending on this error value. Once the training is over, we store the knowledge it acquired. When the train is over, we can detect the abnormal behaviour. The output will express the whether the input pattern is normal behaviour or not.

The algorithm should adjust the network parameters in order to minimize the sum square error. The italic letter t is target value and a is output value in equation 2.

$$sse = \sum_{i=1}^{N} (e_i)^2 = \sum_{i=1}^{N} (t_i - a_i)^2 \tag{2}$$

3.3 Training Network with Test Attack Dataset

The test data of KDD Cup 99 has two data sets. One is complete dataset with 406M, the other amount database is 10 percent of the complete database about 42.6M. Because we use another corrected database different from KDD Cup 99, we can't use the test dataset as training data. But we modify our test attack data set according to the test data structure of KDD Cup 99.

In order to simulate the real attack, we insert noise data into the corrected attack dataset. The noise average value set to zero. The noise mean square error ranges from 0 to 4. We use different noise datasets to simulate different attack. The relationship between the noise level and recognition errors percentage of IDS is shown as figure 2.

Real lines indicate that the recognition neural network is trained by corrected attack dataset and test attack dataset with noise data. Broken lines show that the recognition neural network is only trained by single corrected attack dataset. The multi-training method can decrease the recognition errors percentage.

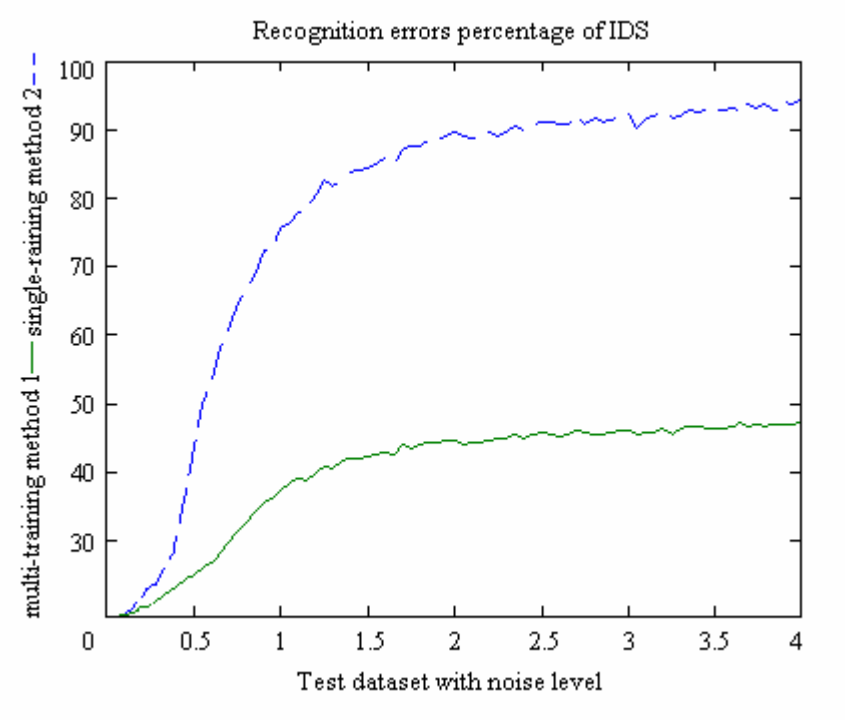


Fig. 2. The relationship between the noise level and percentage of recognition errors

4 Results Analysis

The BP neural network has 64 bits input vectors which indicate different attack feature workflow. We changed these input vectors into double values. The double values are show as the first graph in figure 3.

The network should been trained and tested using different data sets. It is trained with corrected behaviours and test behaviours with some noise. The noise data is produced by random function which is shown as the second graph in figure 3. After being trained, the network can distinguish intruders including attack behaviours users and normal behaviours.

The target vectors have output values 0 or 1 in the 26 output bits. Each one stands for different attack types from KDD cup 99 dataset. The output is shown as the third graph of figure 3.

The fourth graph of the figure 3 indicates that the recognition neural network is trained by corrected attack dataset and test attack dataset with noise data. When the noisy level increase, the recognition errors percentage can grow correspondingly. When the noise level approach zero, the recognition percentage will improve greatly.

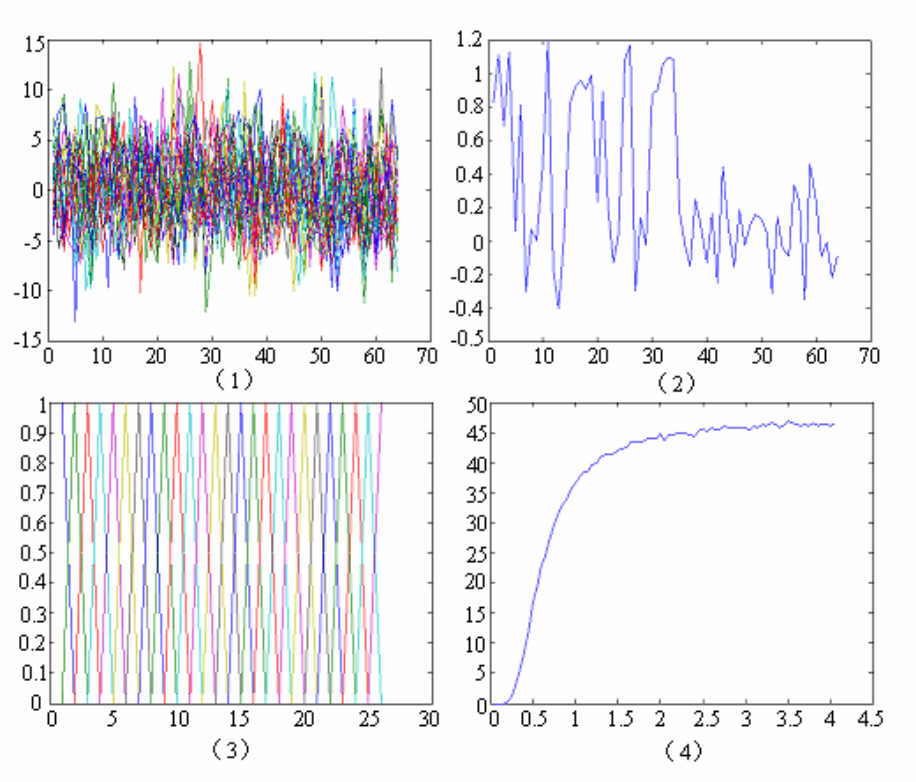


Fig. 3. Results of BP neural network for intrusion detection

5 Conclusions

BP neural networks with work flow features can be applied in IDS. The network can classify normal and abnormal attacks on intrusion detection system by training and learning. The noise level has influence on the recognition percentage. If it satisfies precise corrected dataset with attack workflow feature and low noise level, the results will be quite perfect. However, it's difficulty to create perfect training test dataset. Even the users have wonderful dataset, which does not mean the network can find all kinds of network intrusion attack. Our research on virus using rootkit technology describes that it's a long way for anti-virus software or operating system, even the neural network, to find all unknown attack. Further research might be carried out to build a system with other types of work flow features. Moreover, a combination of more than one type of neural networks and features from malicious code can achieve better results.

Acknowledgments. Supported by National hi-tech research and development project No.2006AA01Z405. Sponsored by the Shanghai leading academic discipline project, China grant No. P1303. Sponsored by Shanghai postdoctoral scientific program

No.08R214131. Innovation Program of Shanghai Municipal Education Commission No. 09YZ346.

References

1. Martin, T., Hagan, H.B., Demuth, M.B.: Neural Network Design. China Machine Press (2003)
2. Michailidis, E., Katsikas, S.K., Georgopoulos, E.: Intrusion Detection Using Evolutionary Neural Networks. In: Panhellenic Conference on Informatics, pp. 8–12. IEEE Press, Los Alamitos (2008)
3. Song, G.J., Zhang, J.L., Sun, Z.L.: The Research of Dynamic Change Learning Rate Strategy in BP Neural Network and Application in Network Intrusion Detection. In: The 3rd International Conference on Innovative Computing Information and Control, p. 513. IEEE Press, Los Alamitos (2008)
4. Chen, R.C., Cheng, K.F., Hsieh, C.: Using Fuzzy Neural Networks and Rule Heuristics for Anomaly Intrusion Detection on Database Connection. In: Proceedings of the Seventh International Conference on Machine Learning and Cybernetics, pp. 3607–3612. IEEE Press, Kunming (2008)
5. Zhou, T.J., Yang, L.: The Research of Intrusion Detection Based on Genetic Neural Network. In: Proceedings of the 2008 International Conference on Wavelet Analysis and Pattern Recognition, pp. 276–281. IEEE Press, Hongkong (2008)
6. Golovko, V., Kachurka, P., Vaitsekhovich, L.: Neural Network Ensembles for Intrusion Detection. In: IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, pp. 578–583. IEEE Press, Dortmund (2007)
7. Ganesh, K.P., Devaraj, D.: Network Intrusion Detection Using Hybrid Neural Networks. In: IEEE - Signal Processing, Communications and Networking, pp. 563–569. IEEE Press, Chennai (2007)
8. Tich, P.T., Jan, T.: Boosted Modified Probabilistic Neural Network (BMPNN) for Network Intrusion Detection. In: International Joint Conference on Neural Networks, pp. 2354–2361. IEEE Press, Vancouver (2006)
9. Tamilarasan, A., Mukkamala, S., Sung, A.H., Yendrapalli, K.: Feature Ranking and Selection for Intrusion Detection Using Artificial Neural Networks and Statistical Methods. In: International Joint Conference on Neural Networks, pp. 4754–4761. IEEE Press, Vancouver (2006)
10. Lazarevic, A., Pokrajac, D., Nikolic, J.: Applications of Neural Networks in Network Intrusion Detection. In: International Joint Conference on Neural Networks, pp. 59–64. IEEE Press, Serbia (2006)

Features of Hodgkin-Huxley Neuron Response to Periodic Spike-Train Inputs

Zeng An, Liu Yan, and Chen Liujun*

Department of Systems Science, School of Management
Beijing Normal University, Beijing 100875, P.R. China
chenlj@bnu.edu.cn

Abstract. Researches on neuron response to external stimulation will provide useful insights into neural mechanism of learning and cognitive function of the brain. In this article, a neuron described by Hodgkin-Huxley model receives periodic spike-train inputs. The responses to inputs with various frequencies and synaptic conductivities are simulated. The results show that mode-locking response pattern is the main type of response to periodic inputs, which is consistent with general knowledge. The mode-locking patterns as the function of the frequency and synaptic conductivity are given and characteristics of mode-locking boundaries are analyzed. Furthermore, how input frequency and synaptic conductivity influence HH neuron response is in detail explained respectively.

Keywords: Hodgkin-Huxley neuron, spike-train inputs, mode-locking, synaptic conductivity.

1 Introduction

The Hodgkin-Huxley neuron, which was first derived as a model of the squid giant axon, shows typical dynamics of a real neuron[1]. Since the model was proposed, the response patterns of HH neuron under specific inputs have received much attention and lots of relevant researches have been completed[2-7]. Some researchers investigated the response patterns and the related bifurcation structure of the neurons stimulated by constant or periodic currents. For instance, Lee and Kim found mode-locking and chaotic response patterns with sinusoidal current inputs[3]. Huber and Braun discussed the dynamical behaviors with external chaotic random inputs[4]. However, A real neuron of the brain receives synaptic current transmitted by synapse instead of constant or sinusoidal current. It has been pointed out that long-term changes in the transmission properties of synapses provide a physiological substrate for learning and memory[5]. So researches on dynamic characteristics of a HH neuron response to synaptic currents will provide useful insights into neural mechanism of learning and memory. Some corresponding researches have been accomplished from this perspective. Hasegawa analyzed response patterns with different kinds of spike-train inputs

* Corresponding author.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 68–74, 2009.

and offered the range of input periods and synaptic conductivities for mode-locking patterns[8]. In this paper, more detailed simulations with some different settings are implemented and a more accurate parameter ranges are given. Specifically speaking, the mode-locking patterns as the function of the frequency and synaptic conductivity are given and characteristics of mode-locking boundaries are analyzed. Furthermore, how input frequency and synaptic conductivity influence HH neuron response are in detail explained respectively.

2 Method

We consider a simple system with a neuron and a synapse as Hasegawa has discussed. The neuron is described by the HH model, whereas the synapse is described by the synaptic model of Destexhe et al which is different to the model of Hasegawa[8-9]. when periodic spike-train inputs are applied to the synapse, a corresponding synaptic currents are transmitted to the HH neuron. We will investigate the response patterns of the HH Neuron to inputs with various frequencies and synaptic conductivities.

HH model is given as follows

$$C\frac{dV}{dt} = -g_{Na}m^3h(V - E_{Na}) - g_K n^4(V - E_K) - g_L(V - V_L) - I_{syn} \quad (1)$$

where V is the membrane potential while I_{syn} is the synaptic current. The other parameters are: $C = 1\mu F/cm^2$, $E_{Na} = 50mV$, $E_K = -77mV$, $E_L = -54.387mV$, $g_{Na} = 120mS/cm^2$, $g_K = 36mS/cm^2$, $g_L = 0.3mS/cm^2$. Details of the HH model can be found in Refs[1].

The spike-train inputs can be given as a train of delta functions,

$$U_i(t) = \sum_{n=1} \delta(t - t_{in}) \quad (2)$$

where t_{in} is the time of the nth spike. For a periodic spike-train input, the period of T_i can be represented by $T_i = t_{in+1} - t_{in}$.

The synaptic current model is different from the model of Hasegawa. The synaptic current is given as

$$I_{syn} = g_{syn}r(V - E_{syn}) \quad (3)$$

where g_{syn} is synaptic conductivity, the parameter r is the proportion of open ion channels, and E_{syn} is the synaptic reversal potential. For an excitatory synapse, $E_{syn} = 0$. The parameter r, different from the alpha function model of Hasegawa, is described as

$$\frac{dr}{dt} = \begin{cases} \alpha T_{\max}(1 - r) - \beta r, & t_{in} \le t < t_{in} + \tau \\ -\beta r, & t_{in} + \tau \le t < t_{in+1} \end{cases} \quad (4)$$

where the parameters $\alpha = 1.1mM^{-1}mS^{-1}$, $\beta = 0.19mS^{-1}$, $T_{max} = 1mM$, $\tau = 1ms$. Details of the synaptic current can be found in Refs[9].

The HH neuron is regarded as firing when the membrane potential $V > 0mV$. The mth firing time of the output spike-train is represented by t_{om}. If the HH neuron fires more than twice, the output ISI (interspike interval) can be defined as $T_{om} = t_{om+1} - t_{om}$. So the average ISI of the spike-train output can be expressed by

$$\mu_o = \sum_{m=1}^{q-1} T_{om}/(q-1) = (T_{oq} - T_{o1})/(q-1) \tag{5}$$

where q is the number of the spikes and $q \geq 2$. if $q < 2$, then $\mu_o \to \infty$, which means that the HH neuron doesn't respond to the input persistently.

The simulation starts from 100ms and lasts 2000ms. Runge-Kutta method is adopted and the step length is set as 0.01ms.

3 Results

For convenience, we define the input frequency $f_i = 1/T_i$ and output frequency $f_o = 1/\mu_o$. In this section, how the f_o reacts to the parameter f_i and g_{syn} will be analyzed.

3.1 Three Types of Response Pattern

To characterize response pattern, we define the ratio of output to input frequency $k = f_o/f_i$, representing the average number of spikes per period of spike-train input. According to the value of k, the response of HH neuron under periodic spike-train inputs can be roughly classified into three types, mode-locking pattern, irregular firing pattern and nonpersistent firing pattern. If the ratio $k = n/m$ where n and m are positive integers, the response is classified into mode-locking pattern. Whereas if the $k \neq n/m$, the response is classified into irregular firing pattern. For nonpersistent firing pattern, the k is almost 0. Two typical mode-locking response patterns and an irregular firing pattern are shown in Fig.1. The input frequencies are the same while the synaptic conductivities are different. The mode-locking response pattern of 1/1 will appear if synaptic conductivity $g_{syn} = 120\mu S/cm^2$, while the pattern of 1/2 will appear if $g_{syn} = 92\mu S/cm^2$. The neuron will respond with irregular firing pattern if $g_{syn} = 95\mu S/cm^2$.

The responses to periodic spike-train inputs with various frequencies and synaptic conductivities are numerically explored and the ratio k is calculated. The input frequencies range from 10Hz to 200 Hz and synaptic conductivity range from $50\mu S/cm^2$ to $250\mu S/cm^2$. The ratio $k = 1$ when synaptic conductivity $g_{syn} = 250\mu S/cm^2$ and input frequency $f_i = 10Hz$. Then with the decreasing of g_{syn} and increasing of f_i, k decreases from 1 to 0. When g_{syn} is low enough, nonpersistent firing pattern with $k = 0$ will appear no matter how high f_i is. According to the calculated value of k, the main form of k is $k = 1/m$ with $m = 1, 2, 3$, representing the mode-locking of 1/1, 1/2 and 1/3 respectively. The response patterns of $k \neq 1/m$ should be classified into the irregular firing patterns. So it can be concluded that the mode-locking pattern is the main type of response to periodic spike-train inputs.

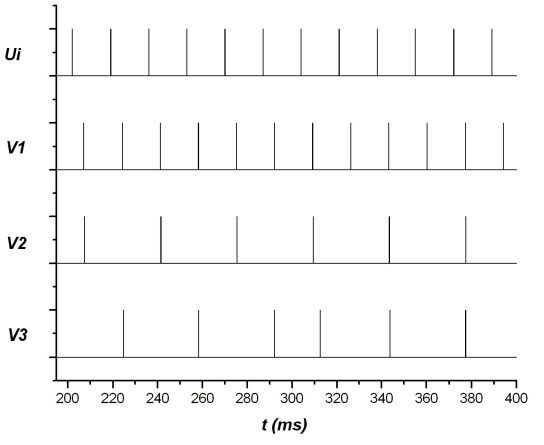


Fig. 1. The two typical mode-locking response patterns and an irregular firing pattern to periodic spike-train inputs with frequency $f_i = 59$Hz. U_i is the periodic spike-train input, $V1$ is 1/1 mode-locking response pattern with $g_{syn} = 120\mu S/cm^2$, $V2$ is 1/2 mode-locking pattern with $g_{syn} = 92\mu S/cm^2$, $V3$ is irregular firing pattern with $g_{syn} = 95\mu S/cm^2$.

3.2 Characteristics of the Pattern Boundary Curves

To show the features of HH neuron response, it is useful to give the parameter ranges of input frequency and synaptic conductivity for each response pattern. So we depict the boundary curves of 1/1, 1/2 and 1/3 mode-locking and non-persistent firing pattern in the parameter space of $f_i - g_{syn}$ as given in Fig. 2. The boundary curves divide the parameter space into the mode-locking and non-persistent firing regions, and the irregular firing regions locate between them.

Let's consider the characteristics of the boundary curve for nonpersistent firing pattern firstly. For excitatory synapse, the higher synaptic conductivity is, the easier the postsynaptic neuron get fired. So the HH neuron can fire persistently only if g_{syn} is more than a minimal value, which can be defined as $\overline{g}_{syn}$, representing the boundary of nonpersistent firing pattern. The $\overline{g}_{syn}$ as a function of the input frequency f_i can be seen in Fig.2. When input frequency $f_i < 50$Hz, the $\overline{g}_{syn} \approx 82\mu S/cm^2$. When $f_i > 50$Hz, the $\overline{g}_{syn}$ rises mainly and decreases occasionally as input frequency increases. For example, the $\overline{g}_{syn}$ increase to $251\mu S/cm^2$ when f_i is 200Hz.

Then consider the characteristics of the boundary curves for mode-locking patterns. There are three kinds of mode-locking regions in the parameter space of $f_i - g_{syn}$, the mode-locking of 1/1, 1/2 and 1/3. It can be seen from Fig. 2 that the upper boundary curve of the mode-locking of 1/2 and 1/3 are all upward smoothly. It is interesting that the lower boundary curve of 1/1, 1/2 and 1/3 are all mainly upward with short uplift segment. Take the mode-locking of 1/2 for example, as input frequency increases, it goes upward at first, then it falls down a little, at last it goes upward again.

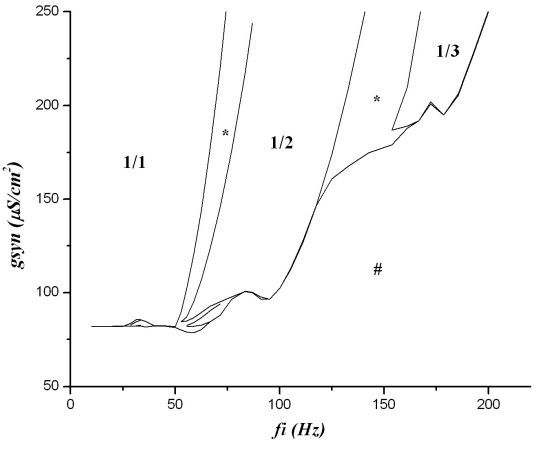


Fig. 2. The 2-dimensional phase diagram of the ratio of k in the parameter space of $f_i - g_{syn}$. The parameter regions labeled 1/1, 1/2 and 1/3 is corresponding to 1/1, 1/2 and 1/3 mode-locking pattern respectively, the regions labeled * is corresponding to irregular firing pattern, and the region labeled # is corresponding to nonpersistent firing pattern.

3.3 How Input Frequency Influences the Response

The phase diagram of the ratio k show how input frequency influences the response of a HH neuron with synaptic conductivity given. For example, with $g_{syn} = 200\mu S/cm^2$, when input frequency increases from 10Hz to 200Hz, the response pattern is the mode-locking of 1/1, to 1/2 and then to 1/3. The different mode-locking response patterns mean the influence of input frequency is nonlinear. It indicates that when a neurons is supplied with low frequency, the neuron inclines to transfer the signal completely; when supplied with high frequency, the neuron may compress the signal. It is especially interesting that bigger input frequency incur fewer spikes output. For example, when the input frequency is 68Hz, the mode-locking is 1/1 meaning one spike input will recur one spike output, so the output spikes 68 times each 1000ms. When the input frequency increase to 86Hz, the mode-locking is 1/2 meaning two spikes input will recur one spike output, so the output spikes only 43 times each 1000ms.

3.4 How Synaptic Conductivity Influences the Response

As we have known, synaptic conductivity determine transmission property of synapse, so the impact of synaptic conductivity on response pattern of a neuron to given inputs is especially important. From Fig. 2, it can be seen that there are more than one mode-locking response patterns as synaptic conductivity varies with input frequency given. For example, given spike-train input with frequency 70Hz, when g_{syn} decrease from $250\mu S/cm^2$ to $220\mu S/cm^2$, the ratio of output to input frequency k is 1/1, representing that the output frequency maintains

70Hz instead drops continuously. With g_{syn} decreasing further, the HH neuron shows irregular firing pattern. When g_{syn} decreases to $146\mu S/cm^2$, the response pattern becomes the mode of 1/2 until g_{syn} is $95\mu S/cm^2$. After that, the HH neuron fire irregularly again. At last, the HH neuron can not fire persistently when the g_{syn} is less than $88\mu S/cm^2$. It is evident that synaptic conductivity have no impact on response of a neuron within one mode-locking pattern, while the impact will be dramatic and discontinuous from one to another mode-locking pattern. Contrary to the situation of direct current input where the average output frequency changes gradually with synaptic conductivity varying, the average output frequency changes discontinuously under periodic spike-train inputs.

4 Conclusion

The response patterns of a HH neuron with periodic spike-train inputs are simulated. The results show that a HH neuron will respond mainly in a mode-locking way. The mode-locking patterns as the function of input frequency and synaptic conductivity are given and characteristics of mode-locking boundaries are investigated. Furthermore, how input frequency and synaptic conductivity influence the response is in detail explained respectively.

How a HH neuron responds to inputs with different frequencies is analyzed. Generally, when a neurons is supplied with low frequency input such as 50Hz, the neuron respond in 1/1 locked mode, which inclines to transfer the input signal completely. When supplied with higher frequency input, the neuron will respond in 1/2 or 1/3 locked mode, which may compress the input signal. The results might provide useful sights into signal transmission in neural network.

How synaptic conductivity influences the response of a HH neuron to inputs with given frequency is analyzed. There are more than one mode-locking response patterns as synaptic conductivity varies, representing that synaptic conductivity has no impact on the response within one mode-locking pattern while the impacts are dramatic and discontinuous from one to another mode-locking pattern. Since synaptic conductivity determines transmission properties of synapse and the long-term changes in synaptic properties provide a physiological substrate for learning and memory, the results are useful for understanding learning process.

The mode-locking response pattern is quite common and plays an important role in synchronization, which deserves further investigation[10].

Acknowledgement

This work was supported by the NSFC Grant (Number: No. 70601002, No. 60534080). The authors thank Dr. Wang Dahui, Dr. Di zengru for their helpful discussion and suggestion.

References

1. Hodgkin, A.L., Huxley, A.F.: A quantitative description of membrane current and its application to conduction and excitation in nerve. J. Physiol. 117, 500–544 (1952)
2. Izhikevich, E.M.: Which Model to Use for Cortical Spiking Neurons. IEEE transactions on neural networks 15(5), 1063–1070 (2004)
3. Lee, S.G., Kim, S.: Bifurcation analysis of mode-locking structure in a Hodgkin-Huxley neuron under sinusoidal current. Physical Review E 73, 41924 (2006)
4. Huber, M.T., Braun, H.A.: Stimulus-response curves of a neuronal model for noisy subthreshold oscillations and related spike generation. Physical Review E 73, 041929 (2006)
5. Abbott, L.F., Regehr, W.G.: Synaptic computation. Nature 431, 796–803 (2004)
6. Matsumoto, G., Aihara, K., Ichikawa, M., Tasaki, A., Theor, J.: Periodic and non-periodic responses of membrane potentials in squid giant axons during sinusoidal current stimulation. Neurobiol. 3, 1–14 (1984)
7. Arcas, B.A.Y., Fairhall, A.L., Bialek, W.: Computation in a single neuron: Hodgkin and Huxley revisited. Neural Computation 15, 1715–1749 (2003)
8. Hasegawa, H.: Responses of a Hodgkin-Huxley neuron to various types of spike-train inputs. Physical Review E 61, 718–726 (2000)
9. Destexhe, A., Bal, T., McCormick, D.A., et al.: Ionic Mechanisms Underlying Synchronized Oscillations and Propagating Waves in a Model of Ferret Thalamic Slices. J. Neurophys. 76, 2049–2070 (1996)
10. Francisco, V., Jean-Philippe, L., et al.: The Brainweb: Phase Synchronization And Large-Scale Integration. Nature Reviews 2, 229–239 (2001)

Multi-start Stochastic Competitive Hopfield Neural Network for *p*-Median Problem

Yiqiao Cai, Jiahai Wang, Jian Yin, Caiwei Li, and Yunong Zhang

Department of Computer Science, Sun Yat-sen University,
No.135, Xingang West Road, Guangzhou 510275, P.R.China
yiqiao00@hotmail.com

Abstract. In this paper, we propose a neural network algorithm—multi-start stochastic competitive Hopfield neural network (MS-SCHNN) for the *p*-median problem. The proposed algorithm combines two mechanisms to improve neural network's performance. First, it introduces stochastic dynamics into the competitive Hopfield neural network (CHNN) to help the network escape from local minima. Second, it adopts multi-start strategy to further improve the performance of SCHNN. Experimental results on a series of benchmark problems show that MS-SCHNN outperforms previous neural network algorithms for the *p*-median problem.

Keywords: *p*-median, competitive Hopfield neural network, stochastic dynamics, multi-start strategy.

1 Introduction

Given a set N of n demand nodes, the objective of the p-median problem is to select a subset F with p facilities (median nodes), where $F \subset N$, such that the total distance from the remaining nodes in $\{N - F\}$ to its closest facility in F is minimized [1]. This problem has been proven to be NP-hard by Kariv and Hakimi [2]. So far, many sophisticate heuristics techniques have been proposed for solving the p-median problem, including constructive algorithms, local search procedures, mathematical programming and metaheuristic methods, such as tabu search (TS), variable neighborhood search (VNS), simulated annealing (SA), hybrid heuristic algorithms and so on (details can be found in the review [3]).

Since Hopfield and Tank [4] proposed a Hopfield neural network (HNN) for solving traveling salesman problem, it have been widely used for classical combinatorial optimization problems, such as assignment problems, constraint satisfaction problems, clustering problems, graph problems, etc [5]. The HNN consists of numerous interconnected neurons, and uses a gradient descent method. This gradient method only needs simple computational requirements, thus can potentially lead to rapid solutions for implementing complex computation. Therefore, HNN can be directly implemented in analog hardware easily due to simple computational requirements and the intrinsic parallel structure.

Domínguez and Muñoz [6] firstly proposed a two-layer Hopfield neural network for the p-median problem based on two types of decision variables. In

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 75–83, 2009.

the two-layer neural model, they organized the network in disjoint groups and adopted the competitive rule, thus proposed the neural algorithms based on the group-parallel dynamics and the layer-parallel dynamics (called NA-G and NA-L respectively)[6]. In order to improve the the performance of NA-L, they also adopted a scattering strategy to maximize the initial facilities' dispersion, and propose NA-L+ [6]. Noted that this maximizing dispersion problem is similar to the maximum diversity problem (MDP), which is proved to be a NP-hard problem [7]. Simulation results showed that the performance of NA-L+ was better than that of VNS. However, the algorithms based on the two-layer neural model [6] have no mechanism to overcome the local minima problem of neural networks.

In this paper, we first propose a stochastic competitive HNN (SCHNN) by introducing stochastic dynamics into the two-layer competitive HNN (CHNN) to help the network escape from local minima. Then, in order to further improve the performance of SCHNN, we adopt the multi-start strategy in the SCHNN to propose a new neural network algorithm, called multi-start stochastic competitive Hopfield neural network (MS-SCHNN) in this paper. The simulation results on the benchmark instances show that MS-SCHNN can remarkably improve the two-layer neural model and outperform previous neural network algorithms in Ref.[6] for the p-median problem.

2 Two-Layer CHNN for the p-Median Problem

The p-median problem is a combinatorial optimization problem, where the feasible solutions are yielded by selecting p facilities to minimize the sum of the weighted distance from the demand nodes and its closest facility.

An integer programming formulation for the discrete p-median problem is given by ReVelle and Swain [8], but the formulation has n^2 variables and n^2+1 constraints. In addition, the constraint of preventing demand points from being assigned to an unopened facility is an obstacle to apply the neural network [3]. Due to these insufficiencies of this model, Domínguez and Muñoz proposed a two-layer neural model for the p-median problem with $2np$ variables and $n+p$ constraints [6]. The new model of the problem is shown as follows:

$$\text{Minimize} \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{q=1}^{p} d_{ij}x_{iq}y_{jq}. \tag{1}$$

$$\text{Subject to} \sum_{q=1}^{p} x_{iq} = 1, \qquad i = 1,\ldots,n, \tag{2}$$

$$\sum_{j=1}^{n} y_{jq} = 1, \qquad q = 1,\ldots,p, \tag{3}$$

where n is the considered number of demand points, p is the number of facilities, d_{ij} is the distance between the demand point i and the facility j. In this model,

these are two types of variables: x_{iq} (allocation variables) and y_{jq} (location variables), and their definitions are as follows:

$$x_{iq} = \begin{cases} 1 & \text{if } i \text{ is assigned to the cluster } q, \\ 0 & \text{otherwise,} \end{cases}$$

$$y_{jq} = \begin{cases} 1 & \text{if } j \text{ is the center of the cluster } q, \\ 0 & \text{otherwise.} \end{cases}$$

Noted constraints (2) and (3) are much simpler to guarantee that one point associates to only one cluster and there is only one facility in each cluster respectively.

In order to make sure the solution validity and avoid parameter tuning problem, they divided the neural network into $n + p$ groups based on the two constraints and introduced the competitive rule into the new model [6]. Thus, a competitive HNN (CHNN) for the p-median problem is proposed, and the energy function is reduced to

$$E = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{q=1}^{p} d_{ij} x_{iq} y_{jq}. \tag{4}$$

The input function of the neurons is calculated as follows:

$$h_{x_{iq}}(k) = -\sum_{j=1}^{n} d_{ij} y_{jq}(k), \tag{5}$$

$$h_{y_{jq}}(k) = -\sum_{i=1}^{n} d_{ij} x_{iq}(k), \tag{6}$$

where k is the iteration time, $h_{x_{iq}}$ and $h_{y_{jq}}$ denote the activation potential of allocation neuron x_{iq} and location neuron y_{jq} respectively. And the output of the neuron is defined as follows:

$$x_{iq}(k+1) = \begin{cases} 1 & \text{if } h_{x_{iq}}(k) = \max\limits_{1 \le j \le n} \{h_{x_{jq}}(k)\}, \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

$$y_{jq}(k+1) = \begin{cases} 1 & \text{if } h_{y_{jq}}(k) = \max\limits_{1 \le i \le n} \{h_{y_{iq}}(k)\}, \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

3 MS-SCHNN for the p-Median Problem

In the previous section, a two-layer CHNN for the p-median problem is proposed. However, the neural network model has no mechanism to escape from local minima. In this section, by incorporating the stochastic dynamics into the model, we first propose a stochastic CHNN (SCHNN) that permits temporary energy ascent to help the CHNN escape from local minima. Then, in order to further improve the performance of the SCHNN, we introduce the multi-start strategy into the SCHNN to develop a multi-start SCHNN (MS-SCHNN) for the p-median problem.

3.1 SCHNN

In SCHNN, we introduced the stochastic hill-climbing dynamics strategy [9] to overcome the local minima, thus the input functions of the neurons are modified as follows:

$$h_{x_{iq}}(k) = \alpha(t) \cdot (-\sum_{j=1}^{n} d_{ij} y_{jq}(k)), \tag{9}$$

$$h_{y_{jq}}(k) = \alpha(t) \cdot (-\sum_{i=1}^{n} d_{ij} x_{iq}(k)), \tag{10}$$

where t is the updating step number, donates that all the neurons are being updated at tth step. $\alpha(t)$ is a random multiplier, given by $\alpha(t) = random(h(t), 1)$. In $\alpha(t)$, $h(t) = 1 - T \cdot e^{-t/\lambda}$, and T is the factor of controlling the stochastic dynamics range. The random multiplier $\alpha(t)$ controlling the value $h_{x_{iq}}(k)$ and $h_{y_{jq}}(k)$ in Eq.(9) and Eq.(10), can be regarded as a random scaling parameter for them. Therefore, it can make the stochastic decision on which neuron within the same group to be fire according to Eq.(7) and Eq.(8). This stochastic dynamics can contribute to the blocking of continuous firings and encourage other neurons to be eventually fired, and ultimately assist the system to escape from a fixed point or a local minimum [10]. During the search process, the stochastic dynamics is gradually vanishing. When $\alpha(t) = 1$, stochastic dynamics vanishes and the whole dynamics of the SCHNN is the same as that of the original CHNN [13].

From the above analysis, the dynamics of SCHNN, differs from the optimal dynamics of the original CHNN, can provide a mechanism for hill climbing by varying the direction of neuron motion.

3.2 MS-SCHNN

The intensification and diversification are the very important mechanism to determine the performance of a metaheuristic, thus the right balance between them is crucial for the algorithm [11]. Therefore, we introduce the multi-start strategy into SCHNN and propose MS-SCHNN algorithm.

In the SCHNN, the stochastic dynamics vanishes gradually and the system finally reaches a stable state. Thus, the balance between intensification and diversification is fixed or only changed into one direction, i.e., continuously decreasing diversification and increasing intensification. In order to further improve the performance of the SCHNN, the balance between intensification and diversification should rather be dynamical. Therefore, a multi-start strategy is introduced into the SCHNN. The multi-start strategy super-imposed on the SCHNN is characterized by alternating phases of cooling and reheating the stochastic dynamics, thus provides a means to achieve an effective dynamic or oscillating balance between intensification and diversification during the search. The MS-SCHNN for the p-median problem can be summarized as follows:

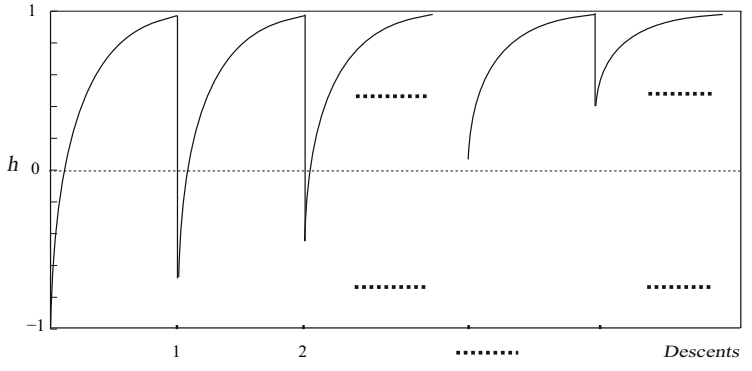


Fig. 1. A schematic diagram for $h(t)$ changes over *Descents* in the multi-start algorithm (where $T = T \cdot StartTFactor$, $StartTFactor < 1$)

Algorithm—MS-SCHNN:

1. Initialize:
  1.1 Initialize the neurons in allocation and location layers (random around zero);
  1.2 Determine the activation potential of the neurons in each groups using Eq.(7) or (8); initialize the best-so-far solution *gb* by using Eq.(4);
  1.3 set the parameters: T, λ, *maxiterations*, *Descents*, $StartTFactor$;
2. MS-SCHNN phases: /*multi-start strategy*/
  FOR $i = 1$ to *Descents* do
  2.1 Update the neurons in the two-layer stochastic competitive neural network
  $t = 0$;
  REPEAT /*SCHNN procedures */
  2.1.1 Update all the neurons in the allocation layer using Eq.(9) and (7);
  2.1.2 Update all the neurons in the location layer using Eq.(10) and (8);
  2.1.3 Calculate the energy using Eq.(4) and update the *gb*;
  2.1.4 $t = t + 1$;
  UNTIL (the energy stable or $t >$ *maxiterations*)
  2.2 Set the current solution as a new starting point for the next search;
  2.3 Set $T = T * StartTFactor$;
  END FOR
3. Output the best solution *gb*.

In the MS-SCHNN, the stochastic dynamics anneals gradually and vanishes finally during one descent neural search, and reheated by restarting the next search process. When the stochastic dynamics vanished completely, the proposed algorithm is equivalence to NA-L.

Fig.1 is a schematic diagram to show how the $h(t)$ changes over *Descents* in the multi-start algorithm. In the MS-SCHNN, the SCHNN performs multiple descent processes starting from different initial T values with the solution found

in the previous processes. Note that the first SCHNN search starts with a random solution using higher T value than that of the subsequent SCHNN process. The decreasing of T can guarantee that the restarting search is biased to only the promising regions and the search is gradually converged around the good solution by performing several SCHNN processes.

Therefore, by combining the stochastic dynamics and multi-start strategy, MS-SCHNN can provide an efficient mechanism for restarting the search for better solutions.

4 Simulation Results and Discussions

In order to test the performance of our algorithm, simulations were implemented on 40 benchmark p-median problems in the ORLIB [12]. We carried out the simulation experiments in Pentium IV 2.4GHz computer with 512Mb, which is similar to Ref.[6]. Every instance was run 50 times with different randomly generated initial neuron states, and then recorded the best solution and the CPU time for comparison.

According to the preliminary tests, we find that a large number of SCHNN restarting with a short run of each SCHNN search can obtain better solutions. In this parameters setting strategy, the parameter λ is set to be a small value. Since this strategy provides a means to achieve an effective dynamic balance between intensification and diversification, it can not only alleviate the sensitivity of λ, but also greatly improve the search performance. Based on the preliminary tests, the parameters for each instance are set as follows: $T = 2$, $\lambda = 15$, $maxiterations = 20$, $Descents = 50$, $StartTFactor = 0.99$.

Table 1 gives the results obtained by MS-SCHNN, NA-G and NA-L for the small scale problems. Since the ORLIB instances are benchmark problems, results of the NA-L and NA-G are directly from the Ref.[6]. The %ERROR denotes the rate of deviation between the best solution found by the algorithms and the optimum value. Simulation results show MS-SCHNN can get much better solutions for all the 10 instances than the NA-G and NA-L, but at the cost of significantly more CPU time. This is due to that most CPU time of MS-SCHNN is mainly used in stochastic search to improve the algorithm's performance. However, it is acceptable for the small scale problems in practice and it can be alleviated by implementing in parallel computing environment owing to the neural networks' parallel structure.

The experiment results suggest that MS-SCHNN can significantly improve the performance of the two-layer neural model within reasonable computation time. The reasons why MS-SCHNN is significantly superior to NA-G and NA-L are: (1) the stochastic strategy of SCHNN provides a strategy to help the network escape from local minima; (2) the proposed algorithm provides an efficient restarting search by using the solution found in the previous iteration as the next' s initial solution, differs from NA-L and NA-G with randomize initial next solutions for each iteration.

The comparison tests between NA-L+ and MS-SCHNN for the ORLIB instances are also carried out. Unlike MS-SCHNN, NA-L+ uses a scattering strategy to

Table 1. Comparison of MS-SCHNN, NA-G and NA-L for 10 ORLIB instances

Instance	Opt.	Best Median Cost			%ERROR			CPU time		
		NA-G	NA-L	MS-S-CHNN	NA-G	NA-L	MS-S-CHNN	NA-G	NA-L	MS-S-CHNN
pmed01	5819	5821	5819	5819	0.03	0.00	0.00	0.84	0.01	4.1
pmed02	4093	4300	4248	4093	5.06	3.79	0.00	1.61	0.01	8.2
pmed03	4250	4276	4278	4250	0.61	0.66	0.00	0.84	0.01	7.5
pmed04	3034	3205	3238	3034	5.64	6.72	0.00	2.91	0.02	14.0
pmed05	1355	1633	1570	1363	20.52	15.87	0.59	1.67	0.02	22.1
pmed06	7824	7867	7995	7824	0.55	2.19	0.00	1.56	0.01	19.3
pmed07	5631	5862	5819	5631	4.10	3.34	0.00	3.05	0.02	33.9
pmed08	4445	4955	4928	4446	11.47	10.87	0.02	4.21	0.02	71.6
pmed09	2734	3179	3179	2761	16.28	13.24	0.99	7.96	0.05	119.2
pmed10	1255	1840	1840	1273	46.61	27.25	1.43	9.71	0.05	171

Table 2. Comparison of MS-SCHNN and NA-L+ for the ORLIB instances

Instance	(n,p)	Opt.	Best Median Cost		%ERROR	
			NA-L+	MS-SCHNN	NA-L+	MS-SCHNN
pmed05	(100,33)	1355	1359	1363	0.30	0.59
pmed08	(200,20)	4445	4448	4446	0.07	0.02
pmed09	(200,40)	2734	2751	2761	0.62	0.99
pmed10	(200,67)	1255	1264	1273	0.72	1.43
pmed14	(300, 60)	2968	2983	2982	0.51	0.47
pmed15	(300, 100)	1729	1751	1768	1.27	2.26
pmed18	(400, 40)	4809	4811	4810	0.04	0.02
pmed19	(400, 80)	2845	2863	2872	0.63	0.95
pmed20	(400, 133)	1789	1815	1807	1.45	1.01
pmed23	(500, 50)	4619	4624	4623	0.11	0.09
pmed24	(500, 100)	2961	2986	2977	0.84	0.55
pmed25	(500, 167)	1828	1865	1906	2.02	4.27
pmed28	(600, 60)	4498	4508	4505	0.22	0.16
pmed29	(600, 120)	3033	3060	3096	0.89	2.08
pmed30	(600, 200)	1989	2016	2039	1.36	2.51
pmed33	(700, 70)	4700	4706	4706	0.13	0.13
pmed34	(700, 140)	3013	3038	3044	0.83	1.03
pmed37	(800, 80)	5057	5071	5065	0.28	0.16
pmed40	(900, 90)	5128	5155	5150	0.53	0.43

improve the performance of neural network. This strategy can generate good initial solutions by maximizing the facilities' dispersion. Results of the NA-L+ are also directly from the Ref.[6]. Table 2 summarizes the comparative results for the test instances. In this table, we eliminate the instances which both algorithms got the optimum value for simplicity. From Table 2, we can find that MS-SCHNN obtains the same number of optimum value as NA-L+ (21 out of 40), and gets better solution than NA-L+ in 9 instances, while worse than NA-L+ in 9 instances. It suggests that MS-SCHNN is competitive with NA-L+ in the quality of solution.

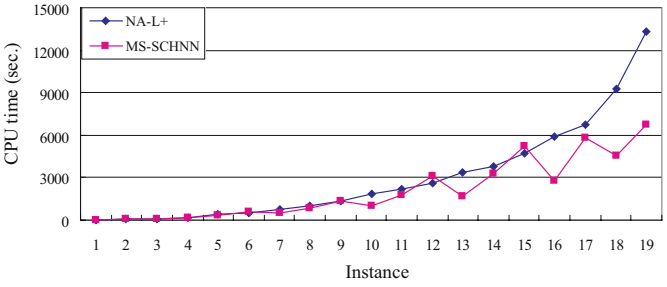


Fig. 2. Comparison of the CPU time of MS-SCHNN and NA-L+ for the instances in Table 2

Fig.2 shows the comparison of the CPU time. From the figure, MS-SCHNN spends approximately the same time as the NA-L+ for the small scale problems, but much less for the large scale problems, especially for the largest instance. The reason of this phenomenon maybe that NA-L+ uses the scattering strategy to construct an initial solution by maximizing the facilities' dispersion. However, the maximizing dispersion problem is similar to the maximum diversity problem (MDP), which is proved to be a NP-hard problem [7]. Therefore, NA-L+'s initial strategy spends much more time in getting the good initial solution, and increases the difficulty to implement in practice. Clearly Fig.2 states MS-SCHNN displays more scalability than NA-L+ for large scale problems. Moreover, MS-SCHNN can be implemented in practice much easily than NA-L+.

In summary, we can conclude that the performance of neural network can be improved efficiently by introducing stochastic dynamics and multi-start strategy into the neural model, and MS-SCHNN can more effectively and efficiently solve the p-median problem than other neural network algorithms in Ref.[6] (NA-G, NA-L and NA-L+).

5 Conclusions

In this paper, multi-start stochastic competitive Hopfield neural network (MS-SCHNN) is proposed for the p-median problem. MS-SCHNN introduces stochastic dynamics and multi-start strategy into the two-layer neural model to improve the performance of neural network. Simulation results show that MS-SCHNN can significantly improve the two-layer neural model and outperforms previous neural network algorithms for the p-median problem. In the future, we will apply MS-SCHNN to solve other combinatorial optimization problems, such as frequency assignment problem (FAP) [14], clustering problems [15], etc.

Acknowledgments. This work was supported in part by the National Natural Science Foundation of China (60805026), the Guangdong Provincial Natural Science Foundation of China (07300630), the Specialized Research Fund for the Doctoral Program of Higher Education (20070558052), and the Scientific

Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry (2007-1108).

References

1. Hakimi, S.: Optimum Distribution of Switching Centers in to Communication Network and Some Related Graph Theoretic Problems. Operations Research 13 (1965)
2. Kariv, O., Hakimi, S.: An Algorithmic Approach to Network Location Problem. Part2: The P-median. SIAM Journalon Applied Mathematics. 37, 539–560 (1979)
3. Mladenović, N., Brimberg, J., Hansen, P., Moreno-Perez, J.A.: The P-median Problem: A Survey of Metaheuristic Approaches. European Journal of Operational Research 179, 927–939 (2007)
4. Hopfield, J.J., Tank, D.W.: "Neural" Computation of Decisions in Optimization Problems. Biological Cybernetics 52, 141–152 (1985)
5. Smith, K.A.: Neural Networks for Combinatorial Optimization: A Review of More than a Decade of Research. INFORMS Journal on Computing 11, 15–34 (1999)
6. Domínguez, E., Muñoz, J.: A Neural Model for the p-median Problem. Computers & Operations Research 35, 404–416 (2008)
7. Kuo, C.C., Glover, F., Dhir, K.S.: Analyzing and Modeling the Maximum Diversity Problem by Zero-one Programming. Decision Sciences 24, 1171–1185 (1993)
8. ReVelle, C., Swain, R.: Central Facilities Location. Geographical Analysis 2, 30–42 (1970)
9. Smith, K., Palaniswami, M., Krishnamoorthy, M.: Neural Techniques for Combinatorial Optimisation with Applications. IEEE Transactions on Neural Networks 9, 1301–1318 (1998)
10. Wang, J., Zhou, Y.: Stochastic Optimal Competitive Hopfield Network for Partitional Clustering. Expert Systems with Applications 31, 2072–2080 (2008)
11. Blum, C., Roli, A.: Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. ACM Computing Surveys 35, 268–308 (2003)
12. Beasley, J.E.: OR-Library: Distributing Test Problems by Electronic Mail. Journal of the Operational Research Society 41, 1069–1072 (1990), http://people.brunel.ac.uk/~mastjjb/jeb/orlib/pmedinfo.html
13. Galán-Marín, G., Muńoz-Pérez, J.: Design and Analysis of Maximum Hopfield Networks. IEEE Transactions on Neural Networks 12, 329–339 (2001)
14. Wang, L.P., Liu, W., Shi, H.: Noisy Chaotic Neural Networks with Variable Thresholds for the Frequency Assignment Problem in Satellite Communications. IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and reviews 38, 209–217 (2008)
15. Huang, J.Z., Ng, M.K., Rong, H., Li, Z.: Automated Variable Weighting in k-means Type Clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence 27, 1–12 (2005)

An Information Theoretic Perspective of the Sparse Coding

Hideitsu Hino and Noboru Murata

School of Science and Engineering, Waseda University,
3-4-1 Ohkubo, Shinjuku, Tokyo 169-8555, Japan
hideitsu.hino@toki.waseda.jp, noboru.murata@eb.waseda.ac.jp

Abstract. The sparse coding method is formulated as an information theoretic optimization problem. The rate distortion theory leads to an objective functional which can be interpreted as an information theoretic formulation of the sparse coding. Viewing as an entropy minimization problem, the rate distortion theory and consequently the sparse coding are extended to discriminative variants. As a concrete example of this information theoretic sparse coding, a discriminative non-linear sparse coding algorithm with neural networks is proposed. Experimental results of gender classification by face images show that the discriminative sparse coding is more robust to noise, compared to the conventional method which directly uses images as inputs to a linear support vector machine.

Keywords: Rate Distortion Theory, Sparse Coding, Neural Network, Gender Classification.

1 Introduction

One of the major focuses of information theory and machine learning research is to extract essential information from data by means of computational and statistical methods. In most cases, compact representations of the information are considered of value. For example, data compression methods [1] explore effective coding of the original data, that is, encoding scheme using as few bits (or other information carrier unit) as possible while maintaining the similarity between the original data and decoded data within a certain allowable distortion level. Sparse coding methods [2] which originate from visual perception mechanisms in brain, try to reconstruct an input signal by a sparse combination of bases chosen from a large number of bases in a dictionary. In these theories, data compression and sparse coding, complexity of the representation is regarded as

- rate (description length for each datum),
- sparseness (the number of active bases used in actual representation),

respectively. In each theory, these complexity is minimized under some constraints on the following accuracy. Data compression methods try to maintain

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 84–93, 2009.

the original data structure (probability distribution of the source) with short description length. The accuracy is measured by decompression errors in the case of lossless compression, or differences between the original and the compressed data in the case of lossy compression. Sparse coding methods also try to minimize data reconstruction errors, while maintaining the number of active bases under a predefined level.

In this paper, we explore the information theoretic approach to the sparse coding, that is, we try to describe the sparse coding from the viewpoint of the rate distortion theory [3]. We then consider to extend the rate distortion, and consequently the sparse coding, to have discriminative ability. This extension is similar to the correspondence between the principal component analysis (PCA) and the Fisher discriminant analysis (FDA), both of which are based on the variance of given data. PCA finds axes of projections which conserve the variance of the original data as much as possible. In other words, it minimizes a reconstruction error of the original data in a lower dimensional subspace (e.g., feature space). As a supervised classification method based on the variance, FDA is the most famous one. FDA finds an axis of a projection which maximizes the ratio of the between-class variance to the within-class variance. The data projection to the axis leads to a good separation between different classes. From the information theoretic viewpoint, the notion of "variance" is replaced by "entropy". We propose a discriminative rate distortion theory or a discriminative sparse coding method which tries to encode the data or to extract the feature suitable for the classification task. As a concrete embodiment of this notion, we will show an example of discriminative sparse coding with neural networks, formalized as an optimization problem of the information theoretic objective functional.

The rest of this paper is organized as follows. In section 2, we briefly describe the rate distortion theory. In section 3, the sparse coding and its information theoretic interpretation are introduced. The rate distortion theory and the information theoretic sparse coding are extended to discriminative variants in section 4. In section 5, we propose combined neural networks for data reconstruction and classification as a concrete example of our discriminative sparse coding. A result of a simple experiment for gender classification by face images is shown in section 6.

2 Rate Distortion Theory

The rate distortion theory [3] provides theoretical foundations for lossy data compression. Let $x \in \mathbb{R}^n$ be an input valuable and $z \in \mathbb{R}^m$ be its compressed (encoded) expression. We use upper cases X, Z to represent the corresponding random variables henceforth. Given a distortion function $d : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ and a distortion upper limit D, the rate distortion problem is formally stated as an optimization problem with respect to the conditional probability $p(Z|X)$:

$$\min_{p(Z|X)} I(X;Z) \quad \text{subject to} \quad E[d(X,Z)] \leq D, \tag{1}$$

where $I(X;Z)$ is the mutual information between X and Z, and expectation in $E[d(X,Z)]$ is taken with respect to the joint distribution $p(X,Z)$. Intuitively speaking, the distortion $d(x,z)$ is an error measure for representing the data x by the decoded data from z. The mutual information $I(X;Z)$ is a measure of the information shared by the variables X and Z. Taking the meaning of these two measurements into consideration, the rate distortion theory is understood as an optimization problem which effectively extracts important information Z from the data X as much as possible, under the constraint that the loss of the information is limited to a certain level.

The rate distortion optimization problem of (1) can also be expressed in a trade-off form:

$$\min_{p(Z|X)} \; I(X;Z) + \beta E[d(X,Z)], \tag{2}$$

where β is a positive trade-off parameter.

3 Sparse Coding

The sparse coding is a method of reconstructing signals by a small number of activated bases. In general, the total number of the bases is very large, but each signal is reconstructed by a combination of a relatively small number of the bases. We denote a set of bases $\{b_i\}_{i=1}^m$ by a matrix $B = \{b_1, \ldots, b_m\} \in \mathbb{R}^{n \times m}$, where n is the dimension of the input signal and m is the number of bases. The main purpose of the sparse coding is to find a set of bases $\{b_i\}_{i=1}^m$ which can reconstruct signals from an information source with small errors with a small number of bases actually used. We write the coefficients of the bases by a vector $z \in \mathbb{R}^m$ and the reconstructed signal by $\hat{x} := Bz$. One of the formalization of this sparse coding objective is as follows:

$$\min_{B} \; E[d(X,Z)] \quad \text{subject to} \quad E[\,||Z||_p^p] \leq T, \tag{3}$$

where $||\cdot||_p$ stands for p-norm and T is a predetermined sparseness parameter. The distortion measure $d(x,z)$ is usually defined as $d(x,z) := ||x - Bz||^2$. Note that the coefficient vector $z \in \mathbb{R}^m$ is also interpreted as a feature vector of the input signal x. A set of bases B is searched so that the (expected) reconstruction error is minimized (see [4] for an example of the specific algorithms).

3.1 Information Theoretic Formalization of the Sparse Coding

Viewing from the information theoretic perspective, we can find a correspondence between the rate distortion theory and the sparse coding. The small reconstruction error naturally corresponds to the small distortion when the distortion function is defined as $d(x,z) = ||x - Bz||^2$. The sparseness of the representation ($E[||Z||_p^p] \leq T$) corresponds to the small rate ($\min I(X;Z)$). In fact, when we suppose the distribution of z to be the generalized Gaussian[1]

[1] when we let $p = 2$, the generalized Gaussian distribution becomes the usual Gaussian distribution.

with zero mean whose density is given by $p(z) = C_1 \exp(-||z/\alpha||_p^p)$, where $C_1 = (2\Gamma(1+1/p)\alpha)^{-1}$, the Shannon (joint) entropy $H(Z) := -\sum_{z\in\mathbb{R}^m} p(z)\log p(z)$ is equal to $E[\,||Z||_p^p]$ up to constant. Intuitively speaking, in the information theoretic literature, the sparseness condition $E[||Z||_p^p] \leq T$ is interpreted as the condition that the joint entropy of the coefficients $H(Z)$ is under a certain value. Thus the sparse coding (3) is essentially equivalent to the dual formulation of the rate distortion theory, which finds a coding scheme within a given rate such that the expected distortion between the source x and the decoded data from z is minimized. To avoid the primal and dual distinction, we adopt the trade-off parameter β in the same way in (2). Then the sparse coding is formulated as

$$\min_{p(Z|X)} \; H(Z) + \beta E[d(X,Z)]. \tag{4}$$

When we consider the deterministic coding scheme $x \to z$, then $I(X;Z) = H(Z) - H(Z|X) = H(Z)$ holds. In this case, the correspondence between the rate distortion and the sparse coding becomes clearer. In general, the objective functional of the rate distortion theory (2) is upper bounded by the objective functional of the sparse coding (4) because $I(X;Z) = H(Z) - H(Z|X) \leq H(Z)$, and they are equivalent when the coding scheme is deterministic.

4 Discriminative Variants of the Rate Distortion Theory

We now consider to extend the rate distortion theory and consequently the sparse coding to the class discriminative data compression or feature extraction technique. The objective functional of the rate distortion problem (2) consists of two distinct terms. The first term $I(X;Z)$ concerns the rate of the compression, and the second term $\beta E[d(X,Z)]$ concerns the distortion between the original and the compressed data. One way to make the rate distortion to have discriminative ability is adding class information to the distortion term, which is known as the information bottleneck method [5]. In the rate distortion theory, the original data x sometimes contains unimportant information, and one should dedicate to keep the class label information y rather than x itself. The information bottleneck method formulates such problems. For input data x with class y, the objective functional is represented as

$$\min_{p(Z|X)} \; I(X;Z) - \beta I(Y;Z), \tag{5}$$

where β is a trade-off parameter for the compression rate and the class information preservation. When we use a Kullback-Leibler divergence D_{KL} as d to measure the discrepancy between class conditional distributions $p(Y|X)$ and $p(Y|Z)$, the functional of the information bottleneck method is naturally induced from that of the rate distortion theory because,

$$\begin{aligned} E_{X,Z}[D_{KL}(p(Y|X)||p(Y|Z))] &= \sum_{x,z} p(x,z) \sum_{y} p(y|x) \log \frac{p(y|x)}{p(y|z)} \\ &= \sum_{x,y,z} p(x,y,z) \log \frac{p(y|x)}{p(y|z)} \\ &= H(Y|Z) - H(Y|X) + H(Y) - H(Y) \\ &= I(X;Y) - I(Y;Z), \end{aligned}$$

where we used Markov chain property $Z \leftarrow X \rightarrow Y$, that is, $p(Y|X,Z) = p(Y|X)$. From this relationship, $I(X;Z) + \beta E[d(X,Z)] = \beta I(X;Y) + \{I(X;Z) - \beta I(Y;Z)\}$ holds. Noting that $I(X;Y)$ is irrelevant to the minimization, we see the equivalence of two theories:

$$\min_{p(Z|X)} I(X;Z) + \beta E[D_{KL}(p(Y|X)||p(Y|Z))] \Leftrightarrow \min_{p(Z|X)} I(X;Z) - \beta I(Y;Z).$$

Another view of the information bottleneck method is an information theoretic formalization of data clustering. It constructs clusters maintaining the similarity of the data within the same cluster as much as possible. However, it does not contains the reconstruction error term explicitly, so it is not suitable for the discriminative variant of the sparse coding discussed here.

Now we add the class label information to the rate concerning term in (2) instead of the distortion concerning term to get the optimization problem:

$$\min I(X;Z|Y) + \beta E[d(X,Z)].$$

Then, we derive a novel discriminative variant of the rate distortion theory. The objective functional we derive here is more intuitive as a modification of the original rate distortion theory, which combines separability criterion and small reconstruction error criterion together. We first condition the rate term $I(X;Z)$ by Y, and rewrite it as follows:

$$I(X;Z|Y) = H(X|Y) + H(Z|Y) - H(X,Z|Y).$$

From the Markov chain property $Z \leftarrow X \rightarrow Y$, we get

$$\begin{aligned} H(X,Z|Y) &= -H(Y) + H(X,Z) + H(Y|X,Z) = -H(Y) + H(X,Z) + H(Y|X) \\ &= -H(Y) + \{H(X) + H(Z|X)\} + \{H(X,Y) - H(X)\} \\ &= -H(Y) + H(Z|X) + H(X,Y) = H(Z|X) + H(X|Y). \end{aligned}$$

Then we get the equality

$$\begin{aligned} I(X;Z|Y) &= H(X|Y) + H(Z|Y) - \{H(Z|X) + H(X|Y)\} \\ &= H(Z|Y) - H(Z|X). \end{aligned}$$

When we suppose $H(Z|X) \simeq 0$ again, the rate distortion theory or the sparse coding method in discriminative setting can be formulated as follows:

$$\min_{p(Z|X)} H(Z|Y) + \beta E[d(X,Z)]. \tag{6}$$

We write this objective functional with respect to $p(Z|X)$ as

$$\mathcal{L}_{rd} := H(Z|Y) + \beta E[d(X, Z)] \tag{7}$$

henceforth. Note that formally, the objective functional $\mathcal{L}_{rd}$ is obtained by replacing the entropy term $H(Z)$ in the information theoretic formalization of the sparse coding (4) by the conditional entropy of Z given Y, $H(Z|Y)$, and it seems natural as a discriminative extension.

5 Non-linear Sparse Coding with Neural Networks

In many applications, linear combinations of bases are not enough to represent fine information of input signals. To overcome the limitations of the linear methods and to get flexible representation power, we consider a non-linear sparse coding model with neural networks (three-layered perceptrons). We will give discriminative ability to this non-linear sparse coding model by applying the results derived in the previous section.

Model Description

We consider non-linear maps $f : x \to z$ and $g : z \to \hat{x}$, which correspond to the feature extraction map $x \mapsto z$ and the signal reconstruction map $z \mapsto Bz = \hat{x}$ in the linear sparse coding, respectively. As a realization of these non-linear maps, we adopt a combination of two neural networks, the one for approximating the feature extraction function f, and the other for the reconstruction function g (Fig. 1). The input of the first neural network (NN_f henceforth) is the original signal x. The output vector $z = (z_1, \cdots, z_m)$ of NN_f is restricted to range $[0, 1]^m$, where m is the number of units in the output layer of NN_f. This output vector z is used as a feature vector for classification task. Also, the vector z is given to the second neural network (NN_g) as an input. The output of the NN_g is considered as the reconstructed signal $\hat{x}$.

NN Learning Procedure

To describe a procedure for minimizing the objective functional $\mathcal{L}_{rd}$, we first need the definition of the distortion $E[d(X, Z)]$ and the conditional entropy

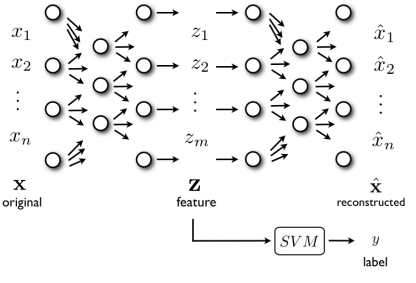


Fig. 1. A schematic diagram of the combined neural networks and SVM

$H(Z|Y)$ in our model. We define the distortion function $d : (x, z) \to d(x, z)$ by the squared difference of the input data vector x and the reconstructed data vector $\hat{x} = g(z)$. Then the distortion term is explicitly written as $E[d(X, Z)] = \frac{1}{N}\sum_x ||x - \hat{x}||^2$, where N is the number of the training data and the summation $\sum_x$ is taken over all the training data. There may be various definitions of entropy of the neural network. We simply define the entropy of the feature vectors $H(Z) = -\frac{1}{m}\sum_{j=1}^{m}\sum_{z\in\mathbb{R}^m} p_j(z)\log p_j(z)$, where $p_j(z)$ is a distribution of the j-th element of the feature vector. To calculate this entropy, we need to estimate the distribution of each element of the feature vector, which is prohibitive for online neural network training. Hence, we introduce a probabilistic interpretation of the input vector of NN_g as an approximation method. Since the output vector z of NN_f is restricted to range $[0, 1]^m$, we let the j-th unit of the input layer of NN_g become 1 with probability z_j, which is the value of the unit of the output layer of NN_f, and become 0 with probability $1 - z_j$. That is, input values to each input unit of NN_g is distributed with a Bernoulli distribution with a parameter z_j. With this approximation, we can calculate the entropy of the j-th element of the feature vector z as $h(z_j) = -z_j \log z_j - (1 - z_j)\log(1 - z_j)$. For the sake of simplicity, we will consider only two-class discrimination tasks henceforth. In order to calculate the conditional entropy of Z given Y, we add two extra bits to the input layer of NN_f. In the training phase, these extra bits are set to $(0, 1)$ or $(1, 0)$ according to the class label of each input. After training the neutral networks, these extra bits are set to $(0, 0)$.

As an usual rate distortion theory, the objective functional $\mathcal{L}_{rd}$ is iteratively minimized. For the minimization of the distortion term, we consider the combination of the NN_f and NN_g as one neural network, and used the conventional error back-propagation algorithm. For the minimization of the conditional entropy term, we only trained the first neural network NN_f by the back-propagation. We iterate the back-propagation procedure for both terms $2,000$ times in our experiments. Note that the minimization of the conditional entropy may increase the error, and vice verse. When we fix the number of iteration, the reconstruction accuracy by this model may be inferior to that of the same neural networks without entropy minimization. Our intention here is, however, not the accurate data reconstruction but the classification using reconstructive features. The sparse coding is often used for denoising images [6]. Thus the main virtue of the classification with sparse, reconstructive features is its expected robustness for the noise in input data. We exemplify the robustness of the proposed method in the next section.

Classification Procedure

We briefly explain the procedure of two class classification using the output vector z as an input vector for a support vector machine (SVM) with a linear kernel. We suppose here that the two neural networks NN_f and NN_g are already trained. The training data used to train the neural networks are transformed to the feature vectors, and used to train the SVM. In the test phase, test data are transformed to the feature vectors by trained NN_f and classified by the trained SVM.

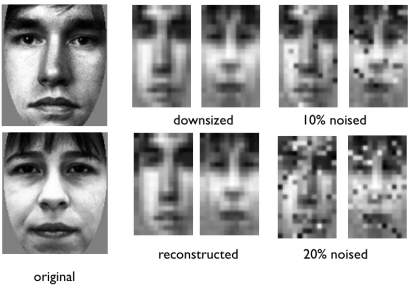


Fig. 2. Examples of the face images from the FERET database

6 Experimental Result

We demonstrate the effectiveness of our method with experiments using FERET face database [7]. The FERET database contains gender information for some of the images. We used 1000 face images from the FERET database, 500 male and 500 female images represented in pgm 256 level gray scale format. We divided it in 5 disjoint subsets for validation. Each subset contains 90 male and 90 female images as the training set, and 10 male and 10 female images as the classification accuracy test set. Each face image is cropped to contain only face region and normalized to have the same eye coordinates. The original images are downsized to 21×12 pixels, the same size adopted by an earlier study that used the FERET database [8]. We show example images in Fig. 2. The combined neural networks are trained by the procedure denoted in the previous section using the training data sets. The numbers of units in the first network NN_f are 252,100,200 for the input, the hidden, and the output layer, respectively. The numbers of units in the second network NN_g are 200,100,252. We note that while the number of the output unit of NN_f, which corresponds to the size of the bases in the sparse coding literature, is less than the input size 252, the bases is still over-complete. This is because the effective dimension of the input data which is estimated by the principal component analysis is about 110[2].

Inputting the training and test images, we can extract feature vectors for training and testing any classifier. There are some researches to discriminate gender from face images, and the method using a SVM is one of the most accurate one [8]. They used 21×12-pixel face images from the FERET database as feature vectors and input to a SVM. They used the RBF kernel for the SVM. However, we just use a linear kernel in order to confirm the usefulness of the extracted feature vector z.

As we mentioned above, classification using sparse reconstructive features are expected to be robust to noises. To see the robustness to the noise, we made two sets of noised test data by replacing 10% and 20% of pixels in each image

[2] With only about 110 largest eigenvalues, the cumulative proportion of power achieved more than 99%.

Table 1. Error rates of gender classification for the FERET database

	FERET		
noise level	0%	10%	20%
NNs & SVM	0.140(±0.082)	**0.210**(±0.042)	**0.280**(±0.027)
SVM	**0.130**(±0.045)	0.280(±0.084)	0.320(±0.104)

with random natural numbers between 0 to 255, respectively. We trained neural networks using noiseless training data, and extract features for the noiseless, 10 % noised, and 20 % noised test data respectively. The error rates of gender classification by our non-linear discriminative sparse coding denoted as "NNs & SVM" and the SVM with a linear kernel denoted as "SVM" are shown in Table 1. For the experiments with the FERET data, we show the mean and one standard deviation of 5-fold validation. From the result of these experiments, we see that the gender classification accuracy of the proposed method is comparative to that of the SVM with a linear kernel. Furthermore, it is notable that the accuracy of the proposed method shows only little degradation of classification accuracy in the presence of noises in test data, while the conventional method is rapidly degraded. From this experiment, we conclude that we can acquire feature vectors with the proposed algorithm that are effective for the classification task, especially when the test data are corrupted by noise.

7 Concluding Remarks

The main contribution of this paper is to present a novel formalization of the discriminative variant of the rate distortion theory. This discriminative rate distortion theory is more suitable for data reconstruction than the information bottleneck method, and motivated us to argue the information theoretic perspective of the sparse coding and its discriminative variants. We formalized an objective functional for the discriminative sparse coding from the viewpoint of the rate distortion theory, and embodied this concept using neural networks. We discussed that there will be at least two approaches to make the rate distortion theory to be discriminative. Adding the class label information to the distortion term is known as the information bottleneck method, and extensively studied mainly by Tishby and his coworkers. To our knowledge, however, another approach which we proposed in this paper have not explored yet, and remains to be investigated further.

In the gender classification experiment, we trained the combined neural networks by minimizing the reconstruction error and the class conditional entropy of the feature vectors. From the classification result, we see that the feature vector is effective for discriminating gender. This result suggests that minimizing the discriminative objective functional leads to a good feature vector, and it is equivalent to find a good kernel for the inner product based classifiers. Kernel learning using information theoretic criteria for classification is another direction of our future work. Though the neural network is flexible and able to realize

various non-linear transformations, we need large amount of data and time to train in general. We think other algebraic implementations of the discriminative sparse coding are possible, and the study on this direction is ongoing.

Acknowledgments. The authors would like to thank the US Army Research Laboratory for providing face databases. The authors also appreciate Yu Ogasawara for helping neural network programming. Finally, the authors express their appreciation to reviewers for their valuable comments.

References

1. Salomon, D.: Data Compression: The Complete Reference. Springer, Heidelberg (2004)
2. Olshausen, B.A., Field, D.J.: Sparse Coding of Sensory Inputs. Current Opinion in Neurobiology 14(4), 481–487 (2004)
3. Cover, T.M., Thomas, J.A.: Elements of Information Theory. John Wiley and Sons, Inc., Chichester (1991)
4. Aharon, M., Elad, M., Bruckstein, A.: K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. IEEE Trans. Signal Processing 54(11), 4311–4322 (2006)
5. Tishby, N., Pereira, F.C., Bialek, W.: The Information Bottleneck Method. In: 37-th Annual Allerton Conference on Communication, Control and Computing, pp. 368–377 (1999)
6. Elad, M., Aharon, M.: Image Denoising via Sparse and Redundant Representations over Learned Dictionaries. IEEE Trans. Image Processing 15(12), 3736–3745 (2006)
7. Phillips, P.J., Moon, H., Rizvi, S.A., Rauss, P.J.: The FERET Evaluation Methodology for Face-recognition Algorithms. IEEE Trans. Pattern Anal. Mach. Intell. 22(10), 1090–1104 (2000)
8. Yang, M.H., Moghaddam, B.: Support Vector Machines for Visual Gender Classification. In: International Conference on Pattern Recognition, vol. 1, p. 5115 (2000)

Information Dynamics and Intelligent Cooperation in Networked Societies

Jing Wang[1], Fujun Ren[2], Mei Zhu[2], and Long Wang[1]

[1] Center for Systems and Control, College of Engineering,
Peking University, Beijing, China
[2] China Research Institute for Science Popularization,
China Association for Science and Technology,
Beijing, China

Abstract. We propose a simple yet effective theoretical model for the evolutionary threshold public goods game with binary contributions (the fair personal share or nothing), incorporating the effect of collective risk. We distinguish two distinct public goods games according to whether to return the contributions when the target is not collected. For the two cases, in order to study the impact of collective risk on cooperation, we analyze dynamics of the population which can be represented by the replicator equations. It shows that high rate of loss can enhance the emergence of social cooperation and the provision of public goods. Furthermore, other elements also can promote the cooperation, such as large initial endowment and small threshold. Interestingly, for large group size, it has a positive impact on cooperation in the case of returning the donation amount, whereas a negative impact in the case of no return.

1 Introduction

Cooperation is essential to the development of human and animal societies [1,2]. However, everyone faces the temptation to defect as cooperator benefits others with a cost to itself. Why social cooperative behavior can emerge in the collective action stays in the central position in the field of evolutionary game theory [3,4]. A classical metaphor investigating this social problem is public goods game (PGG) [5], which concentrates on the origin of cooperation in the conflict between individual interest and collective interest. PGG is played in interaction groups, in which each cooperator contributes to the public pool with a cost to itself while each defector contributes nothing. Then, the public goods are distributed equally among all the individuals. As cooperators are always prone to exploitation by defectors, how to maintain cooperation? Recently, plenty of mechanisms, for example, repeated interactions, direct reciprocity, punishment [6,7,8], spatially structured populations [9,10] and voluntary participation in social interactions [11,12] have been proposed to promote the cooperation.

In absence of all the above mechanisms, we present a new one, collective risk, which might be encountered when the provision of public goods fails [13]. We introduce a target for PGG. The public goods can be provided only if the target is completed [14,15,16]. In this case, the private goods of individuals are at stake with a certain probability if the target is not reached. This mechanism is substantially different from

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 94–103, 2009.

the above referred ones as it is based on the incentive to avoid a loss but not to obtain a gain. Motivated by this new mechanism, we propose a model of PGG incorporating the effect of the collective risk to study the evolution of social cooperation theoretically.

Consider a well-mixed infinite population. From time to time, an interacting group of N agents is chosen at random among the whole population. Each of these N agents is provided with a fixed endowment W. Within such a group, there is a task to be done, for example, constructing a dam or building a defence. In order to complete the target, a final contribution T mainly from donating is required. For simplicity, we restrict participants to binary contributions (the fair personal share, T/N, or nothing, 0). Each player can choose to donate (cooperate) or not (defect). If the final target is achieved, the dam or the defence can be constructed and the private goods of each individual is prevented from losing. Cooperators can keep whatever is left in their private account, $W - T/N$, and defectors own the whole endowment W. If the target is not completed, the dam or the defence cannot be built and the risk happens with probability p ($0 \leq p \leq 1$). In this situation, two distinct scenarios are distinguished according to whether to return the donation amount or not. For both of the two cases, once the danger occurs, all participants including cooperators and defectors lose their whole private goods. Whereas the danger does not happen, in the case of returning the contribution, both defectors and cooperators own the whole endowment W; while in the case of no return, cooperators can hold what they had not invested in their private account, $W - T/N$, and defectors can keep the whole endowment W. Hence, cooperation does not dominate defection. Due to this disadvantage of cooperation, whether or not social cooperation can emerge in some instances? To answer this question, we analyze the evolutionary population dynamics using the replicator equation [17, 18]. We find that high risk rate can readily promote the emergence of social cooperative behavior. In addition, large initial endowment and small final target can advance cooperation, respectively. Most interestingly, we find a group size paradox: large group size enhances the emergence of cooperation in the situation of returning the contribution, but inhibits that in the other situation of no return.

The paper is organized as follows. For the case of returning the donation amount, the model with collective risk is introduced and discussed in Sec. 2. In Sec. 3, for the case of no return, the corresponding model is introduced and investigated. Finally, conclusions are drawn in Sec. 4.

2 Population Dynamics: Returning the Contribution

First, we consider a simple situation that the donation amount will be returned if the final goal is not completed. We suppose that the cooperator chooses to donate the fair personal share T/N and the defector donates nothing. In this case, the target T can be reached in the limiting of all donating. Note that the initial endowment W needs to be larger than T/N (otherwise, the public goods can never be provided). The remainder in the private account of a cooperator and a defector in a group consisting of n many cooperators and $N - n$ many defectors are given by

$$P_C(n) = \begin{cases} W - \dfrac{T}{N} \ , & n = N \\ (1-p)W \ , & 0 < n < N \end{cases} \tag{1}$$

and

$$P_D(n) = (1-p)W, 0 \leq n < N. \tag{2}$$

Based on the remainder, if the danger happens with certainty ($p = 1$), cooperation dominates defection; while if the danger never happens ($p = 0$), both cooperation and defection are the best choice for each participant. The strategy profile of all defection is always a stable Nash equilibrium irrespective of the risk rate.

In order to study the evolutionary behavior of the repeated game, we apply the replicator dynamics. Denote the fraction of cooperators by x and that of defectors by y. There is straightforward $x + y = 1$. The time evolution of this system is governed by the following differential equations

$$\begin{cases} \dot{x} = x(f_C - \bar{f}) \\ \dot{y} = y(f_D - \bar{f}) \end{cases} \tag{3}$$

where f_C is the expected remainder for a cooperator in a group of N players and f_D is that for a defector, $\bar{f} = xf_C + yf_D$ is the average remainder in the population.

In the well-mixed population, a group of N agents is chosen randomly, resulting in a random population composition. For a given cooperator, the probability to find him in an N persons group consisting of j other cooperators and $N - 1 - j$ defectors is

$$\binom{N-1}{j} x^j y^{N-1-j}.$$

In this case, the expected remainder in the private account for a cooperator is

$$f_C = \sum_{j=0}^{N-1} \binom{N-1}{j} x^j y^{N-1-j} P_C(j+1). \tag{4}$$

Similarly, a given defector finds him in a group composed of j many cooperators and $N - 1-$ many other defectors with the same probability

$$\binom{N-1}{j} x^j y^{N-1-j}.$$

Thus, the expected remainder for a defector is

$$f_D = \sum_{j=0}^{N-1} \binom{N-1}{j} x^j y^{N-1-j} P_D(j). \tag{5}$$

Substituting equation $x + y = 1$ and using

$$\sum_{j=0}^{N-1} \binom{N-1}{j} x^j (1-x)^{N-1-j} = 1,$$

Eq. (4) and (5) can be transformed into

$$\begin{aligned} f_C &= (1-p)W + x^{N-1}(pW - \frac{T}{N}) \\ f_D &= (1-p)W. \end{aligned} \tag{6}$$

Further substituting $x + y = 1$ into the first equation of Eq. (3), the dynamics of $x(t)$ is given by

$$\dot{x} = x(1-x)(f_C - f_D) = x^N(1-x)(pW - \frac{T}{N}). \tag{7}$$

Let the right of Eq. (7) equal zero, we get only two fixed points of the population, $x = 0$ and $x = 1$. Note that under the condition of $pW - T/N > 0$, the fraction x increases monotonously in the interval $0 < x < 1$. Hence, if a small perturbation is added to the fixed point $x = 0$, the fraction x tends to another fixed point $x = 1$. It shows that equilibrium $x = 0$ is an unstable fixed point. Similarly, if $pW < T/N$, the fraction x decreases monotonously in the interval $0 < x < 1$. When we add a small perturbation to the equilibrium $x = 0$, the fraction x goes back to $x = 0$ immediately. The fixed point $x = 0$ is stable in this case (see Table 1). In order to judge the stability of the other equilibrium $x = 1$, we analyze the Jacobian of Eq. (7) which is given by

$$J = Nx^{N-1}(1-x)(pW - \frac{T}{N}) - x^N(pW - \frac{T}{N}).$$

Substituting $x = 1$ into the Jacobian, we obtain

$$J(x=1) = -(pW - \frac{T}{N}).$$

If $pW > T/N$, the Jacobian is always below zero, leading to a stable fixed point $x = 1$, otherwise, it is unstable fixed point (see Table 1).

Thus, the condition $pW > T/N$ can lead to all donation, maintaining cooperation. Moreover, increasing the risk rate p, the group size N and the initial endowment W, respectively, and decreasing the target T, can make the inequality $pW > T/N$ be satisfied more easily. Under the influence of these measures, the population is easier to reach the state of all cooperation, as the equilibrium $x = 1$ is stable whereas $x = 0$ is unstable. Therefore, in this case, high risk rate, large group size, big initial endowment as well as small target sum can promote the cooperation.

Table 1. Stability of equilibria in the case of returning the contribution

	$p > \frac{T}{NW}$	$p < \frac{T}{NW}$
$x = 0$	unstable	stable
$x = 1$	stable	unstable

3 Population Dynamics: Not Returning the Contribution

If the contribution is not returned when the final goal is not achieved, the situation is more complex than that discussed above. We also assume that the donator contributes T/N while the defector donates nothing. The public goods can be provided if and only if all participants cooperate. In this case, the remainder in the private account of a cooperator and a defector in a group consisting of n many cooperators and $N-n$ many defectors are given by

$$P_C(n) = \begin{cases} W - \frac{T}{N} & , \quad n = N \\ (1-p)(W - \frac{T}{N}) & , \ 0 < n < N \end{cases}$$

and

$$P_D(n) = (1-p)W, 0 \leq n < N.$$

Based on the remainder, it is found that if the danger happens with certainty ($p = 1$), they are better off donating than defecting. If the danger never happens ($p = 0$), defection is the best choice. However, if the danger happens with an nonzero and non-one probability, some individuals want to save their interests by cooperating, whereas the others are willing to gamble for the danger. If the donation amount exceeds the expected loss arising from the risk, i.e., $T/N > pW$ ($p < T/(NW)$), defection dominates cooperation. The strategy profile of all defection is the unique Nash equilibrium. If $T/N < pW$ ($p > T/(NW)$), everyone is better off if the public goods is provided than not. The "all cooperation" set of strategies is a Nash equilibrium as no one can increase his gains by changing his own strategy while the others stay the same. However, it is an unstable equilibrium because once one player changes his strategy, the remaining players can increase their expected gains by altering their strategies. These changes lead to another Nash equilibrium, the profile of "all defection".

The above discussion is about the one-shot game. Now, we investigate the evolutionary behavior of the repeated game using the replicator dynamics. Analogously to the calculations in Sec. 2, the expected remainder in the private account for a cooperator and a defector is, respectively, given by

$$\begin{aligned} f_C &= \sum_{j=0}^{N-1} \binom{N-1}{j} x^j y^{N-1-j} P_C(j+1) \\ &= (1-p)(W - \frac{T}{N}) + x^{N-1} p(W - \frac{T}{N}), \\ f_D &= \sum_{j=0}^{N-1} \binom{N-1}{j} x^j y^{N-1-j} P_D(j) \\ &= (1-p)W. \end{aligned}$$

Accordingly, the replicator dynamics in the case of no return is derived as

$$\begin{aligned} \dot{x} &= x(1-x)(f_C - f_D) \\ &= x(1-x)[x^{N-1} p(W - \frac{T}{N}) - \frac{T}{N}(1-p)]. \end{aligned} \tag{8}$$

Its Jacobian is written by

$$J = (1-2x)[x^{N-1}p(W-\frac{T}{N}) - \frac{T}{N}(1-p)] + x^{N-1}(1-x)(N-1)p(W-\frac{T}{N}). \quad (9)$$

We focus on the steady state which the system evolves to after the transient behavior. Let $\dot{x} = 0$, we get all fixed points of the system whose stability is dependent on the value of the risk rate p. Hence, we distinguish three situations as follows.

(1) If $p = 0$, Eq. (8) reduces to

$$\dot{x} = -\frac{T}{N}x(1-x).$$

This system has only two boundary fixed points, $x = 0$ and $x = 1$. Its Jacobian is $J = -\frac{T}{N}(1-2x)$. At $x = 0$, the Jacobian becomes $J(x=0) = -T/N < 0$, leading to a stable equilibrium. At $x = 1$, the Jacobian is $J(x=1) = T/N > 0$, leading to an unstable equilibrium. In this case, the population system converges to the state of all defectors. No one wants to contribute his savings to avoid an impossible risk.

(2) If $p = 1$, Eq. (8) reduces to

$$\dot{x} = (W-\frac{T}{N})x^N(1-x).$$

There are also only two boundary fixed points $x = 0$ and $x = 1$. It is worth noting that the initial endowment W is needed to exceed the donation amount T/N. The fraction x is an increasing function of t in the interval $0 < x < 1$ irrespective of the initial state as $\dot{x} > 0$ is always satisfied. Departure of the trajectory from the point $x = 0$ shows that $x = 0$ is an unstable equilibrium. For the equilibrium $x = 1$, the Jacobian is

$$J(x=1) = -(W-\frac{T}{N}) < 0.$$

Thus, the fixed point $x = 1$ is stable, leading to extinction of defectors. We can find that all individuals in this population choose to donate eventually. In fact, if the danger happens with certainty, the best response for each individual is to donate unconditionally.

(3) If $0 < p < 1$, we also obtain two boundary fixed points $x = 0$ and $x = 1$ from $\dot{x} = 0$. For the point $x = 0$, Eq. (9) becomes

$$J(x=0) = -\frac{T}{N}(1-p) < 0.$$

It determines that the equilibrium $x = 0$ is a stable boundary fixed point. For the equilibrium $x = 1$, the Jacobian is

$$J(x=1) = \frac{T}{N} - pW.$$

It is below zero under the condition of $p > T/(NW)$ and exceeds zero in the case of $p < T/(NW)$. Thus, the boundary fixed point $x = 1$ is a stable equilibrium if $p > T/(NW)$, while an unstable equilibrium, otherwise. That is to say, if the expected loss is larger than the donation amount, everyone is likely to donate and the population may be full

of cooperators. Whereas if the expected loss is smaller than the donation amount, no one want to cooperate as they think it is not worthy to protect their savings by donating much more amount.

In addition, it should be illuminated that for $p < T/(NW)$, there is no interior equilibrium and for $p > T/(NW)$, there is only one in the interval (0,1). In order to show this, we set

$$F(x) = x^{N-1}p(W - \frac{T}{N}) - \frac{T}{N}(1-p).$$

The partial derivative of $F(x)$ with respect to x is

$$F'(x) = (N-1)x^{N-2}p(W - T/N) > 0.$$

The polynomial $F(x)$ is an increasing function of the fraction x. It is worth noting that

$$F(0) = -\frac{T}{N}(1-p) < 0$$

and

$$F(1) = p(W - \frac{T}{N}) - \frac{T}{N}(1-p).$$

In the case of $p < T/(NW)$, the result that no root of $F(x)$ exists in the interval (0,1) is derived from $F(1) < 0$ (see Fig. 1). While if $p > T/(NW)$, inequalities $F(1) > 0$, $F(0) < 0$ leads to only one root in (0, 1). Let $F(x) = 0$, we obtain the unique interior fixed point as

$$x^* = \sqrt[N-1]{\frac{T}{N}(1-p)/[p(W - \frac{T}{N})]}.$$

At the interior equilibrium x^*, the Jacobian is

$$J(x = x^*) = \frac{T}{N}(1-x^*)(1-p)(N-1) > 0,$$

leading to an unstable equilibrium.

To sum up, the collective risk brings in a rich dynamics (see Table 2). When there is no risk ($p = 0$), the population is full of defectors ultimately irrespective of the initial state. This phenomenon where defection dominates cooperation persists until the expected loss equals the donation amount ($p = T/(NW)$). With further increase of the risk rate, the evolutionary dynamics transforms to defector and cooperator bistable ($T/(NW) < p < 1$). In this situation, both of strategy "cooperation" as well as "defection" are the best response for each individual. Which state the population approaches to eventually depends on the initial state. Furthermore, when the risk happens with certainty ($p = 1$), cooperator dominates defector and defector vanishes. The population is composed of all donators (Fig. 2).

Moreover, in the case of $p > T/(NW)$, the states $x = 0$ and $x = 1$ are both stable nodes. Which state the system evolves to eventually depends on which attraction basin the initial state is located. The interior equilibrium x^* separates the attraction basins of $x = 0$ and $x = 1$. If the initial state meets $x_0 < x^*$, the system ends up with all individuals donating nothing, otherwise, each individual chooses to donate the fair share. Hence,

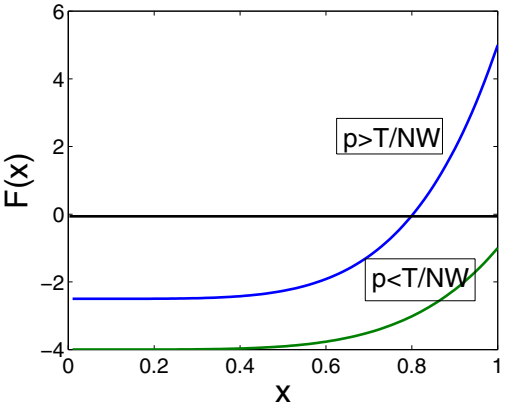


Fig. 1. The difference between the expected remainder of cooperators and defectors $F(x)$ as a function of the fraction of cooperators. If $p > T/(NW)$, there is only one interior fixed point in the interval (0,1), whereas no interior equilibrium, otherwise.

Table 2. Stability of equilibria in the case of no return

	$p = 0$	$p = 1$	$0 < p < 1$	
			$p < \frac{T}{NW}$	$p > \frac{T}{NW}$
$x = 0$	stable	unstable	stable	stable
x^*	—	—	—	unstable
$x = 1$	unstable	stable	unstable	stable

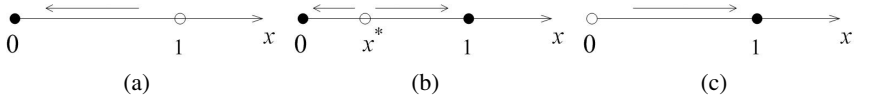


Fig. 2. Evolutionary dynamics under the change of the risk rate p. Filled and open circles represent the stable and the unstable fixed points, respectively. Arrows indicate the evolutionary direction. (a) If $0 \leq p < T/(NW)$, defection dominates cooperation;(b) If $T/(NW) < p < 1$, cooperation and defection are bistable; (c) If $p = 1$, cooperation dominates defection.

decreasing the equilibrium x^* broadens the attraction basin of $x = 1$, making it easier to reach the state of all cooperators and promoting the emergence of cooperation.

Observing $x^* = \sqrt[N-1]{\frac{T}{N}(p^{-1} - 1)/(W - \frac{T}{N})}$, we find that the interior equilibrium x^* is a decreasing function of the risk rate p. Therefore, increasing the risk rate reduces the attraction basin of all defection, meanwhile, broadens that of all donating. Cooperation is favored when the risk rate is high. This result agrees well with the fact in our society. The higher the risk rate, the more the donators, the higher the probability that the population reaches the target sum.

Similarly, the attraction basin of state $x = 0$ is shortened with the increase of the endowment W according to $x^* = \sqrt[N-1]{\frac{T}{N}(1-p)/[p(W-\frac{T}{N})]}$, and the decrease of the final target T derived from $x^* = \sqrt[N-1]{(1-p)/[p(\frac{NW}{T}-1)]}$. Hence, it is easier for the population to be full of donators with larger endowment and smaller target sum. In fact, the larger the remainder left in the personal account, the higher the aspiration to avoid risk, the larger the probability to reach the state of all cooperation.

Besides, noticeably, when the group size N tends to infinite, the interior equilibrium x^* is

$$\lim_{N\to\infty} x^* = \lim_{N\to\infty} \sqrt[N-1]{\frac{T(1-p)}{p(WN-T)}} = \frac{\lim_{N\to\infty} \sqrt[N-1]{T(1-p)}}{\lim_{N\to\infty} \sqrt[N-1]{p(WN-T)}} = 1.$$

It indicates that for sufficiently large group size, the attraction basin of the state $x = 1$ is approximate to zero. Thus, large group size has a negative impact on the emergence of cooperation in the case of no return.

Actually, this result does not contradict the conclusion in Sec. 2. In the above section, increasing the group size decreases the donation amount. It is easier to satisfy the condition that the expected loss is larger than the donation amount, encouraging participants to cooperate. Whereas in Sec. 3, increasing the group size N means that the number of donators needed for the final target is raised. So it becomes harder to meet the number, decreasing the probability to reach the threshold. Too many failure of the target depresses the cooperative behavior. Accordingly, in Sec. 3, large group size hinders the emergence of cooperation.

4 Conclusions

In summary, we have proposed an evolutionary public goods game with threshold incorporating the effects of collective risk, which mimics the characteristics of the collective-risk social dilemma. Compared with the most studied mechanisms [8,10], risk is an original factor introduced in the public goods game. Because the provisions of public goods in the other models are always based on the gains from the public goods, whereas provision of public goods in our model stems from avoiding the loss attributed to the collective risk, meanwhile, participants receive no income. We distinguish two types of public goods game according to whether to return the contributions when the final target is not reached. In both of the two cases, we found that if the rate of loss p is less than $T/(NW)$, the contribution to the public goods exceeds the expected loss. In this situation, there is no incentive to cooperate. The population evolves to the state of all defectors. If $p > T/(NW)$, the expected loss exceeds the donation amount, all cooperation is the final state in the case of returning the contribution. While in the case of no return, the two strategies, donating the fair share and defecting, are bistable. For the two cases, probability to reach the state of all cooperators enhances as p increases. It is worth noting that in the case of no return, with the increase of the risk rate p, a rich population dynamics appears. Scenarios of defection dominance, defection as well

as cooperation bistable, and cooperation dominance occur successively. In addition, we also found cooperation can be significantly enhanced by large initial endowment and small target sum. Interestingly, group size plays an opposite role in cooperation in the two cases. In the case of returning the donation amount, large group size raises the probability to reach the state of all cooperators; while in the other case, large group size suppresses the emergence of cooperation.

Acknowlegement. This work was supported by China Research Institute for Science Popularization.

References

1. Binmore, K.G.: Playing Fair: Game Theory and the Social Contract. MIT Press, Cambridge (1994)
2. Colman, A.M.: Game Theory and Its Applications in the Social and Biological Sciences. Butterworth-Heinemann, Oxford (1995)
3. Hauert, C., Michor, F., Nowak, M.A., Doebeli, M.: Synergy and Discounting of Cooperation in Social Dilemmas. J. Theor. Biol. 239, 195–202 (2006)
4. Okada, D., Bingham, P.M.: Human Uniqueness-Self-Interest and Social Cooperation. J. Theor. Bio. 253, 261–270 (2008)
5. Kagel, J.H., Roth, A.E. (eds.): The Handbook of Experimental Economics. Princeton University Press, Princeton (1995)
6. Sigmund, K., Hauert, C., Nowak, M.A.: Reward and Punishment. Proc. Natl. Acad. Sci. USA 98, 10757–10762 (2001)
7. Fehr, E., Gachter, S.: Altruistic Punishment in Humans. Nature 415, 137–140 (2002)
8. Hauert, C., Traulsen, A., Brandt, H., Nowak, M.A., Sigmund, K.: Via Freedom to Coercion: The Emergence of Costly Punishment. Science 316, 1905–1907 (2007)
9. Szabo, G., Hauert, C.: Phase Transitions and Volunteering in Spatial Public Goods Games. Phys. Rev. Lett. 89, 118101 (2002)
10. Santos, F.C., Santos, M.D., Pacheco, J.M.: Social Diversity Promotes the Emergence of Cooperation in Public Goods Games. Nature 454, 213–216 (2008)
11. Hauert, C., Monte, S.D., Hofbauer, J., Sigmund, K.: Replicator Dynamics for Optional Public Good Games. J. theor. Biol. 218, 187–194 (2002)
12. Hauert, C., Holmes, M., Doebeli, M.: Evolutionary Games and Population Dynamics: Maintenance of Cooperation in Public Goods Games. Proc. R. Soc. B 273, 2565–2570 (2006)
13. Milinski, M., Sommerfeld, R.D., Krambeck, H.J., Reed, F.A., Marotzke, J.: The Collective-Risk Social Dilemma and the Prevention of Simulated Dangerous Climate Change. Proc. Natl. Acad. Sci. USA 105, 2291–2294 (2008)
14. Bagnoli, M., Mckee, M.: Voluntary Contribution Games: Efficient Private Provision of Public Goods. Econ. Inq. 29, 351–366 (1991)
15. Bach, L.A., Helvik, T., Christiansen, F.B.: The Evolution of n-Player Cooperational Threshold Games and ESS Bifurcations. J. Theor. Biol. 238, 426–434 (2006)
16. Pacheco, J.M., Santos, F.C., Souza, M.O., Skyrms, B.: Evolutionary Dynamics of Collective Action in N-person Stag Hunt Dilemmas. Proc. R. Soc. B 276, 315–321 (2009)
17. Taylor, P., Jonker, L.: Evolutionary Stable Strategies and Game Dynamics. Math. Biosci. 40, 145–156 (1978)
18. Hofbauer, J., Sigmund, K.: Evolutionary Games and Population Dynamics. Cambridge University Press, Cambridge (1998)

Statistical Dependency of Image Wavelet Coefficients: Full Bayesian Model for Neural Networks

Xingming Long[1] and Jing Zhou[2]

[1] Department of Physics, Chongqing Normal University, 400047, Chongqing, China
[2] College of Electrical Engineering, Chongqing University, 400044, Chongqing, China
lennydragon@163.com

Abstract. A novel method based Full Bayesian Model for Neural Network (FBMNN) to study the statistical dependency of wavelet coefficients is presented. To overcome the ignorance of the relationship between wavelet coefficients, we introduce the FBMNN to model joint probability density distribution (JPDF) of Child and Parent wavelet coefficients. According to the characteristics of the suggested FBMNN-JPDF model, its parameters are estimated by reversible jump MCMC (rjMCMC) algorithm. Finally, a practical application on denoising image by using the FBMNN-JPDF model is demonstrated and the result shows that the suggested method can express wavelet coefficients dependency efficiently.

1 Introduction

The prior distribution of the wavelet coefficients plays an important role in image processing based Bayesian method in wavelet domains. In the past years, many authors use the Gaussian functions, Laplacian and mixture Gaussian model to analysis the statistical characteristics of wavelet coefficients. However, most of them focus on independent scale while the relationship between the scales is ignored. Recently, M.S.Crouse proposed hidden Markov model (HMM) to study the relationship between the scales[1].

In this paper, we present that dependency of the wavelet coefficients is studied by the JPDF of Child and Parent wavelet coefficients, furthermore, the JPDF is approximated by the FBMNN. As we all know, artificial neural networks (ANN) are powerful nonlinear approximation tools, which rely on structured combination of many parameterized basis function to perform regression, classification and density estimation. They can approximate any continuous function arbitrarily well as the number of neurons increased without bound. In addition, they have been successfully applied to many and varied fields, including speech recognition, financial modeling and medical diagnosis. But it is difficult to estimate the common ANN parameters, especially for the number of the basis functions. So, we propose that FBMNN introduced by C.Andrieu[2] is used to model the JPDF for its efficiency in parameters estimation than the common ANN.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 104–109, 2009.

The remainder of the paper is organized as follows: firstly, we present the principles of the FBMNN. Secondly, we study the JPDF of Child and Parent wavelet coefficients, subsequently, we formalize the FBMNN-JPDF model and rjMCMC algorithm is derived to deal with the suggested model. Thirdly, the performance of the suggested model is illustrated by denoising image. Finally, some conclusions are drawn in conclusion section.

2 FBMNN

Many physical processes may be described by the following nonlinear, multivariate input-output ANN mapping:

$$\mathbf{y}_t = \sum_{j=0}^{k} a_j \phi(\| x_t - \mu_j \|)(1 - \mathrm{Dirc}(k)) + b + \beta^T x_t + \mathbf{n}_t \qquad k = 0,1,2\cdots, \tag{1}$$

where $x_t \in R^d$ corresponds to a group of d-dimensions input variables, $y_t \in R^c$ to the c-dimensions target variables, $\mathrm{Dirc}(\bullet)$ is Dirac Function, $\| \bullet \|$ denotes an Euclidean distance metric, $\phi(\rho)$ denotes basis function, $\mu_j \in R^d$ denotes the j-th RBF center for a model with k RBFs, $a_j \in R^c$ the j-th RBF amplitude , $b \in R^c$ and $\beta \in R^d \times R^c$ are linear regression parameters. The noise sequence $\mathbf{n}_t$ is assumed to be zero-mean and σ_t^2-variance white Gaussian.

For convenience, Eq.(1) is expressed in vector-matrix form:

$$\mathbf{y} = \mathbf{D}(\mu_{1:k,1:d}, x_{1:N,1:d})\alpha_{1:1+d+k,1:c} + \mathbf{n} \tag{2}$$

Given the data set $\{x, y\}$, our objective is to estimate the unknown NN parameters: the number k of RBFs and their parameters $\theta = \{\alpha_{1:m,1:c}, \mu_{1:k,1:d}, \sigma^2_{1:c}\}$.

There are many approaches to reach the objective. Here, considering the Bayesian inference of k, θ ,C.Andrieu setup a natural hierarchical structure called FBMNN [4].According to FBMNN, Eq.(2) can be expressed efficiently by introducing hyper-parameters $\psi = \{\Lambda, \delta^2\}$, here $\delta^2 \sim$ *Inverse* Gama $(\alpha_{\delta^2}, \beta_{\delta^2})$ $(\alpha_{\delta^2} = 2, \beta_{\delta^2} > 0)$ and $\Lambda \sim \mathrm{Gama}(0.5 + \varepsilon_1, \varepsilon_2)(\varepsilon_1, \varepsilon_2 << 1)$. So, Bayesian inference of k, θ conditioned on the data set $\{x, y\}$ is:

$$p(k, \theta, \psi | x, y) = p(k, \alpha_{1:m}, \mu_{1:k}, \sigma^2, \Lambda, \delta^2 | x, y) \propto p(y | k, \alpha_{1:m}, \mu_{1:k}, \sigma^2, \Lambda, \delta^2, x) p(k, \alpha_{1:m}, \mu_{1:k}, \sigma^2, \Lambda, \delta^2). \tag{3}$$

Eq.(3) integrates with respect to $\alpha_{1:m}$ (Gaussian distribution) and with respect to σ^2 (inverse Gamma distribution) to obtain the posterior: $p(k, \mu_{1:k}, \Lambda, \delta^2 | x, y)$.

In a word, We can firstly use the standard methods such as: ML, MAP ,MCMC or rjMCMC to estimate the parameters: $k, \mu_{1:k}$, ,then compute the coefficients $\alpha_{1:m}$ by least squares according to Eq.(2) and use the conventional estimation of the variance to get the White Gaussian noise σ^2.

3 FBMNN-JPDF Model

3.1 JPDF of Nature Image

Researches show that the PDF of nature image intensity is similarity. Based on lots of experiments, we find that the JPDF of the nature image's Child and Parent wavelet coefficients has also similarity. Fig.2 (b) is illustrated the similar characteristics. The JPDF is reached by the following steps: firstly, the nature image $\boldsymbol{f}(\boldsymbol{m},\boldsymbol{n})$ is decomposed (Fig.2(a)) by Mallat Pyramid[3],

$$\begin{aligned} \boldsymbol{f}(\boldsymbol{m},\boldsymbol{n}) = & \sum_{k,l}[(\boldsymbol{A}_{\boldsymbol{M}}^{\boldsymbol{d}}\boldsymbol{f})_{k,l}\cdot\phi_{M,k}(\boldsymbol{m})\phi_{M,l}(\boldsymbol{n})] + \sum_{j=1}^{M}\sum_{k,l}[(\boldsymbol{D}_{\boldsymbol{j}}^{\boldsymbol{H}}\boldsymbol{f})_{k,l}\cdot\phi_{j,k}(\boldsymbol{m})\psi_{j,l}(\boldsymbol{n})] + \\ & \sum_{j=1}^{M}\sum_{k,l}[(\boldsymbol{D}_{\boldsymbol{j}}^{\boldsymbol{V}}\boldsymbol{f})_{k,l}\cdot\psi_{j,K}(\boldsymbol{m})\phi_{j,l}(\boldsymbol{n})] + \sum_{j=1}^{M}\sum_{k,l}[(\boldsymbol{D}_{\boldsymbol{j}}^{\boldsymbol{D}}\boldsymbol{f})_{k,l}\cdot\psi_{j,K}(\boldsymbol{m})\psi_{j,l}(\boldsymbol{n})] \end{aligned} \tag{4}$$

Then, calculate the JPDF of Parent and Child wavelet components.

3.2 JPDF Based FBMNN

Considering the features of JPDF: statistical similarity and nonlinearity, We introduce the FBMNN model described as section 2 to model the relationship of the vertical wavelet coefficients between Child and Parent. that is:

$$\begin{cases} \mathbf{y}_t = \sum_{j=1}^{k} a_j \phi(\| x_t - \mu_j \|) + b + \beta^T x_t + \mathbf{n}_t \qquad k = 1,2\cdots \\ \mathbf{y}_t = [p(V^C, V^P)]_{1:N,1} \\ x_t = [V^C \quad V^P]_{1:N,1:2} \end{cases}, \tag{5}$$

where $\boldsymbol{d}=2,\boldsymbol{c}=1$, the number of RBFs is from 1 to $\boldsymbol{k}_{\max}$ ($\boldsymbol{k}_{\max}=\boldsymbol{N}-(\boldsymbol{d}+1)$), $\boldsymbol{V}^{\boldsymbol{C}}, \boldsymbol{V}^{\boldsymbol{P}}$ denote the vertical wavelet coefficients of Child and Parent respectively, and $p(V^C, V^P)$ denotes the values of JPDF. So, the parameters of the suggested JPDF-FBMNN can visualize with a directed acyclic graphical model(DAGM) as shown in Fig.1

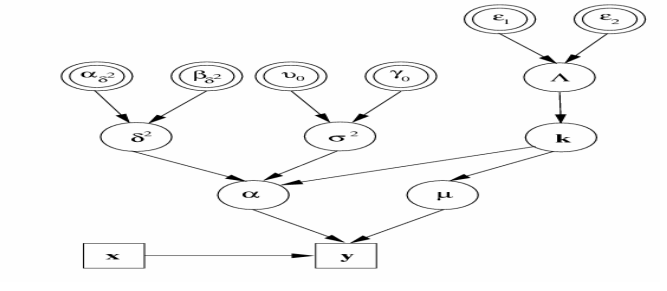


Fig. 1. DAGM for the JPDF- FBMNN

3.3 Parameters Estimation with rjMCMC Algorithm

The estimation of the joint posterior distribution Eq.(5), where k is unknown, is very complex by the Bayesian computation. So, we use the reversible jump MCMC (rjMCMC) which has been introduced by Green[4] and this algorithm is capable of jumping between subspace of different dimensions. For our problem, the following moves have been slected:

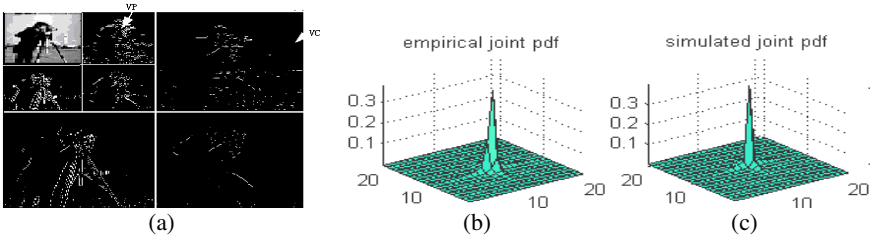
1) grow/decrease the network: Birth of a new RBF and Death of an existing RBF.
2) local adjustment: Merge a RBF with its closest neighbor RBF and Split a RBF into two neighbors.
3) update the RBF centers.
The main steps of the algorithm are as follows:

```
initialization: set {k, μ_{l:k}, Λ, δ^2}^(0)
Loop i
(1) uniformly sample u at [0,1]
(2) if (u<grow/decrease probability),then Birth/death
of a new RBF move
     Else if  (u<local adjustment probability),then
Merge/Split of a new RBF move
     Else Update the RBF centers
End if
  (3) Do α_{l:m} and σ^2 samples
End loop
```

4 Simulation and Result

In the MATLAB6.5,we choose the probabilities of the suggested moves: birth, death, merge, split and update to satisfy the following constraints: 1) $b_k + d_k + m_k + s_k + u_k \in [0.25,1]$ for all $1 \le \boldsymbol{k} \le \boldsymbol{k}_{\max}$; 2) $\boldsymbol{m_k} = 0$,that is merge move is not permitted for k=1; 3) $\boldsymbol{b_k} = \boldsymbol{s_k} = 0$,the birth and split are not allowed for $\boldsymbol{k} = \boldsymbol{k}_{\max}$; 4)except the cased:1)~3),we adopt the $\boldsymbol{s_k} = b_k = 0.25 \min\{1, \frac{p(k+1)}{p(k)}\}$. $m_{k+1} = d_{k+1} = 0.25 \min\{1, \frac{p(k+1)}{p(k)}\}$, where $\boldsymbol{p}(\boldsymbol{k})$ is the prior probability of model.

In our simulation, $\phi(\rho)$ is chosen to be Gaussian $\phi(\rho) = \exp(-\lambda\rho^2)$,We use the 400 observations of the nature image wavelet coefficients data set{ $\boldsymbol{V^C}$, $\boldsymbol{V^P}$ }to train our model. Fig.2 (b) shows the 3D plots of the training data from empirical JPDF and Fig.2 (c) is the simulated JPDF by the suggested method after 3000 iterations. The RMSE of this method is 0.25% less than that of mixture Gaussian method[6].



(a) (b) (c)

Fig. 2. (a) Mallat Pyramid *(M=2)* of "cameraman" by "db4" wavelet function; (b) empirical JPDF data; (c) simulated JPDF

Further, we use the result to denoise a image.Suppose a nature image $\boldsymbol{f}$ is noised by i.i.d. gaussian noise $\boldsymbol{n}$ with zero mean, the noisy image $\boldsymbol{g}$ is $\boldsymbol{g} = \boldsymbol{f} + \boldsymbol{n}$. In the orthogonal wavelet domain, $y = w + n$ where, $\boldsymbol{y}$, $\boldsymbol{w}$ and $\boldsymbol{n}$ is the orthogonal wavelet coefficients of $\boldsymbol{g}$, $\boldsymbol{f}$ and $\boldsymbol{n}$ respectively.

Let $\boldsymbol{y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2\}$, $\boldsymbol{w} = \{\boldsymbol{w}_1, \boldsymbol{w}_2\}$, $\boldsymbol{n} = \{\boldsymbol{n}_1, \boldsymbol{n}_2\}$, $\boldsymbol{w}_1 = \boldsymbol{V}^{\boldsymbol{C}}$, $\boldsymbol{w}_2 = \boldsymbol{V}^{\boldsymbol{P}}$, and y1,y2,n1,n2 denote observed image wavelet coefficients and noise wavelet coefficients respectively. With the help of Bi-variate Bayes rules and Poster Expected estimation algorithm[5], the estimator of the original image wavelet coefficients is:

$$\begin{bmatrix} \hat{\boldsymbol{w}}_1 \\ \hat{\boldsymbol{w}}_2 \end{bmatrix} = \boldsymbol{E}_{\boldsymbol{w}_1,\boldsymbol{w}_2}[\boldsymbol{p}(\boldsymbol{w}_1, \boldsymbol{w}_2 \mid \boldsymbol{y}_1, \boldsymbol{y}_2)] = \boldsymbol{E}_{\boldsymbol{w}_1,\boldsymbol{w}_2}[\boldsymbol{p}(\begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{bmatrix} - \begin{bmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \end{bmatrix}) \boldsymbol{p}(\begin{bmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \end{bmatrix})], \tag{6}$$

where $p(\begin{bmatrix} w_1 \\ w_2 \end{bmatrix})$ is the approximated PDF by Eq.5 and $p(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} - \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}) = \frac{1}{2\sigma^2} \exp(\frac{-(n_1^2 + n_1^2)}{2\sigma_n^2})$,

σ_n^2 means noise square derivation.

Giving a nature image 1024*1024 "remote image" Fig.3 (a) and the observed noisy image Fig.3 (b). In "db4" wavelet domain, the suggested method result is Fig.3 (d) in contrast to Wiener method result in Fig.3 (c). We can see the suggested method can get good visual feelings.

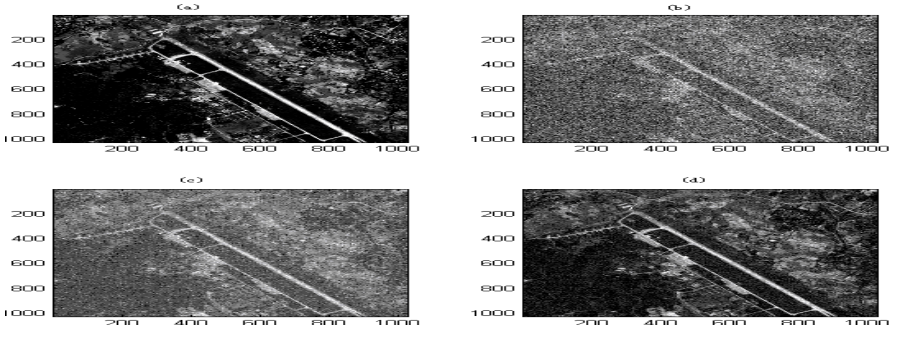


Fig. 3. (a) original image; (b) noisy image*(SNR=3.33dB)*; (c) Wiener ; (d) the suggested method

In addition, the performances by SNR and RMSE criterion are compared between the Wiener method and the suggested method, when the noisy image SNR is 0.36dB, the suggested method improves the performance over Wiener by 4.25dB in SNR and 12.8% in RMSE. With the change of the SNR of the noisy image, the performance is different; when SNR is lower than 6dB, the suggested method's performance improves greatly; On the other hand, when the SNR is higher than 15dB, the suggested method's performance is less efficient than Wiener method.

5 Conclusion

We introduce the FBMNN to model nature image's JPDF of Child and Parent wavelet coefficients. This method overcomes the ignorance of the relationship between the wavelets coefficients and improves the performance of prior distribution. Following the procedure of modeling the JPDF-FBMNN, we do a practical application to share the efficiency of the suggested model.

References

1. Crouse, M.: Wavelet-based Statistical Signal Processing Using Hidden Markov Models. IEEE Trans.Signal Processing. 46, 886–902 (1998)
2. Andrieu, C.: Sequential Bayesian Estimation and Model Selection Applied to Neural Networks. Technical Report CUED/F-INFENG/TR341,Cambridge University
3. Mallat, S.: Singularity Detection and Processing with Wavelets. IEEE Transaction on information theory 38, 617–643 (1992)
4. Green, P.J.: Reversible Jump MCMC Computation and Bayesian Model Determination. Biometrika 82, 711–732
5. Sendur, L.: Subband Adaptive Image Denosing via Bivariate Shrinkage. IEEE ICIP, III577–III 580 (2002)
6. Xing, L., Jing, Z.: A Novel Image-denoising Method Based on Mixture Gaussian Model. In: Proceedings of the 6th international progress on WAAMT. World Scientific Publishing Co., Singapore (2005)

The Bounds on the Rate of Uniform Convergence of Learning Process on Uncertainty Space

Xiankun Zhang, Minghu Ha*, Jing Wu, and Chao Wang

College of Mathematical and Computer Sciences, Hebei University,
Baoding 071002, China
zhangxiankun08@163.com, mhha@mail.hbu.edu.cn,
wujing198405@126.com, wang222chao@163.com

Abstract. Statistical Learning Theory on uncertainty space is investigated. The definitions of empirical risk functional, expected risk functional and empirical risk minimization principle on uncertainty space are introduced. Based on these concepts, the bounds on the rate of uniform convergence of learning process are given, which estimate the value of achieved risk for the function minimizing the empirical risk and the difference between the value of achieved risk and the value of minimal possible risk for a given set of functions.

Keywords: Uncertain measure, Expected risk functional, Empirical risk functional, Empirical risk minimization principle, Bounds of the rate of uniform convergence.

1 Introduction

Statistical Learning Theory (SLT) [1-3] was introduced by Vapnik et al., concerning itself mainly the statistic principles when samples are limited. SLT provides a new framework for some general learning problems, the key idea is to study the generalization abilities of learning machine through controlling the learning machine's capacity. Based on this theory, a novel pattern recognition method - Support Vector Machine (SVM) was provided. Comparing to the machine learning methods in the past, SLT and SVM show great advantages. Many scholars have begun to pay attention to the academic field [1-10], it is believed that SLT and SVM are becoming a new hot area in the field of machine learning [4-10].

The bounds on the rate of uniform convergence of learning process are important parts of the SLT. They determine the generalization abilities of learning machine by using the empirical risk minimization principle (ERM), and are important foundation for analyzing the performance of learning machine and developing new learning algorithms. By estimating the bounds, we can get the relationship between empirical risk and actual risk in ERM, thus we can study the generalization abilities of the learning machine.

Despite SLT shows good characters in dealing with learning problems in the case of the small samples, there are still some disadvantages such as the fact that SLT is

* Corresponding author.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 110–117, 2009.

established on probability space. As we all know, the condition of additivity of probability is very strong, sometimes it can not be satisfied in practical application. Additivity was challenged by non- additivity like theory of capacities by Choquet [11] and fuzzy measure theory by Sugeno [12]. In order to deal with general uncertainty, Liu founded uncertainty theory [13], Uncertain theory provides the commonness of probability theory, credibility theory and chance theory.

In this paper, the research of SLT is extended to uncertainty space, by using uncertain measure – a kind of non-additive measure, the bounds on the rate of uniform convergence of learning process and the relation between the bounds and the capacity of the set of functions are given and proved.

2 Preliminaries

In this section, we review some basic notions, which will be of interest in the ensuing investigation. The reader can refer to [13,14] for further detail.

Let Γ be a nonempty set, and let L be a σ-algebra over Γ. Each element $\Lambda \in \mathrm{L}$ is called an event. In order to present an axiomatic definition of uncertain measure, it is necessary to assign to each event Λ a number $\mathrm{M}\{\Lambda\}$ which indicates the level that Λ will occur. In order to ensure that the number $\mathrm{M}\{\Lambda\}$ has certain mathematical properties, Liu [14] proposed the following four axioms.

Axiom 1. (Normality) $\mathrm{M}\{\Gamma\}=1$.

Axiom 2. (Monotonicity) $\mathrm{M}\{\Lambda_1\} \le \mathrm{M}\{\Lambda_2\}$ whenever $\Lambda_1 \subset \Lambda_2$.

Axiom 3. (Self-Duality) $\mathrm{M}\{\Lambda\}+\{\Lambda^c\}=1$ for any event Λ.

Axiom 4. (Countable Subadditivity) For every countable sequence of events $\{\Lambda_i\}$, we have

$$\mathrm{M}\left\{\bigcup_{i=1}^{\infty} \Lambda_i\right\} \le \sum_{i=1}^{\infty} \mathrm{M}\left(\Lambda_i\right).$$

Definition 1. [14] The set function M is called an uncertain measure if it satisfies the normality, monotonicity, self-duality, countable subadditivity axioms. The triplet $(\Gamma, \mathrm{L}, \mathrm{M})$ is called an uncertainty space.

Throughout this paper, unless otherwise stated, $(\Gamma, \mathrm{L}, \mathrm{M})$ is an uncertainty space.

Definition 2. [14] An uncertain variable ξ is a measurable function from an uncertainty space to the set of real number, i.e., for any Borel set B of real numbers, the set

$$\{\xi \in B\} = \{\gamma \in \Gamma \mid \xi(\gamma) \in B\}$$

is an event.

Definition 3. [14] The uncertainty distribution $\Phi: R \to [0,1]$ of an uncertain variable ξ is defined by

$$\Phi(x) = \mathrm{M}\{\gamma \in \Gamma \mid \xi(\gamma) \le x\}.$$

Definition 4. [14] The uncertainty density function $\phi: R \to [0,+\infty]$ of an uncertain variable ξ is a function such that

$$\Phi(x) = \int_{-\infty}^{x} \phi(y)dy$$

$$\int_{-\infty}^{+\infty} \phi(y)dy = 1$$

where Φ is the uncertainty distribution of ξ.

Definition 5. [14] Let ξ be an uncertain variable. Then the expected value of ξ is defined by

$$E[\xi] = \int_0^{+\infty} \mathrm{M}\{\xi \geq r\}dr - \int_{-\infty}^{0} \mathrm{M}\{\xi \leq r\}dr$$

provided that at least one of the two integrals is finite.

Proposition 1. [14] Let ξ be an uncertain variable whose uncertainty density function ϕ exists. If the Lebesgue integral

$$\int_{-\infty}^{+\infty} x\phi(x)dx$$

is finite, then we have

$$E[\xi] = \int_{-\infty}^{+\infty} x\phi(x)dx\ .$$

Proposition 2. [14] Let ξ be an uncertain variable with finite expected value. Then for any real number a and b, we have

$$E[a\xi + b] = aE[\xi] + b\ .$$

Definition 6. [14] The uncertain variables $\xi_1, \xi_2, \cdots, \xi_n$ are said to be independent if

$$E\left[\sum_{i=1}^{n} f_i(\xi_i)\right] = \sum_{i=1}^{n} E[f_i(\xi_i)]$$

for any measurable functions $f_1, f_2, \cdots, f_n$ provided that the expected values exist and are finite.

Proposition 3. [14] If ξ and η are independent uncertain variables with finite expected values, then we have

$$E[a\xi + b\eta] = aE[\xi] + bE[\xi]$$

for any real numbers a and b.

Definition 7. [14] The uncertain variables ξ and η are identically distributed if

$$\mathrm{M}\{\xi \in B\} = \mathrm{M}\{\eta \in B\}$$

for any Borel set B of real numbers.

3 Main Results

In this part, consider a function set concluding N indicator functions $Q(z,\alpha_k)$ $k=1,2,\cdots,N$, where $z_1, z_2 \cdots, z_l$ are independent and identically distributed samples.

Definition 8. Let $\Phi(z)$ be the distribution function of uncertainty random variable z. Then on the uncertainty space, expected risk functional and empirical risk functional are defined by

$$R_u(\alpha) = E\left[Q(z,\alpha)\right] \tag{1}$$

$$R_{uemp}(\alpha) = \frac{1}{l}\sum_{i=1}^{l} Q(z_i,\alpha) \tag{2}$$

Definition 9. (Empirical Risk Minimization Principle) Let $Q(z,\alpha_0)$ minimize the expected risk , $Q(z,\alpha_l)$ minimize the empirical risk. We take $Q(z,\alpha_l)$ as an approximation of $Q(z,\alpha_0)$. The principle of how to solve uncertain risk minimization problem is called Empirical Risk Minimization Principle on uncertainty spaces (UERM).

In this section we discuss the rate of convergence of learning process , mainly discuss two questions:

1. What actual risk $R(\alpha_l)$ is provided by the function $Q(z,\alpha_l)$ that achieves minimal risk $R_{emp}(\alpha_l)$?

2. How close is this risk to the minimal possible $\inf_\alpha R(\alpha)$, $\alpha \in \Lambda$, for a given set of functions?

In the following we will give the answers of these two questions.

Theorem 1. (Hoeffding inequality) Let the uncertain variable sequence $\xi_1, \xi_2, \cdots, \xi_n$ be independent, and $S_n = \sum_{i=1}^{n} \xi_i$. Then for any $\lambda > 0$, $t > 0$ and $p > 0$, we have

$$\mathrm{M}\left\{S_n - E(S_n) \geq t\right\} \leq e^{-\lambda p t} E\left[e^{\lambda p (S_n - E(S_n))}\right].$$

Proof. We can get it by the property of uncertain variable and Markov inequality [14].

Remark 1. We get the same result as in probability space, credibility space, and chance space for Theorem 1, when $p=1$.

Theorem 2. Let ξ be an uncertain variable whose expected value is 0, and $\xi \in [a,b]$. Then for any $\lambda > 0$, we have

$$E\left[e^{\lambda\xi}\right] \leq e^{\lambda^2 (b-a)^2/8}.$$

Proof. By the property that power function is convex, we have

$$e^{\lambda\xi} \leq \frac{\xi - a}{b-a} e^{\lambda b} + \frac{b-\xi}{b-a} e^{\lambda a}.$$

Taking expected value on both sides, we get

$$E\left(e^{\lambda\xi}\right) \le \frac{be^{\lambda a} - ae^{\lambda b}}{b-a} = \left(1-p+pe^{\lambda(b-a)}\right)e^{-\lambda(b-a)p} = e\left(\phi(u)\right)$$

where $\phi(u) = \ln\left(1-p+pe^{u}\right) - up, u = \lambda(b-a), p = -\frac{a}{b-a}$.

We know $\phi(0) = \phi'(0) = 0$, meanwhile

$$\phi''(u) = \frac{p(1-p)e^{-u}}{\left(p+(1-p)e^{-u}\right)^2} \le \frac{1}{4}.$$

By Taylor expansion, for $\theta \in [0,u]$, we have

$$\phi(u) = \phi(0) + u\phi'(0) + \frac{u^2}{2}\phi''(\theta) \le \frac{u^2}{8} = \frac{\lambda^2(b-a)^2}{8}.$$

Thus

$$E\left[e^{\lambda\xi}\right] \le e^{\lambda^2(b-a)^2/8}.$$

Theorem 3. Let $\xi_i \in [a_i, b_i]$, $i = 1,2,\cdots,n$ be uncertain variables which are bounded. If $\xi_1, \xi_2 ..., \xi_n$ are independent, then for any $t > 0$, $p > 0$ we have

$$\mathrm{M}\left\{S_n - E(S_n) \ge t\right\} \le \exp\left(\sum_{i=1}^{n} \frac{-2p^2t^2}{(b_i-a_i)^2}\right).$$

$$\mathrm{M}\left\{S_n - E(S_n) \le -t\right\} \le \exp\left(\sum_{i=1}^{n} \frac{-2p^2t^2}{(b_i-a_i)^2}\right).$$

Proof. Combining Theorem 1 and Theorem 2, we have

$$\mathrm{M}\left\{S_n - E(S_n) \ge t\right\} \le \exp\left(-\lambda pt + \sum_{i=1}^{n} \frac{\lambda^2 p^2 (b_i-a_i)^2}{8}\right) \tag{3}$$

Let

$$\lambda = \frac{4pt}{\sum_{i=1}^{n}(b_i-a_i)^2}.$$

Minimizing the right side of (3), we have

$$\mathrm{M}\left\{S_n - E(S_n) \ge t\right\} \le \exp\left(\sum_{i=1}^{n} \frac{-2p^2t^2}{(b_i-a_i)^2}\right).$$

Similarly, we have

$$\mathrm{M}\left\{S_n - E(S_n) \le -t\right\} \le \exp\left(\sum_{i=1}^{n} \frac{-2p^2t^2}{(b_i - a_i)^2}\right).$$

The theorem is proved.

Applying Theorem 3 to the model that is discussed, we have

$$\mathrm{M}\left\{\left(R(\alpha) - R_{emp}(\alpha)\right) > \varepsilon\right\} < \exp\left\{-2\varepsilon^2 p^2 l\right\} \tag{4}$$

$$\mathrm{M}\left\{\left(R_{emp}(\alpha) - R(\alpha)\right) > \varepsilon\right\} < \exp\left\{-2\varepsilon^2 p^2 l\right\} \tag{5}$$

where $b_i - a_i = 1$, l is the number of the set of functions.

Theorem 4. Let $Q(z, \alpha_k)$, $k = 1, 2, \cdots, N$ be a set of indicator functions. Then for any $p > 0$, the inequality

$$R(\alpha_k) - R_{emp}(\alpha_k) \le \frac{1}{p}\sqrt{\frac{\ln N - \ln \eta}{2l}} \tag{6}$$

holds with uncertain measure at least $1 - \eta$.

Proof.

$$\begin{aligned}
&\mathrm{M}\left\{sup_{1\le k\le N}\left(R(\alpha_k) - R_{emp}(\alpha_k)\right) > \varepsilon\right\} \\
&\le \mathrm{M}\left\{\bigcup_{k=1}^{N}\left(R(\alpha_k) - R_{emp}(\alpha_k) > \varepsilon\right)\right\} \\
&\le \sum_{k=1}^{N} \mathrm{M}\left\{R(\alpha_k) - R_{emp}(\alpha_k) > \varepsilon\right\} \\
&\le N \exp\left\{-2\varepsilon^2 p^2 l\right\}
\end{aligned} \tag{7}$$

Let

$$0 < \eta \le 1.$$

$$N \exp\left\{-2\varepsilon^2 p^2 l\right\} = \eta.$$

Solving ε, we have

$$\varepsilon = \frac{1}{p}\sqrt{\frac{\ln N - \ln \eta}{2l}}.$$

Then (7) is written by

$$\mathrm{M}\left\{sup_{1\le k\le N}\left(R(\alpha_k) - R_{emp}(\alpha_k)\right) \le \varepsilon\right\} \ge 1 - \eta.$$

For the set of functions $Q(z, \alpha_k), k = 1, 2, \cdots, N$, inequality

$$R(\alpha_k) - R_{emp}(\alpha_k) \le \frac{1}{p}\sqrt{\frac{\ln N - \ln \eta}{2l}}.$$

holds with uncertain measure at least $1-\eta$. The theorem is proved.

Let $Q\left(z,\alpha_{k(0)}\right)$ be the function that minimizes the expected risk, $Q\left(z,\alpha_{k(l)}\right)$ be the function that minimizes the empirical risk, (6) is true for all functions in the set , also holds true for the function $Q\left(z,\alpha_{k(l)}\right)$.

Therefore, the following inequality holds with uncertain measure at least $1-\eta$.

$$R\left(\alpha_{k(l)}\right)-R_{emp}\left(\alpha_{k(l)}\right)\le\frac{1}{p}\sqrt{\frac{\ln N-\ln\eta}{2l}} \tag{8}$$

This inequality answers the first question: What actual risk $R\left(\alpha_l\right)$ is provided by the function $Q\left(z,\alpha_l\right)$ that achieves minimal risk $R_{emp}\left(\alpha_l\right)$?

Theorem 5. For any $p>0$, inequality

$$R\left(\alpha_{k(l)}\right)-R\left(\alpha_{k(0)}\right)\le\frac{1}{p}\left(\sqrt{\frac{\ln N-\ln\eta}{2l}}+\sqrt{\frac{-\ln\eta}{2l}}\right) \tag{9}$$

holds with uncertain measure at least $1-2\eta$.

Proof. For the function $Q\left(z,\alpha_{k(0)}\right)$ which minimizes the expected risk, the following inequality holds

$$\mathrm{M}\left\{R_{emp}\left(\alpha_{k(0)}\right)-R\left(\alpha_{k(0)}\right)>\varepsilon\right\}\le\exp\left\{-2\varepsilon^2p^2l\right\}.$$

This inequality means that the following inequality holds with uncertain measure at least $1-\eta$.

$$R_{emp}\left(\alpha_{k(0)}\right)-R\left(\alpha_{k(0)}\right)\le\frac{1}{p}\sqrt{\frac{-\ln\eta}{2l}} \tag{10}$$

Because $Q\left(z,\alpha_{k(l)}\right)$ is the function that minimizes the empirical risk, the inequality

$$R_{emp}\left(\alpha_{k(0)}\right)-R_{emp}\left(\alpha_{k(l)}\right)\ge 0.$$

holds. Combining this inequality and formulas (8), (10) then the following inequality

$$R\left(\alpha_{k(l)}\right)-R\left(\alpha_{k(0)}\right)\le\frac{1}{p}\left(\sqrt{\frac{\ln N-\ln\eta}{2l}}+\sqrt{\frac{-\ln\eta}{2l}}\right)$$

holds with uncertain measure at least $1-2\eta$. Theorem 2 is proved.

Theorem 5 gives the answer to the second question at the first of this section: How close is this risk to the minimal possible $\inf_\alpha R(\alpha)$, $\alpha\in\Lambda$, for given set of functions?

Remark 2. We get the same result as in probability space, credibility space, and chance space from Theorem 3 to Theorem 5, when $p=1$.

4 Conclusions

In this paper, we provide the bounds on the rate of uniform convergence of learning process on uncertainty space which is broader than probability space, chance space

and credibility space. Further investigations might focus on such fundamental issues as structural risk minimization, VC dimension theory and address the applied aspects such as support vector machines on uncertainty space.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 60773062), the Natural Science Foundation of Hebei Province of China (No. F2008000633), the Key Scientific Research Project of Education Department of Hebei Province of China(No.2005001D) and the Key Scientific and Technical Research Project of the Ministry of Education of China (No. 206012).

References

1. Vapnik, V.N.: Statistical Learning Theory. A Wiley-Interscience Publication, New York (1998)
2. Vapnik, V.N.: The Nature of Learning Theory. Springer, New York (1995)
3. Vapnik, V.N.: An Overview of Statistical Learning Theory. In: 8th IEEE Transaction on Neural Network, pp. 988–999. IEEE Press, New York (1999)
4. Zhang, X.G.: Introduction to Statistical Learning Theory and Support Vector Machines. Acta Automatica Sinica 26(1), 32–42 (2000)
5. Su, C., Yang, C.H.: Feature Selection for the SVM: an Application to Hypertension Diagnosis. Expert Systems with Applications 34, 754–763 (2008)
6. Huang, C.L., Chen, M.C., Wang, C.J.: Credit Scoring with a Data Mining Approach Based on Support Vector Machines. Expert System with Applications 33, 847–856 (2007)
7. Jin, B., Tang, Y.C., Zhang, Y.Q.: Support Vector Machines with Genetic Fuzzy Feature Transformation for Biomedical Data Classification. Information Sciences 177, 476–489 (2007)
8. Ha, M.H., Bai, Y.C., Tang, W.G.: The Sub-key Theorem on Credibility Measure Space. In: Proceeding of 2003 International Conference on Machine Learning and Cybernetics, Xi'an, China, vol. 5, pp. 3264–3268 (2003)
9. Ha, M.H., Li, Y., Li, J., Tian, D.Z.: Key Theorem of Learning Theory and Bounds on the Rate of Convergence Based on Sugeno Space. Science in China. Ser. E Information Sciences 36(4), 398–410 (2006)
10. Ha, M.H., Tian, J.: The theoretical Foundations of Statistical Learning Theory Based on Fuzzy Number Samples. Information Sciences 178(16), 3240–3246 (2008)
11. Choquet, G.: Theory of capacities. Annal de l'Institute Fourier 5, 131–295 (1954)
12. Sugeno, M.: Theory of Fuzzy Integals and its Applications. Ph.D. Dissertation, Tokyo Institute of Technology (1974)
13. Liu, B.: Uncertainty Theory, 2nd edn. Springer, Berlin (2007)
14. Liu, B.: Uncertainty Theory, 3rd edn (2008), http://orsc.edu.cn/liu/ut.pdf

Adaptive Growing Quantization for 1D CMAC Network

Ming-Feng Yeh and Kuang-Chiung Chang

Department of Electrical Engineering,
Lunghwa University of Science and Technology,
Taoyuan 33327, Taiwan
{mfyeh,kcchang}@mail.lhu.edu.tw

Abstract. This study attempts to propose an adaptive growing quantization approach for one-dimensional cerebellar model articulation controller (1D CMAC) network. Even though the target function is unknown in advance, the learning error can be still acquired and then is utilized to determine whether the input space needs to be repartitioned or not. Once the input space is determined to be repartitioned, some new knots are inserted for further quantization, and then the number of the states is increased. Therefore, the proposed approach not only possesses the adaptive quantization ability in the input space, but also has the growing feature in the number of the states. Beside, the linear interpolation scheme is applied to calculate the CMAC output for simultaneously improving the generalization ability and reducing the memory requirement. Simulation results show that the proposed approach not only has the adaptive quantization ability, but also can achieve a better learning accuracy and a faster convergence speed.

Keywords: Cerebellar model articulation controller, Adaptive quantization, Pseudo-inverse, Linear interpolation.

1 Introduction

The cerebellar model articulation controller (CMAC) network, proposed by Albus [1, 2], is a kind of supervised neural network inspired by the human cerebellum. This network learns input-output mappings based on the premise that similar inputs should produce similar outputs. Therefore, unlike multi-layer feedforward networks, CMAC networks store information locally. Owing to its fast learning speed, good generalization ability and ease of implementation by hardware, the CMAC network has been successfully applied in many applications such as control problem, signal processing and pattern recognition [3-5]. Other previous researches concerning about CMAC network have focused mainly on developing the CMAC learning algorithms [6, 7], improving the CMAC topology [8, 9], and selecting the learning parameters [10, 11].

This study attempts to propose an adaptive growing quantization approach for 1D CMAC network. Learning starts with a conventional 1D CMAC using low resolution and equal-sized quantization. The learned information is employed to evaluate whether the input space needs to be repartitioned or not. Once the input space is determined to be repartitioned, some new knots are inserted into the input space for

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 118–127, 2009.

further quantization, where the locations of those newly inserted knots are placed according to the learned information. Inserting new knots also causes some increase in the number of states. Such a mechanism enables the CMAC not only to possess the "adaptive" quantization ability in the input space, but also to have the "growing" feature in the number of the states. Since Albus' CMAC uses a binary basis function as association memory selection vector in the learning and recall processes, the output is constant within each quantized state. That is the main reason to result in the generalization error of CMAC. Without using B-Spline or Gaussian function as the basis function to smoothen the CMAC output [8], the study applies the linear interpolation scheme to the recall process of Albus' CMAC for improving the learning accuracy.

Any newly inserted knot will cause some variations in the block division of CMAC such as the number of states, the entire memory size and the association memory index. Hence, that CMAC must be retrained again for attaining a high learning performance. Such a retraining process may be done several times during the adaptive growth process and then makes the proposed method become very inefficient. In order to achieve the purpose of fast learning speed, this study also proposes an average method to generate some virtual training patterns from the given training dataset. Once every virtual training target is determined, the memory contents, which could minimize the quadratic error, can be obtained by the pseudo-inverse scheme. By this way, a CMAC requires only one training step to obtain the learned information of memory contents. After very adaptive growth step, if necessary, the virtual training patterns must be updated according to the newly quantized space.

2 Preliminaries

In a CMAC network, each state variable is quantized and the problem space is divided into discrete states [1, 2]. A vector of quantized input values specifies a discrete state and is used to generate addresses for retrieving information from memory elements for this state. The basic structure of CMAC is depicted in Fig. 1. In the figure, the association memory **A** is obtained from the input space **S**, the associated data stored in the memory cell **W** are yielded in accordance with each input state. The CMAC sums the mapped data up as its output and feeds the error between the actual and desired outputs back to the memory cell equally. The mathematical expression of 1D CMAC is stated as follows.

Assume that the input space **S** is quantized into N_s states and every state utilizes N_e memory units to store the corresponding memory contents. By this way, the entire memory size is $N_b = N_s + N_e - 1$. See Fig. 2(b) for example, the input space is equally divided into 5 states and every state utilizes 3 memory cells to store its information. Therefore, the entire memory size is 7. The stored data y_k (the actual output of the CMAC) for the state s_k is the sum of stored contents of all addressed blocks and can be expressed as

$$y_k = \mathbf{a}_k^T \mathbf{w} = \sum_{b=1}^{N_b} a_{k,b} w_b, \tag{1}$$

where w_b, b = 1, 2, 3, …, N_b, is the memory content of the bth block, $\mathbf{w} = [w_1, w_2, \ldots, w_{N_b}]^T$, $a_{k,b}$ is the association index indicating whether the bth block is addressed by the state s_k, and $\mathbf{a}_k^T = [a_{k,1}, a_{k,2}, ..., a_{k,N_b}]$. Since each state addresses exactly N_e blocks, only those addressed $a_{k,b}$'s are 1, and the others are 0. The CMAC uses a supervised learning method to adjust the memory contents during each learning cycle. Its updating rule can be described as

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \frac{\eta}{N_e}\mathbf{a}_k[\hat{y}_k - \mathbf{a}_k^T\mathbf{w}(t)], \quad t = 1, 2, 3, ... \tag{2}$$

where $\mathbf{w}(t+1)$ is the stored value of the bth block at time t+1, $\mathbf{w}(t)$ is the one at previous time t, η is the learning rate, $\hat{y}_k$ is the desired value for the state s_k, and $\hat{y}_k - \mathbf{a}_k^T\mathbf{w}(t)$ is the error for the state s_k. Note that only the addressed blocks are updated. On the other hand, the analytical solution of $\mathbf{w}$ is

$$\mathbf{w}^* = \mathbf{A}^+\hat{\mathbf{y}}, \tag{3}$$

where $\mathbf{A}^+ = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}$ is the pseudo-inverse of the association matrix $\mathbf{A}$ formed from the association vector $\mathbf{a}_k$, k = 1, 2, 3, …, N_s, as row vector, and

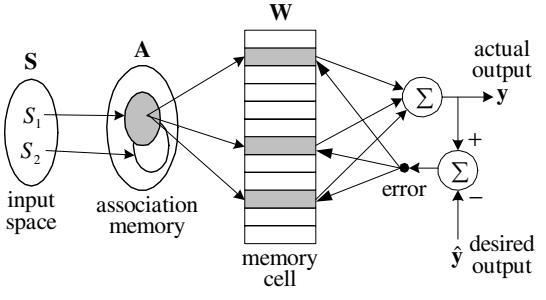


Fig. 1. Block diagram of CMAC

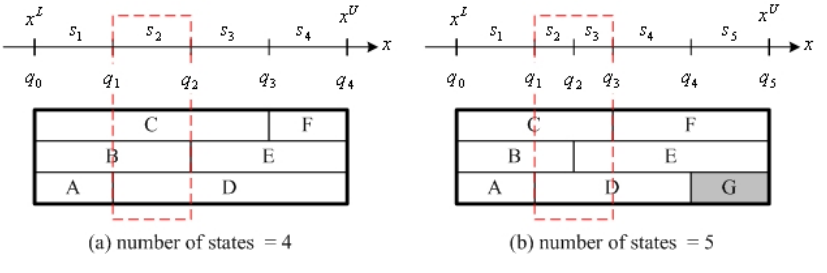


Fig. 2. State growing process

$\hat{\mathbf{y}}^T = [\hat{y}_1, \ \hat{y}_2, ..., \hat{y}_{N_s}]$. That is to say, the updating rule (2) could converge to $\mathbf{w}^*$ if the learning rate is properly selected [12].

3 Adaptive Growing Quantization

Two main assumptions of the proposed approach are stated as follows. One is that the input space is limited in a certain region, but the number of states N_s is allowed to gradually increase at every adaptive growth step. The other is that each state is distributively stored in N_e memory elements, i.e., a complete block covers exactly N_e adjacent states, where N_e is constant.

3.1 Virtual Training Patterns

Assume that there are N_p given patterns (x_j, d_j) generated from an target function $f(x)$, where $d_j = f(x_j)$ and $j = 1, 2, \ldots, N_p$. Generally speaking, $N_p >> N_s$. The input points (x-values) could be randomly or uniformly distributed in the input space $[x^L, x^U]$, where x^L and x^U represent the lower and upper bound, respectively. As the input space is quantized into N_s states, there exist N_s+1 knots, denoted by q_i, $i = 0, 1, 2, \ldots, N_s$, to partition the space as shown in Fig. 2. Moreover, the input x_j belongs to the state s_k if and only if $q_{k-1} < x_j \le q_k$. Without lost of generality, it is assumed that each state contains at least one input.

Define $(\bar{x}_k, \bar{d}_k)$ as the virtual training pattern for the state s_k, where

$$\bar{x}_k = \frac{1}{N(s_k)} \sum_{x_j \in s_k} x_j, \tag{4}$$

$$\bar{d}_k = \frac{1}{N(s_k)} \sum_{x_j \in s_k} d_j, \tag{5}$$

and $N(s_k)$ represents the number of training inputs belonging to s_k. In the above definition, $\bar{x}_k$ and $\bar{d}_k$ represent the virtual training input and the corresponding desired output, respectively. It is obvious that there are exactly N_s virtual training patterns and no two distinct training inputs, say $\bar{x}_i$ and $\bar{x}_j$, $i \neq j$, are associated with the same state. On the other hand, these virtual training patterns not only allow a CMAC to be trained by a few samples, but also could avoid the learning interference [7] during the training process.

3.2 Adaptive Growing Process

The proposed adaptive growing process starts with a CMAC whose input space is uniformly quantized into several states, e.g., $N_s = 4$ in Fig. 2(a). Once all virtual training patterns are generated by (4) and (5), the memory contents $\mathbf{w}$ can be obtained by the pseudo-inverse scheme (3) if let $\hat{y}_k = \bar{d}_k$ for all k's. When the input x_j is applied to the trained CMAC, the corresponding output obtained from (1) will satisfy that

$y(x_j) = y_k$ if $x_j \in s_k$. As can be seen, the generalization error due to the input x_j is $d_j - y_k$. This study adopts two strategies to reduce the generalization error caused by quantization. The first one is applied the linear interpolation scheme to the recall process of Albus' CMAC as follows.

$$y(x_j) = \begin{cases} y_1, & \text{if } x_j \le \bar{x}_1, \\ \dfrac{\bar{x}_k - x_j}{\bar{x}_k - \bar{x}_{k-1}} y_{k-1} + \dfrac{x_j - \bar{x}_{k-1}}{\bar{x}_k - \bar{x}_{k-1}} y_k, & \text{if } \bar{x}_{k-1} < x_j \le \bar{x}_k, \\ y_{N_s}, & \text{if } x_j > \bar{x}_{N_s}, \end{cases} \tag{6}$$

where $1 < k < N_s$.

The second one is the proposed adaptive quantization scheme as described as follows. Define the root mean squared error (RMSE) for the state s_k, termed the state error hereafter, as

$$e_s(s_k) = \sqrt{\frac{1}{N(s_k)} \sum_{x_j \in s_k} (d_j - y(x_j))^2} \ . \tag{7}$$

Those states with larger state error must be further quantized to minimize the generalization error caused by quantization. The evaluation criterion is that a state, say s_k, needs to be repartitioned if $e_s(s_k) \ge \rho$, where ρ is a predefined repartition threshold. Since $q_{k-1} \le \bar{x}_k \le q_k$, inserting the point $\bar{x}_k$ into the interval $[q_{k-1}, q_k]$ could divide that interval into two parts. Hence, the newly inserted knot for the state s_k in this study is directly assigned as the point $\bar{x}_k$ and it can divide the state s_k into two new states as depicted in Fig. 2(b). If there are more than one state which must be repartitioned at the same time, each associated interval must be inserted a corresponding new knot for re-quantization. To do so, the proposed approach could simultaneously attain the purposes of adaptive quantization and state growth. Even though the target function is unknown, the proposed adaptive quantization approach can be still performed by the above evaluation criterion.

The adaptive growing process will stop when a specified performance measure or a pre-specified maximum number of states (or epochs) is reached. To sum up, the algorithm of the proposed adaptive growing quantization can be described as follows.

1) Start with a CMAC whose input space is uniformly quantized into several states.
2) Generate the virtual training patterns by (4) and (5).
3) Obtain the memory contents by the pseudo-inverse scheme (3).
4) Calculate the CMAC output for each virtual training input by (1).
5) Use the linear interpolation scheme (6) to find all the actual outputs.
6) Compute all state errors by (7).
7) Insert a new knot into the state with state error larger than the repartition threshold. If there are more than one state which must be repartitioned, each associated interval must be inserted a corresponding new knot.
8) Repeat steps 2-7 until a stopping criterion is fulfilled.

3.3 Convergence Analysis

At each state growth step, the memory contents are obtained by the pseudo-inverse scheme (3). Since Eq. (3) is the analytic solution of (2), the memory contents $\mathbf{w}^*$ could minimize the quadratic error [12]:

$$E = \frac{1}{2}\sum_{k=1}^{N_p}[d_k - y(x_k)]^2 . \tag{8}$$

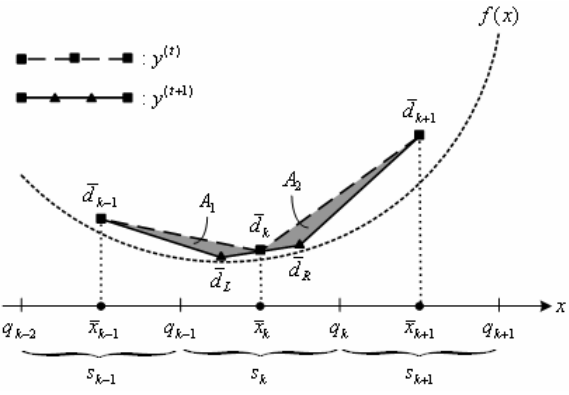
Herein the convergence analysis focuses on showing that the quadratic error is gradually decreased as the number of states is gradually increased. That is, $\Delta E = E(t+1) - E(t) \le 0$, where $E(t)$ represents the quadratic error at the tth epoch.

It can be seen from (6) that the CMAC response is a piecewise linear approximation of the target function $f(x)$. See Fig. 3(a) for example. The broken line $y^{(t)}$ shows a possible piecewise linear approximation of $f(x)$ in the interval $[\bar{x}_{k-1}, \bar{x}_{k+1}]$ at the tth epoch, where the black squares represent the virtual target values as defined in (5). Assume that the state s_k is evaluated to be repartitioned. While the new knot $\bar{x}_k$ is inserted into the state s_k, that state is divided into two states covered by $[q_{k-1}, \bar{x}_k]$ and $[\bar{x}_k, q_k]$, respectively. Subsequently, the original virtual target value $\bar{d}_k$ is replaced by two new ones $\bar{d}_L$ and $\bar{d}_R$ marked by the black triangles in the figure. Since $f(x)$ is concave upward, $\bar{d}_L \le \bar{d}_k$ and $\bar{d}_R \le \bar{d}_k$. Therefore, inserting a new knot could push the piecewise linear approximation toward the target function $f(x)$ as the solid line $y^{(t+1)}$ in Fig. 3(a). The fact also reveals that

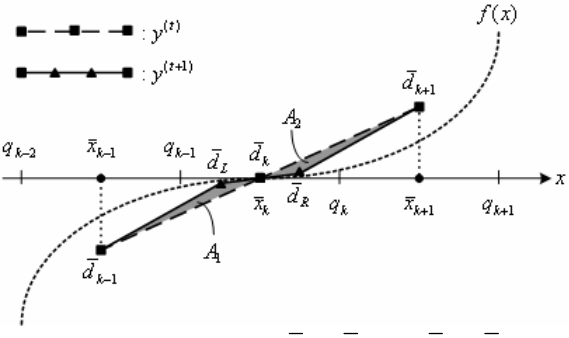
$$\int_{\bar{x}_{k-1}}^{\bar{x}_{k+1}} \left| f(x) - y^{(t)} \right| dx - \int_{\bar{x}_{k-1}}^{\bar{x}_{k+1}} \left| f(x) - y^{(t+1)} \right| dx = A_1 + A_2 \ge 0 , \tag{9}$$

where the term A_1+A_2 represents the area of the region bounded by $y^{(t)}$ and $y^{(t+1)}$ on the interval $[\bar{x}_{k-1}, \bar{x}_{k+1}]$. This also means that the approximation error is reduced by the amount of A_1+A_2. Hence $\Delta E \le 0$ holds for the concave upward case. If $f(x)$ is concave downward (i.e., $\bar{d}_L \ge \bar{d}_k$ and $\bar{d}_R \ge \bar{d}_k$), the fact that $\Delta E \le 0$ is also true.

Fig. 3(b) shows that case of $f(x)$ being increasing, i.e., $\bar{d}_L \le \bar{d}_k$ and $\bar{d}_R \ge \bar{d}_k$. While the new knot $\bar{x}_k$ is inserted into the state s_k, the approximation error is also reduced by the amount of A_1+A_2 as shown in the figure. Hence $\Delta E \le 0$ holds. Analogously, $\Delta E \le 0$ is also true for the decreasing case. To sum up, the quadratic error is gradually decreased as the number of states is gradually increased. This completes the convergence analysis.



(a) Case I: concave upward ($\bar{d}_L \le \bar{d}_k$ and $\bar{d}_R \le \bar{d}_k$)



(b) Case II: increase ($\bar{d}_L \le \bar{d}_k$ and $\bar{d}_R \ge \bar{d}_k$)

Fig. 3. Piecewise linear approximation of $f(x)$

4 Simulation Results

Consider a 1D CMAC network trained to generate a target function $f(x) = 1 + e^{-x} \cdot \sin(3x)$, $x \in [0.0, 10.0]$. There are 201 training patterns, generated from $f(x)$, whose input points (x-values) are uniformly distributed in the input space. In this example, each state is distributively stored in three memory elements, i.e., $N_e = 3$ and the input space is initially quantized into five states. Besides, the repartition threshold is $\rho = 0.01$. The plot of the RMSE for the first several epochs is given in Fig. 4, in which the number shown indicates the number of states used at this epoch. The CMAC with output linear interpolation is converged at the 6th epoch with 30 memory elements and RMSE of 0.0045, whereas the one without linear interpolation, i.e., the CMAC outputs are directly obtained from (1), is at the 7th epoch with 62 memory elements and RMSE of 0.0038. In the figure, one more epoch for each case is to show the corresponding learning result is actually converged. As can be seen, in both cases, the RMSE is gradually reduced as the number of states is gradually increased. However, the former is superior to the latter in both the convergence speed and the memory

requirement. In fact, the memory requirement of the proposed approach is just half of that of the conventional method.

Fig. 5 shows the generalization results for both cases with 1001 verification patterns whose input points are also uniformly distributed in the input space. The RMSEs with and without output linear interpolation are 0.0058 and 0.0087, respectively. This also reveals that applying the linear interpolation scheme to the recall process could effectively reduce the output error with less memory requirement whether the patterns are trained or not. In Fig. 5, the black dots on the *x*-axis represent the knots generated by the proposed approach. The larger the degree of variation in the target function, the more the knots. It is obvious that most of knots are located in the interval [0.0, 2.0] (see the top figure). Since the degree of variation in the interval [4.0, 10.0] is nearly zero (see the bottom figure), no new knot is inserted into that interval. The results could show that the proposed approach has the adaptive quantization ability.

Fig. 6 shows the learning results of the proposed quantization approach with the on-line adaptive quantization method [13]. In the on-line adaptive quantization method, the input space is initially quantized into 40 states and the CMAC is trained with the learning rate η of 0.1 and the generalization size N_e of 3. The corresponding learning result is converged at the 245th epoch with 36 memory elements (34 states) and RMSE of 0.0101. The RMSE of the generalization result for the above 1001 test patterns is 0.0239. Besides, the black dots on the *x*-axis of Fig. 6 represent the knots generated by the on-line adaptive quantization method. Although the on-line quantization method also could adaptively quantize the input space, the proposed quantization approach still outperforms the on-line adaptive quantization method in the learning performance and the generalization ability.

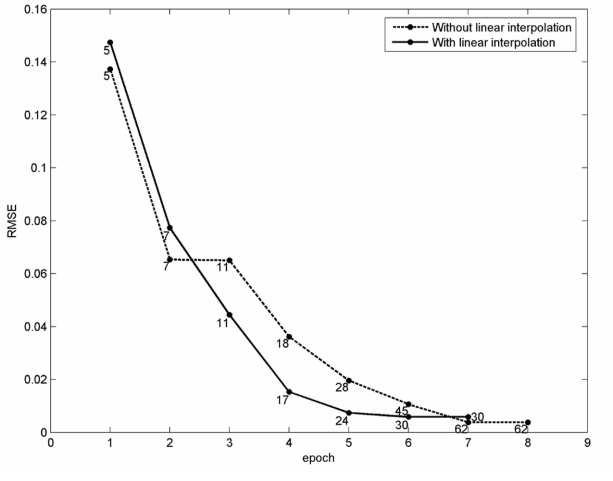


Fig. 4. RMSE and number of states for each epoch

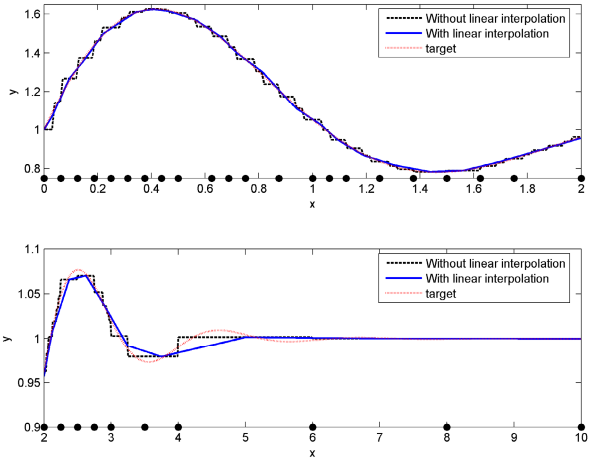


Fig. 5. Generalization results of 1D AG-CMAC

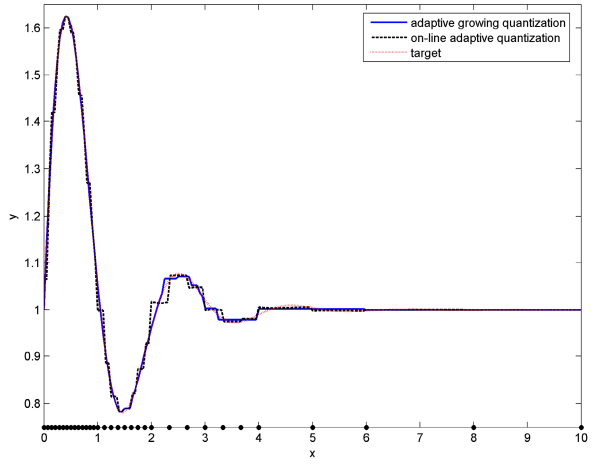


Fig. 6. Comparison of adaptive growing quantization with on-line adaptive quantization

5 Conclusions

This study proposed an adaptive growing quantization approach for 1D CMAC network. With the help of the pseudo-inverse scheme, each adaptive growth step CMAC requires only a single training cycle. The RMSE of each state is used to evaluate which state needs to be repartitioned or not. Even though the target function is unknown in advance, all state errors can be acquired according to the learned information. Once a state is determined to be repartitioned, a new knot is inserted into the corresponding interval. Therefore, the number of states is gradually increased during

the adaptive growing process. Beside, the linear interpolation scheme is applied to calculate the CMAC output for simultaneously improving the generalization ability and reducing the memory requirement. Simulation results for a nonlinear function show that the proposed approach not only has the adaptive quantization ability, but also can achieve a better learning accuracy and a faster convergence speed.

Acknowledgments

This work was supported by the National Science Council, Taiwan, Republic of China, under Grant NSC 97-2221-E-262-009.

References

1. Albus, J.S.: A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC). J. Dyn. Syst., Meas., Contr., Trans. ASME. 97(3), 220–227 (1975)
2. Albus, J.S.: Data Storage in the Cerebellar Model Articulation Controller (CMAC). J. Dyn. Syst., Meas., Contr., Trans. ASME. 97(3), 228–233 (1975)
3. Yeh, M.F.: Single-input CMAC Control System. Neurocomputing 70, 2638–2644 (2007)
4. Glanz, F.H., Miller, W.T., Kraft, L.G.: An Overview of the CMAC Neural Network. In: Proc. 1991 IEEE Neural Netw. Ocean Eng., Washington, DC, pp. 301–308 (1991)
5. Tao, T., Lu, H.C., Hung, T.H.: The CA-CMAC for Downsampling Image Data sSize in the Compressive Dain. In: Proc. 2002 IEEE Int. Conf. Syst., Man and Cybern, Hammamet, Tunisia, vol. 5 (2002)
6. Cotter, N.E., Guillerm, T.J.: The CMAC and A Theorem of Kolmogorov. Neural Netw. 5, 221–228 (1991)
7. Thompson, D.E., Kwon, S.: Neighborhood Sequential and Random Training Techniques for CMAC. IEEE Trans. Neural Netw. 6(1), 196–202 (1995)
8. Lee, H.M., Chen, C.M., Lu, Y.F.: A Self-organizing HCMAC Neural-network Classifier. IEEE Trans. Neural Netw. 14(1), 15–27 (2003)
9. Jan, J.C., Hung, S.L.: High-order MS_CMAC Neural Network. IEEE Trans. Neural Netw. 12(3), 598–603 (2001)
10. Kim, H., Lin, C.S.: Use of Adaptive Resolution for Better CMAC Learning. In: 1992 Int. Joint Conf. Neural Netw., Baltimore, MD, vol. 1, pp. 517–522 (1992)
11. Lin, C.S., Kim, H.: Selection of Learning Parameters for CMAC-based Adaptive Critic Learning. IEEE Trans. Neural Netw. 6(3), 642–647 (1995)
12. Horváth, G., Szabó, T.: Kernel CMAC with Improved Capability. IEEE Trans. Syst., Man, Cybern., B. 37(1), 124–138 (2007)
13. Yeh, M.F., Lu, H.C.: On-line Adaptive Quantization Input Space in CMAC Neural Network. In: Proc. 2002 IEEE Int. Conf. Syst., Man and Cybern., Hammamet, Tunisia, vol. 4, pp. 336–341 (2002)

Qualitative Analysis of General Discrete-Time Recurrent Neural Networks with Impulses

Xinquan Zhao

Information School, Zhongnan University of Economics & Law,
Wuhan 430074, China

Abstract. In this article, the qualitative analysis of general discrete-time recurrent neural networks with impulses is discussed. First, a sufficient condition and a sufficient and necessary condition for existence and uniqueness of the equilibrium point of this neural networks are given with the help of degree theory; second, some sufficient rules for the global exponential stability of this neural networks are obtained by using Lyapunov function; finally the instability of the equilibrium is studied.

Keywords: Discrete-time neural networks, Impulse, Unique equilibrium, Globally exponential stability, Instability.

1 Introduction

The subject of artificial neural networks has become one of the important technical tools for solving a variety of problems in various scientific disciplines. From a view of mathematics, an artificial neural network corresponds to a nonlinear transformation of some inputs into certain outputs. Among the many types of neural networks proposed and studied in the literature, the several discrete-time recurrent neural networks[1-7]described by difference equations have become an important one due to its potential for applications in associative memory, pattern recognition, optimization, model identification, signal processing, etc. The dynamical characteristics of the network are assumed to be governed by the dynamics of the following system of difference equation[8]

$$x(n+1) = Dx(n) + A\sigma(Bx(n) + I). \tag{1}$$

where n is a positive integer, $x=(x_1, x_2, \cdots x_m)^T \in R^m, D=diag(d_1, d_2, \cdots, d_m)$, $|d_i| < 1, \quad I = (I_1, I_2, \cdots, I_m)^T \in R^m, A = diag(a_1, a_2, \cdots, a_m), a_i \neq 0, B = (b_{ij}) \in R^{m\times m}, I_i$denotes a constant external input current to the ith neuron, $\sigma(x) = (\sigma_1(x_1), \sigma_2(x_2), \cdots, \sigma_m(x_m))^T$, $\sigma_i(x_i) = 2^{-1}(|x_i + 1| - |x_i - 1|)$.

When the $D = 0, A = E$, the model becomes BSB model[1-4], so this model has more generalization.

Dynamical systems are often classified into two categories of either continuous-time[9-11] or discrete-time systems. Recently there has been a somewhat a new category of dynamical systems, which is neither purely continuous-time nor

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 128–137, 2009.

purely discrete-time ones; these are called dynamical systems with impulses [12–15].This third category of dynamical systems display a combination of characteristics of both the continuous-time and discrete-time systems. The development of the impulsive dynamical systems has proceeded along two distinct lines. The continuous-time dynamical systems with impulsive have been studied commendably, but the discussion of the discrete-time dynamical systems with impulsive is inchoate. It has been researched widely and deeply, because of the important effect of the model of the discrete-time neural networks with impulses described by difference equations in the pattern recognition and image processing, etc.

Now we consider the system (1) subjected to certain impulsive state displacements at fixed moments of time:

$$\begin{cases} x(n+1) = Dx(n) + Af(x(n) + I) \\ x(n_0) = x_0 \in R^n \\ l_i(x_i(n_k)) = x_i(n_k + 1) - x_i(n_k),\ i = 1, 2, \cdots, m,\ k = 1, 2, 3, \cdots \\ n_0 < n_1 < \cdots < n_k \to \infty \text{ as } k \to \infty \end{cases} \tag{2}$$

By a solution of (2), we mean$x = \{x_1, x_2, \cdots, x_m\}^T \in R^m$;the impulsive functions $l_k(\cdot) : R \longrightarrow R$ are assumed to be discrete, $f(x, I) : R^m \times R^m \to R^m$.

In the following, the qualitative analysis of a discrete-time recurrent neural networks with impulses is discussed. First, a sufficient and a necessary conditions for existence and uniqueness of the equilibrium point of this neural networks is given by the theory of topologic degree[8], where the degree theory of nonlinear analysis is used; second, some sufficient rules for the global exponential stability of this neural networks are obtained by using Lyapunov function; finally the instability of the equilibrium is studied.

2 The Existence of the Equilibria of Discrete-time Systems

The results of this section are obtained in [8]. Here, we introduce the good results and methods causing people's interesting and recommending. Let

$$h(x, I) = (E - D)x - Af(x + I),$$

then the equilibrium point of (1) is equal to the solution of the equation$h(x, I) = 0$. Here, the description of the theory of topologic degree is given: suppose that $\Omega \subset R^m$is a bounded open set, $\bar{\Omega}$ is its closure, and $\partial\Omega$ is its boundary such that $h$is a mapping, $h : \bar{\Omega} \to R^m$.If $h(\partial\Omega) \neq 0$, then, foe 0, the integral value funtion $d(h, \Omega, 0)$ denotes the degree of the function $h$on Ω, $H(\lambda, x) \in C(J \times \Omega, R^n)$is homotopy, $J \in [0, 1]$.

Theorem 1. *For every* $I = (I_1, I_2, \cdots, I_m)^T \in R^m, \|f(x, I)\| \leq L, or \|f(x, I)\| \leq L\|x\|^\alpha, 0 \leq \alpha \leq 1$, *there is* R_0 *that is enough large such that*

$$d(h(x, I), U(R_0), 0) = d(x, U(R_0), 0) = 1. \tag{3}$$

Namely, there is at least a solution of $h(x, I) = 0$ *in the* $U(R_0)$, *so there is at least an equilibrium point of (1).*

Proof. For every $I = (I_1, I_2, \cdots, I_m)^T \in R^m$, making

$$H(\lambda, x) \triangleq \lambda h(x, I) + (1 - \lambda)x, \tag{4}$$

and $d = \max d_i$, then $\forall x \in \partial U(R_0)(\text{namely} \|x\| = R_0)$, have

$$x^T H(\lambda, x) \geq (1 - d)R_0^2 - \sum_{i=1}^{m} a_i x_i f_i(x_i + I_i).$$

(1) If $\|f(x, I)\| \leq L, R_0 > (1 - d)^{-1} L \|A\|$, for $\forall \lambda \in J$

$$x^T H(\lambda, x) \geq (1 - d)R_0^2 - \sum_{i=1}^{m} L_i |a_i| |x_i| \geq (1 - d)R_0^2 - L \|A\| R_0 > 0$$

(2) If $0 \leq \alpha < 1, R_0 > \left((1 - d)^{-1} L \|A\|\right)^{\frac{1}{1-\alpha}}$, for $\forall \lambda \in J$,

$$x^T H(\lambda, x) \geq (1 - d)R_0^2 - \sum_{i=1}^{m} L_i |a_i| x_i^{1+\alpha} \geq (1 - d)R_0^2 - L \|A\| R_0^{1+\alpha} > 0;$$

(3) If $\alpha = 1, 1 - d - L \|A\| > 0$, for $\forall \lambda \in J$,

$$x^T H(\lambda, x) \geq (1 - d)R_0^2 - \sum_{i=1}^{m} L_i |a_i| x_i^2 \geq (1 - d - L \|A\|)R_0^2 > 0.$$

Here $\|A\|$ is a spectral norm of A. Hence$\forall \lambda \in J, x^T H(\lambda, x) \neq 0$, by homotopy invariability,

$$d(H(1, x), U(R_0), 0) = d(H(0, x), U(R_0), 0).$$

Namely (3) is right, the result of the theorem has been gotten.

Theorem 2. *For every $I = (I_1, I_2, \cdots, I_m)^T \in R^m$, a sufficient condition for uniqueness of the equilibrium point of (1) is*

$$a_i^{-1}(1 - d_i) \neq \Delta f_i / \Delta x_i, i = 1, 2, \cdots, m. \tag{5}$$

Proof. Now we are going to prove this uniqueness of the equilibrium point of (1), by reduction to absurdity. If $a_i^{-1}(1 - d_i) \neq \Delta f_i / \Delta x_i, i = 1, 2, \cdots, m$, there are two equilibrium points of (1) $x \neq y$, then

$$(E - D)(x - y) - A[f(x + I) - f(y + I)] = 0.$$

$$\begin{pmatrix} a_1^{-1}(1 - d_1) \\ a_2^{-1}(1 - d_2) \\ \cdots \\ a_m^{-1}(1 - d_n) \end{pmatrix} = \begin{pmatrix} [f_1(x_1 + I_1) - f_1(y_1 + I_1)] / (x_1 - y_1) \\ [f_2(x_2 + I_2) - f(y_2 + I_2)] / (x_2 - y_2) \\ \cdots \\ [f_3(x_3 + I_3) - f_3(y_3 + I_3)] / (x_m - y_m) \end{pmatrix}.$$

Well than

$$a_i^{-1}(1 - d_i) = (x_i - y_i)^{-1} [f_i(x_i + I_i) - f_i(y_i + I_i)] = \Delta f_i / \Delta x_i, i = 1, 2, \cdots, m,$$

illogicality. The sufficient condition (1) has been completed.

3 Globally Exponential Stability of the Equilibrium of Discrete-Time Systems

In this section, we derive sufficient conditions for the globally exponential stability of the equilibrium of a system when the system is subjected to impulsive displacements. In particular we consider the stability of equilibrium of the discrete-time impulsive system (1). Fist two principal definitions and one conclusion are delivered:

1. If there are two positive definite matricesA, Q, such that$A^T PA - P = -Q$, we say that$A$is stable.

2. If there are a positive diagonal matrix$P$and a positive definite matrix$Q$such that $A^T PA - P = -Q$, we say that$A$is diagonal stable.

3. Let $|A| = (|a_{ij}|)_{n\times n}$, by we have

$\rho(|A|) < 1 \Leftrightarrow |A|$is stable$\Leftrightarrow |A|$is diagonal stabal$\Leftrightarrow E - |A|$is a M-matrix

$$\Leftrightarrow E - A \in P \quad \Rightarrow E - A \in P,$$

$A$is diagonal stable$\Rightarrow E - A \in P$(see [19,20,21]). (4)

We can simplify the above system as follows: Suppose $x^*$is a equilibrium point of (1), we let $y_i(n) \equiv x_i(n) - x_i^*,\ i = 1, 2, 3, \cdots, m, F(y(n)) = f((y(n) + x^*) + I) - f(x^* + I)$

and note that the $y_i(\cdot)$are governed by

$$\begin{cases} y(n+1) = Dy(n) + AF(y(n)) \\ y(n_0) = x_0 - x^* = y_0 \in R^n \\ l_i(y_i(n_k)) = y_i(n_k + 1) - y_i(n_k), i = 1, 2, \cdots, m,\ k = 1, 2, 3, \cdots \\ n_0 < n_1 < \cdots < n_k \to \infty \text{ as } k \to \infty \end{cases} . \tag{6}$$

where $F(y(n)) = f(y(n) + x^* + I) - f(x^* + I)$. Obviously, $y = 0$is a trivial equilibrium point of (5), to study the stability of an equilibrium point of (1) is equal to study the stability of a trivial equilibrium point of (5), also for an arbitrary positive diagonal matrix$P$, we have

Theorem 3. *Suppose that* $|F_i(y)| \le L_i |y_i|, i = 1, 2, \cdots, n,$ *and*

(1)$1 > |d_i| + L_i |a_i|, i = 1, 2, \cdots, n,$ or $\max\limits_{1\le i\le m} [1 - |d_i| - L_i |a_i|] > 0;$

(2) The impulsive operators $I_{ik}(y_i(n_k))$ satisfy

$$I_{ik}(y_i(n_k)) = -r_{ik}(y_i(n_k) - x_i^*), 0 < r_{ik} < 2, i = 1, 2, \cdots, m, k \in Z^+,$$

then the solution of (5) is globally exponential stable.

Proof. Consider a Lyapunov function $V_1(y) = V_1(y(\cdot))$defined by

$$V_1(y) = V_1(y(\cdot)) = \sum_{i=1}^{m} |y_i(\cdot)| = \sum_{i=1}^{m} |x_i(\cdot) - x_i^*|. \tag{7}$$

Obviously,$V_1(y) \geq 0$ and $V_1(0) = 0, V_1(y) \to +\infty$as$\|y_i(\cdot)\| \to +\infty$. We find the change $\Delta^+ V_1(y)$along the solution of (5)given by

$$\Delta^+ V_1(y) \leq -\min_{1\leq i\leq m} (1 - |d_i| - L_i |a_i|) \sum_{i=1}^{m} |y_i(n)| = -\alpha_1 V_1(y). \tag{8}$$

Using the inequality $e^r \geq 1 - r(r > 0)$, mathematical induction and (8), we get

$$V_1(y(n)) \leq V_1(y(n-1))e^{-\alpha_1} \leq V_1(y(n-2))e^{-2\alpha_1} \leq \cdots \leq V_1(y(0))e^{-n\alpha_1}. \tag{9}$$

It follows that on each $n = 0, 1, 2, \cdots$, $V(\cdot)$is exponential descending. Also

$$V_1(y(n_k + 1)) \leq \sum_{i=1}^{m} |x_i(n_k) - x_i^*| = V_1(y(n_k)),\ k \in Z^+. \tag{10}$$

We may make the Lyapunov function$V_2(y) = V_2(y(\cdot)) = \max\limits_{1\leq i\leq m} |y_i(\cdot)|$, and obviously,$V_2(y) \geq 0$ and $V_2(0) = 0$,as$V_2(y) \to +\infty$, the change $\Delta^+ V_2(y)$along the solution of (5) is

$$\Delta^+ V_2(y) \leq -\left(\max_{1\leq i\leq m} [1 - |d_i| - L_i |a_i|]\right) \max_{1\leq i\leq m} |y_i(n)| = -\alpha_2 V_2(y). \tag{11}$$

Using the inequality $e^{-r} \geq 1 - r(r > 0)$, mathematical induction and (11), we get

$$V_2(y(n)) \leq V_2(y(n-1))e^{-\alpha_2} \leq V_2(y(n-2))e^{-2\alpha_2} \leq \cdots \leq V_2(y(0))e^{-n\alpha_2}. \tag{12}$$

$$V_2(y(n_k + 1)) \leq \max_{1\leq i\leq m} |x_i(n_k) - x_i^*| = V_2(y(n_k)),\ k \in Z^+. \tag{13}$$

This completes the proof of the globally exponential stability of the solution of the system (5).

Theorem 4. *Suppose that the $|D| + |A|\,|B|$ is stable, and further that the impulsive operators $I_{ik}(y_i(n_k))$satisfy*

$$|I_i\,(y_i(n_k))| \leq C\,|x_i(n_k) - x_i^*|\,, i = 1, 2, \cdots, m, k \in Z^+,$$

then the solution of (2) is globally exponential stable.

Proof. If that the $|D| + |A|\,|B|$ is stable, then we can deduce the one and only equilibrium point of (1) and there are positive diagonal matrix$P_1$and positive definite matrixQ_1, such that

$$(|D| + |A|\,|B|)^T\,P_1\,(|D| + |A|\,|B|) - P_1 = -Q_1, \tag{14}$$

by (4) and Theorem2. Constructing the Lyapunov function$V_3(y) = V_3(y(\cdot)) = y^T P_1 y$, and obviously, $V_3(y) \geq 0$ and $V_3(0) = 0$,as$V_3(y) \to +\infty$, the change $\Delta^+ V_3(y)$along the solution of (5) is

$$\begin{aligned}\Delta^+ V_3(y) \leq & |y(n)|^T \left[|D|\,P_1\,|D| - P_1 + 2\,|D|\,P_1\,|A|\,|B| + |B|^T\,|A|\,P_1\,|A|\,|B|\right] |y(n)| \\ = & -\alpha_3 V_3(y).\end{aligned} \tag{15}$$

Hence

$$V_3(y(n)) \le V_3(y(n-1))e^{-\alpha_3} \le V_3(y(n-2))e^{-3\alpha_3} \le \cdots \le V_3(y(0))e^{-n\alpha_3},$$

and as before

$$V_3(y(n_k+1)) \le (1+C)\sum_{i=1}^{m} p_{1i}\left(x_i(n_k) - x_i^*\right)^2 = (1+C)V_3(y(n_k)),\ k \in Z^+. \quad (16)$$

This enough proves the conclusion of the theorem.

Corollary 1. *Suppose that the $f(y)$is a monotonically increasing function* $\min\limits_{1\le i\le m}\left[\left(1-d_i^2\right) - 2_i\left|a_i d_i\right| L_i - a_i^2 d_i^+ L_i^2\right] > 0$, *the orther conditions are the same as the Theoerm 4, the solutionx* of (2) is globally exponential stable.*

Proof. By (15) and proving process of the Theorem4, we have

$$\begin{aligned}\Delta^+ V_3(y) &\le \sum_{i=1}^{m}\left(d_i^2-1\right)p_i y_i^2(n) + 2\sum_{i=1}^{m}\left|a_i d_i\right| L_i p_i y_i^2(n) + \sum_{i=1}^{m} a_i^2 d_i^+ p_i L_i^2 y_i^2(n)\\ &\le -\min_{1\le i\le m}\left[\left(1-d_i^2\right) - 2_i\left|a_i d_i\right| L_i - a_i^2 d_i^+ L_i^2\right]\sum_{i=1}^{m} p_i y_i^2(n)\\ &= -\alpha_4 V_3\left(y(n)\right).\end{aligned}$$

The conclusion has been gotten effortless.

As we known, the positive solutions of a dynamical system is more useful than the others usually. Now we are going to discuss the stability of the positive solution of the system (2).

Corollary 2. *If the $f_i(y_i), i = 1, 2, \cdots, m$ are monotone increasing functions, $y_i(\cdot) > 0, i = 1, 2, \cdots, m$,* $\min\limits_{1\le i\le m}\left[\left(1-d_i^2\right) - 2_i\left(a_i d_i\right)^+ L_i - a_i^2 d_i^+ L_i^2\right] > 0$, *and the orther conditions are the same as the Theoerm 4, the solutionx* of (2) is globally exponential stable.*

Proof. By (15) and proving process of the theorem4, we have

$$\begin{aligned}\Delta^+ V_3(y) &\le \sum_{i=1}^{m}\left(d_i^2-1\right)p_i y_i^2(n) + 2\sum_{i=1}^{m}\left(a_i d_i\right)^+ L_i p_i y_i^2(n) + \sum_{i=1}^{m} a_i^2 d_i^+ p_i L_i^2 y_i^2(n)\\ &\le -\min_{1\le i\le m}\left[\left(1-d_i^2\right) - 2_i\left(a_i d_i\right)^+ L_i - a_i^2 d_i^+ L_i^2\right]\sum_{i=1}^{m} p_i y_i^2(n)\\ &= -\alpha_5 V_3\left(y(n)\right).\end{aligned}$$

The corollary2 will been proved easily.

The next theorem represents a generalization of Theorem 4.

Theorem 5. *Assume that the* $(E-|D|)^{-1}AB$*is diagonal stabal, and the impulsive operators* $I_{ik}(y_i(n_k))$*satisfy*

$$|I_i(y_i(n_k))| \le e^{\beta n_k}|x_i(n_k)-x_i^*| = e^{\beta n_k}|y_i(n_k)|, i=1,2,\cdots,n, k\in Z^+,$$

$\beta<\alpha_6$, *then the solution of (2) is globally exponential stable.*

Proof. Suppose the condition of the theorem be satisfied, if that the $(E-|D|)^{-1}AB$ is diagonal stabal, then we can deduce the only equilibrium point of (11) and there are positive diagonal matrix$P^*$and positive definite matrixQ_2, such that

$$LA(E-|D|)^{-1}P^*(E-|D|)^{-1}AL-P^*=-Q_2. \tag{17}$$

Making the Lyapunov function $V_4(y)=V_4(y(\cdot))=y^TP_2y$ as the theorem 4, here

$$P_2=P^*(E-|D|)^{-1}=diag(P_{21},P_{22},\cdots,P_{2n}).$$

The change $\Delta^+V_4(y)$along the solution of (5) is

$$\begin{aligned}\Delta^+V_4(y) = {} & y(n)^T(DP_2D-P_2)y(n)+2y(n)^TDP_2(AF(y(n)))\\ & +(AF(y(n)))^TP_2AF(y(n))\end{aligned} \tag{18}$$

Let $G=(g_1,g_2,\cdots,g_m), G^{-1}=(g_1,g_2,\cdots,g_m)^{-1}$, where

$$g_i=\begin{cases}|d_i|(1-|d_i|), d_i\neq 0\\ 0,\ d_i=0\end{cases},\ g_i^{-1}=\begin{cases}|d_i|(1-|d_i|)^{-1}, d_i\neq 0\\ 0,\ d_i=0\end{cases}, \tag{19}$$

thus we have

$$2y^TADP_2F(y)=2\sum_{i=1}^m y_id_iP_{2i}a_iF_i(y_i)\le y^T\left(GDP_2D+G^{-1}LAP_2AL\right)y. \tag{20}$$

It follows that

$$\begin{aligned}\Delta^+V_4(y) &\le y^T\left[(E+G)DP_2D-P_2+(E+G^{-1})LAP_2AL\right]y\\ &\le -\lambda_{\min}(Q_2)\|y\|^2\le -\lambda_{\min}(Q_2)/\lambda_{\max}(P_2)V_4(y(n))\\ &=-\alpha_6V_4(y(n)).\end{aligned} \tag{21}$$

Hence

$$V_4(y(n))\le V_4(y(n-1))e^{-\alpha_6}\le V_4(y(n-2))e^{-3\alpha_6}\le\cdots\le V_4(y(0))e^{-n\alpha_6},$$

and as before

$$\begin{aligned}V_4(y(n_k+1)) &= \sum_{i=1}^m p_i^2(x_i(n_k+1)-x_i(n_k)+x_i(n_k)-x_i^*)^2\\ &\le (1+e^{\beta n_k})e^{-\nu n_k}V_4^{1-\nu}(y(n_k)),\ k\in Z^+, 0<\beta<\nu<\alpha_6.\end{aligned} \tag{22}$$

This completes proof of the theorem.

Remark. In above theorems, the three impulsive operators $I_{ik}(y_i(n_k))$are commutative.

4 Instability of Equilibrium of Systems with Impulses

Criteria for instability of equilibria of this neural networks with impulses have not been systematically investigated in the current literature. Here are some new and simple conditions fore such instability.

Theorem 6. *Suppose that if either of the following conditions holds*

$|d_i| - L_i |a_i| > 1, i = 1, 2, \cdots, n$, or $\min\limits_{1\le i\le m} (|d_i| - L_i |a_i|) > 1$,

then the solution of (5) is unstable.

Proof. Suppose $y(n) \neq 0$, we find the change $\Delta^+ V_1(y) > 0$, along the solution of (5) given by

$$\begin{aligned}\Delta^+ V_1(y) &\ge \sum_{i=1}^{m} \{|d_i| |y_i(n)| - (L_i |a_i| + 1) |y_i(n)|\} \\ &= \sum_{i=1}^{m} (|d_i| - L_i |a_i| - 1) |y_i(n)| > 0.\end{aligned}$$

We can see the change $\Delta^+ V_2(y)$along the solution of (5) is

$$\begin{aligned}\Delta^+ V_2(y) &\ge \max_{1\le i\le m} [|d_i y_i(n)| - L_i |a_i| |y_i(n)|] - \max_{1\le i\le m} |y_i(n)| \\ &\ge \min_{1\le i\le m} [|d_i| - L_i |a_i|] \max_{1\le i\le m} |y_i(n)| - \max_{1\le i\le m} |y_i(n)|. \\ &= \left[\min_{1\le i\le m} (|d_i| - L_i |a_i|) - 1\right] \max_{1\le i\le m} |y_i(n)| > 0.\end{aligned}$$

The conclusion of the theorm6 is proved.

Theorem 7. *Suppose that if either of the following conditions holds*

(1) $|d_i| - L_i |a_i| = 1, i = 1, 2, \cdots, n$, and$|I_i (y_i(n_k))| \ge 2 |x_i(n_k) - x_i^*|$;
(2) $\min\limits_{1\le i\le m} (|d_i| - L_i |a_i|) = 1$, $\max\limits_{1\le i\le m} |I_i (y_i(n_k))| \ge 2 \max\limits_{1\le i\le m} |x_i(n_k) - x_i^*|$

then the equilibrium state$x^*$of systems (2) is unstable.

Proof. Suppose condition (1) holds, for$n \neq n_k$, by Theorem6, one has $\Delta^+ V_1(y) \ge 0$, and. for$n = n_k, k \in Z^+$, has

$$\begin{aligned}V_5(y(n_k + 1)) &\ge \sum_{i=1}^{m} |I_i (y_i(n_k))| - \sum_{i=1}^{m} |x_i(n_k) - x_i^*| \\ &\ge (2 - 1) \sum_{i=1}^{l} |x_i(n_k) - x_i^*| = V_5(y(n_k)),\ k \in Z^+.\end{aligned}$$

Next, suppose condition (2) holds, similarly, we have$V_6(y) \ge 0$, for$n \neq n_k$, and for $n = n_k$, $k \in Z^+$, have

$$\begin{aligned}V_6(y(n_k + 1)) &\ge \max_{1\le i\le m} |I_i (y_i(n_k))| - \max_{1\le i\le m} |x_i(n_k) - x_i^*| \\ &\ge (2 - 1) \max_{1\le i\le m} |x_i(n_k) - x_i^*| = V_6(y(n_k)),\ k \in Z^+.\end{aligned}$$

Implying that the system trajectory escapes to$+\infty$along a path, hence the equilibrium $x^*$of (3.2) is unstable. This completes proof of the theorem..

Acknowledgements. This work is supported by National Natural Science Foundation of Chain under Grant 60474011, and Ministry of Education Special Foundation of Chain under Grant 97048722.

References

[1] Jin, L., Nikiforuk, P.N., Guptam, M.: Absolute Stability Conditions for Discrete-time Recurrent Neural Networks. IEEE Trans. Neural NetworKs 5, 954–963 (1994)
[2] Bose, T., Chen, M.Q.: Overflow Oscillations in State-space Digital Filters. IEEE Trans. Circuits Syst. 38, 807–810 (1991)
[3] Bruco, M., Carnimeo, L., Grassi, G.: Aglobalapp Roach to the Design of Discrete-time Cellular Neural Networks for Associative Memories. Int. J. Cir. Theor. Appl. 24, 489–510 (1996)
[4] Hui, S., Zak, S.H.: Dynamical Analysis of the Brain-State-in-a-Box (BSB), Neural Models. IEEE Trans. Neural Networks 3, 86–94 (1992)
[5] Lillo, W.E., Miller, D.C., Hui, S.: Synthesis of Brain-State-in-a-Box (BSB) Based Associative Memories. IEEE Trans. Neural Networks 5, 730–737 (1994)
[6] Bohner, M., Hui, S.: Brain State in a Convex Body. IEEE Trans. Neural Networks 6, 1053–1060 (1995)
[7] Perfetti, R.: A Synthesis Procedure for Brain-State-in-a-Box Neural Networks. IEEE Trans. Neural Networks 6, 1071–1080 (1995)
[8] Shen, Y., Fan, T.G., Liao, X.X.: Qualitative Analysis of Discrete-time Recurrent Neural Networks. Theory and Practice of Systems Engineering 10, 57–61 (2001)
[9] Kennedy, M.P., Chua, L.O.: Neural Networks for Non-linear Programming. IEEE Trans. Circ. Syst. Theory Appl. 35, 554–562 (1988)
[10] Kaszkurewicz, E., Bhays, A.: On a Class of Globally Stable Nerual Circuits. IEEE Trans. Circuits Syst. *I* 42, 497–499 (1995)
[11] Prufer, M.: Turbulence in Multistep Methods for Initial Value Problems. SIAM J. Appl. Math. 45, 32–69 (1985)
[12] Ushiki, S.: Central Difference Scheme and Chaos. Physica D 4, 407–424 (1982)
[13] Yamaguti, M., Matano, H.: Euler_s Finite Difference Scheme and Chaos. Proc. Jpn. Acad. 55, 78–80 (1979)
[14] Yamaguti, M., Ushiki, S.: Chaos in Numerical Analysis of Ordinary Differential Equations. Physica D 3, 618–626 (1981)
[15] Yee, H.C., Sweby, P.K., Griffiths, D.F.: Dynamical Approach Study of Spurious Steady-state Numerical Solutions of Nonlinear Differential Equations–I: The Dynamics of Time Discretization and Its Implications for Algorithm Development in Computational Fluid Dynamics. J. Comput. Phys. 97, 310–349 (1991)
[16] Hopfield, J.J.: Neural Networks and Physical Systems with Emergent Collective Computational Abilities. Proc. Nat. Acad. Sci. USA 79, 2554–2558 (1982)
[17] Gopalsamy, K.: Stability of artificial neural networks with impulses. Applied Mathematics and Computation 154, 783–813 (2004)
[18] Zhao, X.Q.: Exponential Stability of the Steady State Solution of Hopfield Neural Networks with Reaction-Diffusion and Time-Varying Delay Terms under the L^2 Norm. Dynamics of Continuous, Discrete & Impulsive Systems, Applications and Algorithms, Modelling, Control and Applications 2, 776–781 (2005)

[19] Zhao, X.Q.: Qualitative Analysis of Artificial Neural Networks with Impulses. Dynamics of Continuous, Discrete & Impulsive Systems, Series A: Math. Analysis 13, 713–736 (2006)
[20] Lakshmikantham, V., Liu, X.Z.: Stability Analysis in Terms of Two Measures. World Scientific, Singapore (1993)
[21] Gopalsamy, K.: Stability of Artificial Neural Networks with Impulses. Applied Mathematics and Computation 154, 783–813 (2004)
[22] Haydar, A., Rajai, A., Valery, C.: Continuous-time Additive Hopfield-type Neural Networks with Impulses. J. Math. Anal. 290, 436–451 (2004)
[23] Mickens, R.E.: Nonstandard Finite Difference Models of Differential Equations. World Scientific, Singapore (1994)
[24] Potts, R.B.: Nonlinear Difference Equations. Nonlinear Anal. Theory Methods Appl., 659–665 (1982)
[25] Wang, X., Blum, E.K.: Discrete-time Versus Continuous time Models of Neural Networks. J. Comput. Syst. Sci. 45, 1–19 (1992)
[26] Liao, X.X.: Mathmatic Theory and Application of the Stability. The Publishing Company of Central China Normal University, Wuhan (1998)
[27] Hu, S.G.: The Theory and Mathird of The Non-linear Analysis. The Publishing Company of Huazhong University of Science and Technology, Wuhan (1996)
[28] Kaszkurewicz, E., Bhaya, A.: Robust Stability and Diagonal Lyapunov Functions. SIAM J. Matrix Anal., Appl. 14, 508–520 (1993)

Simulation Study of CPG Model: Exploring of a Certain Characteristics of Rhythm of Gait Movement on the Intelligent Creature

Wei Dong, Rubin Wang, and Zhikang Zhang

Institute for Cognitive Neurodynamics, School of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237, China
020060098@163.com

Abstract. Movement of the intelligent beings is various, which is divided into two major types: the rhythmic movement and the non-rhythmic movement. The generation of the rhythmic movement uncertain is induced by the responding signals of the peripheral neural receivers, but is generated spontaneously by the central pattern generator (CPG). CPG not only can generate the rhythmic movement, but also can change the frequency and pattern of movement. This paper is based on the revised CPG neural network model which has relation with the movement of legs. We carry on a series of computer number analysis and imitation to further elaborate many characteristics of the rhythmic movement. The research results show that three greatest characteristics of the rhythmic movement respectively are: rhythm, coordination and variety. The paper further reveals the biological properties of gait movement on the intelligent beings and the characteristics of various rhythmic movements.

Keywords: The rhythmic movement, Central pattern generator (CPG), Rhythm, Coordination, Variety.

1 Introduction

For the animals, the most familiar locomotion is rhythmic movement, the rhythm is a periodic movement which has the symmetry of time and space, such as walking, running, jumping , swimming, flying and breath, chew etc.. Most of the biologists think, the rhythmic movement is irrelevant to consciousness, however, they think it is a spontaneous behavior generated by the lower neural central system. The rhythmic movement is a kind of time-space motor pattern, which is generated and controlled by the central pattern generator. The central pattern generator is located on spinal cord (vertebrates) or on the neural nods between chest and belly (invertebrates). The rhythmic movement has regular manifestations and dynamic characteristics, such as high stability and adaptability. It is a perfect combination between simplicity and practicality.

CPG is a local oscillation network which is constitute of inter-neurons. It can generate a stable the phase interlocked relationship by reciprocal inhibition between neurons, and it can excite related parts of the body to generate periodic rhythm movement

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 138–148, 2009.

by self-oscillation. The synaptic connections of neurons in the CPG have plasticity, so the CPG neural network can reflect various output ways in order to control animals to realize different movement patterns [1].

2 Mathematical Model of CPG

The animal's movement control system is a complicated network, including central nervous system, receptor, sensory organ and skeleton-muscle executive system, as shown in Fig.1. Among them, the CPG in the spinal cord is a center control unit, it can generate rhythm signals to control effectors to realize movement. The advanced central nervous system (brain, cerebellum) can sent out movement instructions to control the beginning and end of rhythmic movement, and can integrate some signals to monitor movement, such as central feedback information from CPG, proprioceptive information, visual information etc.. By the biological reflex mechanism, the feedback information from proprioceptor can coordinate relations among CPG, environment and essence to regulate the output signals of CPG. The whole control system is a hierarchical modular feedback control system, which can maintain the stability of rhythmic movement and the adaptability to real-time changes of external complicated environment [2].

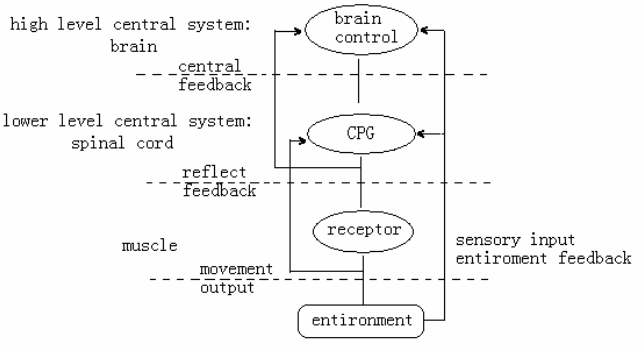


Fig. 1. The control network of the rhythmic movement on animals

For the humans, the human walking is typical rhythmic movement. CPG model in this paper is built on the base of the structure model of the muscle-skeleton system and reciprocal inhibition neural network theory. This kind of neural network model is an application of the interdisciplinary research between biology and biomechanics [3].

The behavior and dynamic characteristics of CPG can be described or imitated in various methods. From engineering point of views, CPG neural circuit can be regarded as a distribute system which consists of inter-coupling nonlinear oscillators. Depending on phase coupling, it can generate different rhythm signals. However, by changing the coupling relations between oscillators, it can generate time-space sequence signals with different phase relationships, for the purpose of realizing various motion patterns.

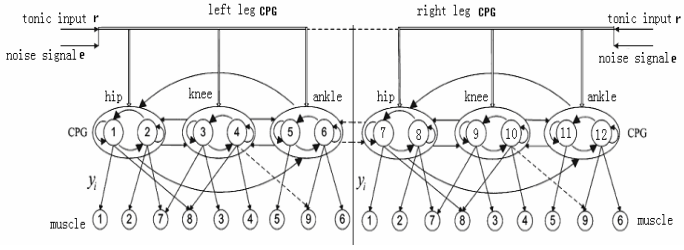


Fig. 2. The leg CPG model

According to simplified three-joint muscle model on leg [4,5] and CPG structure model based on the neural oscillator theory [6,7], this paper will further show the kinematics characteristics of human gait movement and biological significance of various movement properties. Here, the CPG model is shown as follows:

The mathematical model can be written as:

$$\begin{aligned} &\frac{1}{K_f}\tau\dot{X}_i = -X_i - \beta v_i + K_t r + \sum_{i\neq j}\omega_{ij}Y_i + Le \quad i,j = 1,2,\cdots,12 \\ &T\dot{v}_i = -v_i + Y_i \\ &Y_i = f(X_i) = max(X_i, 0) \end{aligned} \tag{1}$$

Here, K_f, K_t respectively are rhythm sensory gain and tonic input gain, i, j represent the 1th-12th neuron, X_i represents the states of the i th oscillator--indicates two states of muscles: flexor and extensor, v_i represents the adaptability of muscles in the recovery process, ω_{ij} is the weight for connections among twelve neurons of two legs, τ is the state constant, T is the adaptability constant, some details appear in references[3,6,7]。

The motor patterns generated by above-mentioned CPG model act on three-joint muscle groups on human legs, and by controlling movements of the leg muscles, it can show a lot of characteristics of the rhythmic movement.

3 Simulation Study on Gait Rhythm Movement

3.1 There Are Rhythmic and Coordinate on the Animal's Movement (Take Constant Input as an Example)

Through proper adjustment to parameters of the CPG model [3,6,7], a stable pattern of rhythmic movement can be obtained. Adjusting to the state constant τ and the adaptability constant T, it can make the output patterns of the model to reach the desired frequency.

In the Matlab environment, taking numerical simulation to the equation (1), the output patterns not only has rhythm (Fig. 3a), but also satisfies the coordination of the gait movement (Fig. 3b).

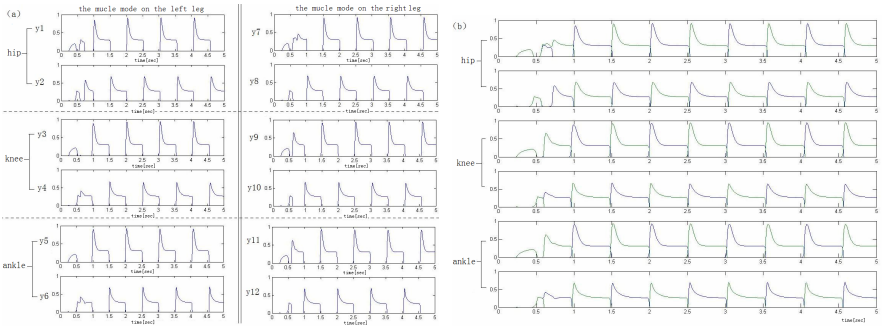


Fig. 3. The outputs of CPG during rhythmic walking. (a The rhythm characteristic of the leg movement; b The harmonized characteristic of the leg movement, the locked out-of-phase).

In Fig.3 it is clear to see that the muscle movement patterns of the corresponding parts are generated with a certain frequency and similar shape. Comparing the phases of the rhythmic motor patterns on two legs, it is obvious to find that the muscle movement patterns between the contralateral corresponding parts have an out-of-phase relationship.

3.2 The Animals' Rhythm Movement Has Variety

To the rhythmic movement, the animals can not only generate a certain motor rhythm but also change their speeds and movement patterns in a certain scope. Some animals not only can change the frequency of the rhythmic movement, meanwhile, they can change the patterns of the rhythmic movement themselves, which embodies the variety of the rhythmic movement. For example, for quadruped, there are four gait types at least: walking, jogging, hiking and running; For insects, there are two gait types at least: walking and jumping; For biped, the variety of gait movement still exists--running, jumping, striding, pacing, sliding, and so on, there are further richer variety than another animals. According to the request of the motion velocity, the animals can choose the proper gait to achieve optimal energy consumption [8], which is applied in the robot motion control system. For the CPG model, an important aspect on the rhythmic movement is that, the motor patterns can be changed by changing the strength of the electrical stimulation signals. However, for the CPG mechanism, besides the interpretation of the local oscillation network, yet, it is not clear to understand the mechanism on changing its frequency and patterns [9].

Gait conversion is the regulation of CPG network outputs in some way to generate some time-space sequence signals with different phase relationships, for the purpose of controlling multiped or biped to generate different movement patterns, which exactly shows the variety of the gait movement. Then, we will research the variety of the rhythmic movement by analyzing each parameter of the model:

(1) The variety of the movement frequency and pattern

Based on our previous research [3], there are the following characteristics on each parameter of the CPG model:

a) The value of compulsory input signal r determines the amplitude of the rhythmic movement.

b) The state constant τ and the time constant T determine the frequency and shape of the rhythmic motor patterns.

c) The weight ω_{ij} includes three types: one is the weight for connections between the neurons in a same joint position, another is the weight for connections between the neurons in the different joints position of ipsilaterala leg, the last is the weight for connections between the neurons of the contralateral legs. The selection of weight for connections will directly influence the rhythm and coordination of rhythmic movement generated by CPG model.

d) The parameter β and the variable v_i reflect the muscle adaptation on the recovery process.

According to the influence degree of each parameter of CPG mathematics model, the gait patterns with various frequencies can be realized by changing CPG network itself. The detailed method is to regulate each parameter, which will generate various patterns. The basic structure diagrams of these patterns are similar to Fig.3. Here, we don't make detailed analysis.

(2) The variety of the input signals

Biological research shows, CPG network can couple with external input signals to deliver input patterns, so the outputs of the network can be regulated by changing input signal in order to realize gait conversion [1]. The neural mechanisms of the rhythmic movement is irrelevant to the advanced neural central on brain, but in the conversion process of rhythmic movement brain plays a regulation role, for which we are trying to use numerical simulation to exhibit different rhythmic movement types: First, the spontaneous rhythmic movement which generated from spinal cord is invested, then the rhythmic movement under the regulation of the cerebral cortex signals is invested.

For the rhythmic movement, by making the lower neural system self-oscillation, the motor control system which generated from spinal cord can spontaneously generate neural signals [9]. It is considered that these neural signals are from some internal-stimulations, in fact, internal-stimulation can be regarded as the constant input signal of CPG model. When the rhythm of the gait movement is changed, the brain can set out some directions which indicate that the motor rhythm has been changed. However, to the limb movement, the input signal generated by cerebral cortex mostly exists in the form of the sine wave [10]. In order to deeply understand the characteristics of various rhythmic movements, we compare the rhythmic movement of the different tonic inputs with the rhythmic movement under the regulation of neural signals.

a) The rhythmic movement with constant input signals

The patterns of the rhythmic movement with constant input signals have been given in the Fig.3. In addition, in this paper what is interested in is:

i) Comparing simulation results with the different values of constant input signals

For the different constant input signals r, the corresponding output Y_i of the model has many variety, some details appear in Fig.4 (Taking the output Y_i of CPG generated by different input signals as an example).

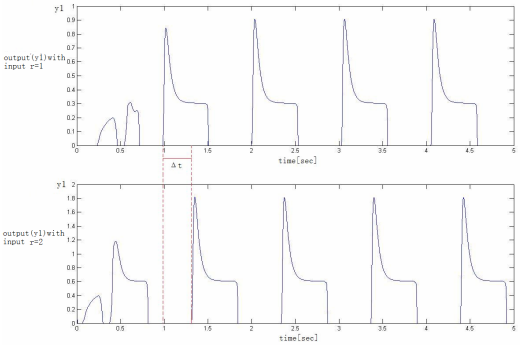


Fig. 4. Comparing the output value y1 when r=1 and r=2

In Fig.4, it shows that, for different constant input signals r, the patterns of the output signals of CPG model almost are same (The frequency is exactly same.), what is different is that the amplitude of the rhythmic movement and there is a time difference Δt on the required time generating a stable rhythmic movement. An important result is got, that is, the rhythmic movement generated by the CPG system in the role of the spontaneous neural signals, with different constant input signals caused by self-oscillation, has different regulation time. If the values of the constant input signals get bigger, the rhythmic movement will not only increase the amplitude of the rhythm, but also there is a lag time. This phenomenon indicates that, due to the increase of input energy (the increase internal stimulation, r=2), the energy of the rhythmic movement in the musculoskeletal system will also increase, which is reflects as the enhanced amplitude strength of the rhythmic movement. However, the value of constant input signals is relatively small before internal stimulation increases (r=1). Comparing with the stronger energy in the musculoskeletal system when internal stimulation increases, the original energy is smaller. Therefore, the increase of the amplitude energy is at the cost of the lag time in the starting time of the rhythm. For example, for human and animals, there is a lag time in the process of speeding up when running and uniform motion, or else do not need the acceleration.

ii) The output patterns with the constant input signals changing suddenly.

In Fig.5, it shows that , when the constant input signal changes suddenly from 1 to 2, the amplitude of periodic patterns of CPG output signals will change accordingly. The input signal changes suddenly if the pattern of the CPG output signals is in a state of excitement, then the output signals also will change immediately. The total time of the excited time of the CPG outputs before the input signal changes suddenly and the excited time of the outputs after changing suddenly is the same as the excited time without changing suddenly (about 0.5s). Some details appear in Fig.5 (Comparing the output signal Y_2 of CPG after changing suddenly in Fig.5). After the input signal changes suddenly, if the CPG's output patterns get back to the new state of rhythmic movement, it needs a certain delay time. In Fig.4 the biological significance of the delay time has been explained in detail. It should be emphasized that, in Fig.5 it shows clearly that the changes of the constant input signal do not change the rhythm frequency of the CPG output signals. It is still a cycle T≈1s. The exciting time and the

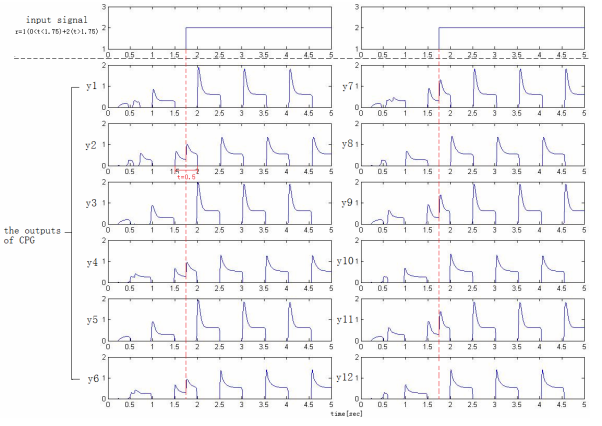


Fig. 5. The output patterns of CPG when the constant input signals changing suddenly

suppression time of the output signals respectively are about 0.5s. It indicates that the rhythmic movement on biped has symmetry and the out-of-phase relationship. It reflects high coordination and consistency of the gait movement.

iii) CPG output signals with the constant input signals from changing suddenly to getting back to the original state

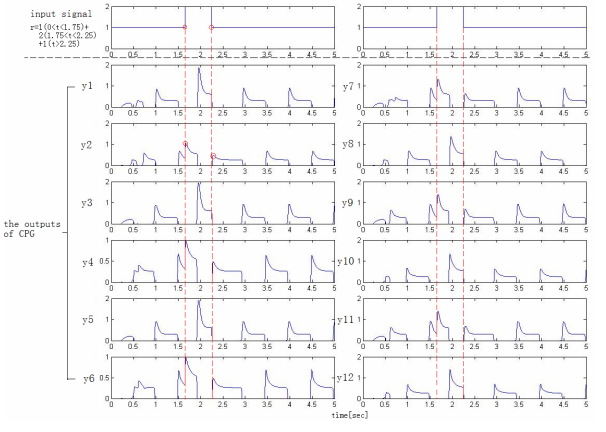


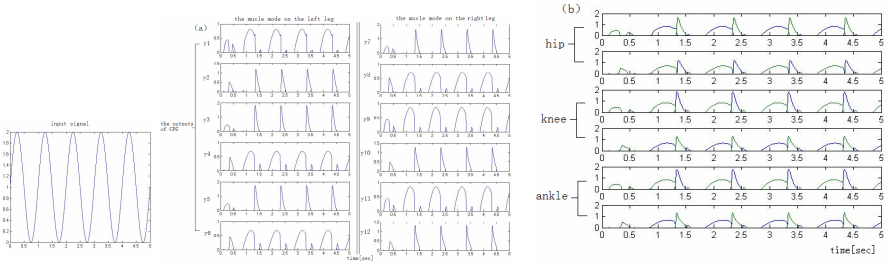
Fig. 6. CPG output signals with the constant input signals from changing suddenly to getting back to the original state

In Fig.6, it shows that , the input signal changes suddenly when the rhythm pattern of signals is a state of excitement, CPG output signals will also change suddenly (see the dots in Fig.6), that is, the output signals change in an instant. At the same time, the amplitude of the output patterns directly relates to the value of the input signal after changing suddenly, and it seems to be irrelevant to the value of the input signal before changing suddenly. After changing suddenly, the output signals can get back to the original rhythm patterns. This phenomenon indicates that the changes of the gait

rhythm patterns are determined by the property of the input signal. Because the characteristic of skeleton-muscle is that, if the stimulation frequency gets higher, the tension arising from muscle will be stronger [11].For the output patterns, whose performance is the increasing amplitude of the rhythmic movements, and the increase is very large. This kind of rhythm output patterns changes with the changes of the input signal, which reflects the characteristic of the rhythmic movement in the case that the human gait rhythmic movement is effected by external environment.

b) The rhythmic movement with the sine input signal

For the sine input signal, it cannot change the rhythm and coordination of CPG output patterns, just to change the basic shape of rhythmic movement.



r=1+sin(2*π*t)

Fig. 7. CPG output signals when the input signal is sine wave

In Fig.7, it shows the CPG output patterns in the role of the sine input signal. It can be seen clearly that, the output patterns of CPG in the role of the sine input signal and the ones in the role of the constant signal are different, but there are still the rhythm (Fig.7a) and the coordination (Fig.7b).It seems that the rhythm period of CPG output patterns with the sine input signal is longer than the one with the constant input signal, the cycle is T> 1s. Comprehensibly, the exciting time of the cells in the role of the motor cortex signals is longer than the exciting time in the role of self-stimulated signal.

In the following figure, it shows that the CPG output patterns in the case that the input signal is from the constant to the sine wave then recover to the original constant. We try to explore the influence and the regulation of neurons in the brain's motor cortex to the rhythmic movement in the process of the normal rhythmic movement.

The sine input signal also will not change the frequency of CPG output patterns (a period is about 1s, the exciting time and the suppression time respectively are 0.5s).It is emphasized that, the input signal changes from the constant signal to the sine signal when the CPG output is in a state of excitement, the total time of the excited time of the CPG outputs before the input signal changes suddenly and the excited time of the outputs after changing suddenly is longer than the excited time without changing suddenly (t> 0.5s), which is different from the case of the self-stimulated signal(comparing Fig.5 with Fig.9).

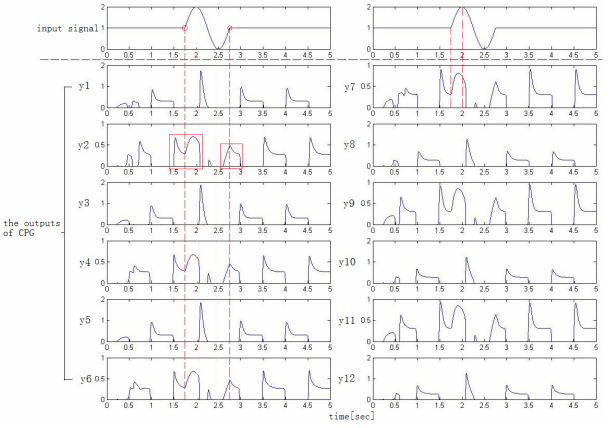


Fig. 8. The CPG output patterns with the input signal from the constant to the sine wave recover to the original constant

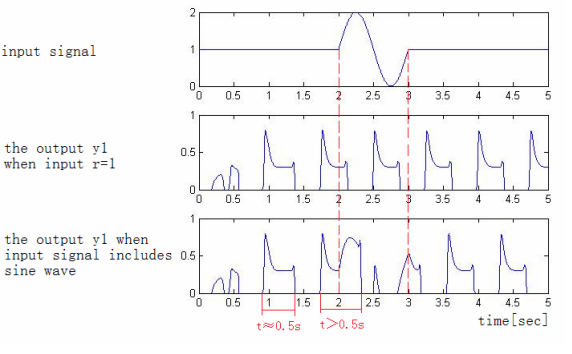


Fig. 9. Comparing the CPG output y1 when the input signal from the constant to the sine wave recover to the original constant to the output y1 with the constant input signal

According to above stimulation results, it can be known that, for the CPG model, when the input signal is the constant or sine signal, if the input signal is changed, the frequency of the output signals cannot change, just is that the amplitude of the resulting patterns occurs changes. Carrying on the computer simulation in the case of having noise, it is found that the rhythm and the coordination of the CPG output signals still are not influenced [3]. However, the recovery time of CPG output signals with sine input signal (simulating the role of neurons in the motor cortex) is longer than the one with the constant input signal(simulating the role of the cell self-stimulated signals), which agrees with the facts. That is, the regulation time of the rhythmic gait movement effected by the feedback signals from brain is longer than the regulation time of the rhythmic movement itself in the case of conditioned reflex. The numerical results prove that, the motion control system which generated from spinal cord generally is realized by conditioned reflex, for the command system of the cerebral cortex cannot or do not need to respond in time. In the mechanics structure of the

musculoskeletal system, the system monitors the movement of the limbs by the spinal control center. For humans and animals, in the nature world it takes an evolution for millions of years to achieve a most reasonable control strategy, which can release the nervous system from the complicated control to deal with more senior control tasks--such as path planning, obstacle avoidance, and making decision etc.--which can reduce the burden of neural computation [1]. This control strategy has given rise to the attention of researchers who research the walking robot, for the control strategy has great significance in the aspect of reducing energy consumption and others, and based on this theory a series of passive dynamic robots have been worked out [8]. The gait rhythmic movement in the participation of the cerebral cortex nerve signals is a kind of more advanced and more complex movement, which needs more feedback signals and neural computations. So the delay time is necessary for restoring stable gait rhythm.

4 Conclusions

By the computer simulation and numerical analysis, based on CPG model which simulates human gait movement, this paper shows the characteristics and biology results of the rhythmic gait movement as follows:

1) The human rhythmic gait movement is divided into: the spontaneous gait movement and the gait movement with the regulation of the central nervous system. Their response times and regulation modes are different. For these two types of gait movements which present different control patterns, the rhythm of the human gait movement is various. Besides presenting rhythm, various patterns of rhythmic movement also present symmetry and out-of-phase relationship, which reflect the high coordination and consistency of gait movement.

2) The rhythmic gait movement generated spontaneously by CPG system in the role of different constant input signals will need different regulation time to reach stable rhythm output patterns. If the values of the constant input signals get bigger, there is a time delay in the starting time of the rhythm and the amplitude of rhythm will also be increased. Generally speaking, the increase of the amplitude energy on the rhythmic movement of intelligent beings is at the cost of a time lag of the starting time of the rhythm.

3) The result of reference [8] has been proven by numerical simulation. That is, the characterization of the skeleton-muscle is that, if the stimulation frequency gets higher, the tension arising from muscle will be stronger. This is because that the values of the input signal which indicates the strength of the internal stimulation can change the rhythmic gait pattern. For the output pattern, the amplitude of the rhythmic movement will increase suddenly when the internal stimulation increases. The rhythm output patterns change with the changes of the input signal, which reflects the characteristic of the human gait rhythmic movement when is effected by external environment.

4) By numerical simulation, it has been proven, the responding time of the rhythmic gait movement in the role of the movement control system which generated from spinal cord, is faster than the one in the regulation of the central nervous system. This is in line with the neurobiology experiment results [11].

In the aspect of the control mechanism of CPG the new theories and new methods for the robot's movement control are proposed, whose application is to improve the robot's movement performance and the adaptation to environment, and to provide helps for solving the problems on the walking of the biped robot. More and more scientists and engineers are interested in the simulation studies of human gait movement, and the analysis of the human gait rhythm still need more thorough researches. In the future, we will continue to study the stability of gait and the influence of external feedback to walking, so as to make these researches closer to the facts, which is significant for the development of the walking robot and the rehabilitation of the physically disabled.

References

1. Zheng, H., Zhang, X., Li, T., Duan, G.: CPG-based Methods for Motion Control of Robot. High Technology Letters 7, 64–68 (2003) (in Chinese)
2. Pandy, M.G.: Computer Modeling and Simulation of Human Movement. Annu. Rev. Biomed Eng. 3, 245–273 (2001)
3. Wei, D., Wang, R., Shen, E., Zhang, Z.: The Simulation Study on the Pattern of Muscles Controlled by CPG in Rhythm Gait Movement. Journal of dynamics and control 6, 327–331 (2008) (in Chinese)
4. Matsuoka, K.: Mechanisms of Frequency and Pattern Control in the Neural Rhythm Generators. Biol. Cybern. 56, 345–353 (1987)
5. Matsuoka, K.: Sustained Oscillations Generated by Mutually Inhibiting Neurons with Adaption. Biol. Cybem. 52, 367–376 (1985)
6. Zhang, D., Zhu, K., Zheng, H.: Model the Leg Cycling Movement with Neural Oscillator. In: IEEE Inter. Conf. on Systems, Man and Cybern., pp. 740–744 (2004)
7. Zhang, D., Zhu, K.: Modeling Biological Motor Control for Human Locomotion with Function Electrical Stimulation. Biol. Cybern. 96, 79–97 (2007)
8. Zhang, J., Wang, R., Shen, E., Zhang, Z.: An Exploration of Dynamics on Neural Control Mechanism of Insect Locomotion. Journal of Dynamics and Control (in press, 2008)
9. Ogihara, N., Yamazaki, N.: Generation of Human Bipedal Locomotion by a Biomimetic Neuro-musculo-skeletal Model. Biol. Cybern. 84, 1–11 (2001)
10. Mitsuo, K.: The Computation Theory of Brain. Industry publishing (1996) (in Japanese)
11. Diao, Y.: The Principle and Application of Biological Mechanics. Tongji University publishing (1991) (in Chinese)

Recognition of Altered Rock Based on Improved Particle Swarm Neural Network

Yunjun Zhan and Yanyan Wu

Department of Resources and Environment Engineering, Wuhan University of Technology, Wuhan 430070, China
zhanyj998@vip.sina.com

Abstract. PSO (particle swarm optimization) algorithm is apt to slow down and prematurity during the evolutionary anaphase. Besides, the algorithm of BP neural network also encounters some problems such as slowness in constringency, longer training time and so on. Aimed at these phenomena, PSO algorithm can be improved in two aspects: reinforcing the diversity of particles and avoiding the prematurity of particle swarm, therefore the algorithm of particle swarm neural network based on improved algorithm is presented here. Finally, this algorithm is applied to the recognition of hyper-spectral altered rock, which overcomes the disadvantage of local minimization for BP algorithm, and trained network shows great generalization ability. The instance indicates that improved PSO-BP algorithm is effective in the recognition of hyper-spectral altered rock.

Keywords: Particle swarm optimization, Neural network, Pattern recognition, Altered rocks.

1 Introduction

Due to its capability of self-study, self-structure, fault – tolerance and simulation of nonlinear relation, neural network is suitable to resolve the problem of complicated pattern classification. Hornik[1] adopted three-layer feed-forward neural network of Sigmoid response function to simulate complex nonlinear relation, but the key point of realizing above capability is to train neural network sufficiently. Therefore, training algorithm has decisive effect on the performance of pattern classification of neural network. Although BP algorithm is the most pervasive among neural network training algorithms, it depends on the selection of initial weights because of the dependency on gradient descent, which causes the slowness in constringency and tends to plunge into local optimization. The limitation of BP results in disagreement and inscrutability of the output of the trained neural network, which brings about reduce of reliability of pattern classification.

Based on swarm intelligent theory, PSO can intelligently guide optimized search by swarm generated by cooperation and competition among particles [2]. It is a rising random global optimized algorithm, which is successfully applied to many practices, such as optimization of nonlinear function [3], voltage stability control [4], dynamic object optimization [5] and so on. Wang Suihua [6] improves the ability of avoiding

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 149–155, 2009.

local extremum in PSO by mending the basic model, and five tests based on norm function revealed that improved algorithm is better than basic PSO and inertia weight model PSO [6].

This paper combines PSO and BP to generate a new PSO-BP training algorithm which integrates both advantages in it. By combining the global optimization ability in PSO and instructed search idea in BP, the algorithm accelerates constringency and avoids local constringency, and trained network represents great generalization power. The results of the experiment shows that, PSO-BP is effective neural network training algorithm, which reaches the purpose of altered rock recognition by optimized network and can be used to deal with the problem of pattern classification.

2 A New PSO-BP Algorithm of Neural Network Training

2.1 The Principle of PSO-BP Algorithm in Training Neural Network

The chief application of PSO in artificial neural network is to optimize the weight of ANN. That is to substitute traditional learning algorithm by PSO. When using PSO to train neural network, the spatial location of particle corresponds to all link weights and thresholds of neural network, and the error sum of square of exact output and expected output of network is used as fitness value. And PSO is used to search optimized particle location, that means optimized weight and threshold of neural network, to make the error sum of square of network output be minimum. PSO-BP algorithm firstly updates speed and location of PSO for once, then learns BP for once, and then repeates like that until satisfying the expected error requirement.

2.2 Improved PSO

According to the standard PSO presented by Shi and Eberhart in 1998 [3,7], its mathematic description is: Suppose the dimension of search space is D, the total number of particle in swarm is N, the location of particle i is expressed as vector $X_i=(x_1, x_2, \ldots, x_d)$, and the flight speed is described as vector $V_i = (v_1, v_2, \ldots, v_d)$, and the optimized location in the flight of particle i (the location has optimized is marked as *pbest*, and optimized individual of all N individuals, that is global optimized individual is marked as *gbest*. After the two extremums (*pbest* and *gbest*) is attained, particle updates its speed and location by the following formula:

$$v_i = \omega \times v_i + c_1 \times rand() \times (pbest_i - x_i) + c_2 \times rand() \times (gbest_i - x_i) \quad (1)$$

$$x_i = x_i + v_i \quad (2)$$

Here, rand() is a random number between 0 and 1, c_1 and c_2 is learning factor, ù is inertia factor, non-minus. If ù is larger, global optimization ability is stronger, and local optimization is weaker.

(1) is composed of three parts. The first one is previous velocity of the particle. It represents the self-confidence in the current state of their own movement, which

conducts inertia movement according to their own rate. The second is about cognitive part. It stands for the thinking about the particle itself and encouraging it flying to the best position which it has ever found. The third is social part. It stands for the information sharing and mutual cooperation among the particle and encouraging it flying to the best position which the particle group has ever found.

In the particle group algorithm, c_1, c_2 are two positive constants, called cognitive and social parameter respectively. In the population-based optimization methods, it always hopes that the individuals in the initial stage can wander through the entire optimizing search space, without early trapped into the local optima. Meanwhile, in the end stage of the algorithm it improves the velocity and accuracy of convergence and effectively finds the global optimum[8]. In the initial stage, it has better to have a large "cognitive" part and a small "social" part for the benefit of the whole algorithm search in the optimization space and to improve the velocity and accuracy of convergence. So, take c1 and c2 as follows:

$$c_{1t} = 4 \times (\left| f_{at} - f_{gt} \right|) / f_{at} \tag{3}$$

$$c_{2t} = 4 - c_{1t} \tag{4}$$

$$f_{gt} = \frac{1}{N} \sum_{i=1}^{n} f_{it} \tag{5}$$

Where, t is for the current iteration algebra, fat is the average fitness of the current generation, fgt is the current global optimum location of the *gbest* fitness. So (1) can be improved as:

$$v_{i(t+1)} = \omega \cdot v_{it} + 4 \cdot rand()[\frac{\left| f_{at} - f_{gt} \right|}{f_{at}} (pbest_{it} - gbest_{it}) + (pbest_{it} - x_{it})] \tag{6}$$

When the group fitness variance is less than the value of the convergence precision square, and the difference between the current global optimum particle *gbest* fitness and the particle theory optimal fitness value or the best experience value is less than the value of precision when convergence, it notes convergence algorithm and the overall situation was not optimal solution, that is to say, it gets a premature convergence. Therefore, it needs to conduct mutation for PSO in order to make the particles escape local optimum and goes with further search in the other regions. The mutation method is: when the particle group trapped into premature convergence, or fitting with the variation conditions, a velocity-changed mutation operator can be applied to break the original gathering status. Taking a fresh start in the solution space for further search can be to conduct mutation in the velocity of the particle group. Equation (6) is updated as:

$$v_{i(t+1)} = \omega \cdot v_{it} + 4 \cdot rand()[\frac{|f_{at} - f_{gt}|}{f_{at}}(pbest_{it} - gbest_{it}) + (pbest_{it} - x_{it})] + \delta \times \mu \times v_{\max} \quad (7)$$

Where, δ is a sign value. It is 0 or 1, by default 0, When the particles is trapped into a premature convergence, δ is set to 1.ìmeans the current extent about mutation operator impact to current velocity. Its value is:

$$\mu = \min\left\{\left|\frac{f_{gt} - f_t}{f_t}\right|, 1\right\} \quad (8)$$

Here: f_t is the best theory value or the best experience value.

2.3 Hybrid PSO-BP Neural Network Training Algorithm Steps

①Initialize a group of particles for the size N, including the initial position and the initial velocity.

②Initialize *pbes*t and *gbest*. The initial location of the particles is set as *pbest* and that of the smallest fitness particles is set as *gbest*.

③Evaluate the desired fitness values for all the particles. Compare the evaluated fitness value of each particle with its *Pbest*. If current value is better than *pbest*, then set the current location as the *pbest* location. Furthermore, if current value is better than *gbest*, then reset *gbest* to the current index in the particle array.

④Loop termination when reaching the largest evolution algebra G, or (and) the extremum *gbest* in the whole group meeting the scheduled minimum fitness value.

⑤The best preserved and the worst removal. Preserve the overall extreme *gbest* until the smaller overall value of extreme value appears. The largest Fitness value will be removed in order to make the search solution toward the optimal direction.

⑥Change the velocity and location of the particle according to (1) and (3).

⑦Decode the space location vectors of the particle into the network connection weights and threshold values. It use BP algorithm to learn neural network and re-coding the weights and threshold values. And then turn to step ③.

3 Application Example

3.1 The Selection of Neural Network Training Samples

Altered rock spectral data by field survey are used to train and identify samples for the network. The measuring instrument is ASD FieldSpec FR spectroradiometer with the spectral range of 350 -2500nm. When the spectral range is from 350nm to1000nm, the spectral resolution is 3nm, while from 1000nm to 2500nm, the spectral resolution is 10nm. Many profiles crossing the whole mining area are studied with the measuring points set every 100 meters. And more than 200 spectral data documents are obtained at last.

Take the sample spectrum of illite, chlorite, gypsum and kaolinite for analysis (shown in Fig. 1).

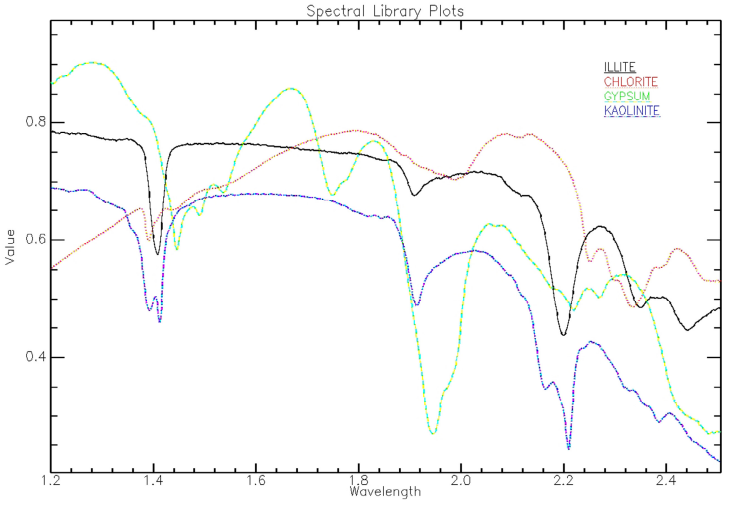


Fig. 1. Spectral curve of illite, chlorite, gypsum and kaolinite

Fig.1 shows that the characteristic peak of the above four altered rocks centralized on the interval of 1220-2500nm, thus the 128 paths can be used as the input of the neural network. In this way, the high spectrum feature information is made full use of on the one hand and the input number of neural network decreases, simplifying the scale of the network on the other hand. The output of illite, chlorite, gypsum and kaolinite are respectively set as [0 0], [0 1], [1 0], [1 1]. The corresponding constructed neural network holds three-layer structure, 128 input neurons and 2 output neurons. For the confirmation of neuron number of hidden layer, the following method can be adopted.

$$h = \sqrt{k + m} + a \tag{9}$$

Where h is the cell number of the hidden layer, k is cell number of the input layer, m is the cell number of the output layer, a is a constant among [1, 10]. Through repeated experimenting and dynamically adjusting the node number of the hidden layer, the neuron number of the hidden layer is confirmed as 16.

3.2 Result Analysis

When using the mixed PSO-BP algorithm to train the above established neural network, the parameters are respectively set as: N=30, G=3000, lr=0.01, V_{max}=0.5, $ù_{ini}$ = 0.9, $ù_{end}$ = 0.4, minfitness=0.01. In the end of running, the current algebrag=106, minfitness=0.0092, elapsed_time=23.610426s, and the training error (minfitness) curve is shown as Fig.2. In order to compare conveniently, Fig.3 shows the training error curve of the BP algorithm. Here, max_epoch=6000, err_goal=0.01, lr=0.01, elapsed_time=513.073500 s.

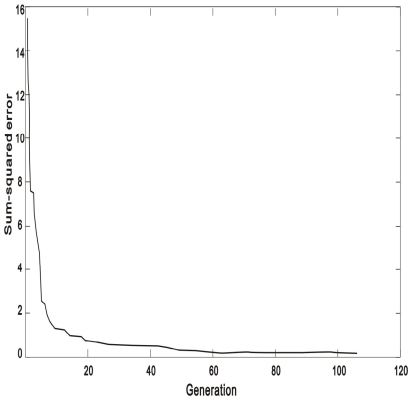


Fig. 2. Network error of mixed PSO-BP Algorithm

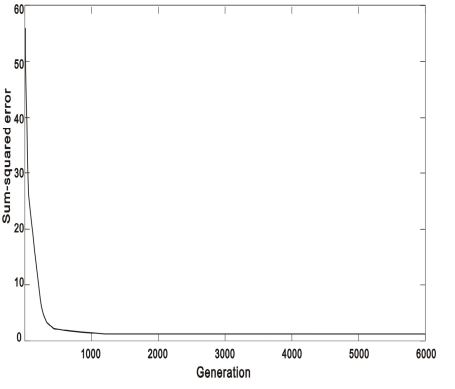


Fig. 3. Network error of BP Algorithm

The above comparison shows that using BP algorithm to train the weights of neural network, the local minimum is reached in advance when the square error sum converged to 0.9656 at the step of 300, thereafter the value doesn't converge even at the maximum step of 6000. BP algorithm falls into local minimum very easily, and it is even more so when the network structure is complex and the training samples are in great number and dimension. While the mixed PSO-BP algorithm has the ability of global optimization, can avoid local minimum and even converge quickly with the training error requirement can be satisfied only when g=106 and elapsed_time=23.610426s.

The result is shown as Fig.4 and Fig.5 by identifying the training samples.

Fig. 4. The recognised result of chlorite and kaolinite

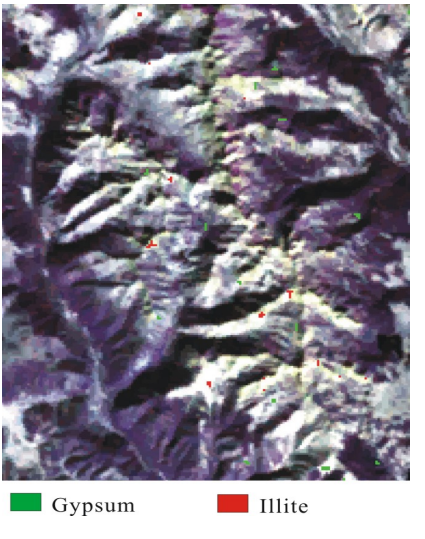


Fig. 5. The recognised result of illite and gypsum

4 Conclusion

The improved and mixed PSO-BP algorithm integrates the strong macroscopic search property of PSO algorithm and the instructive search idea of BP algorithm, and exerts the advantages to avoid local minimum problem which easily appeared in BP algorithm and increase the convergence speed. Identifying the sample set by using the training network, the identification correct rate is 82%, which illustrates that the neural network trained by the improved and mixed PSO-BP algorithm holds great generalization ability. The example shows that it is effective by applying the improved and mixed PSO-BP algorithm in the identification of altered rocks.

Acknowledgement

The work was supported by the National Natural Science Foundation of China (No.40 601076, 40572166), the Natural Science Foundation of Hubei Province(No.2007ABA 322).

References

1. Hornik, K., Stinchcombe, M., White, H.: Multilayer Feed-forward Networks Are Universal Approximators. Neural Networks 2, 359–366 (1989)
2. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: IEEE International Conference on Neutral Networks, pp. 1942–1948. IEEE Press, Australia (1995)
3. Shi, Y.H., Eberhart, R.C.: Empirical Study of Particle Swarm Optimization. In: IEEE International Congress on Evolutionary Computation, pp. 6–9. IEEE Press, USA (1999)
4. Yoshida, H., Kawata, K., Yoshikazu, F.: A Particle Swarm Optimization for Reactive Power and Voltage Control Considering Voltage Security Assessment. IEEE Transaction on Power System 15, 1232–1239 (2000)
5. Carlisle, A., Dozier, G.: Adapting Particle Swarm Optimization to Dynamic Environments. In: IEEE International Conference on Artificial Intelligence, pp. 11–15. IEEE Press, USA (2000)
6. Wang, S., Feng, N., Li, A.: A Novel Particle Swarm Optimization Algorithm. Computer Engineering and Applications 13, 109–110 (2003)
7. Dongsheng, S., Yuming, D., Chunlin, Z.: Application of Particle Swarm Optimization to Identify Gamma Spectrum with Neural Network. Nuclear Techniques 30, 615–618 (2007)
8. Jia, H., Zhihui, C., Yingxin, Y.: Integrative Improved Particle Swarm Optimization Neural Network Arithmetic. Computer Engineering and Design 29, 2890–2896 (2008)

Development of Design Strategy for RBF Neural Network with the Aid of Context-Based FCM

Ho-Sung Park[1], Sung-Kwun Oh[2], and Hyun-Ki Kim[2]

[1] Industry Administration Institute, The University of Suwon, San 2-2 Wau-ri, Bongdam-eup, Hwaseong-si, Gyeonggi-do, 445-743, South Korea
parkhs@suwon.ac.kr
[2] Department of Electrical Engineering, The University of Suwon, San 2-2 Wau-ri, Bongdam-eup, Hwaseong-si, Gyeonggi-do, 445-743, South Korea
{ohsk,hkkim}@suwon.ac.kr
http://autosys.suwon.ac.kr

Abstract. In this paper, we develop a new design strategy of Radial Basis Function (RBF) neural network and provide a comprehensive design methodology and algorithmic setup supporting its development. The architecture of the network is fully reflective of the structure encountered in the training data which are granulated with the aid of clustering techniques. More specifically, the output space is granulated with use of FCM clustering while the information granules in the multidimensional input space are formed by using a so-called context-based Fuzzy C-Means which takes into account the structure being already formed in the output space. A series of numeric studies exploiting synthetic data and data from the Machine Learning Repository provide a detailed insight into the nature of the algorithm and its parameters as well as offer some comparative analysis.

Keywords: Context-based Fuzzy C-Means, Radial Basis Function (RBF), Neural network, FCM clustering, machine Learning data.

1 Introduction

Given the simple topological structure and universal approximation ability, radial basis function (RBF) neural networks have been widely studied and applied to many categories of problems such as those arising in pattern recognition, signal processing, time series prediction, and nonlinear system modeling and control, cf. [1-6]. Since the very inception of this concept in neurocomputing, we can witness a number of interesting and useful expansions of the generic topology of such neural networks. In this study, we view RBF neural networks as predominantly data driven constructs whose processing is based upon an effective usage of experimental data through a prudent process of information granulation. Our ultimate objective is to offer a comprehensive design methodology of neural networks by (a) using fuzzy granulation realized by means of a specialized, output variable - oriented machinery of fuzzy clustering [7], (b) considering number of clusters for each context which reflects a nature of the input space by context-based FCM clustering [8], [9], and (c) realizing functional links of the neuron present in the output layer of the network.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 156–163, 2009.

2 Algorithm of Fuzzy Granular Computing

2.1 Fuzzy C-Means Clustering

The FCM clustering has been applied to a variety of areas, including image and data preprocessing for system modeling [10].

Let $\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_N$ be n-dimensional patterns defined in the space of inputs, $\mathbf{x}_k \in \mathbf{R}^n$, As usual, the basic objective function (performance index) of the original FCM method is defined as a sum of squared errors

$$Q = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^{m} \left\| \mathbf{x}_k - \mathbf{v}_i \right\|^2 \tag{1}$$

With $U=[u_{ik}]$ denotes a partition matrix, $U \in \mathbf{U}$, and "c" stands for the number of the clusters. The parameter "m" used above "m">1 is often referred to as a fuzzification coefficient. U denotes a family c×N partition matrix, namely

$$\mathbf{U} = \left\{ u_{ik} \in [0,1], \quad \sum_{i=1}^{c} u_{ik} = 1 \; \forall k, \quad 0 < \sum_{k=1}^{N} u_{ik} < N \; \forall i \right\} \tag{2}$$

Formally, the optimization problem is expressed in the form

$$\begin{aligned} &\min_{U, \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_c} Q \\ &\text{subject to } U \in \mathbf{U} \end{aligned} \tag{3}$$

The elements of U are computed as

$$u_{ik} = 1 \Big/ \sum_{j=1}^{c} \left(\frac{\left\| \mathbf{x}_k - \mathbf{v}_i \right\|}{\left\| \mathbf{x}_k - \mathbf{v}_j \right\|} \right)^{2/(m-1)} , \; i=1, 2, \ldots.c, \; k=1, 2, \ldots, N. \tag{4}$$

The prototypes of the clusters are obtained in the form of the weighted average of the individual inputs

$$v_i = \sum_{k=1}^{N} (u_{ik})^m \cdot \mathbf{x}_k \Big/ \sum_{k=1}^{N} (u_{ik})^m , \; i=1, 2, \ldots.c. \tag{5}$$

2.2 Context-Based Fuzzy C-Means Clustering

The context-based FCM clustering [8], [9] attempts to reflect upon the output variable while clustering the remaining data. This means that we first agree upon some granulation of the output variable of the model and afterwards produce some information granules being, in fact, induced by the successive fuzzy sets already formed for the output variable [8]. The conditional aspect of the clustering mechanism is introduced into the algorithm by taking into consideration the conditioning variable assuming the values f_1, f_2, …, f_N for the corresponding patterns. More specifically, f_k describes a

level of involvement of $\mathbf{x}_k$ in the constructed clusters. We admit f_k to be distributed additively across the entries of the k-th column of the partition matrix meaning that

$$\sum_{i=1}^{c} u_{ik} = f_k \text{ , } k=1, 2, \ldots, N. \tag{6}$$

We can also request that the maximum of membership values within the corresponding column equals f_k,

$$\max_{i=1,2,\ldots,c} u_{ik} = f_k \text{ , } k=1, 2, \ldots, N. \tag{7}$$

In this study, we confine ourselves to the first method of the distribution of the context values. Bearing this in mind, let us modify the requirements to be met by the original partition matrices and define the new family of matrices

$$\mathbf{U}(f) = \left\{ u_{ik} \in [0,1], \quad \sum_{i=1}^{c} u_{ik} = f_k \ \forall k, \quad 0 < \sum_{k=1}^{N} u_{ik} < N \ \forall i \right\} \tag{8}$$

Note that the standard normalization condition in Eq. (2) is replaced by the involvement (conditioning) constraint. The optimization problem is now reformulated accordingly [8].

$$\begin{aligned} &\min_{U, \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_c} Q \\ &\text{subject to } U \in \mathbf{U}(f) \end{aligned} \tag{9}$$

Again the minimization of the objective function is carried out iteratively where the partition matrix is updated accordingly,

$$u_{ik} = \frac{f_k}{\sum_{j=1}^{c} \left(\frac{\|\mathbf{x}_k - \mathbf{v}_i\|}{\|\mathbf{x}_k - \mathbf{v}_j\|} \right)^{2/(m-1)}} \text{ , } i=1, 2, \ldots.c, \ k=1, 2, \ldots, N. \tag{10}$$

The computations of the prototypes are the same as for the original FCM method. Moreover, the convergence conditions for the method are the same as thoroughly discussed for the original FCM algorithm [7].

3 The Topology of the RBF Neural Network

In this section, we highlight the main structural features of the RBF neural network. The network dwells on the concept of context-based clustering method. The fuzzy partitions formed for all variables gives rise to the topology as visualized in Figure 1. In particular, this structure consists of two clusters for each context.

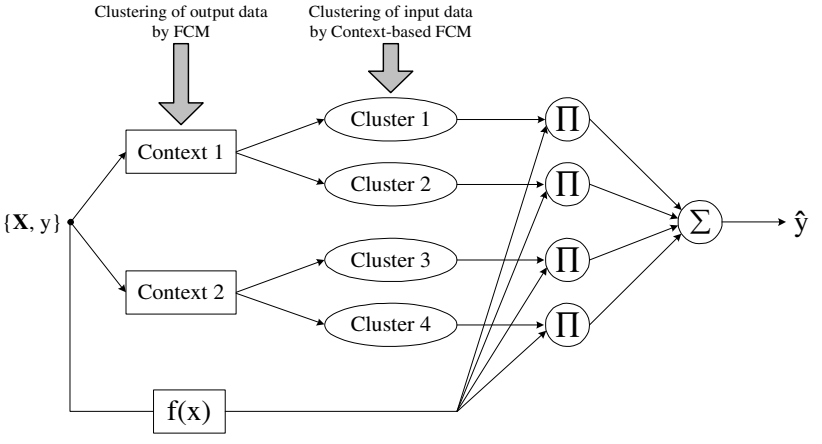


Fig. 1. The overall structure of the RBF neural network

The network has a single hidden layer composed on a basis of the receptive fields built through the use of the proposed algorithm. The local model used here constitutes a linear regression built around the information granules formed in the input and output space which implies the following expression of the model.

$$y_{ij} = u_{ij}\left(\sum_{k=0}^{n} a_{ijk}(x_k)\right) \tag{11}$$

where, y_{ij} is the modal value of the i-th context and j-th cluster.
The overall output of the network is computed by taking a weighted average of all local models

$$\hat{y} = \sum_{i=1}^{p}\sum_{j=1}^{c} y_{ij} \tag{12}$$

Parameter optimization of the local models is completed by solving a standard Mean Square Error (MSE) problem.

$$Q = \sum_{k=1}^{N}\left(y(\mathbf{x}_k) - \hat{y}(\mathbf{x}_k)\right)^2 \tag{13}$$

4 Experimental Studies

4.1 Synthetic One-Dimensional Data

We consider a single-variable nonlinear function of the following form

$$y = 0.6\sin(\pi x) + 0.3\sin(3\pi x) + 0.1\sin(5\pi x) \tag{14}$$

Where the input x is defined in the space X=[-1, 1]. Using (14), 200 pairs of input-output data have been generated. The entire dataset is split randomly into a training dataset (being a 60% of the overall data set) while the rest of the data is used as a testing data. To assess an impact of the varying granularity of information, we carried out experiments for the number of contexts varying in-between 2 to 5.

Table 1 summarizes the performance of each context of the proposed model as optimal combinations of the number of initial clusters(c) used for each context.

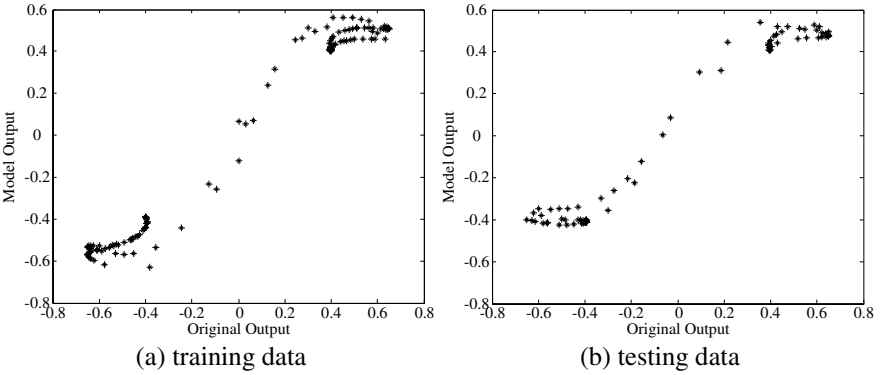
Table 1. Values of the performance index for the synthetic-one dimensional data; PI-performacne index for the training data. EPI-performance index for the testing data.

(a) Values of the PI

Number of clusters (c)	Number of contexts (p)			
	2	3	4	5
2	0.0084 ± 7.7e-4	0.0045 ± 6.2e-4	0.0024 ± 5.3e-4	7.70e-4 ± 5.1e-4
3	0.0051 ± 7.2e-4	0.0026 ± 3.6e-4	0.0011 ± 3.4e-4	4.70e-4 ± 1.8e-4
4	0.0013 ± 2.4e-4	0.0008 ± 2.0e-4	0.0004 ± 9.8e-5	1.30e-4 ± 1.1e-5
5	0.0005 ± 1.1e-4	0.0002 ± 6.5e-5	1.08e-4 ± 2.8e-5	5.44e-5 ± 1.6e-5
6	0.0002 ± 7.4e-5	0.0001 ± 2.6e-5	4.50e-5 ± 1.5e-5	1.71e-5 ± 6.8e-6

(b) Values of the EPI

Number of clusters (c)	Number of contexts (p)			
	2	3	4	5
2	0.0164 ± 0.004	0.0281 ± 0.014	0.1156 ± 0.177	0.0758 ± 0.068
3	0.1222 ± 0.257	0.2100 ± 0.311	0.6547 ± 0.692	0.5889 ± 0.630
4	0.0835 ± 0.089	0.2941 ± 0.260	0.5090 ± 0.298	2.5968 ± 7.037
5	0.3640 ± 0.306	0.3468 ± 0.150	0.4895 ± 0.278	0.4701 ± 0.173
6	0.3077 ± 0.212	0.9894 ± 0.709	0.4783 ± 0.245	0.6036 ± 0.269



(a) training data (b) testing data

Fig. 2. Scatter plots of model output vs. original output (p=2, c=2)

Figure 2 visualizes the approximation and generalization capabilities of the network when the m=2.0, the number of the contexts was set up to 2 and the number of the clusters per context was equal to 2 (here PI=0.0072, EPI=0.0125).

Table 2 presents the performance of the proposed model vis-à-vis other RBF neural networks. RBF NN I is a standard RBF neural network with the receptive fields whose centers are formed using the FCM algorithm while the values of the connections of the output linear neuron are estimated with the use of the standard least square error method.

The spreads of the receptive fields are equal to 1 while the receptive fields themselves are Gaussian function. The second version of the network, denoted here by RBF NN II, is a standard RBF neural network with the receptive fields formed by the FCM algorithm. Furthermore the outputs of receptive fields for the given input **x** are taken as the membership value of **x** in the corresponding clusters. The weights of the output neuron are estimated in the same way as it was completed for the RBF NN I.

Table 2. Comparative analysis of the performance of selected model

Model	Number of nodes in the hidden layer	Performance index	
		PI	EPI
RBFNN I	5	0.0378 ± 0.002	0.0402 ± 0.004
RBFNN II	5	0.0522 ± 0.005	0.0565 ± 0.009
Proposed model	4 (p=2, c=2)	0.0084 ± 7.7e-4	0.0164 ± 0.004
	6 (p=2, c=3)	0.0045 ± 6.2e-4	0.0281 ± 0.014

Table 3. Performance index of the proposed model

(a) Training data

Number of clusters (c)	Number of contexts (p)			
	2	3	4	5
2	2.1433 ± 0.092	1.1359 ± 0.081	0.7551 ± 0.064	0.5653 ± 0.042
3	2.1392 ± 0.116	1.1286 ± 0.059	0.7087 ± 0.030	0.4964 ± 0.048
4	2.0917 ± 0.065	1.0676 ± 0.085	0.6926 ± 0.044	0.4715 ± 0.057
5	2.0494 ± 0.079	1.0997 ± 0.028	0.6678 ± 0.063	0.4532 ± 0.046
6	2.0611 ± 0.097	1.0443 ± 0.062	0.6592 ± 0.032	0.4220 ± 0.050

(b) Testing data

Number of clusters (c)	Number of contexts (p)			
	2	3	4	5
2	2.5311 ± 0.184	1.4745 ± 0.194	1.0036 ± 0.149	0.6721 ± 0.084
3	2.5159 ± 0.213	1.4281 ± 0.177	1.2802 ± 0.303	0.9338 ± 0.227
4	2.7316 ± 0.297	1.9684 ± 0.473	1.4184 ± 0.225	1.0518 ± 0.384
5	2.9612 ± 0.228	1.9545 ± 0.388	1.5174 ± 0.367	1.2214 ± 0.266
6	3.4652 ± 0.907	2.2551 ± 0.638	1.5624 ± 0.244	1.5978 ± 0.279

4.2 Machine Learning Data

We consider the well-known abalone data coming from Machine Learning repository (http://archive.ics.uci.edu/ml/datasets/Abalone).

Table 3 summarizes the optimal performance expressed in terms of the number of contexts and the number of clusters.

The number of contexts impacts the performance of the network as shown in Figure 3. Their increase leads to the reduction of error for the training set while there is a different tendency for the testing set where we note a certain optimal number of the contexts which is due to possible memorization effect.

Table 4 summarizes the performance of the proposed model vis-à-vis other models that is RBF NN I and RBF NN II. We note that we have achieved substantial improvement over the two versions of the RBF NNs.

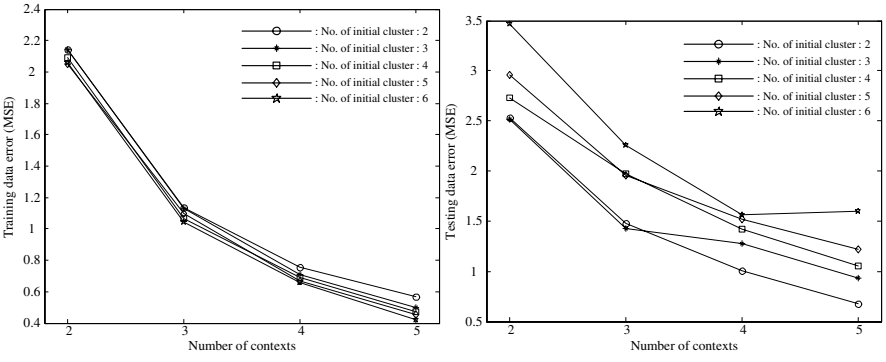


Fig. 3. Performance index vis-à-vis number of contexts

Table 4. Comparative analysis of the performance of selected model

Model	Number of nodes in the hidden layer	Performance index	
		PI	EPI
RBFNN I	10	7.207 ± 0.301	7.763 ± 0.290
RBFNN II	10	6.893 ± 0.186	7.512 ± 0.292
Proposed model	10 (p=5, c=2)	0.565 ± 0.0042	0.672 ± 0.084
	15 (p=5, c=3)	0.496 ± 0.048	0.933 ± 0.227

5 Concluding Remarks

In this study, we have proposed a new architecture of radial basis function neural networks which dwells upon an effective use of experimental data through a prudent process of information granulation. The information granules are developed using a certain context-driven version of the Fuzzy C-Means. This specialized clustering environment emphasizes the role of contexts-fuzzy sets defined in the output space in the formation of the information granules in the input space. The functional character of the network comes with local models that are inherently associated with the modal

values of the fuzzy sets of contexts and the prototypes of the fuzzy clusters formed in the input space.

Along with the experiments embracing the number of commonly encountered architectures of RBF neural networks, we arrived at the quantification of the performance of the introduced network.

Acknowledgements. This work was supported by the GRRC program of Gyeonggi province [GGA0801-45700, Center for U-city Security & Surveillance Technology] and also supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD)(KRF-2008-314-D00376).

References

1. Chang, F.J., Liang, J.M., Chen, Y.C.: Flood Forecasting Using Radial Basis Function Neural Networks. IEEE Trans. Syst. Man Cybern. Part C 31, 530–535 (2001)
2. Tan, Y., Wang, J., Zurada, J.M.: Nonlinear Blind Source Separation Using a Radial Basis Function Network. IEEE Trans. Neural Networks 12, 124–134 (2001)
3. Devaraj, D., Yegnanarayana, B., Ramar, K.: Radial Basis Function Networks for Fast Contingency Ranking. Electrical Power and Energy Systems 24, 387–395 (2002)
4. Fu, X., Wang, L.: Data Dimensionality Reduction with Application to Simplifying RBF Network Structure and Improving Classification Performance. IEEE Trans. Syst. Man Cybern. Part B 33, 399–409 (2003)
5. Han, M., Xi, J.: Efficient Clustering of Radial Basis Perceptron Neural Network for Pattern Recognition. Pattern Recognition 37, 2059–2067 (2004)
6. Vilaplana, J.M., Molina, J.L.P., Coronado, J.L.: Hyper RBF Model for Accurate Reaching in Redundant Robotic Systems. Neurocomputing 61, 495–501 (2004)
7. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithm. Plenum, New York (1981)
8. Pedrycz, W.: Conditional Fuzzy Clustering in The Design of Radial Basis Function Neural Networks. IEEE Trans. Neural Network 19, 601–612 (1998)
9. Pedrycz, W.: Conditional Fuzzy C-Means. Pattern Recognition Letter 17, 625–631 (1996)
10. Park, H.S., Pedrycz, W., Oh, S.K.: Evolutionary Design of Hybrid Self-Organizing Fuzzy Polynomial Neural Networks with The Aid of Information Granulation. Expert Systems with Applications 33, 830–846 (2007)

A Stochastic Complex Dynamical Network and Its Synchronization

Tong-jun He and Zhengping Shi

College of Mathematics and Computer Science,
Fuzhou University, Fuzhou 350108, China
hetongjun@fzu.edu.cn,shizp@fzu.edu.cn

Abstract. In this brief, the authors firstly introduce a complex stochastic dynamical network model. Secondly, the existence and uniqueness of solution of this stochastic complex dynamical network is identified, furthermore, the authors investigate this stochastic complex dynamical network's synchronization in probability and give out two synchronization theorems of this network. Two detail examples are given to verify the theoretical analysis, and numerical simulations verify that the two synchronization theorems are effective.

Keywords: Index Terms–Stochastic dynamical network, Synchronization, Time varying, Time invariant.

1 Introduction

In recent years, the stability of stochastic systems[1-4] has been a focal subject for research due to the uncertainties that exist in the real system, at the same time, an area of particular interest has been the stability and synchronization of stochastic complex dynamical network. To the best of our knowledge, however, there are few works about the stability and synchronization of stochastic complex dynamical network. Actually, chaotic synchronization control and dynamics of stochastic complex dynamical networks [5]-[14] has attracted interesting attention.

Recently, Wang and Chen [12] introduced a simple uniform scale free dynamical network model, and Lü and Chen [13][14] further introduced a more general time-varying dynamical network model, and investigate its synchronization properties. Although their model reflects the complexity from the network structure, it is only a determined dynamical network with the different coupling strength for all connections. In reality, complex networks are more likely affected by external perturbations which in many cases are of great uncertainties and hence may be treated as random noise from the circumstance of network nodes, and other probabilistic causes. Moreover, in real-world complex networks, particularly communication network, random noise influence can't be neglected. Therefore, in this brief, we attempt to introduce a more general complex stochastic dynamical network model which contain the complex network models introduced by Wang and Chen [12] and Lü and Chen [13][14], and further investigate this

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 164–174, 2009.

stochastic complex dynamical network's synchronization in probability. Based on mild conditions, we prove that this stochastic complex dynamical network has an unique solution, and give out the probabilistic synchronization theorem of a time-varying or time-invariant stochastic complex dynamical network. Two detail examples and numerical simulation demonstrate the results obtained in this brief.

2 Stochastic Complex Dynamical Network Model

In this section, we introduce a general stochastic complex dynamical network model and several mathematical preliminaries.

2.1 Mathematical Preliminaries

Consider a nonlinear stochastic dynamical system

$$\frac{dX(t)}{dt} = F(X(t),t) + G(X(t),t)dw(t),\ \ t \in [t_0, T] \tag{1}$$

where suppose that $w(t) = (w_1(t), w_2(t), ..., w_d(t))$ is a d-dimension Brownian motion defined on a complete probability space $(\Omega, \mathcal{F}, P)$ with a natural filtration $\{\mathcal{F}_t\}_{t\geq 0}$ generated by $\{w(s) : 0 \leq s \leq t\}$, where Ω associates with the canonical space generated by $w(t)$, and denote $\mathcal{F}$ the associated σ-algebra generated by $w(t)$ with the probability measure P. Here, the white noise $dw_i(t)$ is independent of $dw_j(t)$ if $i \neq j$, and $G(X(t),t) : R_+ \times R^d \to R^{d\times d}$ is called the noised intensity function matrix.

Lemma 1. [16] *Let $F(X(t),t)$ and $G(X(t),t)$ be continuous and X_0 be a bounded R^d-valued $\mathcal{F}_{t_0}$-measurable random variable. Assume that there exists a continuous increasing concave function $\kappa : R_+ \to R_+$ such that*

$$\int_0^\infty \frac{du}{\kappa(u)} = \infty, \tag{2}$$

and for all $X, Y \in R^d$, $t_0 \leq t \leq T$

$$\|F(X(t),t) - F(Y(t),t)\|^2 \vee \|G(X(t),t) - G(Y(t),t)\|^2 \leq \kappa(\|X - Y\|^2). \tag{3}$$

Then the equation (1) has a unique solution $X(t)$. Moreover, the Caratheodory approximate solutions $X_n(t)$ converge to $X(t)$ in the following sense

$$\lim_{n\to\infty} E(\sup_{t_0\leq t\leq T} \|X_n(t) - X(t)\|^2) = 0. \tag{4}$$

Lemma 2. [16] *Assume that there exists a function $V \in C^{2,1}(R^d \times [t_0, \infty) : R_+$ and constants $p > 0$, $c_1 > 0$, $c_2 \in R$, $c_3 \geq 0$, such that for all $X \neq 0$ and $t \geq t_0$,*

$$c_1\|X\|^p \leq V(X,t), \tag{5}$$

$$LV(X,t) \le c_2 V(X,t), \tag{6}$$

$$c_3 V^2(X,t) \le \|V_X(X,t)g(X,t)\|^2. \tag{7}$$

Then

$$\lim_{t\to\infty} \sup \frac{1}{t} \log \|X(t;t_0,X_0)\| \le -\frac{c_3 - 2c_2}{2p} \quad a.s. \tag{8}$$

for all $X(t_0) \in R^d$. *In particular, if* $c_3 > 2c_2$, *then the trivial solution of (9) is almost surely exponentially stable.*

2.2 Stochastic Complex Dynamical Network Model

Consider a stochastic dynamical network consisting of N linearly and diffusively coupled identical nodes, with each node being an n-dimensional dynamical system. The proposed general stochastic dynamical network is described by

$$\frac{dX_i}{dt} = f(X_i,t) + \sum_{j=1}^{N} c_{ij}(t)\mathbf{A}(t)(X_j - X_i) + g(X_i,t)dw, i = 1,\cdots,N, \tag{9}$$

where $X_i = (x_{i1}, x_{i2}, ..., x_{id})^T \in R^d$ is the state variable of node i, $w(t) = (w_1(t), w_2(t), ..., w_d(t))$ is a d-dimension Brownian motion, $\mathbf{A}(t) = (a_{kl}(t))_{d\times d} \in R^{d\times d}$ is the inner-coupling matrix of the network at time t, $C(t) = (c_{ij}(t))_{N\times N}$ is the coupling configuration matrix representing the coupling strength and the topological structure of the network at time t, in which $c_{ij}(t)$ is defined as follows: If there is a connection from node i to node $j(j \ne i)$ at time t, , then $c_{ij}(t) \ne 0$; otherwise, $c_{ij}(t) = 0(j \ne i)$ and

$$c_{ij}(t) = c_{ji}(t), \quad i,j = 1,2,...,N, \tag{10}$$

condition (10) show that the coupling configuration matrix is symmetrical. Moreover, assume that $\mathbf{A}(t)$, $\mathbf{C}(t)$ are normal bounded. That is, there exist two positive real number K_1, K_2 such that

$$\|\mathbf{A}(t)\| \le K_1, \quad \|\mathbf{C}(t)\| \le K_2. \tag{11}$$

Suppose that stochastic dynamical network (9) is connected in the sense that there are no isolate clusters, that is, $C(t)$ is irreducible.

Remark 1. Compare to Lü and Chen's [13, 14] network model, our network model don't need the diagonal elements of matrix $C(t)$ satisfy

$$c_{ii}(t) = -\sum_{j=1,j\ne i}^{N} c_{ij}(t), \quad i = 1,2,...,N. \tag{12}$$

Moreover, we consider that some stochastic noises affect the deterministic complex dynamical networks (9).

When $g(X_i,t)=0$, stochastic network (9) becomes a determined time-varying dynamical network

$$\frac{dX_i}{dt}=f(X_i,t)+\sum_{j=1}^{N}c_{ij}(t)\mathbf{A}(t)(X_j-X_i),\ i=1,2,...,N \tag{13}$$

Obviously, the network (13) is a deterministic time-varying complex dynamical network of Lü and Chen[13, 14]. Moreover, when $g(X_i,t)=0$, $\mathbf{A}(t)$, $\mathbf{C}(t)$ are constant matrices, stochastic complex dynamical network (9) becomes a deterministic time-invariant dynamical network

$$\frac{dX_i}{dt}=f(X_i,t)+\sum_{j=1}^{N}c_{ij}\mathbf{A}(X_j-X_i),\ i=1,2,...,N, \tag{14}$$

where $\mathbf{C}$ is a $0-1$ matrix and $\mathbf{A}$ is a $0-1$ diagonal matrix, is a special case of stochastic network (13) and the simple uniform dynamical network of Wang and Chen[13]. Therefore, our proposed stochastic complex dynamical network (9) include determined complex dynamical network being introduced by Lü and Chen[13, 14] and Wang and Chen[12].

2.3 Existence and Uniqueness of Solution of Stochastic Complex Dynamical Network Model

Theorem 1. *Let $f(X(t),t)$ and $g(X(t),t)$ be continuous and X_0 be a bounded R^d-valued $\mathcal{F}_{t_0}$-measurable random variable. Assume that there exists N continuous increasing concave function $\kappa: R_+\to R_+$ such that and for all $X,Y\in R^d$, $t_0\le t\le T$*

$$\begin{aligned}&\|f(X(t),t)-f(Y(t),t)\|^2\\ \vee&\|g(X(t),t)-g(Y(t),t)\|^2\le\kappa(\|X-Y\|^2).\end{aligned} \tag{15}$$

Then the stochastic complex dynamical network (9) has a unique solution $X(t)$. Moreover, the Caratheodory approximate solutions $X_n(t)$ converge to $X(t)$ in the following sense

$$\lim_{n\to\infty}E(\sup_{t_0\le t\le T}\|X_n(t)-X(t)\|^2)=0. \tag{16}$$

Proof. Let

$$F(X_i,t)=f(X_i,t)+\sum_{j=1}^{N}c_{ij}(t)\mathbf{A}(t)(X_j-X_i),$$

$G(X_i,t)=g(X_i,t)$. It follows from the assumption (11) that

$$\|c_{ij}(t)\mathbf{A}(t)\|\le K_1K_2\ \ (i,j=1,2,...,N), \tag{17}$$

moreover, using the symmetric (10) of the coupling configuration matrix, condition (15) as well as (17) one can derive that

$$\begin{aligned}
&\|F(X_i(t),t)-F(X_i'(t),t)\|^2\\
&\quad=\|f(X_i(t),t)-f(X_i'(t),t)+(\sum_{i=1}^{N}c_{ij}(t)\mathbf{A}(t))(X_i'-X_i)\|^2\\
&\quad\le 2\|f(X_i(t),t)-f(X_i'(t),t)\|^2+2\|(\sum_{i=1}^{N}c_{ij}(t)\mathbf{A}(t))\|^2\|X_i'-X_i\|^2\\
&\quad\le 2\kappa(\|X_i'-X_i\|^2)+2K_1^2K_2^2\|X_i'-X_i\|^2. \qquad (18)
\end{aligned}$$

Set $\kappa'(u)=2[\kappa(u)+K_1^2K_2^2u]$, since $\kappa(u)$ is a continuous increasing concave functions, so $\kappa'(u)$ is also a continuous increasing concave function. And it follows from (18) that

$$\|F(X_i(t),t)-F(X_i'(t),t)\|^2\le\kappa'(\|X_i'-X_i\|^2), \qquad (19)$$

combining with (15) we can derive that

$$\begin{aligned}
&\|F(X_i(t),t)-F(X_i'(t),t)\|^2\\
\vee&\|G(X_i(t),t)-G(X_i'(t),t)\|^2\le\kappa'(\|X_i-X_i'\|^2). \qquad (20)
\end{aligned}$$

Besides, since $\kappa(u)$ is concave and continuous increasing, there must exist a positive number a such that

$$\kappa(u)\le a(1+u)\quad on\ u\ge 0. \qquad (21)$$

Then using (20) we can derive that

$$\int_0^\infty\frac{du}{\kappa'(u)}\ge\int_0^\infty\frac{du}{2[a(1+u)+K_1^2K_2^2u]}\ge\infty. \qquad (22)$$

Therefore, these conditions (2), (3) of Lemma 2.1 are fulfilled. Applying Lemma 2.1 to stochastic complex dynamical network (9), Theorem 2.4 is complete. □

3 Synchronization Realization of Stochastic Complex Dynamical Network

In this section, we will establish synchronization realization for stochastic complex dynamical network (9) in probability.

Hypothesis 1 (H1). Assume that the coupling configuration matrix $\mathbf{C}(t)=(c_{ij}(t))_{N\times N}$ satisfy the following conditions: If there is a connection from node i to node $j(j\neq i)$ at time t, , then $c_{ij}(t)\neq 0$ and $c_{ij}(t)=a(t)$; otherwise, $c_{ij}(t)=0$ $(j\neq i)$.

Hypothesis 2 (H2). Assume that $f(X,t)$, $g(X,t)$ are linear with respect to $X \in R^d$, that is, for any real numbers α, β and $X, Y \in R^d$

$$f(\alpha X + \beta Y, t) = \alpha f(X,t) + \beta f(Y,t) \tag{23}$$

$$g(\alpha X + \beta Y, t) = \alpha g(X,t) + \beta g(Y,t). \tag{24}$$

Now, we consider synchronization realization for stochastic complex dynamical network (9). For two different fixed nodes i and j in network, we consider their error state

$$e_{ij}(t) = X_i - X_j, \tag{25}$$

it follows from (9), (25), and (23) that

$$\frac{de_{ij}(t)}{dt} = f(e_{ij}(t),t) - Na(t)\mathbf{A}(t)e_{ij}(t) + g(e_{ij}(t),t)dw,\ i,j = 1,...,N, \tag{26}$$

where (26) is a nonlinear stochastic dynamical system for the error state $e_{ij}(t)$. Using Theorem 2.4 we easily see that nonlinear stochastic dynamical system (26) has an unique solution.

Definition 1. *Nonlinear stochastic dynamical system (26) is said to be globally asymptotically stable in probability if for any given condition such that*

$$P\{lim_{t\to\infty}\|e_{ij}(t)\| = 0\} = 1, i, j = 1, 2, ..., N. \tag{27}$$

Definition 2. *If nonlinear stochastic dynamical system (26) is globally asymptotically stable in probability, then stochastic complex dynamical networks (9) is said to realize synchronization in probability.*

3.1 Synchronization of Time-Varying Stochastic Dynamical Networks

We set $g(e_{ij}(t),t) = (D_1, D_2, ..., D_m)^T e_{ij}(t)$ and $w(t) = (w_1(t), w_2(t), ..., w_m(t))$, then $g(e_{ij}(t),t)$ is a linear functional with respect to $e_{ij}(t)$,

$$g(e_{ij}(t),t)dw = \sum_{i=1}^{m} D_i e_{ij}(t)dw_i.$$

If (H1) and (H2) hold, then the nonlinear stochastic dynamical system (26) can be written as

$$\frac{de_{ij}(t)}{dt} = f(e_{ij}(t),t) - Na(t)\mathbf{A}(t)e_{ij}(t) + \sum_{k=1}^{m} D_k e_{ij}(t)dw_k,\ i,j = 1,...,N. \tag{28}$$

Theorem 2. *Assume that there are two constants $\lambda > 0$ and $\iota \geq 0$ such that*

$$\sum_{k=1}^{m} \|D_i e_{ij}(t)\|^2 \leq \lambda \|e_{ij}(t)\|^2 \quad \textit{and}$$

$$\sum_{k=1}^{m} \|e_{ij}(t)^T D_i e_{ij}(t)\|^2 \geq \iota \|e_{ij}(t)\|^4 \tag{29}$$

for all $e_{ij}(t) \in R^d$. Then

$$\limsup_{t\to\infty} \frac{1}{t}\log\|e_{ij}(t;t_0,e_{ij}(t_0))\| \leq -[\iota - (\lim_{t\to\infty}\|f\| + NK_1K_2 + \frac{\lambda}{2})] \quad a.s. \tag{30}$$

for all $e_{ij(t_0)} \in R^d$. *In particular, if* $\iota > \lim_{t\to\infty}\|f\| + NK_1K_2 + \frac{\lambda}{2}$, *then the trivial solution of (28) is almost surely exponentially stable, and the time-invariant stochastic dynamical complex networks (9) is synchronization in probability.*

Proof. Since $f(e_{ij}(t),t)$ is linear with respect to $e_{ij}(t)$ and continuous, so $f(e_{ij}(t), t)$ is a continuous linear functional with respect to $e_{ij}(t)$, therefore,

$$\|f(e_{ij}(t),t)\| \leq \|f\|\|e_{ij}(t)\| \text{ for all } e_{ij}(t) \in R^d, \tag{31}$$

where $\|f\|$ is the norm of the linear function $f(e_{ij}(t),t)$ and also is a function with respect to t. Using (31) and $\|Na(t)\mathbf{A}(t)e_{ij}(t)\| \leq NK_1K_2$ we can derive that

$$\|f(e_{ij}(t),t) + Na(t)\mathbf{A}(t)e_{ij}(t)\| \leq (\|f\| + NK_1K_2)\|(e_{ij}(t)\|. \tag{32}$$

Let Lyapunov function $V(e_{ij}(t),t) = \|e_{ij}(t)\|^2$. Then

$$LV(e_{ij}(t),t) = 2e_{ij}^T f(e_{ij}(t),t) + \sum_{k=1}^{m}\|D_i e_{ij}(t)\|^2 \leq (2\|f\| + \lambda)\|e_{ij}\|^2. \tag{33}$$

Moreover, with $g(e_{ij}(t),t) = (D_1, D_2, ..., D_m)^T e_{ij}(t)$,

$$\|V_{e_{ij}}(e_{ij}(t),t)g(e_{ij}(t),t)\|^2 = \sum_{k=1}^{m}\|e_{ij}(t)^T D_i e_{ij}(t)\|^2 \geq \iota\|e_{ij}(t)\|^4. \tag{34}$$

An application of Lemma 2.2 yields the desired assertion (30), we further see that if $\iota > \lim_{t\to\infty}\|f\| + NK_1K_2 + \frac{\lambda}{2}$,

$$\lim_{t\to\infty}\|e_{ij}(t;t_0,e_{ij}(t_0))\| = 0 \quad \text{a.s.}, \tag{35}$$

then using Definition 3.1 and (35) we see easily that nonlinear stochastic dynamical system (28) is globally asymptotically stable in probability, furthermore, by Definition 3.2 we see that the time-invariant stochastic dynamical complex networks (9) is synchronization realization in probability. □

Example 1. Let $D_i = \sigma_i I$ $(1 \leq i \leq m)$, where I is the $d \times d$ identity matrix and σ_i is a constant. Furthermore, assume that

$$\|f(e_{ij}(t),t)\| \leq K\|e_{ij}(t)\| \quad \text{for all } (e_{ij},t) \in R^d \times R_+. \tag{36}$$

In this case, the nonlinear stochastic dynamical system (28) becomes

$$\frac{de_{ij}(t)}{dt} = f(e_{ij}(t),t) - Na(t)\mathbf{A}(t)e_{ij}(t) + \sum_{k=1}^{m}\sigma_k e_{ij}(t)dw_k, \ i,j = 1,\cdots,N. \tag{37}$$

Moreover,

$$\sum_{k=1}^{m} \|D_k e_{ij}(t)\|^2 = \sum_{k=1}^{m} \sigma_k^2 \|e_{ij}(t)\|^2, \ i,j = 1, \cdots, N \tag{38}$$

and

$$\sum_{k=1}^{m} \|e_{ij}(t)^T D_k e_{ij}(t)\|^2 = \sum_{k=1}^{m} \sigma_k^2 \|e_{ij}(t)\|^4, \ i,j = 1, \cdots, N. \tag{39}$$

By Theorem 3.3, the trivial solution of the nonlinear stochastic dynamical system (35) has the property

$$\lim_{t\to\infty} \sup \frac{1}{t} \log \|e_{ij}(t))\| \leq -[\frac{1}{2} \sum_{k=1}^{m} \sigma_k^2 - (K + NK_1K_2)]$$
$$\text{a.s. } i,j = 1, \cdots, N. \tag{40}$$

Therefore, the trivial solution of the nonlinear stochastic dynamical system (37) is almost surely exponentially stable provided $\frac{1}{2}\sum_{k=1}^{m} \sigma_k^2 > K + NK_1K_2$, and the time-invariant stochastic dynamical complex networks (9) realize synchronization in probability. □

Remark 2. The foregoing example shown that if the deterministic complex dynamical network

$$\frac{dX_i}{dt} = f(X_i, t) + \sum_{j=1}^{N} c_{ij}(t)\mathbf{A}(t)(X_j - X_i), \quad i = 1, 2, ..., N \tag{41}$$

is unstable, then we add a strong enough stochastic noise $g(X_i, t)dw$ to the deterministic complex dynamical networks (37), such that networks (37) realize synchronization.

Remark 3. From $\frac{1}{2}\sum_{k=1}^{m} \sigma_k^2 > K + NK_1K_2$, we see easily that if the node number of the deterministic complex dynamical network (37) N is more larger, then the stochastic noise $g(X_i, t)dw$ must be strong enough, the deterministic complex dynamical networks (37) can realize synchronization.

3.2 Synchronization of Time-Invariant Stochastic Dynamical Networks

In this narrow sense, if we consider the synchronization of time-invariant stochastic dynamical networks, then the more detail and accurate results will be obtained.

Hypothesis 3 (H3). Assume that the inner-coupling matrix and coupling configuration matrix of the network (9) are time-invariant, that is, $\mathbf{A}(t) = \mathbf{A}$ and $a(t) = a$, and further suppose $f(e_{ij}(t), t) = f(t)$, $g(e_{ij}(t), t) = g(t)$.

If (H3) holds, then the nonlinear stochastic dynamical system (26) can be written as

$$de_{ij}(t) = [-Na\mathbf{A}e_{ij}(t) + f(t)]dt + g(t)dw(t), \ i,j = 1, \cdots, N, \tag{42}$$

where

$$\mathbf{A} \in R^{d\times d}, \quad f : R_+ \to R^d, \quad g : R_+ \to R^{d\times m}.$$

Theorem 3. *Assume that the nonlinear stochastic dynamical system (40) with initial value $X(t_0) = X_0 \in \mathbf{R}^d$, satisfies that there is also a pair of positive constants β_1 and λ_1 such that*

$$\|f(t)\|^2 \vee \|g(t)\|^2 \le \beta_1 e^{-\lambda_1 t} \quad \text{for } t \ge 0. \tag{43}$$

(i)If $a > 0$ and the eigenvalues of $\mathbf{A}$ have negative real parts, then the time-invariant stochastic dynamical networks (9) is synchronization in probability.
(ii)If $a < 0$ and the eigenvalues of $\mathbf{A}$ have positive real parts, then the time-invariant stochastic dynamical networks (9) is synchronization in probability.

Proof. Since the eigenvalues of $\mathbf{A}$ have real parts, this is equivalent to that there is a pair of constants $\beta_2(\beta_2 > 0)$ and λ_2 such that

$$\|e^{aN\mathbf{A}t}\|^2 \le \beta_2 e^{-aN\lambda_2 t} \quad \text{for } t \ge 0. \tag{44}$$

Using the result of Mao's book (refer to [16] 141-143), we can derive that

$$\limsup_{t\to\infty} \frac{1}{t} \log \|X(t)\| \le -\frac{\lambda_1 \wedge aN\lambda_2}{2} \quad \text{a.s.} \tag{45}$$

(i) When $a > 0$ and the eigenvalues of $\mathbf{A}$ have negative real parts $-\lambda_2 < 0$, the solution of nonlinear stochastic dynamical system (40) tend to zero exponentially in mean square and almost surely as well. Then the time-invariant stochastic dynamical networks (9) is synchronization in probability.
(i) When $a < 0$ and the eigenvalues of $\mathbf{A}$ have positive real parts $-\lambda_2 > 0$, the solution of nonlinear stochastic dynamical system (40) tend to zero exponentially in mean square and almost surely as well. Then the time-invariant stochastic dynamical networks (9) is synchronization in probability. □

Example 2. Consider the following time-invariant stochastic complex dynamical networks with N nodes

$$\begin{cases} \frac{dX_{i+r}}{dt} = \sum_{k=1}^N aA(X_k - X_{i+r}) + (\frac{i+r}{t+1}, \frac{i-r}{t+1})^T e^{-t} + \frac{i+r}{t^2+1} e^{-t} dw(t) \\ \frac{dX_i}{dt} = \sum_{k=1}^N aA(X_k - X_i) + (\frac{i+\frac{r}{2}}{t+1}, \frac{i-\frac{r}{2}}{t+1})^T e^{-t} + \frac{i}{t^2+1} e^{-t} dw(t), \end{cases}$$

where $i = 1, \cdots, N,\ r = 1, ..., N-i,\ X_k \in R^2,\ k = 1, ..., N$, then the response error system as following

$$\begin{aligned} de_{ii+r}(t) &= [-aN\mathbf{A}e_{ii+r}(t) + (\frac{r}{2t+2}, \frac{-r}{2t+2})^T e^{-t}]dt + \frac{r}{t^2+1} e^{-t} dw(t), \\ & i = 1, \cdots, N,\ r = 1, ..., N-i. \end{aligned} \tag{46}$$

$$\begin{aligned} &\|(\frac{r}{2t+2}, \frac{-r}{2t+2})^T e^{-t}\|^2 \vee \|\frac{r}{t^2+1} e^{-t}\|^2 \\ &= [\frac{r^2}{2(t+1)^2} \vee \frac{r^2}{(t^2+1)^2}] e^{-2t} < r^2 e^{-2t}. \end{aligned} \tag{47}$$

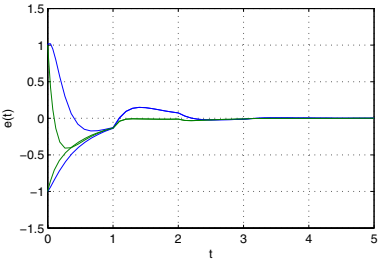


Fig. 1. Transient behavior of $e_{ii+r}(t)$, $e_{ii+r}(0) = 1$, $r = 8$, $a = 1$

(i) Let

$$\mathbf{A} = \begin{pmatrix} 3\sqrt{2} - \frac{5\sqrt{5}}{3} & -\sqrt{2} + \frac{2\sqrt{5}}{3} \\ \frac{15}{\sqrt{2}} - 5\sqrt{5} & -\frac{5}{\sqrt{2}} + 2\sqrt{5} \end{pmatrix}$$

then the matrix $\mathbf{A}$ has two eigenvalues $\frac{\sqrt{5}}{3}$, $\frac{\sqrt{2}}{2}$, the matrix $\mathbf{A}$ has positive real part $-\lambda_2 = \frac{\sqrt{5}}{3}$. When $a < 0$, substituting $\lambda_1 = 2$ and $\lambda_2 = -\frac{\sqrt{5}}{3}$ into (45) we can obtain that

$$\lim_{t\to\infty} \sup \frac{1}{t} \log \|X(t)\| \leq -1 \wedge \frac{\sqrt{5}aN}{6} < 0 \quad \text{a.s.}, \tag{48}$$

using theorem 3.6 we see that this time-invariant stochastic dynamical networks is synchronization in probability.

(ii) Let

$$\mathbf{A} = \begin{pmatrix} -3\sqrt{2} + \frac{5\sqrt{5}}{3} & \sqrt{2} - \frac{2\sqrt{5}}{3} \\ -\frac{15}{\sqrt{2}} + 5\sqrt{5} & \frac{5}{\sqrt{2}} - 2\sqrt{5} \end{pmatrix}$$

then the matrix $\mathbf{A}$ has two eigenvalues , the matrix $\mathbf{A}$ has negative real part $-\lambda_2 = -\frac{\sqrt{5}}{3}$. When $a > 0$, substituting $\lambda_1 = 2$ and $\lambda_2 = \frac{\sqrt{5}}{3}$ into (45) we can obtain that

$$\lim_{t\to\infty} \sup \frac{1}{t} \log \|X(t)\| \leq -(1 \wedge \frac{\sqrt{5}aN}{6}) < 0 \quad \text{a.s.}, \tag{49}$$

using theorem 3.6 we see that this time-invariant stochastic dynamical networks is synchronization in probability. □

4 Conclusions

We have introduced a general stochastic complex dynamical network model and presented this network solution's existence and uniqueness theorem, furthermore, we give out two synchronization theorems of this network. Two detail examples demonstrate that synchronization theorems of this time-varying or time-invariant network are all effective, and numerical simulations verify these results.

Acknowledgements. The authors are partially supported NSFC grant: 60874031 and Science and Technology Development Fund of Fuzhou University grant: 826584.

References

1. Liao, X., Mao, X.: Exponential stability of stochastic delay interval systems with markovian switching. IEEE Transactions on Automatic Control 47, 1604–1612 (2002)
2. Shen, Y., Wang, J.: Noise-induced stabilization of the recurrent neural networks with mixed time-varying delays and markovian-switching parameters. IEEE Transactions on Neural Networks 18, 1857–1862 (2007)
3. Yue, D., Han, Q.: Delay-dependent exponential stability of stochastic systems with time-varying delay, nonlinearity, and markovian switching. IEEE Trans. Automatic Control 50, 217–222 (2005)
4. Chen, W., Guan, Z., Liu, X.: Delay-dependent exponential stability of uncertain stochastic systems with multiple delays: an LMI approach. System & Control Lett. 54, 547–555 (2005)
5. Gade, P.M.: Synchronization in coupled map lattices with random nonlocal connectivity. Phys. Rev. E 54, 64–70 (1996)
6. Volkovskii, A.R., Rulkov, N.F.: Experimental study of bifurcations on the threshold for stochastic locking. Sov. Tech. Phys. Lett. 15, 249 (1989)
7. Kowalski, J.M., Albert, G.L., Gross, G.W.: Symptotic synchronous chaotic orbits insystems of excitable elements. Phys. Rev. A 42, 6260–6263 (1990)
8. Nakagawa, N., Kuramoto, Y.: Collective chaos in a population of globally coupled oscillators. Prog. Theoret. Phys. 89, 313–323 (1993)
9. Wu, C.W., Chua, L.O.: Unified framework for synchronization and control of dynamical systems. Int. J. Bifur. Chaos 4, 979–998 (1994)
10. Chua, L.O.: A paradigm for complexity. World Scientific, Singapore (1998)
11. Erdös, P., Réyi, A.: On the evolution of random graphs, Publ. Math. Inst. Hung. Acad. Sci. 5, 17–60 (1959)
12. Wang, X., Chen, G.: Complex network: small-world, scale-free, and beyond. IEEE Circuits Syst. Mag. 3, 6–20 (2003)
13. Lü, J., Yu, X., Chen, G.: Chaos synchronization of general complex dynamical networks. Physica A 334, 281–302 (2004)
14. Lü, J., Chen, G.: A time-varying complex dynamical network model and its controlled synchronization criteria. IEEE Transactions on Automatic Control 50, 841–846 (2005)
15. Mao, X.: Stochastic differential equations and their applications, 2nd edn. Horwood (2007)

Porosity Prediction Using Bagging of Complementary Neural Networks

Pawalai Kraipeerapun[1], Chun Che Fung[2], and Sathit Nakkrasae[1]

[1] Department of Computer Science, Faculty of Science, Ramkhamhaeng University, Thailand
pawalai@ru.ac.th, sathit@ru.ac.th
[2] School of Information Technology, Murdoch University, Australia
l.fung@murdoch.edu.au

Abstract. This paper presents a novel approach to the regression problem using bagging of complementary neural networks (CMTNN). A bagging technique is applied to an ensemble of pairs of feed-forward backpropagation neural networks created to predict degrees of truth and falsity values. In our approach, uncertainties in the prediction of the truth and falsity values are quantified based on the difference among all the predicted truth values and the difference among all the predicted falsity values in the ensemble, respectively. An aggregation technique based on uncertainty values is proposed. This study is realized to the problem of porosity prediction in well log data analysis. The results obtained from our approach are compared to results obtained from three existing bagging models. These three models are an ensemble of feed-forward backpropagation neural networks, an ensemble of general regression neural networks, and an ensemble of support vector machines. We found that our approach improves performance compared to those three existing models that apply a simple averaging technique based on only the truth porosity values in the ensemble.

1 Introduction

An ensemble of accurate and diverse neural networks was found to provide better results and less error than a single neural network [1]. However, it was found that a diverse ensemble of less accurate classifiers outperforms an ensemble of more accurate classifiers but with less diversity [2]. Diversity can be described as "disagreement" of the classifiers [3]. Disagreement of the neural networks can be created based on varying parameter values such as varying different initial weights, different number of hidden nodes, or different number of input features. Furthermore, varying training data in different manners is another technique to create diverse neural networks. Two well known techniques based on varying training data are bagging and boosting. Bagging is based on bootstrap resampling which provides diversity by randomly resampling the original training data into several training sets [4]. Boosting provides diversity based on iteratively learning classifiers with each respect to a training set created based on the performance of the previous classifier [5].

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 175–184, 2009.

In our previous papers [6,7], an ensemble of pairs of complementary neural networks trained for binary classification was found to provide better performance than a single pair of complementary neural networks and a single neural network. Bagging was used to provide diversity by manipulation of the input data whereas a pair of complementary neural networks was used to provide diversity based on the output data. Each pair of neural networks consists of the truth neural network trained to predict degree of the truth values and the falsity neural network trained to predict degree of the falsity values. The falsity value is supposed to be complement to the truth value for each input pattern. As uncertainty of type vagueness can occur since the boundary between the predicted truth and predicted falsity values is not sharp, the uncertainty values can be computed from the difference between the truth and falsity values. The predicted results together with their uncertainty values can be used to provide a better binary classification compared to the traditional technique of the binary classification based on only a single neural network providing only the truth values. Moreover, it is known that an ensemble of pairs of complementary neural networks was found to provide better results than a single pair of neural networks.

In this paper, an ensemble of pairs of complementary neural networks is trained to solve the regression problem. A novel aggregation technique is proposed based on uncertainty of type vagueness occurred in the ensemble. Vagueness deals with the concept of boundaries which cannot be defined precisely [8]. Instead of considering vagueness based on the boundary between each pair of the truth and falsity values, vagueness in the prediction of the truth and falsity values are considered separately. The precise truth value cannot be obtained from each ensemble component. Also, the precise falsity value cannot be defined from all components in the ensemble. Our proposed technique is realized to the problem of porosity prediction from well log data. Porosity is one of the important rock properties in reservoir engineering. Well logs are measured along the depth of a well using electrical, physical, and radioactive devices. There are several techniques used to estimate the porosity such as backpropagation neural networks [9], probabilistic neural networks [10], ensemble of neural networks [11], and support vector machines (SVM) [12]. However, only few works deal with uncertainty in the porosity prediction.

The rest of this paper is organized as follows. Section 2 explains our proposed techniques used for the prediction and quantification of uncertainty. Section 3 describes the data set and results of our experiments. Conclusions and future work are presented in Section 4.

2 Ensemble of Complementary Neural Networks (CMTNN) Based on Bagging Technique

In general, an ensemble of neural networks can be created in two steps: training several diverse neural networks and aggregating the outputs obtained from the ensemble of networks. In this paper, diverse neural networks are created based on the manipulation of both input and output data. Bagging is applied to manage

diversity based on input data whereas diversity based on output data deals with a pair of complement target outputs used to train a pair of neural networks. The bagging algorithm uses bootstrap resampling to generate multiple training sets in which each generated training set is created by random selection of input patterns from the original training set with replacement. Each generated training set contains the same number of training patterns as the original data set. Therefore, m generated training sets are created and applied to m components in the ensemble.

In order to handle diversity based on output data, each component in an ensemble consists of a pair of neural networks named the truth neural network and the falsity neural network. The truth neural network is trained to predict degree of the truth values whereas the falsity neural network is trained to predict degree of falsity values. In this paper, all pairs of the truth and falsity neural networks have the same architecture, that is, the same number of neurons in each layer, the same initial weight, and the same parameters setting. Both truth and falsity networks in each ensemble component apply the same generated training set. The difference between both neural networks is that the falsity network applies the complement of target outputs used in the truth network. For example, if the target value of the truth network is 0.7 then the target value of the same input pattern for the falsity network is set to 0.3.

After the truth and falsity values are predicted, the next step is to aggregate the outputs. Two aggregation techniques are explained in this paper. The first technique is similar to the simple averaging used for the classification in our previous paper[6], but we apply it to the regression problem in this paper. The second technique is a novel aggregation technique proposed in this study. Let $T_j(x_i)$ and $F_j(x_i)$ be the truth and falsity values predicted for the input pattern i of the component j, where $j = 1, 2, 3, ..., m$. Both techniques can be described below.

1. Equal weight averaging
   For each input pattern, m truth values are averaged and m falsity values are averaged. The average truth value and the complement of the average falsity value for each input pattern are combined using a simple averaging method. Let $T_{avg}(x_i)$ be an average truth value for the input pattern x_i. Let $F_{avg}(x_i)$ be an average falsity value for the input pattern x_i. The combined output $O(x_i)$ can be computed as the following.

$$O(x_i) = \frac{T_{avg}(x_i) + (1 - F_{avg}(x_i))}{2} \tag{1}$$

$$T_{avg}(x_i) = \frac{\sum_{j=1}^{m} T_j(x_i)}{m} \tag{2}$$

$$F_{avg}(x_i) = \frac{\sum_{j=1}^{m} F_j(x_i)}{m} \tag{3}$$

2. Dynamic weight averaging
In this technique, uncertainty of type vagueness is considered. From the experiment, m components provide m different truth values and m different falsity values. Hence, the predicted truth value obtained from each component is not sharp. Also, the predicted falsity value is not sharp. They are not sharp in the sense that each network component cannot provide the precise predicted value. The boundary of the truth and falsity values are vague. In this paper, the vagueness of the truth value is computed as the average of the absolute pairwise difference among all the truth values for each input pattern. Also, the vagueness of the falsity value is computed as the average of the absolute pairwise difference among all the falsity values for each input pattern. Let $V_{\mathrm{T}}(x_i)$ be an average vagueness of the truth values of the input pattern x_i. Let $V_{\mathrm{F}}(x_i)$ be an average vagueness of the falsity values of the input pattern x_i. Both vagueness values are used to weight the combination between the predicted truth and falsity values. The weight for the truth values is computed as the complement of the $V_{\mathrm{T}}(x_i)$. The weight for the falsity value is calculated as the complement of the $V_{\mathrm{F}}(x_i)$. These two types of weight are considered as the certainty in the prediction. In this study, we consider the certainty for predicting the falsity value is equal to the certainty for predicting the non-falsity value, which is the complement of the falsity value. Let $W_{\mathrm{T}}(x_i)$ be the weight for the truth value, and $W_{\mathrm{F}}(x_i)$ be the weight for the falsity value. The dynamic averaging output $O(x_i)$ can be calculated as follows.

$$O(x_i) = (W_{\mathrm{T}}(x_i) \times T_{avg}(x_i)) + (W_{\mathrm{F}}(x_i) \times (1 - F_{avg}(x_i))) \tag{4}$$

$$W_{\mathrm{T}}(x_i) = \frac{1 - V_{\mathrm{T}}(x_i)}{(1 - V_{\mathrm{T}}(x_i)) + (1 - V_{\mathrm{F}}(x_i))} \tag{5}$$

$$W_{\mathrm{F}}(x_i) = \frac{1 - V_{\mathrm{F}}(x_i)}{(1 - V_{\mathrm{T}}(x_i)) + (1 - V_{\mathrm{F}}(x_i))} \tag{6}$$

$$V_{\mathrm{T}}(x_i) = \frac{\sum_{k,h=1}^{m} |T_k(x_i) - T_h(x_i)|}{m(m-1)/2}; \quad k \neq h \tag{7}$$

$$V_{\mathrm{F}}(x_i) = \frac{\sum_{k,h=1}^{m} |F_k(x_i) - F_h(x_i)|}{m(m-1)/2}; \quad k \neq h \tag{8}$$

3 Experiments

3.1 Data Set

In this paper, we deal with porosity prediction from well log data collected from real and practical data in the oil and gas industry. The data set is taken from four wells which are available in a real reservoir. The actual well locations lie approximately on a straight line with the following order: Well 3, Well 1, Well 2 and Well 4. The well logs used in this experiment are gamma ray (GR), deep resistivity (RDEV), shallow resistivity (RMEV), flushed zone resistivity (RXO), bulk density (RHOB), neutron porosity (NPHI), photoelectric factor (PEF), and sonic travel time (DT). This experiment aims to create an estimator in order to predict porosity (PHI) from these eight well logs. The well logs are recorded from different wells at various depths. All variables are normalized within the range of $[0, 1]$. In our data set, 269 data obtained from wells 1, 3, and 4 are used for training and 105 data obtained from well 2 are used for testing.

3.2 Experimental Methodology and Results

An ensemble created in this study consists of thirty components. Thirty generated training data sets are then created based on bootstrap resampling. Hence, each generated training set is used as input for each component in the ensemble. For the complementary neural network (CMTNN) ensmebles, the truth and falsity networks are created based on feed-forward backpropagation neural networks (BPNN). All thirty pairs of neural networks have the same architecture. This experiment deals with eight input features listed in section 3.1. Hence, each network composes of eight input units, one hidden layer constituting of sixteen neurons, and a single output unit. However, the falsity network is trained using the complement of the target output values used to train the truth network. In order to aggregate the output, two proposed aggregation techniques described in the previous section are applied.

In this paper, we do not consider the optimization of the individual predictors but concentrate only on the improvement of the combined prediction. Hence, twenty CMTNN ensembles are created. Twenty different groups of thirty bootstrap resampled training sets are then set up and applied to twenty CMTNN ensembles. Twenty results of the mean square error (MSE) obtained from our proposed techniques are compared in table 1 to those obtained by applying the existing simple averaging technique that uses only the truth value of a single backpropagation neural network (BPNN) instead of a pair of networks in each ensemble component. Furthermore, our results are also compared to the results obtained from other ensemble models which are general regression neural network (GRNN), support vector machine (SVM) with linear, polynomial, and radial basis function (RBF) kernels. For SVM, mySVM [13] is used in the experiment in this study. These ensemble models are created based on the same generated training sets used to train twenty CMTNN ensembles. The comparison of the mean square error (MSE) among the proposed CMTNN ensembles and those ensemble models are also shown in table 1.

Table 1. The comparison among the mean square error (MSE) obtained from the proposed averaging techniques based on CMTNN ensembles and the simple averaging techniques based on SVM ensembles, GRNN ensembles, and BPNN ensembles for the test set of well log data

Ensemble	SVM linear (MSE)	SVM poly (MSE)	SVM RBF (MSE)	GRNN (MSE)	BPNN (MSE)	CMTNN Eq. weight (MSE)	CMTNN Dy. weight (MSE)
1	0.0286	0.0278	0.0327	0.0194	0.0518	0.0139	0.0131
2	0.0288	0.0272	0.0325	0.0186	0.0378	0.0134	0.0130
3	0.0285	0.0266	0.0326	0.0183	0.0253	0.0121	0.0120
4	0.0285	0.0279	0.0325	0.0197	0.0426	0.0136	0.0131
5	0.0283	0.0276	0.0325	0.0185	0.0329	0.0128	0.0127
6	0.0286	0.0287	0.0325	0.0185	0.0336	0.0126	0.0124
7	0.0278	0.0265	0.0324	0.0178	0.0471	0.0141	0.0135
8	0.0294	0.0266	0.0325	0.0187	0.0578	0.0160	0.0151
9	0.0279	0.0272	0.0325	0.0188	0.0377	0.0134	0.0130
10	0.0283	0.0269	0.0326	0.0196	0.0296	0.0121	0.0120
11	0.0278	0.0285	0.0325	0.0188	0.0585	0.0155	0.0147
12	0.0285	0.0275	0.0324	0.0193	0.0452	0.0130	0.0126
13	0.0286	0.0294	0.0327	0.0186	0.0516	0.0150	0.0145
14	0.0284	0.0265	0.0326	0.0194	0.0289	0.0116	0.0115
15	0.0284	0.0271	0.0323	0.0185	0.0546	0.0151	0.0144
16	0.0290	0.0266	0.0327	0.0180	0.0417	0.0127	0.0124
17	0.0283	0.0267	0.0324	0.0197	0.0258	0.0118	0.0118
18	0.0282	0.0275	0.0324	0.0186	0.0321	0.0128	0.0126
19	0.0288	0.0301	0.0325	0.0186	0.0396	0.0133	0.0129
20	0.0295	0.0277	0.0325	0.0190	0.0289	0.0120	0.0118
Avg	0.0285	0.0275	0.0327	0.0188	0.0402	0.0133	0.0130

The average MSE obtained from the existing simple averaging technique based on BPNN ensembles, GRNN ensembles, and SVM ensembles with linear, polynomial, and RBF kernels are 0.0402, 0.0188, 0.0285, 0.0275, and 0.0327, respectively. The average MSE obtained from the equal weight averaging and the dynamic weight averaging techniques based on CMTNN ensembles are 0.0133 and 0.0130, respectively. The results show that our both proposed techniques outperform the simple averaging technique that uses only the truth values in the ensemble of BPNN, GRNN, and SVM. The percent improvement of our techniques compared to the existing techniques can be shown in table 2. We found that the percent improvement of the dynamic weight averaging technique based on CMTNN ensembles compared to the simple averaging technique based on BPNN ensembles, GRNN ensembles, SVM ensembles with linear, polynomial, and RBF kernels are 67.74%, 31.18%, 54.55%, 52.94%, and 60.17%, respectively. Furthermore, the dynamic weight averaging technique is found to provide better performance than the equal weight averaging technique based on CMTNN ensembles in which the percent improvement is 2.9271.

Table 2. The percent improvement of the use of dynamic weight averaging based on CMTNN ensembles compared to BPNN ensembles, GRNN ensembles, and SVM ensembles with linear, polynomial, and radial basis function (RBF) kernels for the test set of well log data

Ensemble	Improvement in percentage with the use of dynamic weight averaging				
	SVM linear	SVM poly	SVM RBF	GRNN	BPNN
1	54.15	52.69	59.83	32.33	74.65
2	54.79	52.05	59.93	29.84	65.58
3	57.96	54.95	63.24	34.58	52.75
4	54.18	53.26	59.83	33.77	69.32
5	55.13	53.99	60.79	31.39	61.41
6	56.42	56.63	61.70	32.86	62.99
7	51.37	49.05	58.33	23.91	71.31
8	48.65	43.14	53.58	19.24	73.89
9	53.23	52.05	59.92	30.80	65.39
10	57.73	55.60	63.35	39.01	59.62
11	47.25	48.39	54.82	21.76	74.91
12	55.72	54.17	61.07	34.49	72.09
13	49.28	50.79	55.68	22.17	71.90
14	59.28	56.35	64.53	40.62	60.02
15	49.45	46.99	55.58	22.51	73.69
16	57.33	53.43	62.19	31.12	70.34
17	58.37	55.99	63.71	40.21	54.48
18	55.41	54.28	61.21	32.39	60.77
19	55.20	57.12	60.30	30.69	67.45
20	59.94	57.31	63.86	37.97	59.06
Avg	54.55	52.94	60.17	31.18	67.74

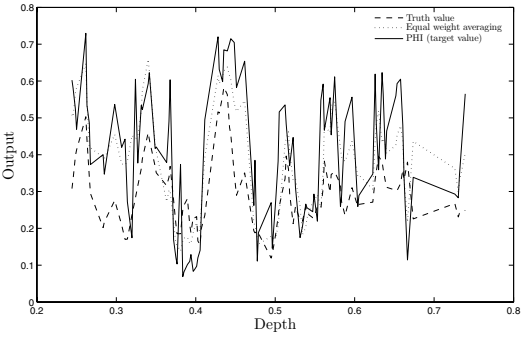


Fig. 1. The comparison between results obtained from the equal weight averaging technique based on CMTNN ensembles and the truth porosity values from BPNN ensembles for the test set of well logs (Ensemble 14)

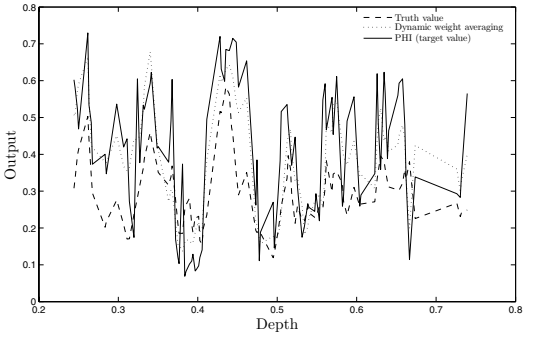


Fig. 2. The comparison between results obtained from the dynamic weight averaging technique based on CMTNN ensembles and the truth porosity values from BPNN ensembles for the test set of well logs (Ensemble 14)

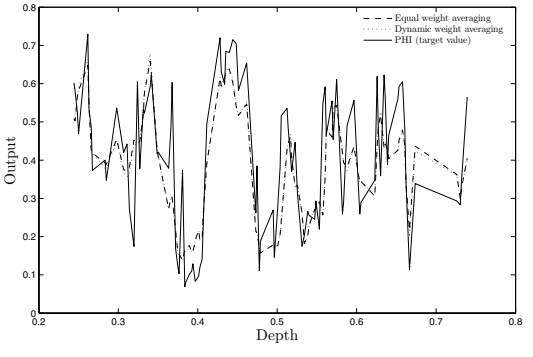


Fig. 3. The comparison between results obtained from the equal weight averaging and dynamic weight averaging based on CMTNN ensembles for the test set of well logs (Ensemble 14)

From table 1, ensemble 14 provides the minimum MSE for the equal weight averaging and the dynamic weight averaging techniques based on CMTNN ensembles, which are 0.0116 and 0.0115 respectively. Therefore, we found that not only the average MSE obtained from twenty runs of the proposed techniques have provided us with better performance compared to the existing techniques, but individual runs obtained from the proposed techniques also provided better accuracy result when compared to the existing techniques. Figure 1, 2, and 3 show the comparison among predicted porosity values obtained from our proposed techniques based on CMTNN ensembles and the simple averaging technique based on BPNN ensembles for the test set of well logs from the ensemble 14. The vertical axis represents the porosity values whereas the horizontal axis is the depth at which the input measurement were taken.

4 Conclusion and Future Work

In this paper, an ensemble of pairs of neural networks named CMTNN ensembles is used to solve a regression problem. The input patterns are derived from the bagging technique whereas pairs of complementary outputs: the truth and falsity values are predicted from pairs of complementary neural networks in an ensemble. Uncertainties in the prediction of the truth and falsity values are considered as the difference among all the predicted truth values and the difference among all the predicted falsity values, respectively. Uncertainty values are used to enhance the prediction results. The proposed averaging techniques based on CMTNN ensembles are compared to the simple averaging technique based on BPNN ensemble, GRNN ensembles, and SVM ensembles. We found that the results obtained from the proposed dynamic weight averaging technique provide better performances than the results obtained from the proposed equal weight averaging and the existing simple averaging techniques that uses only the truth values in the ensemble. In the future, other types of uncertainty will be quantified and used to support the prediction of porosity values in the well log data analysis problems.

References

1. Hansen, L.K., Salamon, P.: Pattern Analysis and Machine Intelligence. IEEE Transactions on Pattern Analysis and Machine Intelligence 12, 993–1001 (1990)
2. Zenobi, G., Cunningham, P.: Using Diversity in Preparing Ensembles of Classifiers Based on Different Feature Subsets to Minimize Generalization Error. In: Flach, P.A., De Raedt, L. (eds.) ECML 2001. LNCS, vol. 2167, pp. 576–587. Springer, Heidelberg (2001)
3. Melville, P., Mooney, R.J.: Constructing Diverse Classifier Ensembles using Artificial Training Examples. In: Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, pp. 505–512 (2003)
4. Breiman, L.: Bagging Predictors. Machine Learning 24, 123–140 (1996)
5. Schwenk, H., Bengio, Y.: Boosting Neural Networks. Neural Computation 12, 1869–1887 (2000)
6. Kraipeerapun, P., Fung, C.C., Brown, W., Wong, K.W.: Neural Network Ensembles using Interval Neutrosophic Sets and Bagging for Mineral Prospectivity Prediction and Quantification of Uncertainty. In: Proceedings of the 2006 IEEE International Conferences on Cybernetics and Intelligent Systems, Bangkok, Thailand, pp. 388–393 (2006)
7. Kraipeerapun, K., Fung, C.C., Wong, K.K.: Ensemble Neural Networks Using Interval Neutrosophic Sets and Bagging. In: Proceedings of the Third International Conference on Natural Computation, Haikou, China, vol. 1, pp. 386–390 (2007)
8. Duckham, M.: Ncertainty and Geographic Information: Computational and Critical Convergence. In: Representation in a Digital Geography. John Wiley, New York (2003)
9. Wong, P.M., Gedeon, T.D., Taggart, I.J.: An Improved Technique in Porosity Prediction: A Neural Network Approach. IEEE Transactions on Geoscience and Remote Sensing 33, 971–980 (1995)

10. Chandra, M., Srivastava, A.K., Singh, V., Tiwari, D.N., Painuly, P.K.: Lithostratigraphic Interpretation of Seismic Data for Reservoir Characterization. In: AAPG International Conference, Barcelona, Spain (2003)
11. Bhatt, A., Helle, H.B.: Committee Neural Networks for Porosity and Permeability Prediction from Well Logs. Geophysical Prospecting 50, 645–660 (2002)
12. Wong, K.W., Ong, Y.S., Gedeon, T.D., Fung, C.C.: Reservoir Characterization Using Support Vector Machines. In: Proceedings of the 2005 International Conference on Computational Intelligence for Modelling, Control and Automation, vol. 2, pp. 354–359 (2005)
13. Rüping, S.: mySVM-Manual (2000), http://www-ai.cs.uni-dortmund.de/SOFTWARE/MYSVM/

Research on the Evaluation Index System for Regional Integration: Analysis and Empirical Study

Xiangzhao Huang [1,2], Qi Fei [1,2], Yangmin Ou[1], and Jian Lu[1]

[1] Institute of Systems Engineering, Huazhong University of Science and Technology, Wuhan 430074, China
xiangzhao.h@gmail.com
[2] Key Lab. for Image Processing and Intelligent control, Huazhong University of Science and Technology

Abstract. In order to implement scientific development view and measure the effect of regional integration, the paper has constructed the evaluation index system for regional integration by means of the combined quantitative and qualitative analysis. Firstly, the paper illustrated the connotation, contents and significance of regional integration. Secondly, the paper put forward that the effect for regional integration is made of the presentation of the static, dynamic and external effects on system theory. Thirdly, the paper proposed the index system including the correlation degree, the degree of development gap and the development goal. Finally, it made an empirical analysis and made a conclusion for the recent problems and future development.

Keywords: System Theory, Regional Integration, Index System, Resource-Saving, Environment-Friendly.

1 Introduction

To carry out Scientific Development is a kind of systems engineering, which is related not only to every aspect of economic and social development, but also to economic and social activities as well as the complex relation of nature. The interaction among human beings, economic, social and natural environment is also involved. It requires that we should adopt systematic and scientific methods to analyze and solve problems. Meanwhile, it is a large system of research on economic and social development from multi-factors, multilayer and multi-aspects. Systems science is to give an analysis on objective world from the prospects of relations between the part and the whole.

At present, the proposal about city-circle overall corresponding reform and stimulating regional communication within the city-circle, demands a fulfillment of scientific development concept. According to the requirements of resource-saving and environment-friendly society, the resource-saving and ecology- protecting systematic mechanism, which offers a stimulus to the coordination among economic development, population, resources and environment, needs come into being as soon as possible. It will explore a brand-new development road for regional integration. This road, different from the traditional industrialization and urbanization, will pave the way for scientific development and harmonious society.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 185–193, 2009.

There is still no general definition for "regional integration" in previous documents. From the prospect of System theory, regional integration refers to a complex system concerning the inter-function, inter-dependence and inter-restriction among social, economic and environmental factors in many sub-regions (This paper will focus on two sub-regions). In this sense, it includes various components ranging form micro to macro, static to dynamic, inner to outer, temporal to spatial and physical to mental. All these components compose the integrity of regional system by interaction and inter-relation, which consists in a huge scale, multi-factors and a complex structure.

Therefore, how to realize regional integration with the limitation of "two types of society" is a huge and complex systematic project. Meanwhile, how to carry out regional integration scientifically under the guidance of Scientific Development View is a new challenge. However, there are no available theories and instruments for these problems.

During the process of constructing "two types of society" and evaluating the effects of regional integration, it is urgently necessary to build up evaluation index system for regional integration to provide basic theories for corresponding governmental policies. Besides, the evaluation index system shows great significance for exploring an regional integration mode which keeps in accordance with characteristics of a city and for programming regional integration of the city-circle.

2 Particularity of Constructing the Evaluation Index System for Regional Integration

2.1 The Definition of Regional Integration

In this paper, regional integration refers to the interaction between two or among more than two cities to achieve a clear assignment, a reasonable resource collocation and harmonious coordination on the whole, with the restriction of constructing two types of society and exploring a new industrialization and a new urbanization.

Therefore, regional integration is a process that the economic development of sub-region come to a fine and interactive state, in which sub-regions will achieve mutual-opening, frequent economic communication and a reasonable assignment. In this way, regional economy can keep a steady and efficient increase.

2.2 The Contents of Regional Integration

According to the definition of regional integration, combined with current development of city-circle, the main content of regional integration can be described as follows:

(1) The infrastructure integration. It mainly refers to high-speed and network, including the construction of road, railway, water carriage and aviation.
(2) The industrial development and layout integration. It mainly refers to a perfect industrial chain of related industries concerning labor division and cooperation, products matching, raw materials supply and technical service. Industrial docking which based on the development of industrial specialized labor division, in nature, is to achieve "competitive and cooperative" relation of regional economy, and aims to improve utilization of resources and retum of industries. In this way, a highly integrated regional economy will be realized.

(3) The market integration. It focuses on constructing a market consisting of capital, technology, communication, property rights, and human resources. Besides, it also requires to develop an integrated technical market, promote communication, network docking and resource co-share of personnel market, stimulate personnel market and labor market. Last but not least, a market with integrated property will come into being.
(4) The urban-rural integration. It aims to realize a harmonious development in urban-rural region through innovating its development mechanism.

The resources and ecology integration. Resources and ecology is a cross-regional problem. It proposes that we should adopt a reasonable exploration and utilization of cross-regional resources to boost the overall renovation of cross-regional pollution of rivers and lakes together with the construction of ecological system such as mountain, forest, marsh and so on.

2.3 The Significance of Regional Integration

(1) Every administrative region, like different organs making up a whole body, will cooperate and unite to achieve the aim of a prosperous, resource-saving and environment-friendly region.
(2) Every docking sub-region is supposed to completment, service and offer support for one another with its own characteristics. Therefore, regional integration is inevitable.
(3) Every sub-region will develop its own resource superiority, adapt industrial structure, enhance industrial complement and improve industrial arrangements. In this way, it will be favorable for saving resources, reducing cost and boosting regional productivity, scientific power, creativity, circulation and business. As it is, regional integration is practical and essential.

3 The Qualitative Analysis of Evaluation Index

According to 2, regional integration should be equipped with following four characteristics with the aim of constructing "two types" society: A closer relation among sub-regions; A reasonable regional assignment and cooperation; A narrower development distance among sub-regions with certain limit; An efficient development of the whole docking region together with keeping the balance among economy, social and ecological development. This paper, based on Systems, will present the regional integration outcomes from static, dynamic and outer aspects.

(1)The static characteristics of regional integration – the relation degree among sub-regions

Based on the theories of Systems of relativity, which defines an interrelated and interactive relation between each element within certain system or the relation between system and environment, there must be an interactive and inter-influential relation between two subsystems within a main system. On the contrary, the two are not possible to belong to the same main system. Therefore, the two subsystems may belong to the same main system only when they are with interrelated foundation and conditions.

Before regional integration is taken into consideration, there should be certain conditions and foundation for the two sub-regions. The first one is a relatively near geographical location and transportation infrastructure, which is reasonable for regional integration. A convenient transportation will bring about advantages for markets and trades. This paper will describe the relation between sub-regions, which is the static performance for regional integration.

(2)The dynamic characteristics of regional integration – the disparity among sub-regions

Based on the stability and dynamic state of a system, the movement of the subregions or components will keep the whole system in a certain state, and this stable state is carried out by regulation. Meanwhile, the relation among every component of the system and the relation between system and environment is time function, in other words, it changes with the passing of time.

According to the definition of regional integration in this paper, in the process of regional integration, the developmental disparity among sub-regions changed with time and within certain limits. Therefore, this paper depicted dynamic characteristics of regional integration based on the developmental disparity among sub-regions.

This paper found out that the less disparity between sub-regions showed a better interaction between two sub-regions, which means a better regional integration.

(3)The interior characteristics of regional integration – the coming results of regional integration

Based on the integrity of a system, the two sub-systems belonging to the same system have a mutual influence on each other and relate to each other. In this way, the system will be equiped with some functions that belong to none of its sub-systems, which will result in an effect called "integrity is larger than the addition of all the parts." That is to say, the improved function is the outer representation and the ultimate aim of the coordinance among the sub-systems. The better the regional integration of two sub-regions is, the better the results will be, because region is also a system. Therefore, this paper will depict the interior effects of regional integration through the results of regional integration.

According to the introduction, the aim of regional integration is to fulfill scientific development view, build up a resource-saving, environmental friendly society and explore a new industrialized and urbanized road, which is also the standard for evaluating the outer effects of regional integration.

4 Constructing of the Evaluation Index System for Regional Integration

According to its qualitative description and construction principles, evaluation index system of regional integration can be divided into three hierarchies: target hierarchy, criterion hierarchy and index hierarchy. The target hierarchy refers to the final effects of regional integration; Criterion hierarchy refers to three criterions including the association of sub-regions, the developmental disparity of sub-regions, and target level of regional integration; Index hierarchy includes some quantitative indexes, which keep in accordance with the requirements of each criterion. It is shown in the following figure:

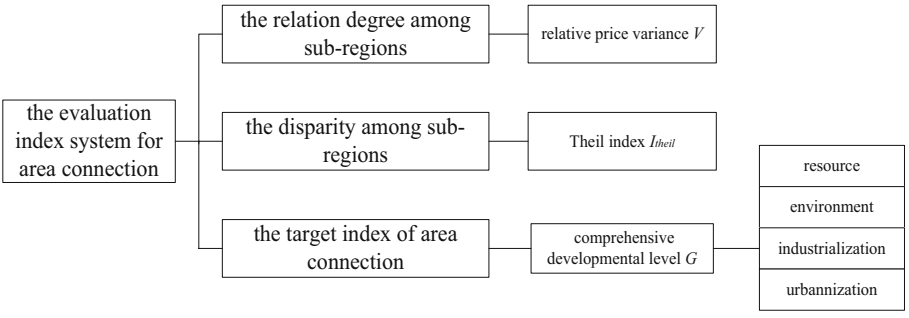


Fig. 1. The evaluation index system for regional integration

4.1 The Relation Index of Sub-regions

There is a close relation between association index of sub-regions and the integrity of regional economy. The nature of the integrity of regional economy is a free flow of components among sub-regions and a free trade. A very important precondition is an open market and an environment of free investment and trade. Because of this, some scholars assumed that regional economy integrity is a process of market integrity. It is clear that the economic market association is a precondition for the integrity of regional economy. Therefore, this paper will show the association degree of sub-regions and economic integrity by adopting market integrity.

There are various methods to measure market integrity and market division, but many are not practical when measuring market integrity. Parsley and Wei adopted empirical study which took relative price variance $VAR(P_i/P_j)$ as object. QIHan has also put Parsley and Wei's research into his own empirical analysis and took relative price variance as a dynamic index for market integrity. Based on this, Glacier Theory, which is a strong theoretical foundation for Price Law, will make up for other methods to some extent. Therefore, this research adopted Price Law to measure the integrity of regional market to show the association of sub-regions.

Suppose P^t_i ,P^t_j was the price of certain goods at time t in two regions. P^t_i / P^t_j was first difference of relative price,namely $ln(P^t_i / P^t_j) - ln(P^{t-1}_i / P^{t-1}_j)$. The original data we used was chain index of goods retail price. At last, we chose the absolute value of relative price to measure variance, namely:

$$V = \left| \ln(P_i^t / P_j^t) - \ln(P_i^{t-1} / P_j^{t-1}) \right| = \left| \ln(P_i^t / P_i^{t-1}) - \ln(P_j^t / P_j^{t-1}) \right| \quad (1)$$

It became narrow as time, reflecting the fluctuation of relative price became smaller and the obstacles in market integrity turned less. According to these, it is reasonable to deduce that market was equipped with association. Because retail price of goods is a comprehensive index, which will keep stable with a relatively long time, relative price variance V is considered as index of sub- regions association to measure static characteristics of regional integration level.

4.2 The Developmental Disparity Index of Sub-Regions

There are many ways to measure regional developmental disparity. This paper adopted Theil Index based on dynamic level of disparity between sub-regions and

Hui Lui's evaluation analysis on evaluation measures for regional disparity. Theil analyzed the individual disparity based on the concepts of information quantity and entropy. Theil index could be used to measure regional disparity. The less disparity between sub-regions, the smaller the Theil index is and vice versa. The calculation formula is as follows:

$$I_{theil} = \sum_{i=1}^{n} p_i (y_i / u) \lg(y_i / u) \quad c=1 \tag{2}$$

$$I_{theil} = \sum_{i=1}^{n} p_i \lg(y_i / u) \quad c=0 \tag{3}$$

In these formulas, y_i (i = 1, 2, 3,…, n) was GDP Per capita in sub-region i; u was GDP Per capita in all regions; p_i was the proportion that sub-region took up in all regions; Parameter C was to measure the change of index changes. Generally speaking, when c < 2, the change of corresponding index is flexible. (In this paper, $c = 0$)

Table 1. The target index of regional integration

Factors	index	calculation formula
Resource	energy consumption per unit of GDP	total energy consumption / area GDP
	water consumption per unit of GDP	water consumption / area GDP
	Consumption of electricity per unit of GDP	Consumption of electricity / area GDP
	coal consumption per unit of GDP	coal consumption / area GDP
Environment	the amount of waste water per ten thousand yuan industrial production value	The amount of industrial waste water / industrial production value
	the amount of solid waste per ten thousand yuan industrial production value	the amount of industrial solid waste / industrial production value
	the amount of SO_2 emission per ten thousand yuan industrial production value	The amount of industrial SO_2 emission / industrial production value
Industrialization	Industrial Added Value Rate	Industrial Added Value / industrial output value
	whole-society-productivity	Industrial Added Value / Average Number of Employment (person)
	The proportion of environmental protection input to GDP	The funds of environmental protection / area GDP
	the proportion of manufacturing industry to GDP	value-added of manufacturing industries / area GDP
Urbanization	urbanization level	the urban population / total population
	per capita gdp	area GDP / total population
	GDP increase rate	(Current GDP - previous GDP)/ previous GDP
	The proportion of value added of service industry to GDP	value added of service industry / area GDP

4.3 The Target Index of Regional Integration

The target index of regional integration in this paper is defined by its target restriction. G, the comprehensive developmental level of regions, included resources, environment, industrialization and urbanization. Table 1 showed calculation formula for each index.

5 The Empirical Analysis of Regional Integration

City Wuhan and city Huanggang in Hubei were the chosen sub-regions in this paper to test the rationality and efficiency of evaluation system of regional integration. In the calculation of V and I_{theil}, only the year 1985 to 2006 was taken into account considering the availability of data and long-time rationality of index. In calculation of G, the year 1995 to 2006 was taken into account, considering that some data was not available.

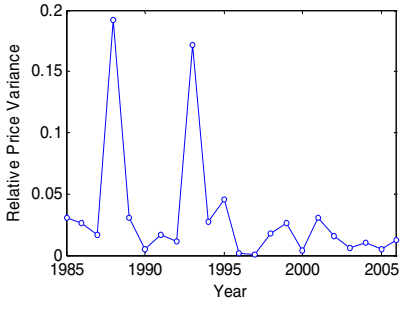


Fig. 2. The relation degree between two cities

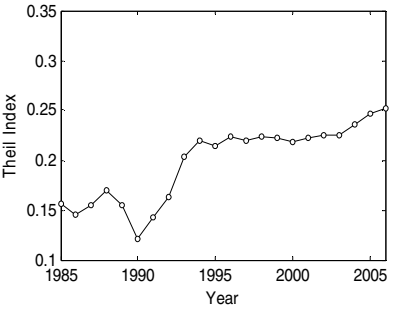


Fig. 3. The disparity between two cities

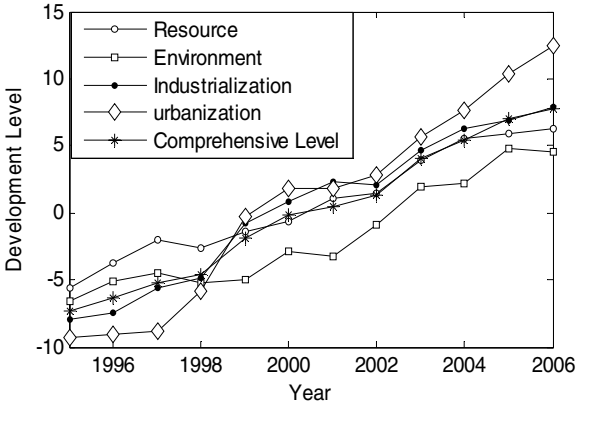


Fig. 4. The comprehensive development level between two cities

The relative price variance showed the relevance of sub-regions, representing the historical relation of regions, regional division and cooperation and static effects of resource allocation, which was shown in figure 2. Obviously, the line became narrower with the change of time, and it reflected a smaller relative price fluctuation between city Wuhan and city Huanggang with fewer obstacles in market integrity. On the whole, the association level of city Wuhan and city Huanggang was improved and tended to be stable from the year 1985 to 2007 in spite of some fluctuation to some degree. Therefore, these two cities were equipped with conditions and foundation for regional integration.

Theil index was to measure the dynamic changes of developmental disparity among sub-regions, which was shown in figure 3. Although there was a growing disparity between city Wuhan and city Huanggang, there were still few changes from the year 1994 to 2004, while relatively more changes in the year 2005 and 2006. Therefore, it is necessary to adopt some measures concerning industrial development and integration of layout to realize a better and faster development.

When calculating the comprehensive development level, city Wuhan and city Huanggang were considered as one region. Each index in the whole region was calculated. Then, the smallest index (the smaller the better) was changed to the largest index (the larger, the better) to achieve the uniformization and standardization of indexes. Factor analysis was adopted to obtain resource saving, friendly environment, industrialization and urbanization in each year from the year 1995 to 2006. Then comprehensive development level G will be obtained by weighted average,in which the weight of each component was 0.25. The changes were as figure 4. The larger slope of the overall industrialization and urbanization between city Wuhan and city Huanggang showed a faster increase of development.

6 Conclusions

It is an important and great systems engineering problem concerning how to carry out regional integration efficiently in the process of constructing “two types” society. The first task is to establish evaluation index system of regional integration to measure its advantages and disadvantages and to provide theoretical foundation for regional integration.

This paper began from the definition and significance to construct the outline of regional evaluation index system, which put forward that the effects of regional integration consisted of the static and dynamic characteristics of regional interior docking together with its outer effects. Then, from the methods of constructing index system, it showed that target was to measure the effects of regional integration, principle was made up from regional association, developmental disparity of sub-regions and the target of regional integration, which referred to relative price variance, Theil index and regional comprehensive development that three principles required. Besides, it also pointed out that comprehensive development was obtained by index system from four aspects: resource saving, environment friendly, new industrialization and new urbanization. At last, two neighboring cities within a province would be chosen as sub-regions, and did study on the index system to analyze regional evaluation index. In this way, it tested the efficiency and rationality evaluation index system.

As in our country, there is still a long way to go to construct evaluation index system of regional integration. In this paper, index system was proposed initially, which was significant for measuring the effects of regional integration. However, since there was no combination with future changes of index system of practical case in this paper, the strategic analysis is the main task for the future research.

References

1. Qian, X.S.: Creating Systems Science. Shanxi Press of Science and Technology, Taiyuan (2001)
2. Golley, J.: Regional Patterns of Industrial Development during China's Economic Transition. Economic Transition 10(3), 761–801 (2002)
3. Poncet, S.: Measuring Chinese Domestic and International Integration. China Economy (14), 1–21 (2003)
4. Peng, R.S.: On the Meaning, Mechanism and Evaluation of Coordinative Development of Interregional Economy. In: Doctoral Dissertation, Henan University, Zhengzhou, China (2007)
5. Peng, Q.M.: The Concept and Mechanism of Regional Economy Integration. Research On Development 16(2), 47–49 (2001)
6. Young, A.: The Razor's Edge: Distortions and Incremental Reform in the People's Republic of China. Quarterly Journal of Economics CXV, 1091–1135 (2000)
7. Feng, Y., Genna, G.M.: Regional Integration and Domestic Institutional Homogeneity: a Comparative Analysis of Regional Integration in the Americas. Pacific Asia and Western Europe, Review of International Political Economy 13(5), 278–309 (2003)
8. Xu, X.P.: Have the Chinese Provinces Become Integrated under Reform. China Economic Review 13(1), 116–133 (2002)
9. Parsley, D.C., Wei, S.J.: Limiting Currency Volatility to Stimulate Goods Market Integration: A Price Based Approach, NBER Working Paper 8468 (2001)
10. Gui, Q.H., Chen, M.: Do China Domestic Commodity Markets Tend to Segmentation or Integration. The Journal of World Economy 66(2), 20–30 (2006)
11. Samuelson, P.: Theoretical Note on Trade Problem. Review of Economics and Statistics 8(46), 145–164 (1954)
12. Chen, X.S., Xu, Y.: Factors Affecting China's Regional Gap: An Empirical Study. Social Sciences in China 25(5), 117–129 (2004)
13. Sui, Y., Zhu, K.L.: Indicator System of Regional Disparity Measurement and Quantitative Calculation. Statistical Research 17(10), 37–45 (2000)
14. Liu, H.: Regional Inequality Measurement: Methods and Evaluations. Geographical Research 25(4), 710–718 (2006)
15. Shang, W.P.: A Study on Inequality Degree of City Residents Expenditures in China. China Soft Science 17(8), 102–104 (2002)
16. Guo, Y.J.: Theory and Methods of Comprehensive Evaluation. Science Press, Beijing (2002)

A Type of Integrating Formula without Derivative

Aifang Long and Di Ning*

College of Computer Science, South-Central University for Nationalities, Wuhan 430074, People's Republic of China
lll1aaa1fff1@tom.com

Abstract. In this paper, A type of integrating formula is constructed, which requires function value at nodes without derivative. In addition, the formula needs less calculation than the Trapezoidal Rule, but the order of convergence is much better. Therefore a more effective .result is obtained.

Keywords: Numerical Integration, Derivative, Interpolation.

1 Introduction

Numerical integration formulas contain interpolation integration formula and Gaussian integration , but Gaussian integration has limitation. Although the Trapezoidal Rule has no same limitation, the order of convergence is bad. In order to obtain the better result, the authors substitute integrand with the Hermite polynomial in the paper [3] and [4]. But the integration formula requires not only the function value and first-order derivative at nodes, but also the second derivative. Therefore, when the integrand is complex and the derivative is not easy to calculate, the formula will have great limitation. In this paper, A type of integrating formula is constructed, which requires function value at nodes without derivative. And a more effective result is obtained.

2 Notations and Preliminaries

Let's choose some positive integer n and break the interval $[a, b]$ into equal n pieces. The width of each piece is $\frac{b-a}{n}$, and the nodes are $x_k = a + kh, k = 0, \cdots, n$. Assume $x_{\frac{k+1}{2}} = x_k + \frac{h}{2}$, and in the interval $[x_k, x_{k+1}]$, we can get the zero degree interpolation polynomial $p(x) = f(x_{\frac{k+1}{2}})$ via $(x_{\frac{k+1}{2}}, f(x_{\frac{k+1}{2}}))$, and the error of the zero degree interpolation polynomial is $f(x) - p(x) = f^{'}(\eta)(x - x_{\frac{k+1}{2}})$, where $\eta \in (x, x_{\frac{k+1}{2}})$. And in the $[x_k, x_{k+1}]$, we calculate the integration using the Rectangle Formula, i.e. $\int_a^b f(x)dx \approx \int_a^b p(x)dx = f(x_{\frac{k+1}{2}})h$. By the

* Corresponding author.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 194–200, 2009.

generalized mean value theorem of Algebra and differential mean value theorem, we obtain the error:

$$\begin{aligned}
&\int_a^b f(x)dx - f(x_{\frac{k+1}{2}})h = \int_a^b [f(x) - p(x])dx \\
&= \int_a^b f^{'}(\eta)(x - x_{\frac{k+1}{2}})dx \\
&= \int_{x_k}^{x_{\frac{k+1}{2}}} f^{'}(\eta)(x - x_{\frac{k+1}{2}})dx + \int_{x_{\frac{k+1}{2}}}^{x_{k+1}} f^{'}(\eta)(x - x_{\frac{k+1}{2}})dx \\
&= f^{'}(\xi_1)\int_{x_k}^{x_{\frac{k+1}{2}}}(x - x_{\frac{k+1}{2}})dx + f^{'}(\xi_2)\int_{x_{\frac{k+1}{2}}}^{x_{k+1}}(x - x_{\frac{k+1}{2}})dx \\
&= -\frac{h^2}{8}f^{'}(\xi_1) + \frac{h^2}{8}f^{'}(\xi_2) \\
&= \frac{h^2}{8}f^{''}(\xi_k)(\xi_2 - \xi_1).
\end{aligned}$$

Where $\xi_1 \in (x_k, x_{k+1})$, $\xi_2 \in (x_{\frac{k+1}{2}}, x_{k+1})$, $\xi_k \in (\xi_1, \xi_2)$, for $|\xi_2 - \xi_1| < h$, we have

$$|\int_{x_k}^{x_{k+1}} f(x)dx - hf(x_{\frac{k+1}{2}})| \leq \frac{h^2}{8}\max_{x_k \leq x \leq x_{k+1}} |f^{'}(x)|. \tag{1}$$

We calculate the approximation I_k of the integration by the Rectangle Formula in the sub-interval, and approximate the integration $I = \int_a^b f(x)dx$ with $\sum_{k=0}^{n-1} I_k$, then we get the formula :

$$I(h) = \sum_{k=0}^{n-1} hf(x_{\frac{k+1}{2}}). \tag{2}$$

By the formula (1), we deduce the error estimation of the formula (2):

$$|I - I(h)| = |\sum_{k=0}^{n-1}[\int_{x_k}^{x_{k+1}} f(x)dx - hf(x_{\frac{k+1}{2}})]| \leq \frac{h^2}{8}\max_{x_k \leq x \leq x_{k+1}} |f^{'}(x)|$$

$$\leq \frac{b-a}{8}h^2 \max_{x_k \leq x \leq x_{k+1}} |f^{''}(x)|$$

In order to consider it's remainder formula. we get the following Taylor expansion:

$$f(x) = f_{\frac{k+1}{2}} + f^{'}_{\frac{k+1}{2}}(x - x_{x_{\frac{k+1}{2}}}) + \frac{f^{''}_{\frac{k+1}{2}}}{2!}(x - x_{x_{\frac{k+1}{2}}})^2 + \cdots. \tag{3}$$

Here $f^j_{\frac{k+1}{2}}, j = 1, \cdots, n$ denotes $f^j(x_{\frac{k+1}{2}}), j = 1, \cdots, n$, and in the interval $[x_k, x_{k+1}]$, we obtain the integration

$$\int_{x_k}^{x_{k+1}} f(x)dx = f_{\frac{k+1}{2}}h + \frac{f^{''}_{\frac{k+1}{2}}}{3!}\frac{h^3}{2^2} + \frac{f^{(4)}_{\frac{k+1}{2}}}{5!}\frac{h^5}{2^4} + \frac{f^{(6)}_{\frac{k+1}{2}}}{3!}\frac{h^7}{2^6} + \cdots$$

and for all k sum from 0 to $n-1$,

$$I = h\sum f_{\frac{k+1}{2}} + \frac{h^3}{3!\times 2^2}\sum f''_{\frac{k+1}{2}} + \frac{h^5}{5!\times 2^4}\sum f^{(4)}_{\frac{k+1}{2}} + \frac{h^7}{7!\times 2^6}\sum f^{(6)}_{\frac{k+1}{2}} + \cdots \quad (4)$$

$$I = I(h) + \frac{h^3}{3!\times 2^2}\sum f''_{\frac{k+1}{2}} + \frac{h^5}{5!\times 2^4}\sum f^{(4)}_{\frac{k+1}{2}} + \frac{h^7}{7!\times 2^6}\sum f^{(6)}_{\frac{k+1}{2}} + \cdots \quad (5)$$

By the formula (4) and $\int_a^b f''(x)dx = f'(b) - f'(a)$, we obtain

$$h\sum f''_{\frac{k+1}{2}} = f'(b) - f'(a) - \frac{h^3}{3!\times 2^2}\sum f^{(4)}_{\frac{k+1}{2}} + \frac{h^5}{5!\times 2^4}\sum f^{(6)}_{\frac{k+1}{2}} + \cdots$$

Substitute the above formula to (5), we thus have

$$I = I(h) + \frac{h^2}{24}[f'(b) - f'(a)] - \frac{7h^5}{360\times 2^4}\sum f^{(4)}_{\frac{k+1}{2}} - \frac{h^7}{5!\times 7\times 2^6}\sum f^{(6)}_{\frac{k+1}{2}} + \cdots \quad (6)$$

Using formula (4) to $f^{(4)}(x)$, we obtain

$$h\sum f^{(4)}_{\frac{k+1}{2}} = f'''(b) - f'''(a) - \frac{h^3}{3!\times 2^2}\sum f^{(6)}_{\frac{k+1}{2}} + \cdots$$

And substitute the above formula to (6), we have

$$I = I(h) + \frac{h^2}{24}[f'(b) - f'(a)] - \frac{7h^4}{5760}[f'''(b) - f'''(a)] + \frac{3!h^7}{5!\times 126\times 2^6}\sum f^{(6)}_{\frac{k+1}{2}} + \cdots$$

We repeat the above operation, and get

$$I = I(h) + b_1h^2 + \cdots + b_kh^{2k} + \cdots$$

Where $b_i, i = 1, 2, \cdots$ is independent of h.

Theorem 1. Assume $f(x) \in C^\infty[a,b]$, then $I = I(h) + b_1h^2 + \cdots + b_kh^{2k} + \cdots$, where $b_i, i = 1, 2, \cdots$ is independent of h, and $b_1 = \frac{1}{24}[f'(b) - f'(a)]$, $b_2 = \frac{7}{5670}[f'''(b) - f'''(a)]$, $b_3 = \frac{31}{967680}[f^{(5)}(b) - f^{(5)}(a)]$.

By Theorem 1, we can obtain the following three numerical formula calculating $I = \int_a^b f(x)dx$:

$$I^{(1)}(h) = I(h) - \frac{h^2}{24}[f'(a) - f'(b)] \quad (7)$$

$$I^{(2)}(h) = I(h) - \frac{h^2}{24}[f'(a) - f'(b)] + \frac{7h^4}{5760}[f'''(a) - f'''(b)] \quad (8)$$

$$I^{(2)}(h) = I(h) - \frac{h^2}{24}[f'(a) - f'(b)] + \frac{7h^4}{5760}[f'''(a) - f'''(b)] - \frac{31h^6}{967680}[f^{(5)}(a) - f^{(5)}(b)] \quad (9)$$

Here, we respectively approximate to $I = \int_a^b f(x)dx$ with $I^{(1)}(h), I^{(2)}(h)$ and $I^{(3)}(h)$, which errors are $o(h^4), o(h^6)$ and $o(h^8)$. By the errors, we know that the order of convergence is good, but the first derivative, third derivative, and fifth derivative must be calculated beforehand.

3 The Integration Formula without Derivative

For $f(x) \in C^{2k+1}[a,b]$, we have Taylor expansion:

$$f(x+jh) - f(x) = \sum_{i=1}^{2k} \frac{f^{(i)}}{i!}(jh)^i + \frac{f^{(2k+1)(\xi_j)}}{(2k+1)!}(jh)^{2k+1} \tag{10}$$

Where $j = 1, 2, \cdots, 2k, \xi_j \in [x, xjh]$, and denotes $\xi_j(x) = \frac{f^{(2k+1)(\xi_j)}}{(2k+1)!} j^{2k+1}, y_i = f^{(i)}(x)h^i, E = (\varepsilon_1(x), \cdots, \varepsilon_{2k}(x))^T, Y = (y_1, \cdots, y_{2k}), f_i^- = f(x+ih) - f(x), F = (f_1^-, \cdots, f_{2k}^-), A = (a_{ij})), a_{ij} = \frac{i^j}{j!}, i, j = 1, 2, \cdots, 2k$, so the formula (10) can be written to

$$F = AY + h^{2k+1}E \tag{11}$$

Theorem 2. In the formula (11) , the matrix A is nonsingular.

Proof:

$$A = \begin{pmatrix} 1 & \frac{1}{2!} & \frac{1}{3!} \cdots & \frac{1}{(2k)!} \\ 2 & \frac{2^2}{2!} & \frac{2^3}{3!} \cdots & \frac{2^{2k}}{(2k)!} \\ 3 & \frac{3^2}{2!} & \frac{3^3}{3!} \cdots & \frac{3^{2k}}{(2k)!} \\ \vdots & \vdots & \ddots & \vdots \\ 2k & \frac{(2k)^2}{2!} & \frac{(2k)^3}{3!} \cdots & \frac{(2k)^{2k}}{(2k)!} \end{pmatrix},$$

So

$det(A) = \frac{2\cdot 3\cdots 2k}{2!3!\cdots(2k)!} Vandermode(1, 2, \cdots, 2k) = \frac{2\cdot 3\cdots 2k}{2!3!\cdots(2k)!} \prod_{1\le j\le i\le 2k}(i-j)$. Because the inverse matrix of A exists, the matrix A is nonsingular. By the formula (11), we obtain

$$F = A^{-1}Y - h^{2k+1}A^{-1}E \tag{12}$$

Where $A^{-1} = (b_{ij})$, and the formula (12) is written to

$f^i(x) = \sum_{j=1}^{2k} b_{ij}[f(x+jh) - f(x)] - h^{2k+1}\sum_{j=1}^{2k} b_{ij}\varepsilon_j(x), i = 1, 2, \cdots, 2k$

Let $x = a$, and we have

$$f^i(a)h^i = \sum_{j=1}^{2k} b_{ij}[f(a+jh) - f(a)] - h^{2k+1}\sum_{j=1}^{2k} b_{ij}\varepsilon_j(a) \tag{13}$$

Let $x = b$, and substitute h with $-h$, we deduce

$$-f^i(b)h^i = \sum_{j=1}^{2k} b_{ij}[f(b-jh) - f(b)] - h^{2k+1}\sum_{j=1}^{2k} b_{ij}\varepsilon_j(b) \tag{14}$$

Adding the formula (13) to the formula (14), we get

$$h^i[f^i(a) - f^i(b)] = \sum_{j=1}^{2k} b_{ij}[f(a+jh) + f(b-jh)] - [f(a) - f(b)]\sum_{j=1}^{2k} b_{ij}$$
$$+h^{2k+1}\sum_{j=1}^{2k} b_{ij}[\varepsilon_j(b) - \varepsilon_j(a)] \tag{15}$$

1 For $i = 1, k = 1$, we obtain

$$A = \begin{pmatrix} 1 & \frac{1}{2!} \\ 2 & \frac{2^2}{2!} \end{pmatrix},$$

$b_{11} = 2, b_{12} = -\frac{1}{2}$. And by the formula (15), we have

$$\begin{aligned} & h[f^{'}(a) - f^{'}(b)] \\ &= -\frac{3}{2}[f(a) + f(b)] + 2[f(a+h) + f(b-h)] - \frac{1}{2}[f(a+2h) + f(b-2h)] + o(h^3) \end{aligned}$$

By the above formula and formula (7), we get the first numerical integrating formula without the derivative:

$$I_1(h) = I(h) - \frac{h}{48}[-3(f(a)+f(b)) + 4(f(a+h)+f(b+h)) - (f(a+2h)+b(a+2h))] \tag{16}$$

The error is $o(h^4)$.

2 For $k = 2$, we have $b_{11} = 4, b_{12} = -3, b_{13} = \frac{4}{3}, b_{13} = \frac{4}{3}, b_{14} = -\frac{1}{4}, b_{31} = 9, b_{32} = -12, b_{33} = 7, b_{34} = -\frac{3}{2}$.

Where

$$A = \begin{pmatrix} 1 & \frac{1}{2!} & \frac{1}{3!} & \frac{1}{4!} \\ 2 & \frac{2^2}{2!} & \frac{2^3}{3!} & \frac{2^4}{4!} \\ 3 & \frac{3^2}{2!} & \frac{3^3}{3!} & \frac{3^4}{4!} \\ 4 & \frac{4^2}{2!} & \frac{4^3}{3!} & \frac{4^4}{4!} \end{pmatrix}$$

and when $i = 1$, we get

$$\begin{aligned} & h[f^{'}(a) - f^{'}(b)] \\ &= -\frac{25}{12}[f(a) + f(b)] + 4[f(a+h) + f(b-h)] - 3[f(a+2h) + f(b-2h)] \\ &\quad + \frac{4}{3}[f(a+3h) + b(b-3h)] - \frac{1}{4}[f(a+4h) + f(b-4h)] + o(h^5) \end{aligned}$$

Let $i = 3$, by the (15), we obtain

$$\begin{aligned} & h^3[f^{'''}(a) - f^{'''}(b)] \\ &= -\frac{5}{2}[f(a) + f(b)] + 9[f(a+h) + f(b-h)] - 12[f(a+2h) + f(b-2h)] \\ &\quad + 7[f(a+3h) + b(b-3h)] - \frac{3}{2}[f(a+4h) + f(b-4h)] + o(h^5) \end{aligned}$$

By the above formula and formula (8), we get the second numerical integrating formula without the derivative:

$$\begin{aligned} & I_2(h) \\ &= I(h) + \frac{h}{11520}[965(f(a) + f(b)) - 1794(f(a+h) + f(b-h)) \end{aligned}$$

$$+1272(f(a+2h)+f(b-2h))-542(f(a+3h)+b(b-3h))$$
$$+99(f(a+4h)+f(b-4h))]$$

Where the error is $o(h^6)$.

3 By the above same method, let $k=3$, and we obtain the third numerical integrating formula having good accuracy.

$$\begin{aligned}&I_3(h)\\&=I(h)+\frac{h}{1935360}[183379(f(a)+f(b))-416872(f(a+h)+f(b-h))\\&+472211(f(a+2h)+f(b-2h))-395496(f(a+3h)+b(b-3h))\\&+21477(f(a+4h)+f(b-4h))-67184(f(a+5h)+f(b-5h))\\&+9185(f(a+6h)+f(b-6h))]\end{aligned}$$

Where the error is $o(h^8)$.

4 Numerical experimentation

Example. Calculate the integration $\int_0^{\frac{\pi}{2}} \sin x dx$.

Solution: The exact value of the above integration is 1. Let $T(h)$ denote the integration approximation by the formula , and the integration interval is divided by n equal parts . ε is the absolute error of the above methods. The result as follows:

When the equal parts are 8, $T(h)=3.2148\cdot 10^{-3}$,
When the equal parts are 16, $T(h)=8.0331\cdot 10^{-4}$,
When the equal parts are 32, $T(h)=2.0080\cdot 10^{-4}$.
When the equal parts are 8, $I(h)=1.6018\cdot 10^{-3}$,
When the equal parts are 16, $I(h)=4.0170\cdot 10^{-4}$,
When the equal parts are 32, $I(h)=1.0040\cdot 10^{-4}$.
When the equal parts are 8, $I_1(h)=1.5537\cdot 10^{-5}$,
When the equal parts are 16, $I_1(h)=1.0781\cdot 10^{-6}$,
When the equal parts are 32, $I_1(h)=7.0547\cdot 10^{-8}$.
When the equal parts are 8, $I_2(h)=2.1600\cdot 10^{-7}$,
When the equal parts are 16, $I_2(h)=4.5494\cdot 10^{-9}$,
When the equal parts are 32, $I_2(h)=7.9492\cdot 10^{-11}$.
When the equal parts are 8, $I_3(h)=2.9389\cdot 10^{-9}$,
When the equal parts are 16, $I_3(h)=2.2646\cdot 10^{-11}$,
When the equal parts are 32, $I_3(h)=1.0824\cdot 10^{-13}$.

So, the formula introduced in the paper has better accuracy than the Trapezoidal Formula, and the formal absolute error is half of the latter. Calculated by the formulas (16),(17),(18), both the order of convergence and accuracy are more accurate, so numerical result is more effective.

References

1. Zhang, D., Wang, X.: Computational Method Algorithmic Language. Higher Education Press, Beijing (1981)
2. Chen, H., Chen, S.: Numerical Computing Methods. Wuhan University Press, Wuhan (2002)
3. Liu, C.: A Type of Integrating Formula with Derivative. Journal of Xi'an Technological University 14, 232–237 (1994)
4. Liu, C.: A Type of Integrating Formula with Second Derivative. Journal of Xi'an Technological University 18, 247–250 (1998)

L1-norm Regularization Based Nonlinear Integrals

JinFeng Wang, KinHong Lee, and KwongSak Leung

Department of Computer Science & Engineering, The Chinese University of Hong Kong, Shatin, NT, Hong Kong SAR
{jfwang,khlee,ksleung}@cse.cuhk.edu.hk

Abstract. Since Nonlinear Integrals, such as the Choquet Integral and Sugeno Integrals, were proposed, how to get the Fuzzy Measure and confirm the unique solution became the hard problems. Some researchers can obtain the optimal solution for Fuzzy Measure using soft computing tools. When the Nonlinear Integrals can be transformed to a linear equation with regards to Fuzzy Measure by Prof. Wang, we can apply the L1-norm regularization method to solve the linear equation system for one dataset and find a solution with the fewest nonzero values. The solution with the fewest nonzero can show the degree of contribution of some features or their combinations for decision. The experimental results show that the L1-norm regularization is helpful to the classifier based on Nonlinear Integrals. It can not only reduce the complexity of Nonlinear Integral but also keep the good performance of the model based on Nonlinear Integral. Meanwhile, we can dig out and understand the affection and meaning of the Fuzzy Measure better.

Keywords: Nonlinear Integral, L1-Norm Regularization, Classification, LASSO.

1 Introduction

Nonlinear Integrals is known to have good results on classification and regression despite of the large computational complexity. Since Fuzzy Measure is introduced firstly by Sugeno [1], Nonlinear Integrals with respect to Fuzzy Measure had been proposed many versions by researchers and applied to classification and regression on real world data [2, 3, 4, 5]. In these methods, the Nonlinear Integrals are used as confidence fusion tools. Given an object $X=\{ x_1, x_2, \ldots, x_n \}$, for each class C_k, $k=1,2,\ldots m$, a Fuzzy Measure is needed to fuse the n degrees of confidence for statement : 'X belongs to class C' based on the value of each x_i, $i=1,2,\ldots,n$.

In all models [6, 7, 8, 9] with respect to Nonlinear Integrals, it is always a hard problem to confirm the Fuzzy Measure. The majority of researchers have used some soft computing tools to get optimal solutions. In this research, we use L1-norm regularization method [10] to obtain a solution of Fuzzy Measure with the fewest nonzero values. It can not only reduce the complexity of Nonlinear Integral but also keep the good performance of the model based on Nonlinear Integral. Meanwhile, we can dig out and understand the affection and meaning of the Fuzzy Measure better. This paper is organized as follows.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 201–208, 2009.

In section 2, the fundamental concepts with respect to Fuzzy Measures and Nonlinear Integral are introduced. Section 3 introduces the transformation of Nonlinear Integrals to linear equation. Then the main algorithm of Nonlinear Integrals for optimal solutions is presented in section 4. In next section describe the experimental results are showed and the detailed analyses are given. Finally, some conclusions are summarized.

2 Fundamental Concepts and Classifier Model

We are given a data set consisting of L example records, called training set, where each record contains the value of a decisive attribute, Y, and the value of predictive attributes $x_1, x_2, \ldots, x_n$. Positive integer L is the data size. The classifying attribute indicates the class to which each example belongs, and it is a categorical attribute with values coming from an unordered finite domain. The set of all possible values of the classifying attribute is denoted by $C = c_1, c_2, \ldots, c_m$, where each C_k, $k = 1, 2, \ldots, m$, refers to a specified class. The feature attributes are numerical, and their values are described by an n-dimensional vector, $(f(x_1), f(x_2), \ldots, f(x_n))$. The range of the vector, a subset of n-dimensional Euclidean space, is called the feature space. The j^{th} observation consists of n feature attributes and the classifying attribute can be denoted by $(f_j(x_1), f_j(x_2), \ldots, f_j(x_n), Y_j)$, $j = 1, 2, \ldots, L$. Before introducing the model, we give out the fundamental concepts as follows.

2.1 Fuzzy Measure

Let $X = x_1, x_2, \ldots, x_n$, be a nonempty finite set of feature attributes and $P(X)$ be the power set of X.

Definition 2.1. A Fuzzy Measure [8], μ, is a mapping from $P(X)$ to $[0, \infty)$ satisfying the following conditions:

1) $\mu(\varphi) = 0$;

2) $A \subset B \Rightarrow \mu(A) \leq \mu(B), \forall A, B \in P(X)$.

To further understand the practical meaning of the Fuzzy Measure, let us consider the elements in a universal set X as a set of predictive attributes to predict a certain objective. Then, for each individual predictive attribute as well as each possible combination of the predictive attributes, a distinct value of a Fuzzy Measure is assigned to describe its influence to the objective. Due to the nonadditivity of the Fuzzy Measure, the influences of the predictive attributes to the objective are dependent such that the global contribution of them to the objective is not just the simple sum of their individual contributions.

Set function μ is nonadditive in general. If $\mu(X)=1$, then μ is said to be regular. The monotonicity and non-negativity of Fuzzy Measure are too restrictive for real applications. Thus, the signed Fuzzy Measure, which is a generalization of Fuzzy Measure, has been defined [15, 16] and applied.

Definition 2.2. A set function $\mu : P(X) \to (-\infty,+\infty)$ is called a signed (non-monotonic) Fuzzy Measure provided that $\mu(\varnothing)=0$.

A signed Fuzzy Measure allows its value to be negative and frees monotonicity constraint. Thus, it is more flexible to describe the individual and joint contribution rates from the predictive attributes in a universal set towards some target.

2.2 Nonlinear Integrals

Definition 2.3. Let μ be a non-monotonic Fuzzy Measure on $P(X)$ and f be a real-valued function on X. The Nonlinear Integral of f with respect to μ is obtained by

$$\int f d\mu = \int_{-\infty}^{0} [\mu(F_\alpha) - \mu(X)]d\alpha + \int_{0}^{\infty} \mu(F_\alpha)d\alpha \tag{1}$$

where $F_\alpha = \{x | f(x) \geq \alpha\}$, for any $\alpha \in (-\infty,\infty)$, is called the $\alpha-$cut of f.

To calculate the value of the Nonlinear Integral of a given real-valued function f, usually the values of f, i.e., $f(x_1), f(x_2), \cdots, f(x_n)$, should be sorted in a nondecreasing order so that $f(x_1') \leq f(x_2') \leq \ldots \leq f(x_n')$, where $(x_1', x_2', \ldots x_n')$ is a certain permutation of $(x_1, x_2, \ldots x_n)$. So the value of Nonlinear Integral can be obtained by

$$\int f d\mu = \sum_{i=1}^{n} [f(x'_i) - f(x'_{i-1})]\mu(\{x'_i, x'_{i+1}, \ldots x'_n\}), \text{ where } f(x_0') = 0 \tag{2}$$

The Nonlinear Integral is based on linear operators to deal with nonlinear space.

3 Transformation of Nonlinear Integral

To be convenient, Wang [11] proposed a new scheme to calculate the value of a Nonlinear Integral with real-valued integrand by the inner product of two (2^n-1)-dimension vectors as

$$\int f d\mu = \sum_{j=1}^{2^n-1} z_j \mu_j \tag{3}$$

where

$$z_j = \begin{cases} \min\limits_{i:frc(\frac{j}{2^i}) \in [\frac{1}{2},1)} f(x_i) - \max\limits_{i:frc(\frac{j}{2^i}) \in [0,\frac{1}{2})} f(x_i), & \text{if it is bigger than zero or } j \text{ is } 2^n-1; \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

for $j = 1,2,\cdots,2^n-1$

with a convention that the maximum on the empty set is zero. Here, $frc(\frac{j}{2^i})$ denotes the fractional part of $\frac{j}{2^i}$. In the above formula, if we express j in the binary form $j_n j_{n-1} \cdots j_1$, then $\left\{ i \middle| frc(\frac{j}{2^i}) \in [\frac{1}{2}, 1) \right\} = \{ i | j_i = 1 \}$ and $\left\{ i \middle| frc(\frac{j}{2^i}) \in [0, \frac{1}{2}) \right\} = \{ i | j_i = 0 \}$.

A significant advantage of this new calculation scheme is that it can easily discover the coefficients matrix of a system of linear equations with the unknown variables μ when the Choquet integral is applied in further applications, such as regression and classification [7, 11, 12]. In those practical applications, values of the signed Fuzzy Measure are usually considered as unknown parameters which are to be estimated using the training data sets. The adoption of this new scheme make it convenient for using an algebraic method, such as the least square method, to estimate the value of μ, and furthermore, to reduce complexity of computation.

After having this transformation, we can obtain the Fuzzy Measure for a known dataset by using L1-norm Regularization.

4 Solutions of the Fuzzy Measure

For determining the Fuzzy Measure, researchers have proposed many methods. In our past work, we used GA to learn the value of Fuzzy Measure for each concrete dataset. In this paper, we propose a new method based on L1-norm regularization.

In many regression problem, the most popular function used is the Least Squares estimate, alternately referred to as minimizer of the residual sum of squared errors (RSS)[10]: $RSS = \sum_{i=1}^{n} (y_i - \omega_0 - \sum_{j=1}^{p} x_{ij} \omega_i)^2$. Regularization addresses the numerical instability of the matrix inversion and subsequently produces lower variance models. It is easy to see that the following penalized RSS function with respect to ω and ω_0: $\sum_{i=1}^{n} (y_i - \omega_0 - \sum_{j=1}^{p} x_{ij} \omega_i)^2 + \lambda \sum_{j=1}^{p} \omega_j^2$. This is referred to as L2 regularization. In order to simplify the notation used, we reduce it to the following problem (in matrix notation): $\|X\omega - y\|_2^2 + \lambda \|\omega\|_2^2$. While L2 regularization is an effective means of achieving numerical stability and increasing predictive performance, it can not address another important problem with Least Squares estimates, parsimony of the model and interpretability of the coefficient values. It does not encourage sparsity in some cases [13]. So a trend has been to replace L2-norm with an L1-norm recently. This L1 regularization has many of the beneficial properties of L2 regularization, but obtains sparse solutions that are more easily interpreted [10]. This property is what our algorithm wants. In Nonlinear Integrals, determining the Fuzzy Measure is the key procedure in the whole model. Fuzzy Measure represents the importance of features and the interaction degree of features combined.

We hope get a solution of Fuzzy Measure with the fewest nonzero values to find the most important features and feature combinations. Using L1-norm regularization,

we can minimize the following formula to reduce the size of nonzero in Fuzzy Measure: $\left\| \sum_{j=1}^{2^n-1} z_j \mu_j - y \right\|_2^2 + \lambda \|\mu\|_1$. We can control the condensation compress degree for Fuzzy Measure by adjusting the parameter λ. Author of [14] proposed the Least Absolute Selection and Shrinkage Operator (LASSO) model based on Gauss-Seidel method. The obvious advantages of the Gauss-Seidel approach are its simplicity and its low iteration cost. We applied this kind of LASSO to solve the above L1-Norm problem. Finally, the optimal Fuzzy Measure can be obtained.

5 Experiments and Analysis

We applied our model for classification to several datasets selected from UCI repository [17] which contains several biomedical data. They are 2-class datasets. The detailed information is shown in Table 1. Two of these datasets, Wisconsin Prognostic Breast Cancer and Echocardiogram, have noisy data labeled as ?. We process the noise to be substituted by the most common value or mean value.

Table 1. Description of Data sets

Datasets	Abbr.	Examples	Attributes	Classes	Reducts
Monk1	Monk1	556	6	2	{1, 2, 4}
Monk2	Monk2	601	6	2	{1, 2, 3, 4, 5, 6}
Monk3	Monk3	554	6	2	{1, 2, 3, 4, 5, 6}
Heart	Hear	270	13	2	{1, 8, 13}
Pima	Pima	768	7	2	{2, 6, 8}
Wisconsin Diagnostic Breast Cancer	Wdbc	569	30	2	{23, 24}
Wisconsin Prognostic Breast Cancer	Wpbc	699	9	2	{3, 5, 6, 7}
Echocardiogram	Echo	132	13	2	{1, 3, 9}
Australian Credit Approval	Aust	690	15	2	{2, 4, 5}

Table 2. The results of the data sets without reduct

Datasets	Accuracy	λ						
		0	1	5	10	20	50	100
Monk1	Train	0.953	0.954	0.953	0.935	0.880	0.765	0.516
	Test	0.946	0.949	**0.951**	0.913	0.858	0.750	0.501
Monk2	Train	0.992	0.993	0.993	0.988	0.910	0.684	0.657
	Test	0.991	0.992	**0.993**	0.973	0.904	0.672	0.657
Monk3	Train	0.897	0.891	0.870	0.871	0.854	0.773	0.700
	Test	0.855	0.854	**0.859**	0.858	0.844	0.773	0.700

We can see that the number of attributes of some datasets is rather large for Nonlinear Integrals to deal with. It will take very long time to learn the Fuzzy Measure. So the feature selection is a necessary step. Based on previous research, we adopt reduct in Rough Sets to process the data before classification. As we all known, there may be many reducts in Rough Sets for one database. We just pick out the one which have more overlap with that selected by Information gain Ranking method. The feature subsets selected are shown in Table 1. We can see the size of feature subsets from Rough Sets is greatly smaller than original one. This can greatly advance the efficiency of Nonlinear Integrals because the time of learning the signed Fuzzy Measure is reduced greatly.

We tested two sets experiments for original data and processed data with reduct separately. The results of the former one are shown in Table 2. We can see that the accuracy is decreasing as the value of λ is increasing. λ is the parameter for controlling the degree of compression for Fuzzy Measure. We set the value of λ as 0, 1, 5, 10, 20, 50 and 100 respectively. The larger the value of λ is, the fewer the number of zero in solution is. The compressing the Fuzzy Measure simplify the computation of Nonlinear Integrals at the cost of performance. We can select an appropriate value for λ to balance the complexity and the performance.

The Table 3 list the results of some data sets with reduct. These data have been processed once by feature selection. The features have been compressed to a small set. We can see that the accuracy of each data set have no too much fluctuation with different values of λ . It means that the compressing by controlling λ have the same affection as feature selection.

Table 3. The results of the data sets with reduct

Datasets	Accuracy	λ						
		0	1	5	10	20	50	100
Monk1	Train	0.953	0.953	0.953	0.953	0.953	0.953	0.953
	Test	0.953	0.953	0.953	0.953	0.953	0.953	0.953
Monk2	Train	0.992	0.993	0.993	0.988	0.910	0.685	0.685
	Test	0.991	0.993	0.993	0.975	0.906	0.673	0.673
Monk3	Train	0.897	0.890	0.870	0.871	0.854	0.773	0.773
	Test	0.858	0.858	0.862	0.862	0.842	0.773	0.773
Pima	Train	0.647	0.649	0.649	0.649	0.648	0.647	0.647
	Test	0.641	0.645	0.644	0.645	0.642	0.640	0.640
Wdbc	Train	0.851	0.851	0.851	0.851	0.851	0.851	0.851
	Test	0.851	0.851	0.851	0.851	0.851	0.851	0.851
Wpbc	Train	0.807	0.806	0.805	0.802	0.796	0.796	0.795
	Test	0.801	0.800	0.799	0.798	0.795	0.794	0.794
Echo	Train	0.888	0.888	0.887	0.886	0.887	0.848	0.849
	Test	0.882	0.883	0.882	0.883	0.879	0.821	0.821
Heart	Train	0.771	0.771	0.771	0.770	0.768	0.767	0.769
	Test	0.767	0.766	0.765	0.765	0.764	0.761	0.767
Aust	Train	0.844	0.843	0.843	0.842	0.844	0.815	0.800
	Test	0.839	0.839	0.839	0.837	0.834	0.807	0.797

As for describing the interaction of features, we can observe the solution by L1-Norm regularization method to acknowledge the trend and the regulation of Fuzzy Measure. We take the Monk2 as an example to introduce briefly relation of Fuzzy Measure and the control parameter λ . we can see that the number of nonzeros is decreasing as the value of λ increases. The order of each Fuzzy measure according to the value is not changed for each λ as long as it is not zero. When we select a value of λ to keep the balance of complexity and performance, those combinations of features with nonzero value are the relative important for contributing to decision. For this example, when λ =5 and λ =10, the accuracies are highest on Training set and Testing set respectively. Apparently the latter is simpler than the former which shift some subsets with small values to be ignored. In the table, negative value means the effect of the subset to contribution is just negative, which will not affect the decision.

6 Conclusions

Due to the great number of Fuzzy Measures to be determined, the computational complexity of Nonlinear Integrals is very large. To finding the values of each Fuzzy Measure is a hard work for those huge data sets. In this paper, we use the L1-norm method to solve the problem of complexity. L1-norm method can obtain the solution with the relative fewest nonzero values. We can get the Fuzzy Measure with small size by compressing the solution using L1-norm regularization, which can reduce the complexity greatly without losing performance. Experimental results show that we can select one value of parameter λ to keep a balance between complexity and performance. The detailed values of Fuzzy Measure can be confirmed to describe the interaction of features with respect to contribution for decision by using L1-norm regularization.

In the future work, we can learn the value of parameter λ to control the L1-norm's operation using a cross-validate method.

References

1. Sugeno, M.: Theory of Fuzzy Integrals and Its Applications. Doctoral Thesis, Tokyo Institute of Technology (1974)
2. Grabisch, M.: The Representation of Importance and Interaction of Features by Fuzzy Measures. Pattern Recognition Letters 17, 567–575 (1996)
3. Grabisch, M., Nicolas, J.M.: Classification by Fuzzy Integral: Performance and Tests. Fuzzy Stes and Systems 65, 255–271 (1994)
4. Keller, J.M., Yan, B.: Possibility Expectation and Its Decision Making Algorithm. In: 1st IEEE Int. Conf. On Fuzzy Systems, San Diago, pp. 661–668 (1992)
5. Mikenina, L., Zimmermann, H.J.: Improved Feature Selection and Classification by the 2-additive Fuzzy Measure. Fuzzy Sets and Systems 107, 197–218 (1999)
6. Xu, K.B., Wang, Z.Y., Heng, P.A., Leung, K.S.: Classification by Nonlinear Integral Projections. IEEE Transactions on Fuzzy System 11(2), 187–201 (2003)
7. Wang, W., Wang, Z.Y., Klir, G.J.: Genetic Algorithm for Determining Fuzzy Measures from Data. Journal of Intelligent and Fuzzy Systems 6, 171–183 (1998)
8. Wang, Z.Y., Klir, G.J.: Fuzzy Measure Theory. Plenum, New York (1992)

9. Wang, Z.Y., Leung, K.S., Wang, J.: A Genetic Algorithm for Determining Nonadditive Set Functions in Information Fusion. Fuzzy Sets and Systems 102, 463–469 (1999)
10. Hastie, T., Tibshirani, R., Friedman, J.H.: The Elements of Statistical Learning (Spring, 2001)
11. Wang, Z.: A new genetic algorithm for nonlinear multiregressions based on generalized Choquet integrals. In: Proc. 12th IEEE Intern. Conf. Fuzzy Systems, vol. 2, pp. 819–821 (2003)
12. Leung, K.S., Wong, M.L., Lam, W., Wang, Z., Xu, K.: Learning nonlinear multiregression networks based on evolutionary computation. IEEE Trans. On Systems, Man and Cybernetics, Part B 32(5), 630–644 (2002)
13. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. R. Statist. Soc. B 58, 267–288 (1996)
14. Shirish, K.S., Sathiyakeerthi, S.: A simple and efficient algorithm for gene selection using sparse logistic regression. Bioinformatics 19(17), 2246–2253 (2003)
15. Murofushi, T., Sugeno, M., Machida, M.: Non Monotonic Fuzzy Measures and the Choquet integral. Fuzzy Sets and Systems 64, 73–86 (1994)
16. Grabisch, M., Murofushi, T., Sugeno, M. (editors):Fuzzy Measures and Integrals: Theory and Applications. Physica-Verlag (2000)
17. Merz, C., Murphy, P.: UCI Repository of Machine Learning, Databases (1996), `ftp://ftp.ics.uci.edu/pub/machine-learning-databases`

Function of EEG Temporal Complexity Analysis in Neural Activities Measurement

Xiuquan Li[1], Zhidong Deng[1,*], and Jianwei Zhang[2]

[1] State Key Laboratory of Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science, Tsinghua University, Beijing 100084, China
lixq06@mails.tsinghua.edu.cn, michael@tsinghua.edu.cn
[2] TAMS - Technical Aspects of Multimodal Systems
Department of Informatics, Hamburg University
Vogt-Koelln-Strasse 30, 22527 Hamburg, Germany
zhang@informatik.uni-hanburg.de

Abstract. We investigate the correlation between temporal complexity of EEG signal and the underlining neural activities. Fractal geometry has been proved useful in quantifying complexities of dynamical signals. Temporal fractal dimension of EEG signals provides a new neurophysiological measure. In order to better understand what the complexity measure reveals about the underling brain process, a further exploration on the neuronal generators of fractal geometry characteristics of EEG is conducted in this study. Our investigation suggests that the temporal fractal measure of EEG signals can be related to the activity diversity of neuronal population activities. The complexity measure also gives an indication on the change in synchronization state under certain mental conditions. These assumptions are supported by experimental evidence from the visual cortex and sensorimotor cortex. This work helps give an interpretation of the obtained results of the temporal complexity analysis on EEG signals and may be useful in further investigating the covert steps of brain information processing.

Keywords: Electroencephalography(EEG), Event-related potentials(ERPs), Temporal complexity, Fractal analysis, Event-related desynchronization (ERD).

1 Introduction

Electroencephalography (EEG) is the electrical signal recorded from the surface of the scalp, produced as a result of the synthesis of electric fields emitted by individual neurons in the neuronal mass activity. When appropriately processed, they have the potential to facilitate the understanding of brain mechanisms and neurocognitive processes [1]. For EEG signal processing, various data analysis techniques have been proposed, such as time-frequency analysis, event-related potentials (ERPs), event-related desynchronization/synchronization (ERD/ERS), and so on. At the same time, the investigation on the nonlinear aspect of brain activities has also occupied numerous researchers in the neuroscience field.

* Corresponding author.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 209–218, 2009.

As an effective tool for the characterization of nonlinear signals, deterministic chaos plays an important role. A straightforward method to distinguish different mental states from recorded EEG signals through a dynamic perspective is to estimate nonlinear parameters of the dynamic system, like correlation dimension of the attractor, Lyapunov exponent, etc. These attractor-based methods mainly work based on the phase space reconstruction procedure according to Taken's theorem [2]. These measures have been shown to be useful for differentiating dynamic properties of neuronal networks [3-5]. Babloyants and his co-researchers analyzed the EEG data of the human brain during the sleep cycle and suggested the existence of chaotic attractors for sleep stages two and four [6]. Significant differences have been reported in these nonlinear measures between different mental states [7, 8, 9].

However, there have also been heated debates about the preconditions of these attractor-based methods for EEG analysis [10]. There were experts who argued that some early 'demonstrations' of deterministic chaos based on weak experimental evidence were accepted without sufficient analysis. They also presented their mathematic and neural physiologic evidence to show that it was not sufficient to prove that the EEG signal can represent the observation values of strange attractors [11, 12]. They did not support the attempts to identify strange attractors in brain signals and to measure their parameters.

Regardless of the dispute about the existence of strange attractors in the behavior of the EEG signals, there is no doubt that the neurosciences should benefit greatly from nonlinear science. EEG signals are nonlinear in themselves. Recent advancements in nonlinear science are expected to provide more insight into the underlining neural activities and thus further progress neuroscience research. Alternative methods have been developed in recent years for the analysis of neural activity.

Fractal geometry has been proved useful in quantifying the complexity of dynamical signals in biological systems [13-14]. Fractal analysis has also been successfully applied in quantifying the temporal complexity of dynamic fluctuations in time-series signals [15-17]. Thus the temporal fractal property provides a good measure for characterizing the nonlinear behavior of the EEG signals as it describes their irregular shape. V. Cabukovski et al. developed a real-time method employing fractal character and gave their results in real-time EEG data analysis; Liu et al. found that the temporal fractal dimension of EEG signals during different physiological stages of a handgrip task showed different behaviors, and increased linearly with hand grip force during the movement and holding periods; Bashashati et al. used fractal dimension as a feature to build a biofeedback system in a brain computer interface system. These results strongly suggest that the morphological aspects of the EEG can shed light on the activity change of the neuronal population.

The temporal geometrical property provides a promising way to explore the nonlinear characteristic of the EEG. It does not depend on the desired attractor. Besides, it does not rely on phase space reconstruction, which is critical for attractor-based measures such as correlation dimension and the largest Lyapunov exponent. Thus it provides an effective tool for measuring the neural activities in neuroscience research. However, it is still not totally clear what the complexity measure could possibly reveal about the underlining neural processing. In this paper, we further explore the correlations between the temporal complexity aspect of the EEG and the neuronal population activities. The neuronal generators of the fractal property of the EEG

signal are discussed. This work may help to give an interpretation of the obtained experimental results and facilitate investigating the mechanism of brain information processing.

2 Methods

Fractals are mathematical sets with a high degree of geometrical complexity that can model many natural phenomena. Fractal features represent the morphology of the signals. It has been well established that fractal analysis can effectively and quantitatively characterize the temporal complexity properties of a nonlinear signal [13-18]. The temporal fractal property can be evaluated in different ways. The Higuchi method gives a good approximation to the fractal dimension (FD), using the length of the irregular curve, from a small number of points [19]. This method is applicable and may be a better solution for fractal dimension estimation in the case of EEG data analysis, where the small number of points is the main limitation [15, 16]. It can be formulated as

Given a discrete time series observations $X(n)=\{X(1),X(2),\cdots,X(N)\}$, a new time series X_k^m can be constructed as

$$X_k^m; X(m), X(m+k), X(m+2k),\cdots, X\left(m+\left[\frac{N-m}{k}\right]\cdot k\right), (m=1,2,\cdots,k)$$

where [ ] denotes the Gauss's notation and both m and k are integers, indicating the initial time and the interval time, respectively. The length of the curve X_k^m can be defined as follows,

$$L_m(k)=\frac{N-1}{\left[\frac{N-m}{k}\right]*k}*\left(\sum_{i=1}^{\left[\frac{N-m}{k}\right]}\left|x(m+i*k)-x(m+(i-1)*k)\right|\right). \quad (1)$$

The term $N-1/[(N-m)/k]\cdot k$ represents the normalization factor for the curve length of the subset time series.

The length of the curve for the time interval k, $\langle L(k)\rangle$, is defined as the average value over k sets of $L_m(k)$, i.e.,

$$\langle L(k)\rangle=\left[\sum_{m=1}^{k}L_m(k)\right]/k.$$

If $\langle L(k)\rangle \propto k^{-D_H}$, then the curve $X(n)$ is fractal with the dimension D_H.

The length of the optimal epoch, denoted by N in (1), needs to be decided when estimating the fractal dimension for EEG signals using the Higuchi method. A long epoch helps to get stable estimation for actual fractal dimension. In the meantime, long epochs are inevitably liable to submerge the subtle change in the fractal property

and thus reduce the temporal resolution of the analysis. So this is a case-dependant problem where the stability and temporal resolution should be balanced. Analysis has been conducted in [16] using fractal sets of the Henon map, the Modified Sierpinski triangle and the Weierstrass cosine function. They reported that an epoch size between 100 and 150 points is enough for a reliable estimation of the fractal dimension.

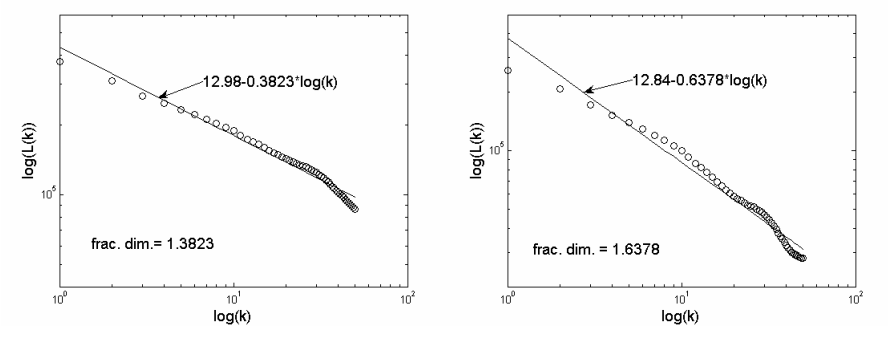
3 Experiments

3.1 Temporal FD and Alpha Blocking Phenomenon

Neuronal networks can display different states of synchrony, with oscillations at different frequencies. An evident alpha rhythms synchronization phenomenon in the visual cortical area can be observed in the visual information processing [20, 21]. To explore the neuronal generators of the temporal complexity measure, we first conducted visual cortex experiments to investigate how the fractal property will change with the alternating of different mental conditions.

The EEG signal was recorded at electrode O1 in the occipital area according to the extended international 10-20 system, using an EEG machine provided by the Biosemi Company. The sampling rate was 1024Hz, and no filter was applied to the sampled data. Then the recorded single trial EEG signals were analyzed in our study. Obvious difference in the fractal property of the EEG was observed for close/open eyes conditions. For illustration, the estimated fractal dimension can be 1.3823 and 1.6378, respectively, for the two conditions as shown in Fig. 1. There is significant increase in signal complexity when the eyes opened.

Fig. 2 shows the time course of the complexity of the single trial EEG signal using a sliding window scheme, where the window length is 2s and the step length is 0.1s. The character 'O' indicates the open-eyes periods and 'C' denotes closed-eyes periods. Significant and regular fluctuation can be observed while the two condition periods alternate.



(a) (b)

Fig. 1. Fractal analysis result for (a) closed-eyes conditions (b) open-eyes conditions

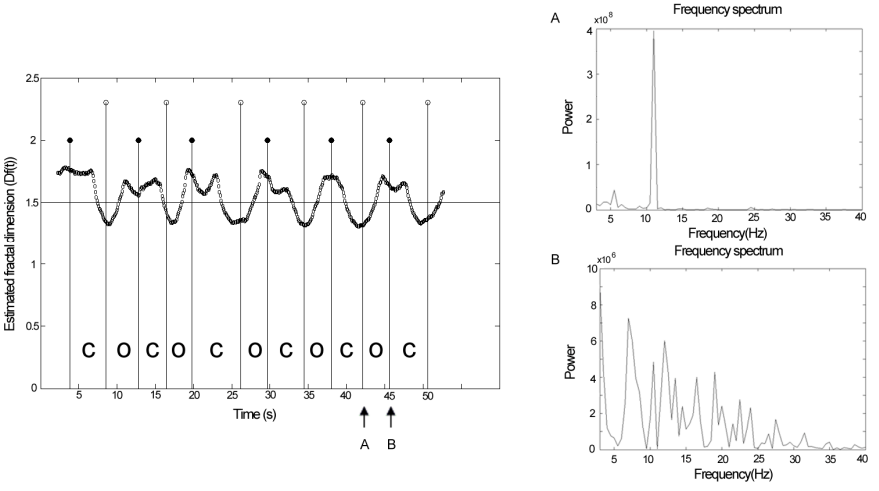


Fig. 2. Time course of the estimated temporal complexity measure

A theoretical approach to brain function is based on Hebb's concept of cell assemblies [22]. According to this view, strongly coupled neuronal groups are the functional units of the brain. The amplitude of oscillations is proportional to the number of synchronously active neural elements [23].

We can see obvious alpha rhythm synchronization phenomenon in the experimental result from Fig. 2 (A). This may imply that, according to the cell assembly theory, when the eyes close, more oscillating cell assemblies change their oscillation patterns to a predominant one, leading to great enhancement in the amplitude of the corresponding frequency component. The appearance of predominant or governing factors in neural activity patterns makes EEG signals more regular in the morphological aspect and thus reduces its temporal complexity. So the estimated fractal dimension is relatively low as indicated by 'A' in the left subfigure of Fig. 2. In other words, a lower fractal dimension of EEG signal may relate to a decrease of the diversity of oscillation patterns of mass neuronal activities in the visual cortical area.

On the other hand, when the eyes open, the task of processing large amount of visual information received may require more oscillating cell assemblies to change their oscillation patterns away from the preceding predominant one. It leads to the attenuation or disappearance of the predominant factors in frequency spectrum, the energy is distributed more equally to various frequency components as illustrated in Fig.2 (B). This change eventually enhances the fractal property of the EEG signal as indicated by 'B' in the left subfigure of Fig. 2. The experimental results suggest that the temporal fractal properties of the EEG signal reflect effectively such changes in the oscillation diversity of neuronal populations.

3.2 Temporal FD and ERD

The event-related desynchronization/synchronization (ERD/ERS) is a high-frequency band-specific technique. Frequency specific changes of the ongoing EEG activity in

event-related phenomenon may consist either of decreases or increases of power in given frequency bands. This may be due to decrease or increase of the synchrony of the underlying neuronal populations. ERD/ERS offers useful information about the neural dynamics in processing mental tasks through frequency-specific monitoring of the brain oscillations [20].

As suggested in the preceding visual cortex experiments, the temporal fractal dimension can effectively reflect the change in the diversity of the oscillation patterns of neural activities. If this assumption holds, the frequency-specific synchronization phenomena, revealed by the ERD/ERS method, should also come within the purview of temporal complexity measuring. Because ERD indicates decrease of the strength of a specific oscillation pattern, cell assembly oscillations change from this pattern to more diverse ones. This will give rise to a reorganizing process among the power of frequency components of the ongoing EEG and will thus affect the temporal complexity measurement.

To testify this hypothesis, we investigated the temporal FD properties in motor imagery (MI) tasks. Hand motor imagery lead to perturbations of the ongoing pericentral rhythm, demonstrating typical ERD/ERS phenomena. Cortical activation related to movement preparation and execution has been shown to desynchronize the mu rhythm and results in an event-related desynchronization (ERD). ERD in hand movement is more prominent over the contralateral sensorimotor areas during motor preparation and extends bilaterally after movement initiation [20, 25, 26].

The dataset we employed for analysis is the dataset IVa of BCI competition III. More detail about the experimental design and EEG recording can be found in [24]. Fig. 3 and Fig. 4 show the analysis results in right hand imagining task for subject al and ay in the datasets, respectively. The time courses of the power of 8-12Hz frequency band are illustrated in Fig. 3 (a) and Fig. 4(a). They are percentage values relative to a baseline of -1s to 0s before the onset of the visual cue, which is indicated by the vertical line in the figures. Significant contralateral pre-movement mu ERD can be seen at electrode C3 for right hand imagination, as denoted by arrows. Fig. 3(b) and Fig. 4(b) display the averaged ERD spatial mappings of the right hand MI task, averaged between 1s-2s after the appearance of the visual cue. The prominent ERD effects can be observed over the left sensorimotor areas for right hand MI tasks, as indicated by the white crosses.

The preparation of movement induces a decrease in the power of mu band, represented by the ERD, over the motor cortex. In this process, more cell assemblies may change their oscillation patterns away from the preceding mu synchrony, resulting in an increase of the diversity of the neuronal population activities. This transformation should enhance the complexity of the signal according to our assumption.

The time courses of the change in the temporal complexity property are shown in Fig. 3(c) and Fig. 4(c), measured by fractal dimension difference relative to a baseline of -1s to 0s before the onset of the visual cue. A sliding window scheme is also employed here with the window length of 1s and the step length of 0.1s. The topography mappings of the temporal fractal dimensions variation are shown in Fig. 3(d) and Fig. 4(d). Evident negative correlations are observed in our experimental results, which strongly support our hypothesis on the correlation between temporal complexity and ERD.

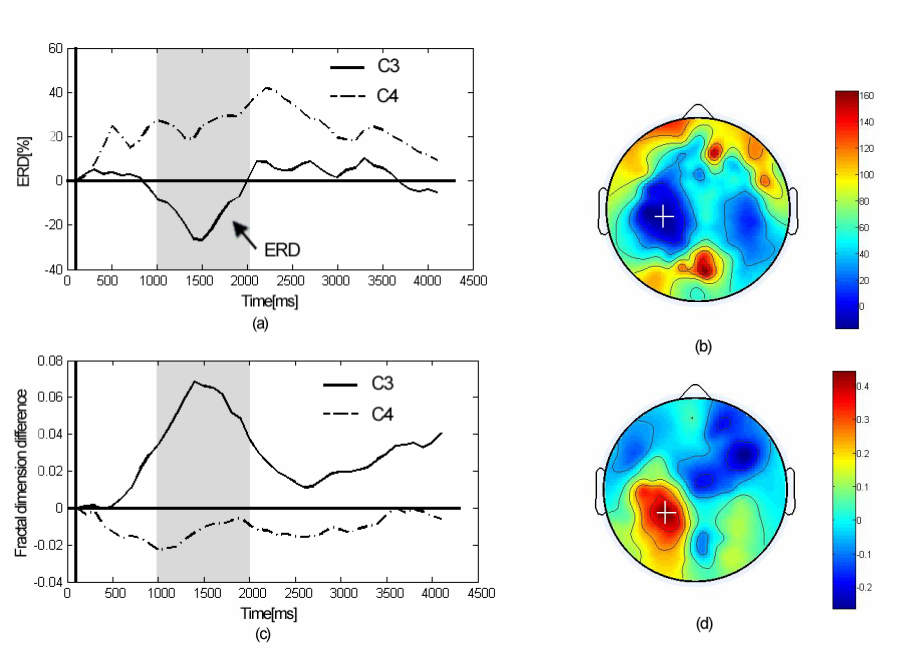


Fig. 3. ERD and temporal complexity variation in hand movement imagination from subject al

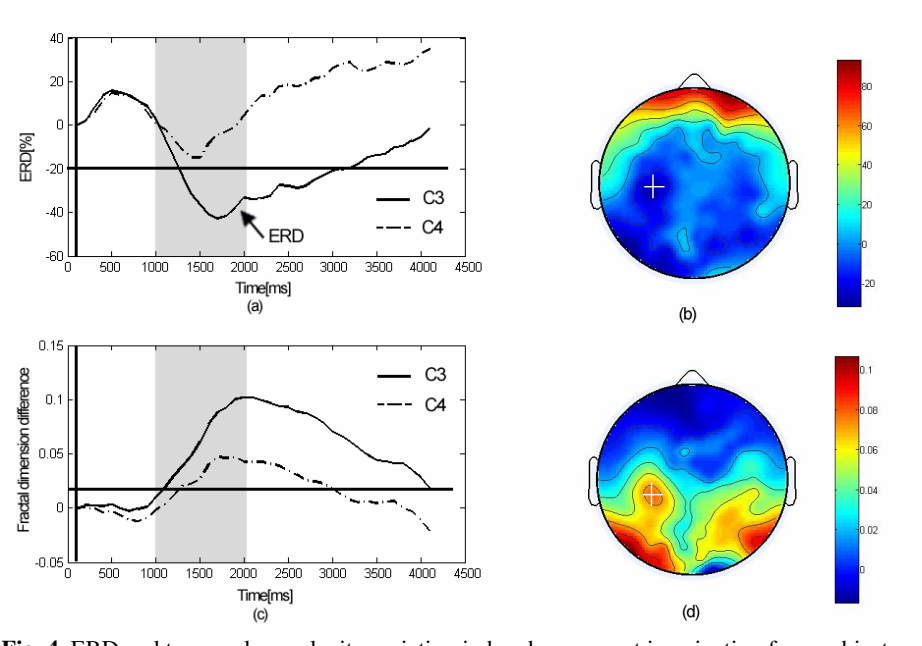


Fig. 4. ERD and temporal complexity variation in hand movement imagination from subject ay

4 Discussions and Conclusions

Nonlinear mechanisms are crucial in neural systems and the temporal fractal property provides a good measure of the nonlinear behavior of the EEG signals by describing their irregular shape. The temporal fractal dimension has advantages over the attractor's fractal dimension in that it depends neither on the desired attractor assumption nor on the phase space reconstruction process.

In this study, we discussed the correlation between the temporal fractal characters of the EEG signal and the underlying neural activities, trying to give an assumption on the neuronal generators of the temporal complexity properties of EEG signals.

EEG signals are believed to be synthesized from individual cell assembly activities. When most of the neuronal groups within a cortex area oscillate at a certain frequency, the neuronal networks display a state of synchrony, for example in the alpha blocking phenomenon with eyes closing. The resultant EEG signal is regular with a lower temporal complexity degree. On the other hand, when a more complex task is executed, for example in preparing the hand movement, more cell assemblies change their oscillation patterns away from the preceding consensus rhythm to fulfill the mental task. The power of the preceding consensus rhythm is reduced, as can be revealed by ERD. At the same time, the transformation in the oscillation patterns inevitably causes a significant enhancement in the diversity of the neuronal population activities, and the overall energy is distributed over more frequency components, leading to an increase in the temporal complexity of the EEG signal.

The experimental results indicate that the temporal complexity properties of EEG possibly relate to the diversity of the neuronal population activities. The change in the diversity of mass neuron activities may induce significant variations in the morphological aspect of the EEG signals. These morphological differences can be picked up and employed by the temporal complexity measure to provide a physical measure of human brain activities. This work may help to better understand the way of measuring the internal states of neural activities through a temporal morphologic view.

Acknowledgments. This work was supported in part by the National Science Foundation of China (NSFC) grant 60621062 and 60775040 and the DFG German Research Foundation (grant #1247) – International Research Training Group CINACS (Cross-modal Interactions in Natural and Artificial Cognitive Systems). Great thanks to Dan Zhang for his informative discussion.

References

1. Epstein, C.M.: Introduction to EEG and Evoked Potentials. Lippincott Williams & Wilkins, Philadelphia (1983)
2. Takens, F.: Detecting Strange Attractors in Turbulence. In: Rand, D.A., Young, L.S. (eds.) Dynamical Systems and Turbulence, Warwick. LNM, vol. 898, pp. 361–381. Springer, Heidelberg (1981)
3. Faure, P., Korn, H.: Is There Chaos in the Brain? I. Concepts of Nonlinear Dynamics and Methods of Investigation. C. R. Acad. Sci. Paris, Ser. III 324, 773–793 (2001)

4. Korn, H., Faure, P.: Is There Chaos in the Brain? II. Experimental Evidence and Related Models. C. R. Biologies 326, 787–840 (2003)
5. Lutzenberger, W., Preissl, H., Pulvermiiller, F.: Fractal Dimension of Electroencephalographic Time Series and Underlying Brain Processes. Biological Cybernetics 73, 477–482 (1995)
6. Babloyantz, A., Salazar, J.M., Nicolis, G.: Evidence of Chaotic Dynamics of Brain Activity During the Sleep Cycle. Phys. Lett. A 111, 152–156 (1985)
7. Kannathal, N., Acharya, R.U., Alias, F., Tibelong, T., Sadasivan, P.K.: Nonlinear Analysis of EEG Signals at Different Mmental States. Biomedical Online Journal 16, 3–7 (2004)
8. Soe, N.N., Nakagawa, M.: Chaos and Fractal Analysis of Electroencephalogram Signals during Different Imaginary Motor Movement Tasks. Journal of the Physical Society of Japan 77, 1–8 (2008)
9. Jing, H., Takigawa, M., Benasich, A.A.: Relationship of Nonlinear Analysis, MRI and SPECT in the Lateralization of Temporal Lobe Epilepsy. European Neurology 48, 11–19 (2002)
10. Preissl, H., Aertsen, A.: Reconstruction and Characterization of Neuronal Dynamics: How Attractive is Chaos? In: Information Processing in the Cortex: Experiments and Theory, pp. 285–297. Springer, Heidelberg (1992)
11. Theiler, J., Rapp, P.E.: Re-examination of the Evidence for Low-dimensional Non-linear Structure in the Human Electroencephalogram EEG. Clin. Neurophysiol. 98, 213–222 (1996)
12. Mandell, A.J., Slez, K.A.: The Impact of Chaos on Science and Society. Hunan Science & Technology Press, Changsha (2001)
13. Fernandez, E., Jelinek, H.F.: Use of Fractal Theory in Neuroscience: Methods, Advantages, and Potential Problems. Methods 24, 309–321 (2001)
14. Heymans, O., Fissette, J., Vico, P., Blacher, S., Masset, D., Brouers, F.: Is Fractal Geometry Useful in Medicine and Biomedical Sciences? Med. Hypotheses 54, 360–366 (2000)
15. Bashashati, A., Ward, R.K., Birch, G.E., Hashemi, M.R., Khalilzadeh, M.A.: Fractal Dimension-Based EEG Biofeedback System. In: 25th Annual International Conference of the IEEE EMBS, pp. 17–23. IEEE Press, New York (2003)
16. Cabukovski, V., Rudolf, N.M., Mahmood, N.: Measuring the Fractal Dimension of EEG Signals: Selection and Adaptation of Method for Real-time Analysis. Transactions on Biomedicine and Health 1, 285–292 (1993)
17. Liu, J.Z., Yang, Q., Yao, B., Brown, R.W., Yue, G.H.: Linear Correlation between Fractal Dimension of EEG Signal and Handgrip Force. Biological Cybernetics 93, 131–140 (2005)
18. Mandelbrot, B.B.: The Fractal Geometry of Nature. W. H. Freeman, New York (1983)
19. Higuchi, T.: Approach to an Irregular Time Series on the Basis of Fractal Theory. Physica D 31, 277–283 (1988)
20. Pfurtscheller, G., Lopes da Silva, F.H.: Event-related EEG/MEG Synchronization and Desynchronization: Basic Principles. Clinical Neurophysiology 110, 1842–1857 (1999)
21. Dewan, E.M.: Occipital Alpha Rhythm Eye Position and Lens Accommodation. Nature 214, 975–977 (1967)
22. Hebb, D.O.: The Organization of Behavior: A Neuropsychological Theory. Wiley, New York (1949)
23. Elul, R.: The Genesis of the EEG. International Review of Neurobiology 15, 227–272 (1972)

24. Dornhege, G., Blankertz, B., Curio, G., Müller, K.R.: Boosting Bit Rates in Non-invasive EEG Single-trial Classifications by Feature Combination and Multi-class Paradigms. IEEE Trans. Biomed. Eng. 51, 993–1002 (2004)
25. Zhang, D., Wang, Y.J., Gao, X.R., Hong, B., Gao, S.K.: An Algorithm for Idle-State Detection in Motor-Imagery-Based Brain-Computer Interface. Computational Intelligence and Neuroscience 2007, Article ID 39714, 9 (2007)
26. Chatrian, G.E., Magnus, C.P., Lazarte, J.A.: The Blocking of the Rolandic Wicket Rhythm and Some Central Changes Related to Movement. Electroencephalogr. Clin. Neurophysiol. 11, 497–510 (1959)

A PDF-Matched Modification to Stone's Measure of Predictability for Blind Source Separation

Mahdi Khosravy[1], Mohammad Reza Alsharif[1,⋆], and Katsumi Yamashita[2]

[1] Department of Information Engineering, Faculty of Engineering, University of the Ryukyus, 1 Senbaru, Nishihara, Okinawa 903-0213, Japan
k078661@eve.u-ryukyu.ac.jp,asharif@ie.u-ryukyu.ac.jp
[2] Graduate School of Engineering, Osaka Prefecture University, 1-1 Gakuen-cho, Sakai, Osaka, Japan
yamashita@eis.osakafu-u.ac.jp

Abstract. This paper presents a PDF-matched modification to Stone's measure of predictability. The modified measure of predictability is a measure of non-gaussianity too. It is an extent of signal predictability by two different prediction terms. One prediction term is based on a normal gaussian PDF assumption for signal. In contrast, the other one is based on a unit variance supergaussian PDF assumption for signal. By contrastive deployment of the above prediction terms, the modified measure of predictability enables BSS to follow a high kurtosis PDF assumption for signals. As an advantage, not only signals with maximized predictability are recovered, but also with increased non-gaussianity too. Deploying the modified measure of predictability concludes more independent recovered signals. The dominance of BSS based on the modified measure to the previous one has been demonstrated by many tests performed over mixtures of realistic audio signals (music and speech) and over mixtures of gray-scale images.

Keywords: Blind source separation, Gaussianity, Eigenvalue routine, Predictability, Short-term linear predictors.

1 Introduction

Blind Source Separation (BSS) [1,2] is separation of blind sources from mixed observed data without any prior information of the source signals and mixing process. As basic model of BSS, K unknown source signals upon transmission through a medium have been linearly mixed together and mixture signals are collected by M sensors. The source signals are not observed and also no information is available about the mixture. In mathematical model, the $M \times 1$ vector $\boldsymbol{x}$ of observed signals $\boldsymbol{x}(n) = (x_1(n)\ x_2(n)\ \cdots\ x_M(n))^T$ is multiplication of $K \times 1$

⋆ Also, Adjunct Professor at School of Electrical and Computer Engineering, University of Tehran.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 219–228, 2009.

vector $\mathbf{s}$ of unknown source signals $\boldsymbol{s}(n) = (s_1(n)\ s_2(n)\ \cdots\ s_K(n))^T$ by unknown $M \times K$ mixing matrix $\boldsymbol{A}$:

$$\boldsymbol{x}(n) = \boldsymbol{A} \times \boldsymbol{s}(n) \tag{1}$$

Given only the observation signals $\boldsymbol{x}(n) = (x_1(n)\ x_2(n)\ \cdots\ x_M(n))^T$, the solution of BSS problem seeks for the best $K \times M$ matrix W as un-mixing matrix to extract signals as much as possible close to unknown source signals as follows

$$\boldsymbol{y}(n) = \boldsymbol{W} \times \boldsymbol{x}(n) \tag{2}$$

where $\boldsymbol{y}$ is $M \times 1$ vector of extracted signals $\boldsymbol{y}(n) = (y_1(n)\ y_2(n)\ \cdots\ y_K(n))^T$. Each row of $\boldsymbol{W}$ is a $1 \times M$ un-mixing vector $\boldsymbol{w}_i$ related to one of extracted signals.

Generally, BSS methods aim to find a separation matrix which minimizes the mixing effect on some signals properties. The desired signal property depends on the BSS method. One of BSS strategies is projection pursuit [3] that is based on the central limit theorem (CLT) [4] and seeks a weight vector such that the signal extracted from a set of signal mixtures is as non-gaussian as possible. A BSS method like Independent Component Analysis (ICA) looks for maximization of statistical independence between signals [5]. ICA based on Infomax [6] maximizes entropy (uniformity) of distribution of signals instead of their independence and it achieves the same result.

Stone has introduced the BSS method based on temporal predictability of signal for instantaneous mixtures [7,8]. The Stone's BSS has been extended to convolutive [9] as well as nonlinear mixtures [10]. Also a probabilistic short term version of Stone's BSS has been presented in [11]. Despite most of BSS methods which are suffered by their computational complexity, Stone's BSS has a fast algorithm. It has scaling characteristics of $O(M^3)$, where M is the number of signal mixtures. Its extended version to complex domain has been used in [12] for reducing complexity of ICA based MIMO-OFDM receiver. However, compared to projection pursuit and ICA, it has lower separation performance.

In this paper, we propose a modification to Stone's measure of predictability to increase the separation performance of BSS. The modified measure is not only a measure of predictability, but also it is a measure of non-gaussianity too. BSS based on the modified measure extracts signals with maximized predictability as well as increased non-gaussianity. Thus, the recovered signals are more independent, and the modified measure concludes a BSS with higher separation performance. Its dominance to Stone's BSS has been derived over mixtures of realistic audio signals (music and speech) and over mixtures of gray-scale images.

The structure of the paper is as follows: Section 2 presents the proposed modification to Stone's measure of predictability. This section first explains the Stone's measure. Then its modification is presented. Section 2.3 establishes BSS based on the modified measure. Section 3 discusses the performance of the modified Stone's BSS in separation of signal mixtures and image mixtures made of different sources. Finally section 4 concludes the paper.

2 The PDF-Matched Measure of Predictability

Stone has presented the predictability conjecture of BSS [7], and it has been modified by Xie [13]. This conjecture indicates that any mixture signal has a predictability lying between the least and the most predictability values of the source signals. Stone defines a measure of temporal predictability of signal that has been used as merit function for BSS. Here we propose a probabilistic modification to Stone's measure of predictability.

2.1 Stone's Measure of Predictability

The measure of temporal predictability for a N-sampled signal z has been defined as

$$F_{Stone}(z) = log\frac{V_z}{U_z} = log\frac{\sum_{n=1}^{N}(z_{long}(n) - z(n))^2}{\sum_{n=1}^{N}(z_{short}(n) - z(n))^2} \tag{3}$$

where $z(n)$ is the value of the signal at sample n. The term U_z reflects the extent to which $z(n)$ is predicted by a short-term moving average z_{short} of values in z. In contrast, the term V_z is a measure of the overall variability in z, as measured by the extent to which $z(n)$ is predicted by a long-term moving average z_{long} of values in z. The long-term predicted value $z_{long}(n)$ and the short-term predicted value $z_{short}(n)$ of $z(n)$ are both exponentially weighted sums of signal values measured up to sample $(n-1)$, such that recent values have larger weighting than those in the distant past. As an essential requirement, the long-term prediction z_{long} must be obtained over much longer averaging window than the short-term prediction z_{short}, typically at least 100 times longer.

2.2 The Proposed Measure of Predictability

In Stone's measure of predictability the short term prediction mean squared error U_z is the main part. The other term V_z is used for to be insured that by maximization of the measure, the recovered signals not to be so predictable as to be constant. While the long-term prediction error is not accurately the variance of the signal, it is used to bring some amount of variance for extracted signals.

Here by using a probabilistic view to short-term prediction we presents a modification to Stone's measure of predictability. The proposed measure wherein the coefficients of short-term predictors are obtained by a probabilistic objective method, concludes higher performance of BSS. The modified measure of predictability for an N-sampled signal z is defined as

$$F_{proposed}(z) = log\frac{cov(\widetilde{e_z}, \widetilde{e_z})}{cov(\widehat{e_z}, \widehat{e_z})} = log\frac{\sum_{n=1}^{N}(\widetilde{z}(n) - z(n))^2}{\sum_{n=1}^{N}(\widehat{z}(n) - z(n))^2} \tag{4}$$

where $\widehat{z}(n)$ is the short-term predicted value of $z(n)$ by a unit variance super-gaussian PDF assumption for z, and $\widetilde{z}(n)$ is the short-term predicted value of $z(n)$ by a normal gaussian PDF assumption for z. The terms $\widetilde{e_z}(n)$ and $\widehat{e_z}(n)$ are

error signals of these prediction signals, which their covariances are respectively considered as measures of gaussianity and non-gaussianity of z. The $F_{proposed}$ is arranged so that it gives us not only a measure of predictability of the signal but also its non-gaussianity. $\widetilde{z}(n)$ and $\widehat{z}(n)$ are obtained as follows

$$\widetilde{z}(n) = \sum_{i=1}^{L} \widetilde{a}_i z(n-i) \tag{5}$$

$$\widehat{z}(n) = \sum_{i=1}^{L} \widehat{a}_i z(n-i) \tag{6}$$

where $\widetilde{a}_i$ and $\widehat{a}_i$ are coefficients of the short-term predictors, and L is the length of the predictors. In a probabilistic view to Eqs.(5)(6), $\widetilde{z}(n)$ and $\widehat{z}(n)$ can be considered as expectation of $z(n-i)$ with respect to i, and $\widetilde{a}_i$ and $\widehat{a}_i$ respectively as probability weight of $z(n-i)$ in linear combination of $\widetilde{z}(n)$ and $\widehat{z}(n)$. We select the coefficient $\widetilde{a}_i$ proportional to $\widetilde{p}_i$ that is the conditional probability of on-coming current sample of the signal to be equal to its i^{th} prior sample with the precondition of assuming a gaussian PDF for z. The coefficient $\widehat{a}_i$ is selected in the same way, but the corresponding conditional probability $\widehat{p}_i$ is obtained by the precondition of assuming a supergaussian PDF for z.

$$\widetilde{p}_i = Prob.\{\ \{z(n) = z(n-i)\} \mid \{\text{PDF of } z \text{ is gaussian}\}\ \} \tag{7}$$

$$\widehat{p}_i = Prob.\{\ \{z(n) = z(n-i)\} \mid \{\text{PDF of } z \text{ is supergaussian}\}\ \} \tag{8}$$

Evidently $\widetilde{p}_i$ and $\widehat{p}_i$ are the most for the closest sample to the current sample of signal, and by going further through the past samples, they are gradually decreased. For obtaining decreasing function of $\widetilde{p}_i$ and $\widehat{p}_i$, we have used respectively the assumed corresponding normal gaussian PDF and unit variance supergaussian PDF for z. The Laplacian distribution has been used as supergaussian PDF. The deployed PDFs are as follows

$$\widetilde{f}_z(z) = \frac{1}{\sqrt{2\pi}} exp\left(-\frac{z^2}{2}\right) \tag{9}$$

$$\widehat{f}_z(z) = \frac{1}{2} exp\left(-|z|\right) \tag{10}$$

Fig.1 shows the above PDFs. The same PDF is approximately considered for short-length segments of the signal under prediction window, and the required probabilities are obtained by sampling from the PDF. L probability values are sampled in the distance from the peak-point of PDF to its tail-end point that can be a point like p-value of .001. the $\widetilde{p}_i$ and $\widehat{p}_i$ are obtained as follows

$$\widetilde{p}_i = \frac{1}{\sqrt{2\pi}} e^{-\frac{(i-1)^2 {E_1}^2}{2(L-1)^2}} \quad (i = 1, \cdots, L) \tag{11}$$

$$\widehat{p}_i = \frac{1}{2} e^{-(i-1)\frac{E_2}{L-1}}, \quad (i = 1, \cdots, L) \tag{12}$$

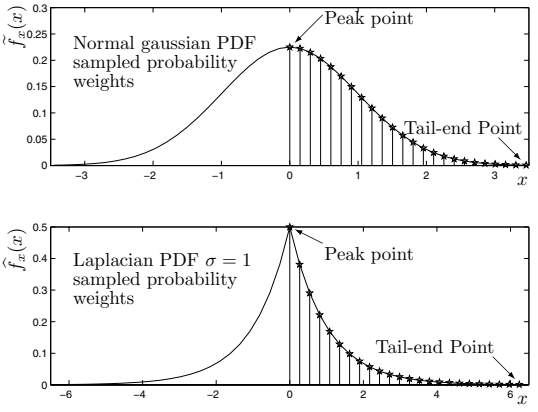


Fig. 1. PDF-matched selection of coefficients of short-term predictors by picking the probability weights through sampling the assumed PDFs in the interval from the peak point to the tail-end point

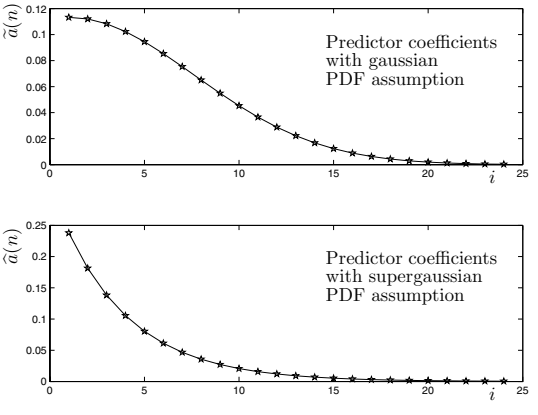


Fig. 2. PDF-matched obtained coefficients for short-term predictors

where E_1 and E_2 are tail-end points of PDFs. $\frac{E_1}{L-1}$ and $\frac{E_2}{L-1}$ are sampling periods in which L is the length of prediction window. The summation of obtained probabilities is normalized to one to conclude probability weights as well as predictors coefficients. Fig.1 shows the PDF-matched selection of coefficients for 24-points length of prediction window ($L = 24$). Fig.2 shows the PDF-matched obtained coefficients for short term predictors.

The proposed measure is not only a measure of predictability, but also it is a measure of non-gaussianity too. It leads BSS to an advantage beside deploying temporal predictability maximization. This advantage is that by maximizing the proposed measure of predictability of a set of signals, their non-gaussianity is increased too. As we expect central limit theorem (CLT)[4], increasing the nongaussianity leads to more independent recovered signals.

2.3 BSS Based on the Proposed Measure of Predictability

Consider the recovered signal $y_i(n)$ formed by applying the separating vector $\boldsymbol{w}_i$ to the set of M mixtures; $y_i(n) = \boldsymbol{w}_i \boldsymbol{x}(n)$. Since $y_i(n) \in LS(x_1(n), x_2(n), \cdots, x_M(n))$ that is the linear space spanned by mixture signals, the proposed measure in Eq.(4) for $y_i(n)$ can be written as

$$F_{proposed}(y_i) = log\frac{cov(\widetilde{e}_{y_i}, \widetilde{e}_{y_i})}{cov(\widehat{e}_{y_i}, \widehat{e}_{y_i})} = log\frac{\boldsymbol{w_i}\widetilde{C}\boldsymbol{w_i}^T}{\boldsymbol{w_i}\widehat{C}\boldsymbol{w_i}^T}. \tag{13}$$

The matrices $\widetilde{C}$ and $\widehat{C}$ are obtained as follows

$$\widetilde{C} = cov(\widetilde{\mathbf{e}}_x, \widetilde{\mathbf{e}}_x) \tag{14}$$

$$\widehat{C} = cov(\widehat{\mathbf{e}}_x, \widehat{\mathbf{e}}_x) \tag{15}$$

where $\widetilde{\mathbf{e}}_x(n) = (\widetilde{e}_{x_1}(n), \widetilde{e}_{x_2}(n), \cdots, \widetilde{e}_{x_M}(n))^T$ and $\widehat{\mathbf{e}}_x(n) = (\widehat{e}_{x_1}(n), \widehat{e}_{x_2}(n), \cdots, \widehat{e}_{x_M}(n))^T$, that $\widetilde{e}_{x_j}(n) = \widetilde{x}_j(n) - x_j(n)$ and $\widehat{e}_{x_j}(n) = \widehat{x}_j(n) - x_j(n)$. The signals $\widetilde{x}_j$ and $\widehat{x}_j$ are obtained by applying the two proposed PDF-matched predictors introduced in Eqs. (5)(6) to observed mixture signal x_j. The elements of $\widetilde{C}$ and $\widehat{C}$ for N-sampled mixture signals are respectively as follows

$$\widehat{C}(ij) = \sum_{n=1}^{N} ((x_i(n) - \widehat{x}_i(n))(x_j(n) - \widehat{x}_j(n))) \tag{16}$$

$$\widetilde{C}(ij) = \sum_{n=1}^{N} ((x_i(n) - \widetilde{x}_i(n))(x_j(n) - \widetilde{x}_j(n))). \tag{17}$$

So, the requirements for obtaining $\widetilde{C}$ and $\widehat{C}$ are available from mixture signals. These matrices should be computed only once. The following derivative form of the measure in Eq.(13) can be used for obtaining the $\boldsymbol{w}_i$ by gradient ascent learning;

$$\bigtriangledown_{\boldsymbol{w}_i} F_{proposed} = \frac{2\boldsymbol{w}_i}{cov(\widetilde{e}_i, \widetilde{e}_i)}\widetilde{C} - \frac{2\boldsymbol{w}_i}{cov(\widehat{e}_i, \widehat{e}_i)}\widehat{C}. \tag{18}$$

Also, the ratio of Eq.(13) has the form of Rayleigh quotient [13] which its extrema points correspond to eigensystem of the generalized eigenproblem. The point that the gradient of F is zero leads to the following equation:

$$\widehat{C}^{-1}\widetilde{C}\boldsymbol{w}_i = \boldsymbol{w}_i \left(\frac{cov(\widetilde{e}_i, \widetilde{e}_i)}{cov(\widehat{e}_i, \widehat{e}_i)} \times I\right) \tag{19}$$

where I is the identity matrix. It can be easily seen that the equation (19) has the form of a generalized eigenvalue problem. So, BSS solutions for $\boldsymbol{w}_i$ can be obtained as eigenvectors of $\widehat{C}^{-1}\widetilde{C}$ matrix with corresponding eigenvalues $d_i = \frac{cov(\widetilde{e}_i, \widetilde{e}_i)}{cov(\widehat{e}_i, \widehat{e}_i)}$. Therefore by solving the generalized eigenvalue routine, $\boldsymbol{W}$ composed of eigenvectors the same as un-mixing vectors $\boldsymbol{w}_i$ is obtained ($i = 1, 2, \ldots, M$). All source signals are simultaneously extracted by $\boldsymbol{W}$.

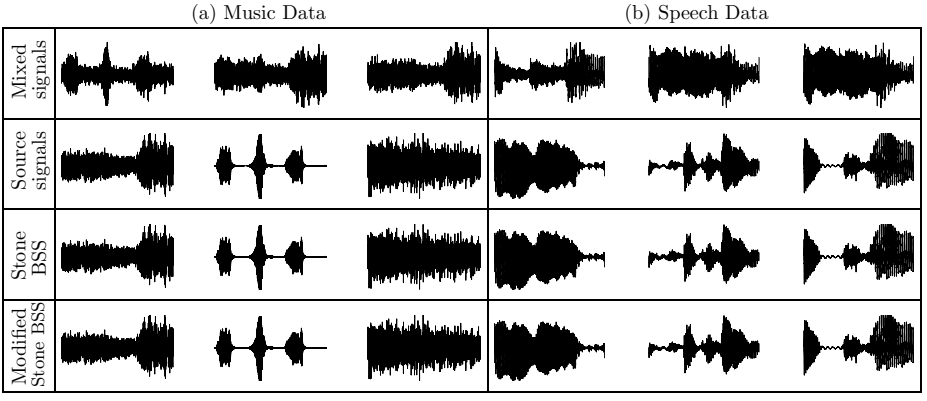


Fig. 3. (a) BSS experiment over music data, (b) BSS experiment over speech data, where respectively from up to down is as follows: mixed music signals, source music signals, recovered signals by Stone BSS and recovered signals by modified stone BSS

Table 1. Comparison of separation performances of Stone BSS and modified Stone BSS related to Fig.3; Over music data (a) and over speech data (b) by correlation values between recovered signals and their corresponding sources

(a) Music Data

Stone BSS

	y_1	y_2	y_3
s_1	**.9963**	.0041	.0863
s_2	.0005	**1.0000**	.0032
s_3	.0879	0.0011	**.9961**

Modified Stone BSS

	y_1	y_2	y_3
s_1	**.9999**	0.0001	0.0145
s_2	0.0042	**1.0000**	0.0045
s_3	0.0136	0.0023	**.9999**

(b) Speech Data

Stone BSS

	y_1	y_2	y_3
s_1	**.9986**	.0504	.0160
s_2	.0854	**.9960**	.0275
s_3	.0077	.0074	**.9999**

Modified Stone BSS

	y_1	y_2	y_3
s_1	**.9999**	.0041	.0077
s_2	.0457	**.9990**	.0243
s_3	.0040	.0122	**1.0000**

3 Results and Discussion

The performance of the BSS based on the modified measure of predictability has been driven with many evaluation tests over mixtures of audio signals (music and speech) and mixtures of gray-scale images. In our simulation the length of predictors is acquired 24 samples. We have compared the modified measure with Stone measure of predictability [7]. The sources, mixtures and recovered signals/images have been normalized to have the same maximum and minimum values. Also, the permutation and negative phase of recovered signals/images have been corrected too. In all comparisons of this paper such normalization have been done, unless stated otherwise. Fig. 3(a) shows BSS performance comparison over three linearly mixed music signals. The correlation values between recovered music signals and source music signals in Table 1(a), indicate a better

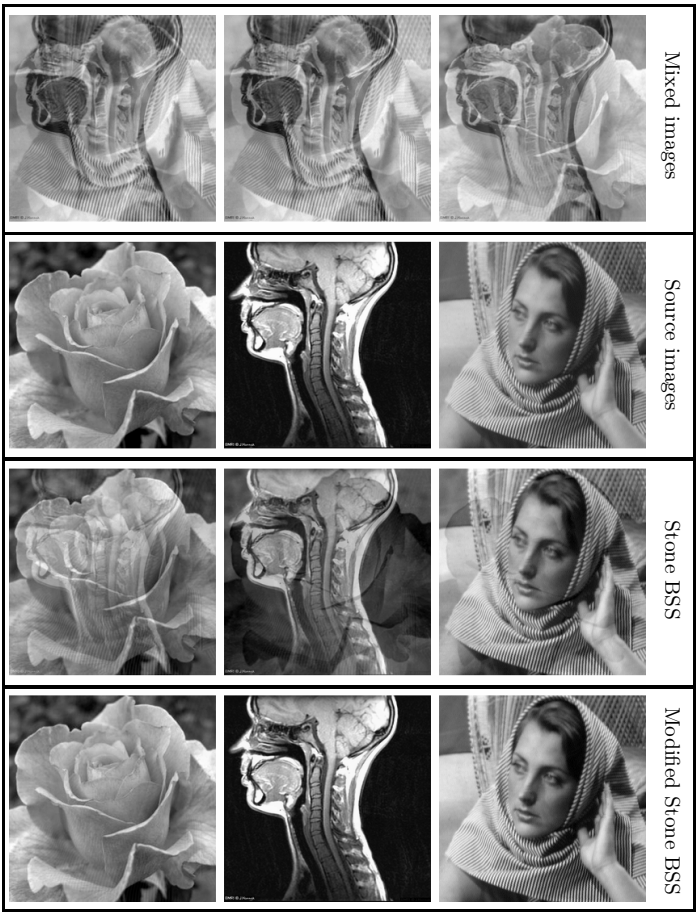


Fig. 4. gray-scale image mixtures (first row), their source images (2^{nd} row), separated images by Stone BSS (3^{rd} row) & separated images by modified Stone BSS (4^{th} row)

separation performance for modified Stone measure. Fig. 3(b) shows such experiment over linear mixture of speech signals. Table. 1(b) conveys that there are higher correlations between the recovered speech signals and their corresponding sources, while the modified measure of predictability has been used.

More than the above experiment cases, we have done 1000 times of the same experiment over mixtures made of different randomly selected music signals from a set of one hundred different music segments. The mixing matrix is also randomly selected and it is the same for both BSS methods in each experiment. The average result of correlation between source music signals and corresponding recovered ones for Stone BSS and its modification are respectively .9963 and .9975. A similar 1000 times experiments has been done over speech signals too. Averaged correlation values for Stone BSS and modified Stone BSS are respectively .9793 and .9822. The numerical results indicate

Table 2. Comparison of separation performances of Stone BSS and modified Stone BSS over gray-scale mixtures related to Fig.4

Stone BSS

	y_1	y_2	y_3
s_1	**.9777**	.1995	.2174
s_2	.1050	**.9105**	.4806
s_3	.1449	.3525	**.8789**

Modified Stone BSS

	y_1	y_2	y_3
s_1	**.9994**	0.0724	.1451
s_2	.1124	**.9999**	.0721
s_3	.0531	0.0439	**.9965**

higher separation performance of modified Stone BSS over both music and speech data.

We have done the evaluation over mixtures of gray-scale images too. For the both methods, the BSS implementation is row-wise that is followed by column-wise. Fig.4 shows the Stone BSS and modified Stone BSS performances in separation of a set of linear mixtures of gray-scale images. For this case of image mixtures, correlation values between recovered images by both methods and their corresponding source images approve the higher performance of modified Stone BSS with respect to Stone BSS (Table 2).

To get a better evaluation of efficiency of BSS methods, we have done 100 different experiments of separating mixtures of gray-scale images. Source images were randomly selected from a set of different images. Mixing matrix was randomly selected too, and it was the same for both BSS methods in each experiment. The average result of correlation values between recovered images and their corresponding source images are respectively .9167 and .9803 for Stone BSS and modified Stone BSS.

4 Conclusion

This paper presents a PDF-matched modification to Stone's measure of signal predictability [7] that is used as a merit function for blind sources separation (BSS). The modified measure of predictability is a measure of non-gaussianity too. Maximization of signals temporal predictability in parallel to increasing their non-gaussianity leads to more independent recovered signals. BSS based on the modified measure of predictability has been compared to BSS based on Stone's measure over mixtures of realistic audio signals (music and speech) as well as mixtures of gray-scale images. Correlation values between sources and their corresponding recovered signals/images demonstrate the higher separation performance of BSS based on the modified measure of predictability. Especially, it is much better over mixtures of gray-scale images.

References

1. Haykin, S.: Unsupervised Adaptive Filtering. Blind Source Separation, vol. 1. John Wiley & Sons, New York (2000)
2. Cichocki, A., Amari, S.: Adaptive Blind Signal and Image Processing. John Wiley & Sons, New York (2000)

3. Friedman, J.: Exploratory Projection Pursuit. Journal of the American Statistical Association 82, 249–266 (1987)
4. DeGroot, M.H.: Probability and Statistics. Addison-Wesley, Reading (1986)
5. Jutten, C., Herault, J.: Independent Component Analysis versus PCA. In: Proc. EUSIPCO, pp. 643–646. IEEE Press, New York (1988)
6. Bell, A.J., Sejnowski, T.J.: An Information-maximization Approach to Blind Separation and Blind Deconvolution. J. Neural computation 7, 1129–1159 (1995)
7. Stone, J.V.: Blind Source Separation using Temporal Predictability. Neurocomputing 13, 1559–1574 (2001)
8. Stone, J.V.: Independent Component Analysis: A Tutorial Introduction. A Bradford Book, London (2004)
9. Stone, J.V.: Blind Deconvolution using Temporal Predictability. Neurocomputing 49, 79–86 (2002)
10. Martinez, D., Bray, A.: Nonlinear blind source separation using kernels. IEEE Trans. Neural Networks 14, 228–335 (2003)
11. Khosravy, M., Asharif, M., Yamashita, K.: A Probabilistic Short-length Linear Predictability Approach to Blind Source Separation. In: 23rd International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 2008), Yamaguchi, Japan, pp. 381–384 (2008)
12. Khosravy, M., Asharif, M., Yamashita, K.: An Efficient ICA Based Approach to Multiuser Detection in MIMO OFDM Systems. In: Plass, S., Dammann, A., Kaiser, S., Fazel, K. (eds.) Multi-Carrier Systems and Solutions MCSS 2009. LNEE. Springer, Heidelberg (2009)
13. Xie, S., He, Z., Fu, Y.: A Note on Stone's Conjecture of Blind Signal sSeparation. Neural Computation 17, 321–330 (2005)
14. Borga, M., Landelius, T., Knutsson, H.: A Unified Approach to PCA, PLS, MLR and CCA. Report LiTH-ISY-R-1992, ISY, SE-581 83 Linkoping, Sweden (1997)

Feedforward Neural Network with Multi-valued Connection Weights

Arit Thammano and Phongthep Ruxpakawong

Computational Intelligence Laboratory, Faculty of Information Technology,
King Mongkut's Institute of Technology Ladkrabang, 10520 Bangkok Thailand
arit@it.kmitl.ac.th, thep@psru.ac.th

Abstract. This paper introduces a new concept of the connection weight to the multi-layer feedforward neural network. The architecture of the proposed approach is the same as that of the original multi-layer feedforward neural network. However, the weight of each connection is multi-valued, depending on the value of the input data involved. The backpropagation learning algorithm was also modified to suit the proposed concept. This proposed model has been benchmarked against the original feedforward neural network and the radial basis function network. The results on six benchmark problems are very encouraging.

Keywords: Feedforward Neural Network, Learning algorithm, Data mining, Classification.

1 Introduction

Classification is the process of finding a set of models that describe and distinguish data classes or concepts, for the purpose of being able to use the model to predict the class of objects whose class label is unknown [1]. A variety of techniques have been applied to deal with the classification problems, such as artificial neural networks, a decision tree, and statistical methods. However, many previous research works [2, 3, 4, 5, 6] show that neural network classifiers have better performance, lower classification error rate, and more robust to noise than other classification methods. Among a number of neural network classifiers, the multi-layer feedforward neural network with backpropagation learning algorithm, often called backpropagation neural network, is the most widely used model. Arbach et al. [7] used the backpropagation neural network to classify mammographic masses. The classification performance was measured by computing the sensitivity, specificity, and area under the receiver operating characteristic (ROC) curves. The results showed that the backpropagation neural network performance was slightly better than the expert radiologists, and significantly better than the residents. Grip et al. [8] used the backpropagation neural network to classify neck movement patterns related to Whiplash-associated disorders. The backpropagation neural network with six hidden nodes achieved a predictivity of 0.89, a sensitivity of 0.90, and a specificity of 0.88, which are very promising results. Another research work by Arbach et al. [9] used the

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 229–237, 2009.

backpropagation neural network for classification of breast MRI lesions. The results illustrated the promise of using the backpropagation neural network as a physician's assistant for breast MRI classification.

Even though the backpropagation neural network has been responsible for numerous successes, it is not without disadvantage. The backpropagation neural network suffers from shortcomings, such as slow convergence rate and easily being trapped in a local minimum [10]. Numerous algorithms that improve on the backpropagation learning algorithm have been proposed. In [11], Sarkar and Yegnanarayana proposed a method of embedding fuzzy objective functions into the original backpropagation learning algorithm. Rimer and Martinez [12] combined the standard backpropagation learning algorithm with lazy training [13]. The combined approach achieved higher accuracy and more robust solutions than either standard backpropagation or lazy training alone.

This paper introduces a new concept of the connection weight to the original multi-layer feedforward neural network. Typically, each connection weight of the multi-layer feedforward neural network is a single real number. However, in this paper the weight of each connection is multi-valued, depending on the value of the input data involved. The performance of the proposed approach is evaluated against the original multi-layer feedforward neural network and the radial basis function network.

This paper is divided into 4 sections. Following this introduction, section 2 presents the architecture of the proposed approach and its learning algorithm. A brief description of the experimental data and the experimental results are given in section 3. Finally, section 4 is the conclusions.

2 The Proposed Approach

In this paper, the three-layer feedforward neural network is used to demonstrate how the proposed approach works. The architecture of the three-layer feedforward neural network is shown in Figure 1. The first layer is the input layer, which consists of N nodes. Each node represents a feature component of the input data. The second layer is the hidden layer. The nodes in the hidden layer are fully connected to the nodes in the input layer and the output layer. The third layer is the output layer. Each node in the output layer represents a class. In this paper, the input vector is denoted by $X = (x_1, x_2, \ldots, x_i, \ldots, x_N)$, where N is the number of features in X. Once the network receives the input and its associated target output (X, Y), the training process will begin. The detailed procedure of the proposed approach is as follows:

a. Define the number of nodes in the hidden layer.
b. Normalize the input data to a value between [0, 1]. Divide the normalized input data range into S segments by specifying the point(s) where the segments will be split, called the splitting point(s). Then, initialize all weight values of each connection, w_{ij} and w_{jk}, with random numbers between -1 and 1.

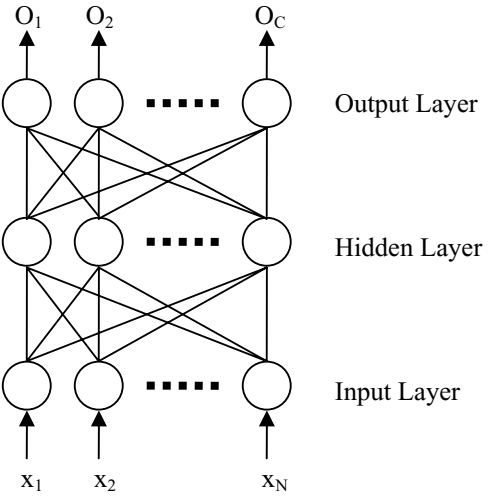


Fig. 1. Architecture of the three-layer feedforward neural network

For example, in Figure 2 the splitting point is at 0.5; as a result, x_i and O_j are divided into 2 equal segments (S = 2). The first segment (s_1) covers the range between 0 and 0.5 while the second segment (s_2) covers the range between 0.5 and 1. Then, for each connection, randomly pick S+1 numbers between -1 and 1. In Figure 2.1, three numbers – 0.55, -0.05, and -0.7 – are randomly picked and assigned to be the weights at the points where $x_i = 0$, $x_i = 0.5$, and $x_i = 1$ respectively – $w_{ij}(0)$, $w_{ij}(0.5)$, and $w_{ij}(1)$.

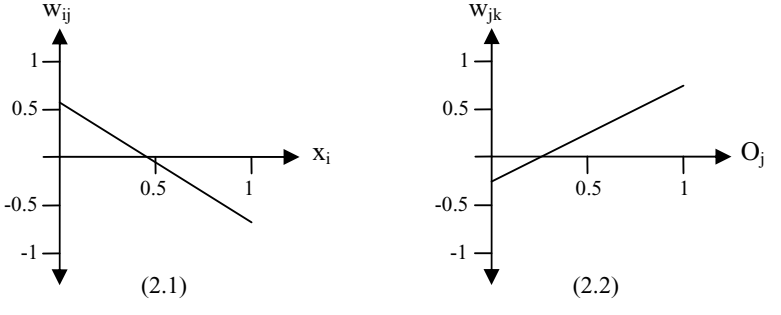


Fig. 2. Example of the new connection weights

c. Calculate the output of each hidden node (O_j) as follows:

$$O_j = f(net_j). \tag{1}$$

$$\text{net}_j = \sum_{i=1}^{N} x_i w_{ij}(x_i) . \tag{2}$$

$$w_{ij}(x_i) = w_{ij}(L_i) + \frac{(x_i - L_i)}{(U_i - L_i)}(w_{ij}(U_i) - w_{ij}(L_i)) . \tag{3}$$

where f(•) is the activation function. In this paper, the sigmoid function is used as the activation function in both the hidden layer and the output layer.

N is the total number of nodes in the input layer.

L_i is a lower bound of the segment that x_i falls into.

U_i is an upper bound of the segment that x_i falls into.

d. Calculate the output of each output node (O_k) as follows:

$$O_k = f(\text{net}_k) . \tag{4}$$

$$\text{net}_k = \sum_{j=1}^{M} O_j w_{jk}(O_j) . \tag{5}$$

$$w_{jk}(O_j) = w_{jk}(L_j) + \frac{(O_j - L_j)}{(U_j - L_j)}(w_{jk}(U_j) - w_{jk}(L_j)) . \tag{6}$$

where M is the total number of nodes in the hidden layer.

L_j is a lower bound of the segment that O_j falls into.

U_j is an upper bound of the segment that O_j falls into.

e. Compare the desired output (Y) with the actual output (O_k) and determine the error at the output layer by

$$E = \frac{1}{2}\sum_{k=1}^{C}(y_k - O_k)^2 . \tag{7}$$

f. Update the connection weights between the hidden layer and the output layer by using the equation

$$w_{jk}^{\text{new}}(L_j) = w_{jk}^{\text{old}}(L_j) + \eta\left(\frac{\partial E}{\partial w_{jk}(O_j)}\right)\left(\frac{(U_j - O_j)}{(U_j - L_j)}\right). \tag{8}$$

$$w_{jk}^{\text{new}}(U_j) = w_{jk}^{\text{old}}(U_j) + \eta\left(\frac{\partial E}{\partial w_{jk}(O_j)}\right)\left(\frac{(O_j - L_j)}{(U_j - L_j)}\right). \tag{9}$$

$$\frac{\partial E}{\partial w_{jk}(O_j)} = (y_k - O_k)(O_k(1 - O_k))O_j . \tag{10}$$

where η is the learning rate, which has a value between 0 and 1.

g. Update the connection weights between the input layer and the hidden layer by using the equation

$$w_{ij}^{new}(L_i) = w_{ij}^{old}(L_i) + \eta\left(\frac{\partial E}{\partial w_{ij}(x_i)}\right)\left(\frac{(U_i - x_i)}{(U_i - L_i)}\right). \tag{11}$$

$$w_{ij}^{new}(U_i) = w_{ij}^{old}(U_i) + \eta\left(\frac{\partial E}{\partial w_{ij}(x_i)}\right)\left(\frac{(x_i - L_i)}{(U_i - L_i)}\right). \tag{12}$$

$$\frac{\partial E}{\partial w_{ij}(x_i)} = (O_j(1-O_j))x_i \sum_{k=1}^{C}(y_k - O_k)(O_k(1-O_k))w_{jk}\ . \tag{13}$$

h. Repeat steps c through g for all remaining input patterns in the training data set.
i. Repeat steps c through h until a predetermined number of iteration is reached or the root mean square error reaches an acceptable level.

During testing, each testing vector is applied in turn and its class is predicted. The class whose output node returns the maximum output value is the result of the prediction.

$$K = \arg\max\{O_k : k = 1, 2, ..., C\}. \tag{14}$$

3 Experimental Results

To test the performance of the proposed approach, the experiments were conducted on 4 artificial data sets and 2 real-life data sets. The four artificial data sets are called Fan, Flower 1, Flower 2, and Sawtooth. The names imply the shape of the classes. Fan data has 4 classes, while the other three have 2 classes. In each of the four data sets, the 4,000 data points in the database were randomly divided into a training set of 2,000 data points and a testing set of 2,000 data points.

The two real-life data sets are the iris data and the image segmentation data. The iris data [14] has been widely used in the classification problem. The sepal length, sepal width, petal length, and petal width of 150 iris flowers from 3 species (Iris-setosa, Iris-versicolor, and Iris-virginica) are measured in centimeters, and are used as the input of the problem. The training set contains 90 records, while the testing set contains 60 records.

The image segmentation data was retrieved from the UCI machine learning database repository [15]. Nineteen continuous attributes are used to predict the output class. There are 7 classes (brick face = 1, sky = 2, foliage = 3, cement = 4, window = 5, path = 6, and grass = 7) with 330 examples per class. In this paper, the training set contains 210 examples, while the testing set contains 2,100 examples.

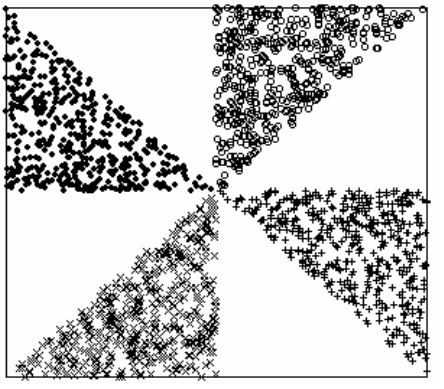


Fig. 3. Fan

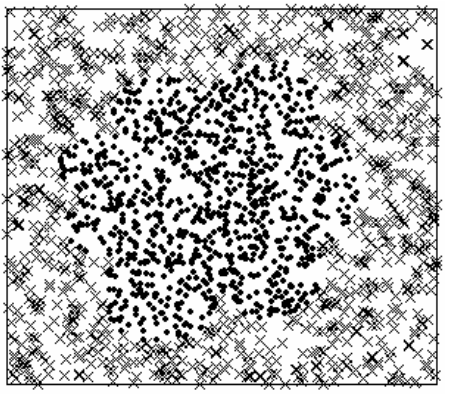


Fig. 4. Flower 1

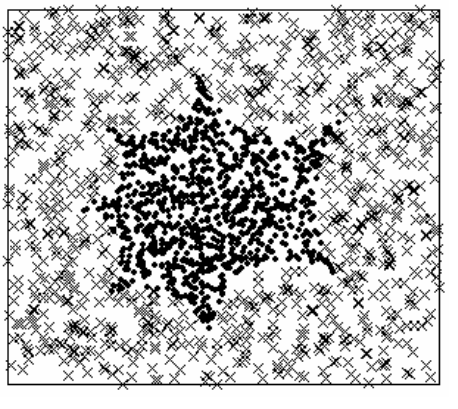


Fig. 5. Flower 2

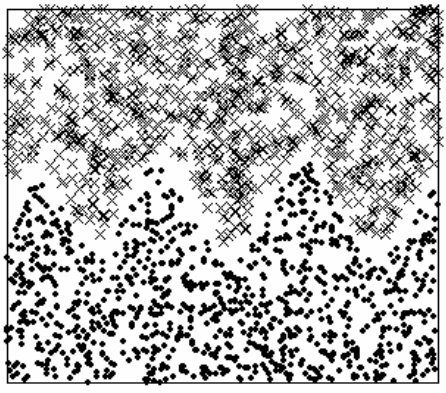


Fig. 6. Sawtooth

Results of the experiments are shown in Table 1. For Fan data (data set 1), the proposed approach and the radial basis function network (RBFN) win out by a small margin over the multi-layer feedforward neural network (MFNN). The proposed approach and the RBFN achieve a perfect accuracy (100%) while the MFNN misses only 5 out of 2,000 patterns.

For Flower 1, Flower 2, and Sawtooth, the proposed approach outperforms the MFNN by a wide margin. However, the performance of the proposed approach is only slightly better than that of the RBFN. The training parameters of the proposed approach are M = 18 and S = 10 for Flower 1, M = 13 and S = 10 for Flower 2, and M = 2 and S = 10 for Sawtooth. The best prediction performance of MFNN are obtained from the networks with M = 23 for Flower 1, M = 29 for Flower 2, and M = 18 for Sawtooth. The best prediction performance of RBFN are obtained from the networks with 93 hidden nodes for Flower 1, 89 hidden nodes for Flower 2, and 91 hidden nodes for Sawtooth.

For the iris data (data set 5), all three methods – the MFNN, the RBFN, and the proposed approach – achieve a perfect accuracy (100%). These best prediction performances are obtained from the MFNN with 10 nodes in the hidden layer, from the RBFN with 32 hidden nodes, and from the proposed approach with 2 hidden nodes and S = 1.

For the image data (data set 6), the proposed approach produces 102 misclassifications out of a total of 2,100 examples; this corresponds to an accuracy of 95.14%. The number of nodes in the hidden layer is 11 and S is 10. On the other hand, the MFNN with 14 hidden nodes produces 137 misclassifications, and the RBFN with 70 hidden nodes produces 109 misclassifications.

4 Conclusions

This paper introduces an improvement on the multi-layer feedforward neural network. In the proposed approach, the weight of each connection in the multi-layer feedforward neural network is modified. To test the performance, the proposed approach, the

original feedforward neural network, and the radial basis function network were used to classify six benchmark problems. With respect to the classification accuracy, the proposed approach is found to be far superior to the original feedforward neural network, and marginally better than the radial basis function network. In addition, the size of the proposed network is much smaller than the sizes of the other two methods.

Table 1. Experimental results

Data Set	Accuracy (%)		
	Feedforward Neural Network	Radial Basis Function Network	Proposed Approach
1. Fan	99.75	100.0	100.0
2. Flower 1	95.35	97.40	99.35
3. Flower 2	95.40	98.25	99.45
4. Sawtooth	95.90	99.80	100.0
5. Iris	100.0	100.0	100.0
6. Image	93.48	94.81	95.14

References

1. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann, San Manteo (2001)
2. Quinlan, J.R.: Comparing Connectionist and Symbolic Learning Methods. In: Workshop on Computational Learning Theory and Natural Learning Systems, pp. 445–456. MIT Press, Cambridge (1994)
3. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall, New Jersey (1995)
4. Shavlik, J.W., Mooney, R.J., Towell, G.G.: Symbolic and Neural Learning Algorithms: An Experimental Comparison. Machine Learning 6, 111–143 (1991)
5. Danaher, S., Herries, G., Selige, T., Mac Suirtan, M.: A Comparison of the Characterisation of Agricultural Land Using Singular Value Decomposition and Neural Networks. In: Neurocomputation in Remote Sensing Data Analysis. Springer, Heidelberg (1997)
6. Lee, J., Weger, R.C., Sengupta, S.K., Welch, R.M.: A Neural Network Approach to Cloud Classification. IEEE Transactions on Geoscience and Remote Sensing 28, 846–855 (1990)
7. Arbach, L., Reinhardt, J.M., Bennett, D.L., Fallouh, G.: Mammographic Masses Classification: Comparison Between Backpropagation Neural Network (BNN), K Nearest Neighbors (KNN), and Human Readers. In: Canadian Conference on Electrical and Computer Engineering, pp. 1441–1444. IEEE Press, New York (2003)
8. Grip, H., Öhberg, F., Wiklund, U., Sterner, Y., Karlsson, J.S., Gerdle, B.: Classification of Neck Movement Patterns Related to Whiplash-Associated Disorders Using Neural Networks. IEEE Transactions on Information Technology in Biomedicine 7, 412–418 (2003)
9. Arbach, L., Stolpen, A., Reinhardt, J.M.: Classification of Breast MRI Lesions Using a Backpropagation Neural Network (BNN). In: IEEE International Symposium on Biomedical Imaging: Nano to Macro, pp. 253–256. IEEE Press, New York (2004)
10. Go, J., Han, G., Kim, H., Lee, C.: Multigradient: A New Neural Network Learning Algorithm for Pattern Classification. IEEE Transactions on Geoscience and Remote Sensing 39, 986–993 (2001)

11. Sarkar, M., Yegnanarayana, B.: Incorporation of Fuzzy Classification Properties into Backpropagation Learning Algorithm. In: Sixth IEEE International Conference on Fuzzy Systems, pp. 1701–1706. IEEE Press, New York (1997)
12. Rimer, M., Martinez, T.: Softprop: Softmax Neural Network Backpropagation Learning. In: 2004 IEEE International Joint Conference on Neural Networks, pp. 979–983. IEEE Press, New York (2004)
13. Rimer, M., Anderson, T., Martinez, T.R.: Improving Backpropagation Ensembles through Lazy Training. In: IEEE International Joint Conference on Neural Networks, pp. 2007–2112. IEEE Press, New York (2001)
14. Fisher, R.A.: The Use of Multiple Measurements in Taxonomic Problems. Annual Eugenics 7, 179–188 (1936)
15. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository, http://www.ics.uci.edu/~mlearn/MLRepository.html

A Method of Geometric Analysis of Condorcet Function

Xiaodong Xu[1] and Xu Liu[2]

[1] College of Public Administration, Huazhong University of Science. & Technology, 430074 Wuhan, China
[2] Department of Control Science & Engineering, Huazhong University of Science. & Technology, 430074 Wuhan, China

Abstract. Voting is usually used in democratic society for decision-making, but majority circle is inevitable, if choosing that method. It is complex to by combinatory to compute the probability of majority circle, instead of that, geometry is an intuitive and simple method, besides, it can show the relationship between the different combination and majority circle. First, compute the probability of majority circle by combinatory, and then denote candidate triangle and result-profile triangle. Finally analyze each preference combination and its voting result, what's more, if voting majority circle occurs, compute the probability.

Keywords: Majority circle, Condorcet winner, Result-profile triangle.

1 Introduction

Voting is an ancient way of decision-making, and it has a long history. From the Middle Ages up to now, research in this field has never been ceased. In democratic modern society, it is widely used in all aspects of social and economic life, and contributes greatly no matter to political and social activities, or to the evaluation and decision-making of governments, international organizations.

To most people, voting seems simple. After all, to vote we just need a sheet of paper or a card, then count how many people favor each candidate. But voting may go wrong with such two elementary procedures. Maybe, the winner was not the one in the eyes of voters, for example, Al Gore was the loser in the U.S. presidential election who won the popular vote by approximately 500,000 votes, but ultimately lost the electoral college to Republican candidate George W. Bush when the legal controversy over the Florida election recount was eventually settled in the U.S. Supreme Court by a 5-4 margin in favor of Bush.

In 18th century, Condorcet pointed out there is only one way that can strictly elect whom they really wanted. The "correct" voting procedure is the winner only can be the candidates who beat all other candidates in pairwise elections, and the winner called "Condorcet winner" [1]. It seems to be unquestionable, but actually it can be questioned. While three or more than three candidates, outcome appears majority circle, so elections reach an impasse. A simple explanation of this phenomenon is the ranking of entire group of voters does not have transitivity.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 238–243, 2009.

2 Combinatory Method

Permutation and combination is a traditional way to analysis the majority circle. This old method proved to be very useful. We illustrate with an example, where voters select one person from {A, B, C} three candidates. The set of voters' preferences is called profiles which strictly ranks the candidates, thus here appears six profiles. "$A \succ B$" means "A is preferred B", "$A \succ_G B$" means "the entire group of voters considers A is preferred B". For convenience, assign these types by the following numbers:

Considering each one can select one type from the six profiles, if there are three voters, the situation is up to 63 combinations. If the first voter chooses type 1, the second chooses type 5, and the third chooses type 5, comes the bad outcome "$A \succ B \succ C \succ A$". This makes it impossible to select an "optimal" candidate. Similarly, while the three voters with the profiles of {2,6,4}, {1,3,5}, {2,4,6}, {5,1,3}, {6,2,4}, {6,4,2}, {5,3,1}, {4,2,6}, {3,1,5}, {4,6,2}, {3,5,1} also cause majority circle. Through the analysis of the above, we compute the probability of three voters voting for three candidates is 12 divided by 216 equals 0.056. From this we see that it is too complicated to adopt combinatory method, especially the growth of the voters and candidates would multiply the complexity of the problem. Niemi and Garman (1968) computed the probability of different voters voting for different candidates [2].

Table 1. Type of preferences

Type	Preference	Type	Preference
1	$A \succ B \succ C$	4	$C \succ B \succ A$
2	$A \succ C \succ B$	5	$B \succ C \succ A$
3	$C \succ A \succ B$	6	$B \succ A \succ C$

Table 2. Probability of different voters voting for different candidates

Candidate	Voter						
	3	5	7	11	15	25	∞
3	0.0556	0.0694	0.075	0.0798	0.082	0.0843	0.0877
4	0.1111	0.14	0.15				0.1755
5	0.16	0.2	0.22				0.2513
6	0.2	0.25	0.27				0.3152
8	0.28	0.33	0.37				0.4151
10	0.32	0.4					0.4887

3 Candidate Triangle of Three Candidates

Faced with all profiles, progress has been seriously hindered by the complexity of combinatory method. This traditional way to construct profiles has a great failing. In recent years, the emphasis of the research has shifted to geometrical method. It reduces these previous complicated problems into simple graph that can be presented.

Denote a candidate triangle where each of candidates {A, B, C} is identified with a vertex. The candidates have the same status, so the triangle is equilateral [3]. In it, there exist many voting points which reflect voters' preferences to the candidates. If the voting points is much closer to the candidate's vertex, the ranking of the candidate is in the voter's profile is closer to the top. Doing a vertical line along the vertex C, divide the candidate triangle into two parts. Voting points is closer to A compared with B in the left part, so the voter consider $A \succ B$, otherwise $B \succ A$, and all of the points on the vertical line means A and B are indifferent between their options. Similarly, the triangle is divided into the six same size and shape sections which correspond with the types of profiles by the three vertical lines along the vertexes. Illustrate an example, all points in the region 1 are the closest to A, next closest to B, and farthest to C, so profile is $A \succ B \succ C$.

If n is the number of the voters, n_i is the number of the voters type i, so $n_1+ n_2+ n_3+ n_4+ n_5+ n_6 =n$, n and n_i are nonnegative integers.

There exists a simple case that if all voters have a common preference, their voting points only will concentrate in one region of the triangle, so the group profile is easily defined. If all voters only have two preferences, the situation is $C_6^2=15$, but considering symmetry and A, B, C are the same status, it reduce to four which is shown in Fig. 2.

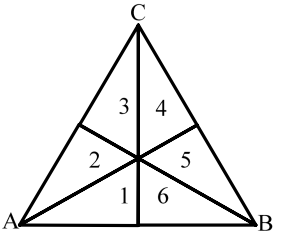


Fig. 1. Candidate triangle and profile region

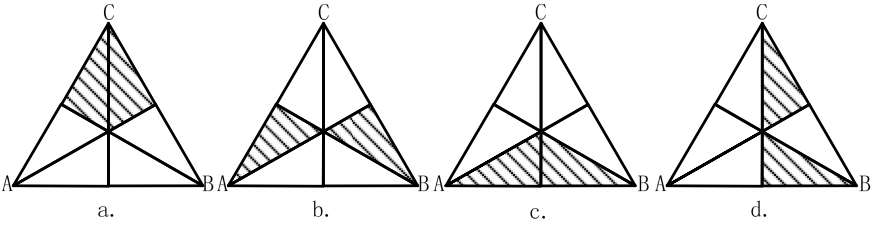


Fig. 2. Two preference candidate triangle

For instance of Fig. 2a, it is shown that all voting points are closest to C, so we only need care about the choice between A and B, if $n_3 > n/2$, thus $n_3 > n_4$, the preference of the group is $C \succ_G A \succ_G B$, otherwise $C \succ_G B \succ_G A$. Similarly, make the same analysis, the outcome is in Table 3.

Table 3. Outcomes of different preferences

	Qualification	Outcome	Qualification	Outcome
Fig. 2a.	$n_3 > n/2$	$C \succ_G A \succ_G B$	$n_3 < n/2$	$C \succ_G B \succ_G A$
Fig. 2b.	$n_2 > n/2$	$A \succ_G B \succ_G C$	$n_2 < n/2$	$B \succ_G A \succ_G C$
Fig. 2c.	$n_1 > n/2$	$A \succ_G B \succ_G C$	$n_1 < n/2$	$B \succ_G A \succ_G C$
Fig. 2d.	$n_4 > n/2$	$C \succ_G B \succ_G A$	$n_4 < n/2$	$B \succ_G C \succ_G A$

If all voters only have two preferences**,** the situation is $C_6^3 = 20$, but symmetry reduces the number to 4.

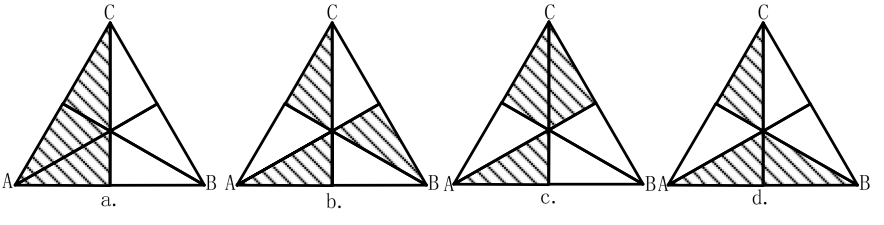


Fig. 3. Three preference candidate triangle

Fig. 3a shows a simplest case that all voting points are far from B, so if $n_3 < n/2$, the voters consider C is preferred to A, thus the group preference is preference $A \succ_G C \succ_G B$, otherwise $C \succ_G A \succ_G B$.

In Figure 3b, we divide n_1, n_3, n_5 by n to facilitate the research, let $x=n_1/n$, $y=n_3/n$, $z=n_5/n$, thus $z=1-x-y$, so the candidate triangle is transformed into result-profile triangle, as in Fig. 4. Compared A and B, only voting points in region 5 think B is preferred to A, if $x+y>1/2$, the group considers $A \succ_G B$, similarly, if $x>1/2$, $A \succ_G B$; if $y>1/2$, $C \succ_G B$. Divide the result-profile triangle into four parts and the center region is $A \succ_G B \succ_G C \succ_G A$, so majority circle happens. There are $(n+1)(n+2)/2$ points in the result-profile triangle. When n is odd the number of points in the center is $(n+1)(n-2)/8$; when even, it is $(n+1)(n-2)/8$[4]. So, if n is odd the probability of majority circle is

$$\frac{1}{4}(1-\frac{3}{n+2}) \tag{1}$$

If n is even, the probability is

$$\frac{1}{4}(1-\frac{9}{(n+2)(n+1)}) \tag{2}$$

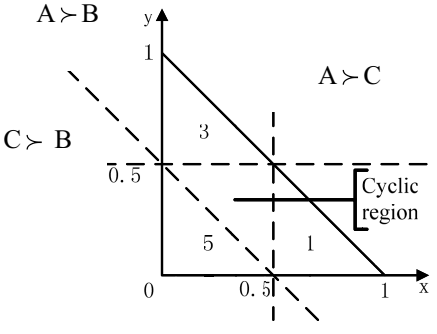


Fig. 4. Result-profile triangle of Fig. 3b

As n approaches infinite, the probability limit of majority cycle is 1/4. For n=3, there are only 1/10 that comes majority circle. Considering the possibility of three preferences is $P_6^3 / 6^3$ =120/216, the probability of majority circle of three voters voting for three candidates is

$$\frac{1}{10}\times\frac{120}{216}=\frac{12}{216}$$

The result agrees to combinatorial method. Majority circle does not occur in Figure 3c and 3d, as is shown in Fig. 5.

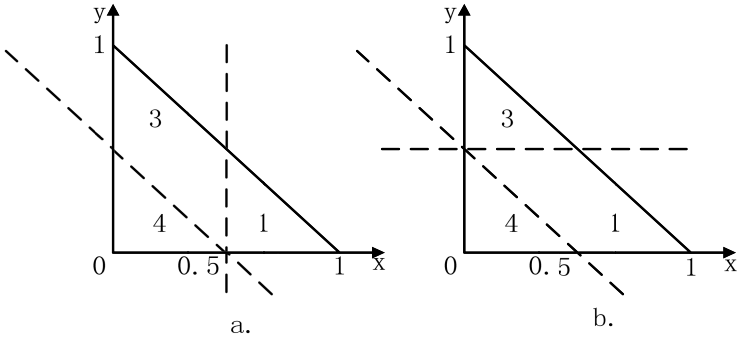


Fig. 5. Result-profile Triangle

The above analysis is the geometry model of Condorcet function of three candidates and three preferences, and it calculates in two-dimensional graph that three vertex is (1, 0), (0, 1), (0, 0). While the preference is more than three, the model becomes multi-dimensional graph. Except the origin, each vertex is assigned as a row of the matrix, for instance, if the number of voter types is five, the vertex coordinate is

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

As the growth of the candidates, the maximal number of preference type increases, so the dimensions of the model also go up, if set all voting theory into geometry, use vector space to make analysis of voting rules, not only do we come to a deeper understanding of the existing theory, but also help to redesign the theory.

4 Summary

Compared with traditional combinatorial method, geometry has great draw on analysis of voting problem. As shown, the geometry offers a simple solution in voting theory for a complex problem. What's more, it allows to see whether the voting paradox occur or not in a specific figure and calculate the likelihood of each profile easily. Geometry is a significant theory as a complement of the solution of voting paradox. We can find the distribution of strategy-proof preference profile by the analysis of different preference profile triangles and predict the strategy-voting behavior on this basis.

References

1. Luo, Y.F., Xiao, R.B.: Theory and Progress of Social Choice. Science Press, Beijing (2003)
2. Yue, C.Y.: Decision Theory and Manners. Science Press, Beijing (2000)
3. Saari, D.G.: Basic Geometry of Voting. Springer, Heidelberg (1995)
4. Saari, D.G.: Geometry Voting and Paradoxes. Mathematics Magazine 71, 243–260 (1998)

Exponential Asymptotic Stability of a Two-Unit Standby Redundant Electronic Equipment System under Human Failure

Xing Qiao[1,2], Zhaoxing Li[1], and Dan Ma[1]

[1] Department of Math., Daqing Normal University, Daqing 163712, China
xiaoqiao1502@163.com
[2] Beijing Institute of Information and Control, Beijing 100037, China

Abstract. In this paper, we deal with the exponential asymptotic stability of a two-unit standby redundant electronic equipment system under human failure. First we prove that the positive contraction c_0-semigroup $\{T(t)\}_{t\geq 0}$ which is generated by the operator corresponding to these equations is a quasi-compact operator. Then by using that 0 is an eigenvalue of the operator with algebraic index one and the c_0-semigroup $\{T(t)\}_{t\geq 0}$ is contraction, we deduce that the spectral bound of the operator is zero. By using the above results we obtain easily the exponential asymptotic stability of the solution of the redundant system.

Keywords: Exponential asymptotic stability, Quasi-compact operator, c_0-semigroup, Electronic equipment system.

1 Introduction

As the development of science and technology, electron productions and network are used everywhere. And redundancy plays an important role in enhancing system reliability. So the stability analysis of the redundant system becomes more and more important. In Ref.[1], the authors have developed a mathematical model, which is a two-unit standby redundant electronic equipment system under human failure. Furthermore, in Ref.[1], using the supplementary variable technique, Laplace transforms of various state probabilities have been obtained which further yield time dependent probabilities by inversion process. In Ref.[2], author proved the asymptotic stability of the system and the steady-state solution is shown to be the eigenvector of the system operator corresponding to the eigenvalue 0. The aim of our present work is to study the velocity of the time-dependent solution converging to the steady-state solution. In this article, we will study the converging velocity. We first convert the model into an abstract Cauchy problem in a Banach space, then show that the operator corresponding to this system model generates a positive contraction c_0-semigroup. We then prove that c_0-semigroup $\{T(t)\}_{t\geq 0}$ is a quasi-compact operator, and that spectral bound of this operator is zero. Thus by Theorem 2.1(see [3]) we have our desired consequence.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 244–252, 2009.

1.1 Assumptions

The following assumptions are associated with this system(see Fig.1)

(I) Initially the system is in good state.
(II) The system has two states i.e. good and failed.
(III) The standby unit starts operating just after the failure of the main unit.
(IV) The repair is undertaken only when the complex system is in state 4.
(V) After repair, the system works like new.
(VI) Both the units suffer two types of failures, namely constant failure and human failure.
(VII) The failure and repair times for the system follow exponential and general distribution respectively.
(VIII) Human failure rates from states 1 to 3 and 2 to 3 are different.
(IX) In the complex system, at any time, only one change can take place in the state of the system.
(X) The system has two units connected in standby redundancy where of course of each unit has two states i.e. good and failed.

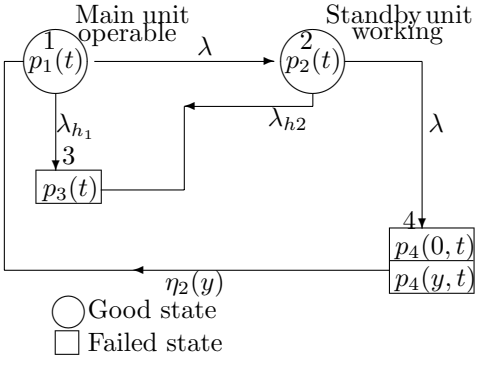


Fig. 1. State transition diagram

1.2 Notations

The following symbols are used in this article:

(I) 1-good state (i.e. both the units are good);
(II) 2-one unit is good and other is failed;
(III) 3-failed state;
(IV) 4-failed state and under repair;
(V) λ-exponential constant failure rate for one unit from states 1 to 2 and states 2 to 4;
(VI) $\eta_2(y)$-transition rate of repair for the two units from states 4 to 1;
(VII) λ_{h1}-constant human failure rate from states 1 to 3;
(VIII) λ_{h2}-constant human failure rate from states 2 to 3;
(IX) $p_1(t)$-Prob{system is in state 1; at time t};

(X) $p_2(t)$-Prob{system is in state 2; at time t};
(XI) $p_3(t)$-Prob{system is in state 3; at time t};
(XII) $p_4(y,t)$-pdf{system is in state 4 and is under repair; elapsed repair time is y ; at time t};

According to [1], the system of a set of integro-differential equations for the stochastic process which is continuous in time and discrete in space associated with this mathematical model is as follows:

$$[\frac{d}{dt} + \lambda + \lambda_{h1}]p_1(t) = \int_0^\infty p_4(y,t)\eta_2(y)dy \tag{1}$$

$$[\frac{d}{dt} + \lambda_{h2} + \lambda]p_2(t) = \lambda p_1(t) \tag{2}$$

$$\frac{d}{dt}p_3(t) = \lambda_{h2}p_2(t) + \lambda_{h1}p_1(t) \tag{3}$$

$$[\frac{\partial}{\partial t} + \frac{\partial}{\partial y} + \eta_2(y)]p_4(y,t) = 0 \tag{4}$$

the boundary and initial conditions are

$$p_4(0,t) = \lambda p_2(t) \tag{5}$$

$$p_1(0) = 1 \tag{6}$$

$$p_2(0) = p_3(0) = p_4(y,0) = 0 \tag{7}$$

where

$$0 \le \eta_2(y) < \infty, \int_0^\infty e^{-\int_0^y \eta_2(s)ds}dy < \infty,$$

$$0 < \eta_2 = \lim_{y\to\infty}\frac{1}{y}\int_0^y \eta_2(s)ds < +\infty, \int_0^\infty \eta_2(y)dy = \infty$$

For simplicity, let

$$m_1 = \lambda + \lambda_{h1}, m_2 = \lambda + \lambda_{h2}$$

Taking state space X as follows:

$$X = \{z \in R \times R \times R \times L^1[0,\infty) | \|z\| = \sum_{i=1}^{3}|z_i| + \|z_4(y)\|_{L^1[0,\infty)}\}$$

It is obvious that $(X, \|\cdot\|)$ is a Banach Space.
In the following we define operator A and its domain:

$$AP = \begin{pmatrix} -m_1 & 0 & 0 & 0 \\ 0 & -m_2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{d}{dy} - \eta_2(y) \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4(y) \end{pmatrix}$$

$D(A) = \{P \in X | \frac{dp_4(y)}{dy} \in L^1[0,\infty), p_4(y)$ is absolutely continuous functions, $p_4(0) = \lambda p_2\}$
define operator $B : X \to X$

$$BP = \begin{pmatrix} 0 & 0 & 0 & \int_0^\infty \cdot \eta_2(y)dy \\ \lambda & 0 & 0 & 0 \\ \lambda_{h1} & \lambda_{h2} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4(y) \end{pmatrix}$$

Then the above equations (1)-(7) can be described as an abstract Cauchy problem in Banach space X:

$$\frac{dP(t)}{dt} = (A + B)P(t),\ t \in [0,\infty) \tag{8}$$
$$P(0) = (1,0,0,0) \tag{9}$$

Theorem 1. *A generates a positive contraction c_0-semigroup $\{S(t)\}_{t\geq 0}$.*

Theorem 2. *$A + B$ generates a positive contraction c_0-semigroup $\{T(t)\}_{t\geq 0}$.*

The proof of Theorem 1.1 and Theorem 1.2 see in Ref. [2,4].

In this article, we first prove that $S(t)$ is a quasi-compact operator by studying two operators $U(t)$ and $V(t)$, then we obtain that $T(t)$ is a quasi-compact operator by using the compactness of B, and finally by using [2], 0 is an eigenvalue of $A + B$ and $(A + B)^*$ with geometric multiplicity one. Thus by Theorem 2.1 in Ref. [3], we can obtain our desired result.

2 Conclusions

Proposition 1. *For $\Phi \in X$, $P(y,t) = (S(t)\Phi)(y)$ is a solution of the following Cauchy problem:*

$$\begin{cases} \frac{d}{dt}P(t) = AP(t), \\ P(0) = \Phi,\ \Phi \in X \end{cases} \tag{10}$$

then

$$(S(t)\Phi)(y) = \begin{cases} \begin{pmatrix} \Phi_1 e^{-m_1 t} \\ \Phi_2 e^{-m_2 t} \\ \Phi_3 \\ p_4(0, t-y)e^{-\int_0^y \eta_2(\tau)d\tau} \end{pmatrix}, y < t \\ \begin{pmatrix} \Phi_1 e^{-m_1 t} \\ \Phi_2 e^{-m_2 t} \\ \Phi_3 \\ \Phi_4(t-y)e^{-\int_{y-t}^y \eta_2(\tau)d\tau} \end{pmatrix}, y \geq t \end{cases}$$

where $p_4(0, t-y)$ is given by (5).

Proof. Since P is a solution of the following equations:

$$\frac{d}{dt}p_1(t) = -m_1 p_1(t) \tag{11}$$

$$\frac{d}{dt}p_2(t) = -m_2 p_2(t) \tag{12}$$

$$\frac{d}{dt}p_3(t) = 0 \tag{13}$$

$$[\frac{\partial}{\partial t} + \frac{\partial}{\partial y} + \eta_2(y)]p_4(y,t) = 0 \tag{14}$$

$$p_4(0,t) = \lambda p_2(t) \tag{15}$$

$$p_1(0) = \Phi_1,\ p_2(0) = \Phi_2,\ p_3(0) = \Phi_3, p_4(y,0) = \Phi_4(y) \tag{16}$$

If we set $\zeta = y - t$ and define $Q_4(t) = p_4(t+\zeta, t)$, then from (14) we get that

$$\frac{dQ_4(t)}{dt} = -\eta_2(\zeta + t)Q_4(t) \tag{17}$$

If $\zeta < 0$, then integrating (17) from $-\zeta$ to t, and using $Q_4(-\zeta) = p_4(0, -\zeta) = p_4(0, t-y)$, we have

$$p_4(y,t) = Q_4(t) = Q_4(-\zeta)e^{-\int_{-\zeta}^{t}\eta_2(\zeta+\tau)d\tau} \tag{18}$$

$$p_4(y,t) = p_4(0,t-y)e^{-\int_0^y \eta_2(\tau)d\tau} \tag{19}$$

From (11)-(13), we have

$$p_1(t) = \Phi_1 e^{-m_1 t}, p_2(t) = \Phi_2 e^{-m_2 t}, p_3(t) = \Phi_3 \tag{20}$$

If $\zeta \geq 0$, then integrating (17) from 0 to t, and then using relations $Q_4(0) = p_4(\zeta, 0) = Q_4(y-t)$, and by similar argument to (18)-(19) we get

$$(S(t)\Phi)(y) = \begin{cases} \Phi_1 e^{-m_1 t} \\ \Phi_2 e^{-m_2 t} \\ \Phi_3 \\ \Phi_4(t-y)e^{-\int_{y-t}^{y}\eta_2(\tau)d\tau} \end{cases}, y \geq t$$

The proof of Prop. 1 is completed. □

By Theorem 1.1 and similar argument in Ref. [2] or [4], we know that the system (10) has a unique nonnegative solution $P(y,t) = (S(t)\Phi)(y)$, by using the c_0-semigroup theory in Ref. [5], we can know that $P(y,t)$ is not only the weak solution of the system (10) but also the strong solution.

If we define two operators as follows, for $P \in X$

$$(U(t)P)(y) = \begin{cases} 0, & y \in [0,t), \\ (S(t)P)(y), & y \in [t,\infty), \end{cases} \tag{21}$$

$$(V(t)P)(y) = \begin{cases} (S(t)P)(y), & y \in [0,t), \\ 0, & y \in [t,\infty), \end{cases} \tag{22}$$

then $S(t)P = U(t)P + V(t)P$.

Lemma 1. *A closed and bounded subset $Y \subset X$ is compact if and only if the following two conditions hold:*

(i) $\lim_{h\to 0} \int_0^\infty |\Phi_4(y+h) - \Phi_4(y)|dy = 0$, *uniformly for* $\Phi = (\Phi_1, \Phi_2, \Phi_3, \Phi_4) \in Y$;
(ii) $\lim_{h\to 0} \int_h^\infty |\Phi_4(y)|dy = 0$, *uniformly for* $\Phi = (\Phi_1, \Phi_2, \Phi_3, \Phi_4) \in Y$.

The proof of the Lemma 2.1 is easy (see [6]), so we omit the details.

Theorem 3. *$V(t)$ is a compact operator on X.*

Proof. According to the definition of $V(t)$, it suffices to prove condition (ii) in Lemma 2.1. For bounded $\Phi \in X$, we set $P(y,t) = (S(t)\Phi)(y), y \in [0,t)$, then $P(y,t)$ is a general solution of the system (10). So by Prop.2.1, we have, for $y, h \in [0,t)$, $y+h \in [0,t)$,

$$\begin{aligned}
&\int_0^\infty |p_4(y+h) - p_4(y,t)|dy = \int_0^t |p_4(y+h,t) - p_4(y,t)|dy \\
&= \int_0^t |p_4(0,t-y-h)e^{-\int_0^{y+h}\eta_2(\tau)d\tau} - p_4(0,t-y)e^{-\int_0^y \eta_2(\tau)d\tau}|dy \\
&\le \int_0^t |p_4(0,t-y-h)| \cdot |e^{-\int_0^{y+h}\eta_2(\tau)d\tau} - e^{-\int_0^y \eta_2(\tau)d\tau}|dy \\
&+ \int_0^t |p_4(0,t-y-h) - p_4(0,t-y)|e^{-\int_0^y \eta_2(\tau)d\tau}dy \qquad (23)
\end{aligned}$$

We will estimate each term in (23). By using (15) and Theorem 1.1 we have

$$\begin{aligned}
|p_4(0,t-y-h)| &= |\lambda p_2(t-y-h)| \le \lambda \|p(\cdot,t-y-h)\|_X \\
&= \lambda \|S(t-y-h)\Phi(\cdot)\|_X \le \lambda \|\Phi\|_X \qquad (24)
\end{aligned}$$

By using (24), we estimate the first term in (23) as follows:

$$\begin{aligned}
&\int_0^t |p_4(0,t-y-h)| \cdot |e^{-\int_0^{y+h}\eta_2(\tau)d\tau} - e^{-\int_0^y \eta_2(\tau)d\tau}|dy \\
&\le \lambda \|\Phi\|_X \int_0^t |e^{-\int_0^{y+h}\eta_2(\tau)d\tau} - e^{-\int_0^y \eta_2(\tau)d\tau}|dy \to 0, \\
&as\ |h| \to 0, uniformly\ for\ \Phi \qquad (25)
\end{aligned}$$

By using (15), (20) we have

$$\begin{aligned}
&|p_4(0,t-y-h) - p_4(0,t-y)| = \lambda |p_2(t-y-h) - p_2(t-y)| \\
&= \lambda |\Phi_2(e^{-m_2(t-y-h)} - e^{-m_2(t-y)})| \to 0, \\
&as\ |h| \to 0, uniformly\ for\ \Phi \qquad (26)
\end{aligned}$$

(26) imply that the second term in (23) satisfies

$$\begin{aligned}
&\int_0^t |p_4(0,t-y-h) - p_4(0,t-y)|e^{-\int_0^y \eta_2(\tau)d\tau}dy \to 0, \\
&as\ |h| \to 0, uniformly\ for\ \Phi \qquad (27)
\end{aligned}$$

Combining (25), (27) with (23), for $y+h \in [0,t)$, we have

$$\int_0^\infty |p_4(y+h,t) - p_4(y,t)|dy \to 0, as\ |h| \to 0, uniformly\ for\ \Phi \tag{28}$$

If $h \in [-t,0)$, $y \in (0,t)$, then from the relation $p_4(y+h,t) = 0$, for $y+h < 0$, we have

$$\begin{aligned}
&\int_0^\infty |p_4(y+h,t) - p_4(y,t)|dy = \int_0^t |p_4(y+h,t) - p_4(y,t)|dy \\
&= \int_{-h}^t |p_4(y+h,t) - p_4(y,t)|dy + \int_0^{-h} |p_4(y+h,t) - p_4(y,t)|dy \\
&= \int_{-h}^t |p_4(y+h,t) - p_4(y,t)|dy + \int_0^{-h} |p_4(y,t)|dy
\end{aligned} \tag{29}$$

Since $y+h \in [0,t)$, for $y \in [0,t)$, $h \in [-t,0)$, for the first term in (29), similar way to (28) we have

$$\int_{-h}^t |p_4(y+h,t) - p_4(y,t)|dy \to 0,\ as\ |h| \to 0, uniformly\ for\ \Phi \tag{30}$$

By using Prop.2.1 and (24), we estimate the second term in (29) as follows:

$$\begin{aligned}
&\int_0^{-h} |p_4(y,t)|dy = \int_0^{-h} |p_4(0,t-y)e^{-\int_0^y \eta_2(\tau)d\tau}|dy \\
&\le \lambda \|\Phi\|_X \int_0^{-h} e^{-\int_0^y \eta_2(\tau)d\tau} dy \to 0, as\ |h| \to 0, uniformly\ for\ \Phi
\end{aligned} \tag{31}$$

Combining (30)-(31) with (29), for $h \in [-t,0)$, we have

$$\int_0^\infty |p_4(y+h,t) - p_4(y,t)|dy \to 0,\ as\ |h| \to 0, uniformly\ for\ \Phi \tag{32}$$

From (28) and (32) we know that the result of the following theorem holds. □

Theorem 4. *Assume there exists one positive constant η_2 such that*

$$0 < \lim_{y\to\infty} \frac{1}{y} \int_0^y \eta_2(s)ds = \eta_2 < +\infty,$$

then $U(t)$ satisfies

$$\|U(t)\Phi\|_X \le e^{-\min\{m_1, m_2, \eta_2\}t} \|\Phi\|_X,\ \forall\, \Phi \in X \tag{33}$$

Proof. For any $\Phi \in X$, from the definition of $U(t)$ and (20), we have

$$\begin{aligned}
&\|U(t)\Phi\|_X\| = |p_1(t)| + p_2(t)| + |p_3(t)| + \int_t^\infty |\Phi_4(t-y)e^{-\int_{y-t}^y \eta_2(\tau)d\tau}|dy \\
&\le |\Phi_1|e^{-m_1 t} + |\Phi_2|e^{-m_2 t} + |\Phi_3| + |e^{-\eta_2 t}|| \int_t^\infty \Phi_4(t-y)dy| \\
&\le e^{-\min\{m_1, m_2, \eta_2\}t} \|\Phi\|_X,\ \forall\, \Phi \in X
\end{aligned} \tag{34}$$

(34) shows that the result of the theorem holds. □

From Theorem 2.1 and 2.2, we have

$$\|S(t) - V(t)\| = \|U(t)\| \leq e^{-\min\{m_1, m_2, \eta_2\}t} \to 0, as \ \ t \to \infty.$$

From which together with Definition 2.7 in Ref.[3], we have the following result.

Theorem 5. *$S(t)$ is a quasi-compact operator on X.*

Since B is a compact operator on X by Theorem 2.3 and Prop.2.9 in Ref.[3], we deduce the following corollary.

Corollary 1. *$T(t)$ is a quasi-compact operator on X.*

Remark 1. $0 \in \sigma_p(A+B)$ and its algebraic index is one, $0 \in \sigma_p(A+B)^*$ and its geometric multiplicity is one, and set $\{r \in \mathcal{C} | Re\, r > 0$ *or* $r = ia, a \neq 0, a \in R\}$ belongs to the resolvent set of $(A+B)$ (in Ref.[2]).

From the above Remark 2.1, we conclude that the spectral bound of $(A+B)$ is zero. Thus by using Remark 2.1, Theorem 1.1 and Corollary 2.1 and Theorem 2.1 (see [3]), we have the following result.

Theorem 6. *If there exists one positive constant η_2 such that* $0 < \lim\limits_{y\to\infty} \frac{1}{y} \int_0^y \eta_2(s)ds = \eta_2 < +\infty$, *then exists a positive projection L of rank one, and suitable constants $\delta > 0$, $M \geq 0$ such that*

$$\|T(t) - L\| \leq Me^{-\delta t},$$

where $L = \frac{1}{2\pi i} \int_{\overline{T}} (2I - A - B)^{-1} dz$, $\overline{T}$ *is a circle with center* 0 *and sufficiently small radius.*

Combining Remark2.1 with Theorem 14 (see [5]), Theorem 2.4, Theorem 2.10 (in Ref.[3]), we have the following result.

Theorem 7. *If there exists one positive constant η_2 such that* $0 < \lim\limits_{y\to\infty} \frac{1}{y} \int_0^y \eta_2(s)ds = \eta_2 < +\infty$, *then the time dependent solution of the system (1)-(7) strongly converges to its steady-state solution that is*

$$\lim_{t\to\infty} P(y,t) = \widehat{p}$$

and $\|P(y,t) - \widehat{p}\| \leq Ce^{-\varepsilon t}$, $\varepsilon > 0$, $C \leq 1$, *where $\widehat{p}$ is the eigenvector corresponding to* 0.

Proof. By using Theorem 2.10 (see [3]), and Theorem 2.4, we have

$$T(t) = T_1(t) + R(t)$$

where $T_1(t) = L$, L is the positive projection of 0, $\|R(t)\| \leq Ce^{-\varepsilon t}$, $\varepsilon > 0$, $C \geq 1$. Then

$$P(y,t) = T(t)P(0) =< P(0), Q > \widehat{p} + R(t)P(0) = \widehat{p} + R(t)P(0)$$

where $P(0) = (1, 0, 0, 0)$, $Q = (1, 1, 1, 1)$ is the eigenvector corresponding eigenvalue 0 of the adjoint matrix $(A + B)^*$.

So, the exponential asymptotical stability of the solution of the electronic equipment system with two units standby is obtained. □

Acknowledgements. This work is supported by Daqing Normal University Young Fund (No.YZQ008) and Heilongjiang Province Natural Science Fund (No.A200813).

References

1. Gupta, P.P., Kumar, A.: Point-wise Availability of a Two-Unit Standby Redundant Electronic Equipment under Human Failure. Microelectron. Reliab. 26(2), 225–228 (1986)
2. Hu, W.W.: Asymptotic Stability Analysis of a Parallel Repairable System with Warm Standby under Common-Cause Failures. Acta Anal. Funct. Appl. 8(1), 5–20 (2006)
3. Nager, R.: One-Parameter Semigroup of Positive Operators. Springer, New York (1986)
4. Xu, H.B., Liu, H.L., Yu, J.Y., Zhu, G.T.: The Asymptotic Sability of a Man-Machice System with Critical and Non-Critical Human Error. System Sci. Math. 25(5), 513–524 (2005)
5. Gupur, G., Li, X.Z., Zhu, G.T.: Functional Analysis Method in Queueing Theory. Research Information Ltd., Hertfordshire (2001)
6. Webb, G.F.: Theory of Nonlinear Age-Dependent Population Dynamics. Marcel Dekker, New York (1985)

Globally Exponentially Attractive Set and Synchronization of a Class of Chaotic Finance System

Jigui Jian, Xiaolian Deng, and Jianfeng Wang

Institute of Nonlinear and Complex Systems, China Three Gorges University, Yichang, Hubei 443002, China
jiguijian65@hotmail.com

Abstract. This paper treats the globally exponentially attractive set and synchronization problem of a chaotic finance system. Firstly, based on the definition of globally exponentially attractive set and Lyapunov stability theory, a sufficient condition for the globally exponentially attractive set was given. Secondly, two control approaches, namely nonlinear feedback control of partial states and transmitted signal method of a single variable, are investigated. In both cases, sufficient conditions for the globally exponential synchronization of two chaotic finance systems are obtained analytically. Finally, numerical simulation results indicates the effectiveness of the proposed methods.

Keywords: chaotic finance system, globally exponentially attractive set, globally exponentially synchronization, nonlinear feedback control, partial state feedback, transmitted signal method.

1 Introduction

Chaos, as a very interesting nonlinear phenomenon, has been intensively studied in the last decades. It is found to be either useful or has great potential in many fields, such as in engineering, biology, physics, chemistry and secure communication. It has been well known that a chaotic system is a nonlinear deterministic system with complex and unpredictable behavior. Chaotic systems exhibit sensitive dependence on initial conditions. Because of this property, chaotic systems are difficult to be synchronized or controlled. From the earlier works, the researchers have realized that synchronization of chaotic motions are possible, synchronization of chaos was of great interest in these years [1-5, 14-15].

Recently, there are two main approaches for controlling chaos: non-feedback control and feedback control. The concept of chaos synchronization involves making two chaotic systems which oscillate in a synchronized manner. Generally speaking, the synchronization phenomenon has the following feature: the trajectories of the drive and response systems are identical notwithstanding starting from different initial conditions. However, slight errors of initial conditions, for chaotic dynamical systems, will lead to completely different trajectories. Therefore, how to control of two chaotic systems to be synchronized has been a flurry

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 253–261, 2009.

of research activities for over a decade. Many approaches [1-5, 14-15] have been presented for the synchronization of chaotic systems such as linear and nonlinear feedback control. At the same time, a chaotic system is also a dissipative system, the global dissipativity of neural networks is studied by Liao and Wang [12] and the globally attractive set and positive invariant set of Lorenz system are further investigated by Liao [13,14].

In paper [6], a new system named the finance system is set up, which is a third order autonomous system exhibiting very complex dynamical behaviors with some interesting characteristics [7-9]. Its dynamical behaviors were further explored and achieved chaotic synchronization for the finance system by applied adaptive synchronization [15].

The layout of the rest of this paper is as follows. In Section 2, a brief description of a new chaotic finance system is introduced. In Section 3, based on the Lyapunov stability theory, a generic conditions of globally exponentially attractive set for the finance system is derived. In Section 4, the globally exponential synchronization of the chaotic finance system via nonlinear feedback control partial states and transmitted signal method of a single variable are investigated. In Section 5, numerical simulations for given chaotic finance system are presented. Finally, the conclusion of this paper is given in Section 6.

2 Chaotic Finance System and Preliminaries

Chaotic finance model consisted of production, currency and labor force can be described by the following system of differential equations: [6]

$$\begin{cases} \dot{x} = -ax + xy + z, \\ \dot{y} = 1 - by - x^2, \\ \dot{z} = -x - cz, \end{cases} \tag{1}$$

where $x(t)$ is interest rate, $y(t)$ is the investment demand, $z(t)$ is the price exponent; a, b and c are parameters, and $a > 0$ is the saving amount, $b > 0$ is the per-investment cost, $c > 0$ is the elasticity of demands of commercials. When the system parameters $a = 0.9$, $b = 0.2$, $c = 1.2$, and the time history of the finance system with initial condition $(x(0), y(0), z(0)) = (2, 1, 2)$, this system (1) exhibits chaotic behavior [5,15] and see Figure 1. Without the particular statement, these values are adopted in this whole paper.

Definition 1.[13] For generalized radially unbound and positive definite function $V_\lambda(X) = V_\lambda(x, y, z)$ with $\lambda \geq 0$, if there exists a constant number $L_\lambda > 0$ such that for $V_\lambda(X_0) > L_\lambda$ and $V_\lambda(X) > L_\lambda$ imply $\lim_{t\to+\infty} V_\lambda(X(t)) = L_\lambda$, then $\Omega_\lambda = \{X|V_\lambda(X(t)) \leq L_\lambda\}$ is said to be a globally attractive set of system (1). If for any $X_0 \in \Omega_\lambda$ and any $t \geq t_0$ imply $X(t, t_0, X_0) \in \Omega_\lambda$, then Ω_λ is said to be positive invariant set. If there exists constant numbers $L_\lambda > 0$, $r_\lambda > 0$ and $\forall X_0 \in R^3$ such that for $V_\lambda(X_0) > L_\lambda$ and $V_\lambda(X(t)) > L_\lambda$ imply $V(X(t)) - L_\lambda \leq (V(X_0) - L_\lambda)e^{-r_\lambda(t-t_0)}$, then $\Omega_\lambda = \{X|V_\lambda(X(t)) \leq L_\lambda\}$ is said to be a globally exponentially attractive set of system (1). Where $X = (x, y, z)$.

3 Globally Exponentially Attractive Set of Chaotic Finance System

Theorem 1. For $\forall\lambda \geq 0$, let $L_\lambda = \frac{2ac^2-(c-1)^2}{2(2a-1)(2c-1)}\lambda^2 + \frac{1}{2(2b-1)} > 0$ with $a > \frac{1}{2}$, $b > \frac{1}{2}$ and $c > \frac{1}{2}$, $V_\lambda = \frac{1}{2}x^2 + \frac{1}{2}y^2 + \frac{1}{2}(z-\lambda)^2$, then the system (1) has the following estimate inequality for globally exponentially attractive set

$$V(X(t)) - L_\lambda \leq (V(X_0) - L_\lambda)e^{-(t-t_0)}.$$

In specially, $\Omega_\lambda = \{X|V_\lambda(X) \leq L_\lambda\}$.

Proof. Consider the following generalized radially unbound and positive definite Lyapunov function

$$V_\lambda = \frac{1}{2}x^2 + \frac{1}{2}y^2 + \frac{1}{2}(z-\lambda)^2.$$

Let $F(X) = \frac{1}{2}(1-2a)x^2 + \frac{1}{2}(1-2b)y^2 + \frac{1}{2}(1-2c)z^2 + \lambda x + y + \lambda(c-1)z + \frac{1}{2}\lambda^2$. Evaluating the time derivative of V_λ along the positive semi-trajectory of (1), we have

$$\begin{aligned}\tfrac{dV_\lambda}{dt}|_{(1)} &= x\dot{x} + y\dot{y} + (z-\lambda)\dot{z} \\ &= x(-ax + xy + z) + y(1 - by - x^2) + (z-\lambda)(-x - cz) \\ &= -V_\lambda(X) + \tfrac{1}{2}(1-2a)x^2 + \tfrac{1}{2}(1-2b)y^2 + \tfrac{1}{2}(1-2c)z^2 \\ &\quad +\lambda x + y + \lambda(c-1)z + \tfrac{1}{2}\lambda^2 \\ &= -V_\lambda(X) + F(X).\end{aligned}$$

Let $\frac{\partial F}{\partial x} = (1-2a)x + \lambda = 0$, $\frac{\partial F}{\partial y} = (1-2b)y + 1 = 0$, $\frac{\partial F}{\partial z} = (1-2c)z + \lambda(c-1) = 0$, we get

$$x = \frac{\lambda}{2a-1}, y = \frac{1}{2b-1}, x = \frac{\lambda(c-1)}{2c-1}.$$

From $a > \frac{1}{2}$, $b > \frac{1}{2}$, $c > \frac{1}{2}$, we have

$$\frac{\partial^2 F}{\partial x^2} = 1 - 2a < 0, \frac{\partial^2 F}{\partial y^2} = 1 - 2b < 0, \frac{\partial^2 F}{\partial z^2} = 1 - 2c < 0, \frac{\partial^2 F}{\partial x \partial y} = \frac{\partial^2 F}{\partial y \partial z} = \frac{\partial^2 F}{\partial z \partial x} = 0.$$

Because $F(X) = F(x, y, z)$ is a quadratic function, its local maximum is also its global maximum, so we get

$$\begin{aligned}\sup_{X \in R^3} F(x, y, z) &= F(x, y, z)|_{x=\frac{\lambda}{2a-1}, y=\frac{1}{2b-1}, x=\frac{\lambda(c-1)}{2c-1}} \\ &= \tfrac{\lambda^2}{2(2a-1)} + \tfrac{\lambda^2(c-1)^2}{2(2c-1)} + \tfrac{1}{2(2b-1)} + \tfrac{1}{2}\lambda^2 \\ &= \tfrac{2ac^2-(c-1)^2}{2(2a-1)(2c-1)}\lambda^2 + \tfrac{1}{2(2b-1)} = L_\lambda > 0,\end{aligned}$$

and

$$\frac{dV_\lambda}{dt}|_{(1)} \leq -V_\lambda(X) + L_\lambda \leq 0,$$

$$V_\lambda(X) \leq V_\lambda(X_0)e^{-(t-t_0)} + \int_{t_0}^{t} e^{-(t-s)}L_\lambda ds = V_\lambda(X_0)e^{-(t-t_0)} + L_\lambda(1 - e^{-(t-t_0)}).$$

When $V_\lambda(X_0) > L_\lambda$ and $V_\lambda(X(t)) > L_\lambda$, we get the following globally exponential estimate

$$V_\lambda(X(t)) - L_\lambda \leq (V_\lambda(X_0) - L_\lambda)e^{-(t-t_0)},$$

and

$$\overline{\lim_{t\to+\infty}} V_\lambda(X(t)) \leq L_\lambda,$$

i.e. $\Omega_\lambda = \{X|V_\lambda(X(t)) \leq L_\lambda\} = \{X|x^2+y^2+(z-\lambda)^2 \leq \frac{2ac^2-(c-1)^2}{2(2a-1)(2c-1)}\lambda^2+\frac{1}{2(2b-1)}\}$ is the globally exponentially attractive set of system (1).

4 Chaos Synchronization of Coupled Finance Systems

4.1 Globally Exponential Synchronization via Nonlinear Feedback of Partial System States

We assume that we have two systems and that the drive system with the subscript 1 is to control the response system with subscript 2, then the drive system (2) and the response system (3) are defined as follows, respectively:

$$\begin{cases} \dot{x_1} = -ax_1 + x_1y_1 + z_1, \\ \dot{y_1} = 1 - by_1 - x_1^2, \\ \dot{z_1} = -x_1 - cz_1, \end{cases} \tag{2}$$

and

$$\begin{cases} \dot{x_2} = -ax_2 + x_2y_2 + z_2 + u_1(t), \\ \dot{y_2} = 1 - by_2 - x_2^2 + u_2(t), \\ \dot{z_2} = -x_2 - cz_2 + u_3(t). \end{cases} \tag{3}$$

We have introduced three control functions $u_1(t)$, $u_2(t)$ and $u_3(t)$ in (3). Our goal is to determine these functions $u_1(t)$, $u_2(t)$ and $u_3(t)$. In order to estimate the control functions, we subtract (2) from (3). We define the error system as the differences between the finance systems (2) and (3) that is to be controlled and the controlling system using $e(t) = (e_1, e_2, e_3)^T = (x_2-x_1, y_2-y_1, z_2-z_1)^T$. Using this notation, we obtain

$$\begin{cases} \dot{e}_1 = -ae_1 + e_3 + e_1e_2 + y_1e_1 + x_1e_2 + u_1(t), \\ \dot{e}_2 = -be_2 - e_1^2 - 2x_1e_1 + u_2(t), \\ \dot{e}_3 = -e_1 - ce_3 + u_3(t). \end{cases} \tag{4}$$

Suppose that one of the following controllers is chosen as the control law for the system (4):
(A_1) $u_1 = x_1e_2 - y_1e_1 - ke_1, u_2 = 0, u_3 = 0$;
(A_2) $u_1 = -y_1e_1 - k_1e_1, u_2 = x_1e_1 - k_2e_2, u_3 = 0$.

Theorem 2. The origin of system (4) is exponentially asymptotically stable, and consequently, the two finance systems (2) and (3) can be exponentially synchronized if there exists constant $k > \frac{(d-1)^2}{4cd} - a$ for any $d > 0$ such that (A_1) can be satisfied or there exist constant $k_1 > \frac{(d-1)^2}{4cd} - a$ for any $d > 0$ and $k_2 > -b$ such that (A_2) can be satisfied.

Proof. Construct a Lyapunov function in the form of $V = \frac{1}{2}e_1^2 + \frac{1}{2}e_2^2 + \frac{1}{2}de_3^2$. If (A_1) is satisfied, we have

$$\begin{aligned}\frac{dV_\lambda}{dt}|_{(4)} &= e_1\dot{e_1} + e_2\dot{e_2} + de_3\dot{e_3} \\ &= -(a+k)e_1^2 - be_2^2 - dce_3^2 + (1-d)e_1e_3 \\ &= -(e_1,e_2,e_3)^T Q_1(e_1,e_2,e_3),\end{aligned}$$

where $Q_1 = \begin{pmatrix} a+k & 0 & \frac{d-1}{2} \\ 0 & b & 0 \\ \frac{d-1}{2} & 0 & dc \end{pmatrix}$. It is easy to show that the matrix Q_1 be positive definite if and only if the following inequalities hold:

$$k + a > 0, 4dc(k+a) > (d-1)^2.$$

Then the matrix Q_1 is positive definite and $\dot{V}|_{(4)}$ is negative definite, we have

$$\dot{V}|_{(4)} \leq -\frac{2\lambda_{min}(Q_1)}{max\{1,d\}}V,$$

i.e.

$$e_1^2 + e_2^2 + e_3^2 \leq (e_1^2(t_0) + e_2^2(t_0) + e_3^2(t_0))\frac{max\{1,d\}}{min\{1,d\}}e^{-\frac{2\lambda_{min}(Q_1)}{max\{1,d\}}(t-t_0)}. \quad (5)$$

If (A_2) is satisfied, we have

$$\begin{aligned}\frac{dV_\lambda}{dt}|_{(4)} &= e_1\dot{e_1} + e_2\dot{e_2} + de_3\dot{e_3} \\ &= -(a+k_1)e_1^2 - (b+k_2)e_2^2 - dce_3^2 + (1-d)e_1e_3 \\ &= -(e_1,e_2,e_3)^T Q_2(e_1,e_2,e_3),\end{aligned}$$

where $Q_2 = \begin{pmatrix} a+k_1 & 0 & \frac{d-1}{2} \\ 0 & b+k_2 & 0 \\ \frac{d-1}{2} & 0 & dc \end{pmatrix}$. It is easy to show that the matrix Q_2 be positive definite if and only if the following inequalities hold:

$$k_1 + a > 0, k_2 + b > 0, 4dc(k_1+a) > (d-1)^2.$$

Then the matrix Q_2 is positive definite and $\dot{V}|_{(4)}$ is negative definite, we have

$$\dot{V}|_{(4)} \leq -\frac{2\lambda_{min}(Q_2)}{max\{1,d\}}V,$$

i.e.

$$e_1^2 + e_2^2 + e_3^2 \leq (e_1^2(t_0) + e_2^2(t_0) + e_3^2(t_0))\frac{max\{1,d\}}{min\{1,d\}}e^{-\frac{2\lambda_{min}(Q_2)}{max\{1,d\}}(t-t_0)}. \quad (6)$$

The inequality (5) and (6) imply that the origin of the error system (4) is exponentially asymptotically stable. Therefore, the drive system (2) is exponentially synchronizing with the response system (3). This concludes the proof.

4.2 Globally Exponential Synchronization via Using x_1 as the Transmitted Signal

Theorem 3. Let x_1 is the transmitted signal, for the following response system

$$\begin{cases} \dot{x_2} = -ax_2 + x_1y_2 + z_2, \\ \dot{y_2} = 1 - by_2 - x_1^2, \\ \dot{z_2} = -x_1 - cz_2. \end{cases} \tag{7}$$

Then the drive system (2) is exponentially synchronizing with the response system (7) and the exponential convergence rate is at lest $r = \min\{a, b, c\}$.

Proof. Let $e_1 = x_2 - x_1, e_2 = y_2 - y_1, e_3 = z_2 - z_1$, then the error system can be expressed as follow

$$\begin{cases} \dot{e_1} = -ae_1 + x_1e_2 + e_2, \\ \dot{e_2} = -be_2, \\ \dot{e_3} = -ce_3. \end{cases} \tag{8}$$

From (8), we have

$$e_2(t) = e_2(0)e^{-bt}, e_3(t) = e_3(0)e^{-ct},$$

$$e_1(t) = e_1(0)e^{-at} + \int_0^t e^{-a(t-s)}(x_1(s)e_2(s) + e_3(s))ds.$$

Let $\delta = \min\{b, c\}, \mu = M_x|e_2(0)| + |e_3(0)|$, where M_x is the upper boundary of the absolute values of variables x_1, then

$$\begin{aligned} |e_1(t)| &\le |e_1(0)|e^{-at} + \textstyle\int_0^t e^{-a(t-s)}(M_x|e_2(0)|e^{-bs} + |e_3(0)|e^{-cs})ds \\ &\le |e_1(0)|e^{-at} + \mu e^{-at} \textstyle\int_0^t e^{(a-\delta)s}ds \\ &= |e_1(0)|e^{-at} + \frac{\mu}{a-\delta}(e^{-\delta t} - e^{-at}). \end{aligned}$$

So $e_1(t)$ also exponentially tend to zero when t tend to infinity, i.e. the drive (2) is exponentially synchronizing with the response (7) using x_1 as the transmitted signal.

5 Numerical Simulation

To verify the theoretical results given in the previous sections, we will discuss the simulation results for the finance systems with the parameters $a = 0.9, b = 0.2, c = 1.2$. Let $d = 5, k = k_1 = k_2 = 1$. Then the response system (3) and (7) synchronize with the drive system (2) as shown in Fig. 2 and Fig. 3, respectively.

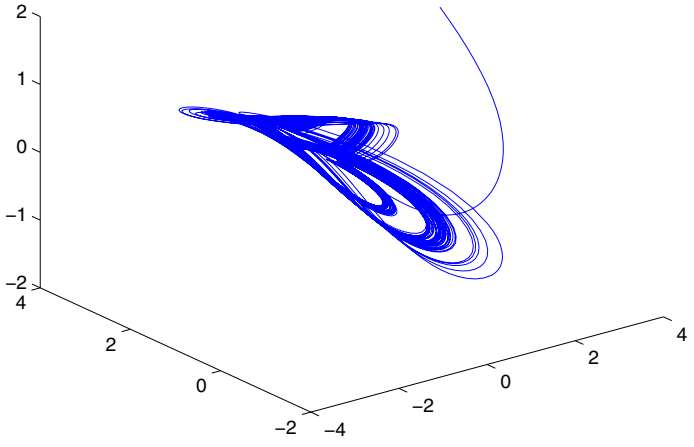


Fig. 1. Finance system (1) exhibits chaotic behavior

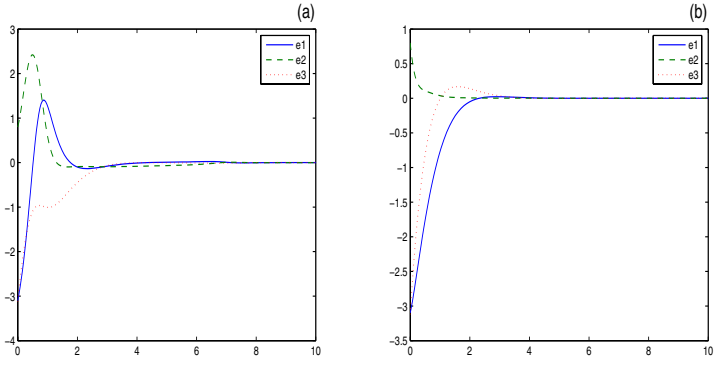


Fig. 2. (a) Shows the behaviors of the trajectory e_1, e_2, e_3 of the error system with the control (A_1); (b) Shows the behaviors of the trajectory e_1, e_2, e_3 of the error system with the control (A_2)

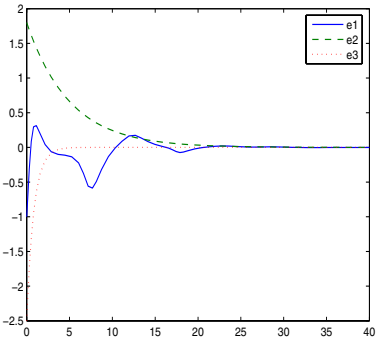


Fig. 3. The trajectory e_1, e_2, e_3 of the error system using x_1 as the transmitted signal

6 Conclusion

In this paper, according to the parameters, detailed estimations of exponential attractive sets for chaotic finance system are presented without any hypothesis on the existence. Meanwhile, two control approaches, namely nonlinear feedback control of partial states and transmitted signal method of a single variable, are investigated and some sufficient conditions for the globally exponential synchronization of two chaotic finance systems are obtained analytically. All the numerical simulation results are in line with the theoretical analysis.

Acknowledgments

This work was partially supported by the Scientific Innovation Team Project T200809 and the Scientific Research Projects D20081306 of Hubei Provincial Department of Education, and the Doctoral Pre-Research Foundation of China Three Gorges University.

References

1. Li, G.H.: Generalized Projective Synchronization between Lorenz System and Chen's system. Chaos, Solitions and Fractals 32, 1454–1448 (2007)
2. Li, C.D., Liao, X.F.: Lag Synchronization of Rossler System and Chua Circuit via A Scalar Signal. Physics Letters A 392, 301–308 (2004)
3. Wang, Y.W., Wen, C.Y., Soh, Y.C., Xiao, J.W.: Adaptive Control and Synchronization for A Class of Nonlinear Chaotic Systems using Partial System States. Physics Letters A 351, 79–84 (2006)
4. Li, Y.N., Chen, L., Cai, Z.S., Zhao, X.Z.: Experimental Study of Chaos Synchronization in the Belousov-Zhabotinsky Chemical System. Chaos, Solitons and Fractals 22, 767–771 (2004)
5. Mei, X.H., Yu, J.N., Zhang, J.G.: The Linear and Nonlinear Feedback Synchronization of a Class of Chaotic Finance System. Journal of Tianjin Normal University (Natural Science Edition) 28(3), 49–51 (2008)
6. Huang, D.S., Li, H.Q.: Theory and Method of the Nonlinear Economics. Publishing House of Sichuan University, Chengdu (1993)
7. Ma, J.H., Chen, Y.S.: Study for the Bifurcation Topological Structure and the Global Complicated Character of a Kind of Non-Linear Finance System (I). Applied Mathematics and Mechanics 22(12), 1119–1128 (2001)
8. Ma, J.H., Chen, Y.S.: Study for the Bifurcation Topological Structure and the Global Complicated Character of a Kind of Non-Linear Finance System (II). Applied Mathematics and Mechanics 22(12), 1236–1242 (2001)
9. Shang, R.B., Wang, X.D., Wang, Q.Y., Ma, J.H.: Study on the Complexity and Topology Structure of Bifurcation of a Kind Finance System (III). Journal of Tianjin University 36(2), 234–238 (2003)
10. Yu, P., Liao, X.X.: New Estimates for Globally Attractive and Positive Invariant Set of the Family of the Lorenz. Int. J. Bifurcation and Chaos 16(11), 3383–3390 (2006)

11. Li, D.M., Lu, J.A., Wa, X.Q., Chen, G.R.: Estimating the Bounded for the Lorenz Family of Chaotic System. Chaos, Solutions Fractals 23, 529–534 (2005)
12. Liao, X.X., Wang, J.: Global dissipativity of continuous-time recurrent neural networks. Phys. Rev. E 68, 161181–161187 (2003)
13. Liao, X.X., Luo, H.G., Fu, Y.L., Xie, S.L., Yu, P.: Positive Invariant Set and the Globally Exponentially Attractive Set of Lorenz System Group. Science in China-E 37(6), 757–769 (2007)
14. Liao, X.X.: New Results for Globally Attractive Set and Positive Invariant Set of Lorenz System and Application of Chaos Control and Synchronization. Science in China-E 34(12), 1404–1419 (2004)
15. Sun, M., Tian, L.X.: Adaptive synchronization of nonlinear chaotic finance system. Journal of Jiangsu University (Natural Science Edition) 26(6), 488–491 (2005)

Continuous and Discrete Halanay Delayed Inequalities and Their Applications in Stability of Neural Networks

Jinhua Huang and Jiqing Liu

Department of Electric and Electronic Engineering
Wuhan Institute of Shipbuilding Technology, Wuhan, Hubei, 430050, China
Angela_icec@yahoo.com.cn,
LJQ6521@public.wh.hb.cn

Abstract. In this paper, a slight generalization of the celebrated continuous Halanay inequality is developed, a discrete analogue of the continuous Halanay inequality is proved. Using continuous and discrete Halanay time-delayed inequalities, we derived some sufficient conditions for the global exponential (asymptotic) stability of equilibrium point of neural networks.

Keywords: Halanay inequality, Discrete, Continuous, Stability.

1 Introduction

Stability theory plays an important role in systems theory and engineering. The direct method of Lyapunov for ordinary differential equations has been generalized to delayed differential equations, serving as the basic tool for stability investigations for delayed differential equations. In addition, the method of differential (difference) inequalities (different from the Lyapunov method) is also a basic tool in the qualitative analysis of the solutions to differential (difference) systems.

In [1], Halanay proved an asymptotic formula for the solutions of a differential inequality involving the "maximum" functional, and applied it in the stability theory of linear systems with delay. Such inequality was called Halanay inequality in several works [2]-[13], in which some generalizations and new applications can be found. In particular, in [6], Cooke and Ivanov consider discrete Halanay-type inequalities in order to study some discretized versions of functional differential equations. In [13], Liz and Ferreiro give a simple discrete version of Halanay's lemma, and to apply it to obtain results on the global asymptotical stability of certain generalized difference equations, in addition, they show that the classical criterion on the absolute stability (or delay-independent stability) in certain delay equations holds for the discretized equation using the Euler scheme if the discretization step is small enough.

In addition, recurrent neural networks have shown their promise and power in the variety of important applications. As dynamic systems, recurrent neural

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 262–271, 2009.

networks frequently need to be analyzed for stability. The stability criteria of equilibrium points are established in a series of papers; e.g., [14]-[16].

In this paper, we obtain a slight generalization of the celebrated continuous Halanay inequality and a discrete analogue of the continuous Halanay inequality. Using continuous and discrete Halanay delayed inequalities, we derived some sufficient conditions for the global exponential (asymptotic) stability of equilibrium point of neural networks.

2 Continuous Halanay Inequality and Its Application

2.1 Continuous Halanay Inequality

Consider the following differential inequalities with time-varying delays

$$D^+x_i(t) \le \sum_{j=1}^{n} a_{ij}x_j(t) + \sum_{j=1}^{n} b_{ij}x_j(t-\delta_j(t)), \tag{1}$$

$$D^+x_i(t) \ge \sum_{j=1}^{n} a_{ij}x_j(t) + \sum_{j=1}^{n} b_{ij}x_j(t-\delta_j(t)), \tag{2}$$

$$\begin{cases} D^+x_i(t) \le a_{ii}x_i(t) + \sum_{j=1,j\neq i}^{n} a_{ij}|x_j(t)| + \sum_{j=1}^{n} b_{ij}|x_j(t-\delta_j(t))|, \\ D^+x_i(t) \ge a_{ii}x_i(t) - \sum_{j=1,j\neq i}^{n} a_{ij}|x_j(t)| - \sum_{j=1}^{n} b_{ij}|x_j(t-\delta_j(t))|, \end{cases} \tag{3}$$

where for all $i, j \in \{1, 2, \cdots, n\}$, $a_{ii} < 0; a_{ij} \ge 0, (i \neq j); b_{ij} \ge 0$, and

$$D^+x_i(t) = \lim_{h\to 0^+} \sup \frac{x_i(t+h) - x_i(t)}{h}$$

is the right-hand upper Dini derivative of $x_i(t)$.

In this paper, we always assume that

H) For $i = 1, 2, \cdots, n$, delay $\delta_i(t) : \Re \to \Re_+$ is a continuous function such that $\delta_i(t) < \delta$ (constant), where $\Re_+ = [0, +\infty), \Re_- = (-\infty, 0]$.

Denote $A = (a_{ij})_{n\times n}, B = (b_{ij})_{n\times n}$, $\bar{x}_i(t_0) = \sup_{t_0-\delta_i(t_0)\le\xi\le t_0} |x_i(\xi)| > 0, \bar{x}(t_0) = \max_{1\le i\le n}\{\bar{x}_i(t_0)\}$, Then we have the following slight generalization of the Halanay differential inequality.

Theorem 1. If for all $i \in \{1, 2, \cdots, n\}$, $x_i(t), (t \ge t_0 - \delta_i(t_0))$ are continuous functions satisfying (1), and $-A-B$ is a nonsingular M-matrix, then there exist positive constants β and θ such that for $\forall i \in \{1, 2, \cdots, n\}, \forall t \ge t_0$,

$$x_i(t) \le \beta\bar{x}(t_0)\exp\{-\theta(t-t_0)\}, \tag{4}$$

Proof. Since $-A-B$ is a nonsingular M-matrix, there exist $\gamma_1, \gamma_2, \cdots, \gamma_n$ such that for $\forall i \in \{1, 2, \cdots, n\}$,

$$\gamma_i a_{ii} + \sum_{j=1,j\neq i}^{n} \gamma_j a_{ij} + \sum_{j=1}^{n} \gamma_j b_{ij} < 0.$$

Let

$$\eta_i(\vartheta) = \gamma_i(a_{ii} + \vartheta) + \sum_{j=1, j\neq i}^{n} \gamma_j a_{ij} + \sum_{j=1}^{n} \gamma_j b_{ij} \exp\{\vartheta\delta\},$$

then $\eta_i(0) = \gamma_i a_{ii} + \sum_{j=1,j\neq i}^{n} \gamma_j a_{ij} + \sum_{j=1}^{n} \gamma_j b_{ij} < 0$, $\eta_i(-a_{ii}) \geq 0$. Hence, there exists $\theta_i \in (0, -a_{ii}]$ such that $\eta_i(\theta_i) = 0$, (we can choose the value of $\theta_i \in (0, -a_{ii})$ satisfying this equation since $\eta_i(\vartheta)$ is an increasing function), and $\eta_i(\vartheta) \leq 0$, $\vartheta \in (0, \theta_i]$. Choose $\theta = \min_{1\leq i\leq n}\{\theta_i\}$, then for $\forall j \in \{1, 2, \cdots, n\}$,

$$\eta_j(\theta) \leq 0. \tag{5}$$

Let $y_i(t) = x_i(t)/\gamma_i$, then according to (1),

$$D^+ y_i(t) \leq [\sum_{j=1}^{n} a_{ij}\gamma_j y_j(t) + \sum_{j=1}^{n} b_{ij}\gamma_j y_j(t - \delta_j(t))]/\gamma_i. \tag{6}$$

Denote $\bar{y}_i(t_0) = \sup_{t_0-\delta_i(t_0)\leq\xi\leq t_0} |y_i(\xi)| > 0, \bar{y}(t_0) = \max_{1\leq i\leq n}\{\bar{y}_i(t_0)\}$, $g_i(t) = y_i(t) - \bar{y}(t_0)\exp\{-\theta(t-t_0)\}, g(t) = (g_1(t), g_2(t), \cdots, g_n(t))^T$. We will prove that for $\forall t \geq t_0$, $g(t) \leq 0$. Otherwise, let $t_1 = \min_{1\leq i\leq n}\{\inf\{t, g_i(t) > 0\}\}$, on the one hand, since $g(t)$ is a continuous vector function, there exist $k \in \{1, 2, \cdots, n\}$, $t_2 > t_1 > t_0$ and small enough $\epsilon > 0$ such that

$$g_k(t_2) = \epsilon, g_k(t_1) = 0 \tag{7}$$

$$g_j(t) \leq 0, \text{ for } j \in \{1, 2, \cdots, n\}, t \in [t_0 - \delta_j(t_0), t_1], \tag{8}$$

$$g_j(t) \leq \epsilon, \text{ for } j \in \{1, 2, \cdots, n\}, t \in [t_1, t_2], \tag{9}$$

and

$$D^+ g_k(t)|_{t=t_2} > 0. \tag{10}$$

On the other hand, (6) implies that

$$\begin{aligned} D^+ g_k(t)|_{t=t_2} &= D^+ y_k(t)|_{t=t_2} + \theta\bar{y}(t_0)\exp\{-\theta(t_2 - t_0)\} \\ &= D^+ y_k(t)|_{t=t_2} + \theta(y_k(t_2) - \epsilon) \\ &\leq [a_{kk} + \theta]y_k(t_2) + [\sum_{j=1,j\neq k}^{n} a_{kj}\gamma_j y_j(t_2) \\ &\quad + \sum_{j=1}^{n} b_{kj}\gamma_j y_j(t_2 - \delta_j(t_2))]/\gamma_k - \theta\epsilon. \end{aligned}$$

By (8) and (9), for $j = 1, 2, \cdots, n$, $y_j(t_2) \leq \bar{y}(t_0)\exp\{-\theta(t_2 - t_0)\} + \epsilon$, $y_j(t_2 - \delta_j(t_2)) \leq \bar{y}(t_0)\exp\{-\theta(t_2 - \delta_j(t_2) - t_0)\} + \epsilon$. Hence, $a_{kj} \geq 0, (k \neq j)$; $b_{kj} \geq 0$

imply that

$$D^+g_k(t)|_{t=t_2} \le \{(a_{kk}+\theta)+[\sum_{j=1,j\neq k}^{n}\gamma_j a_{kj} + \sum_{j=1}^{n}\gamma_j b_{kj}\exp\{\theta\delta_j(t_2)\}]/\gamma_k\}\bar{y}(t_0)\exp\{-\theta(t_2-t_0)\} + \{a_{kk}+[\sum_{j=1,j\neq k}^{n}\gamma_j a_{kj}+\sum_{j=1}^{n}\gamma_j b_{kj}]/\gamma_k\}\epsilon. \quad (11)$$

Then, by (5), it follows that $D^+g_k(t)|_{t=t_2} \le 0$. This contradicts (10). Hence $x_i(t) = \gamma_i y_i(t) \le \gamma_i \bar{y}(t_0)\exp\{-\theta(t-t_0)\}$. Choose $\beta = \max_{1\le i\le n}\{\gamma_i\bar{y}(t_0)\}/\bar{x}(t_0)$, then (4) holds, this complete the proof.

Theorem 2. If for all $i \in \{1,2,\cdots,n\}$, $x_i(t), (t \ge t_0 - \delta_i(t_0))$ are continuous functions satisfying (2), and $-A-B$ is a nonsingular M-matrix, then there exist positive constants β and θ such that for $\forall i \in \{1,2,\cdots,n\}, \forall t \ge t_0$,

$$x_i(t) \ge -\beta\bar{x}(t_0)\exp\{-\theta(t-t_0)\}. \quad (12)$$

Proof. Since $-A-B$ is a nonsingular M-matrix, it is similar to the proof of Theorem 1 that there exist positive constants $\gamma_1,\gamma_2,\cdots,\gamma_n,\theta$ such that for $\forall i \in \{1,2,\cdots,n\}$,

$$\gamma_i(a_{ii}+\theta)+\sum_{j=1,j\neq i}^{n}\gamma_j a_{ij}+\sum_{j=1}^{n}\gamma_j b_{ij}\exp\{\theta\delta\} \le 0.$$

Let $z_i(t) = x_i(t)/\gamma_i$. Denote $\bar{z}_i(t_0) = \sup_{t_0-\delta_i(t_0)\le\xi\le t_0}|z_i(\xi)| > 0$, $\bar{z}(t_0) = \max_{1\le i\le n}\{\bar{z}_i(t_0)\}$. It is similar to the proof of Theorem 2 that $\bar{g}_i(t) = z_i(t) + \bar{z}(t_0)\exp\{-\theta(t-t_0)\} \ge 0$. Choose $\beta = \max_{1\le i\le n}\{\gamma_i\bar{z}(t_0)\}/\bar{x}(t_0)$, then (12) holds.

Theorem 3. If for all $i \in \{1,2,\cdots,n\}$, $x_i(t), (t \ge t_0 - \delta_i(t_0))$ are continuous functions satisfying (3), and $-A-B$ is a nonsingular M-matrix, then there exist positive constants β and θ such that for $\forall i \in \{1,2,\cdots,n\}, \forall t \ge t_0$,

$$|x_i(t)| \le \beta\bar{x}(t_0)\exp\{-\theta(t-t_0)\}.$$

Proof. It is similar to the proof of Theorems 1 and 2, hence it is omitted.

2.2 Exponential Stability of Neural Networks

Consider neural networks with time-varying delay

$$\frac{dx_i(t)}{dt} = a_{ii}x_i(t)+\sum_{j=1,j\neq i}^{n}a_{ij}f_j(x_j(t))+\sum_{j=1}^{n}b_{ij}g_j(x_j(t-\delta(t))), \quad (13)$$

where $i = 1, 2, \cdots, n, x(t) = (x_1(t), \cdots, x_n(t))^T$, $f(x(\cdot)) = \big(f_1(x_1(\cdot)), f_2(x_2(\cdot)), \cdots, f_n(x_n(\cdot))\big)^T$ and $g(x(\cdot)) = \big(g_1(x_1(\cdot)), g_2(x_2(\cdot)), \cdots, g_n(x_n(\cdot))\big)^T$ are activation functions which satisfies

$$\begin{aligned} &|\sum_{j=1,j\neq i}^{n} a_{ij} f_j(x_i(t)) + \sum_{j=1}^{n} b_{ij} g_j(x_j(t-\delta(t)))| \\ &\quad \leq \sum_{j=1,j\neq i}^{n} a_{ij}|x_j(t)| + \sum_{j=1}^{n} b_{ij}|x_j(t-\delta_j(t))|. \end{aligned} \tag{14}$$

For $H > 0$ and $t_0 \geq 0$, let $C_H(t_0)$ is the set of continuous function $\phi : [t_0-\delta, t_0] \to \Re^n, \phi = (\phi_1, \cdots, \phi_n)^T$, and it satisfies $||\phi||_{t_0} = \sup_{s\in[t_0-\delta,t_0]}\{\max_{1\leq i\leq n}\{|\phi_i(s)|\}\} < H$. For $t_0 \geq 0$, denote $x(t; t_0, \phi)$ be state of neural network (13), it means that $x(t; t_0, \phi)$ is continuous and satisfies (13) and $x(s; t_0, \phi) = \phi(s)$, for $s \in [t_0-\delta, t_0]$. Also simply denote $x(t)$ be state of neural network (13).

Theorem 4. If $-A - B$ is a nonsingular M-matrix, then there exist positive constants β and θ such that $\forall i \in \{1, 2, \cdots, n\}, \forall t \geq t_0$,

$$|x_i(t)| \leq \beta \bar{x}(t_0) \exp\{-\theta(t - t_0)\};$$

i.e., neural network (13) is globally exponentially stable.

Proof. From (13) and (14),

$$\begin{cases} D^+x_i(t) \leq a_{ii}x_i(t) + \sum_{j=1,j\neq i}^{n} a_{ij}|x_j(t)| + \sum_{j=1}^{n} b_{ij}|x_j(t-\delta_j(t))|, \\ D^+x_i(t) \geq a_{ii}x_i(t) - \sum_{j=1,j\neq i}^{n} a_{ij}|x_j(t)| - \sum_{j=1}^{n} b_{ij}|x_j(t-\delta_j(t))|. \end{cases}$$

According to Theorem 3, Theorem 4 holds.

3 Discrete Halanay Inequality and its Application

In this section, we give a discrete version of the original Halanay inequality, and we always assume that $t \in \mathcal{N}$, where $\mathcal{N}$ is the set of all natural number.

Consider difference equations

$$\Delta x_i(t) = f_i(t, x(t), x(t-1), \cdots, x(t-r)), \tag{15}$$

where $i = 1, 2, \cdots, m, \Delta x_i(t) = x_i(t+1) - x_i(t), x(t) = (x_1(t), x_2(t), \cdots, x_m(t))^T$, and $f_i : \mathcal{N} \times \Re^{(r+1)\times m} \to \Re$. Equation (15) is a class of generalized difference equation. The initial value problem for this equation requires the knowledge of initial data $\{x(-r), \cdots, x(0)\}$. This vector is called initial string in [6]. For every initial string, there exists a unique state $\{x(t)\}_{t\geq -r}$ of (15) that can be calculated by the explicit recurrence formula

$$x_i(t+1) = x_i(t) + f_i(t, x(t), x(t-1), \cdots, x(t-r)), \quad t \geq 0. \tag{16}$$

3.1 Discrete Halanay Inequality

Consider the following difference inequalities with delays

$$\Delta x_i(t) \le \sum_{j=1}^{m} a_{ij}x_j(t) + \sum_{j=1}^{m} b_{ij} \max_{1\le k\le r}\{x_j(t-k)\}, \tag{17}$$

$$\Delta x_i(t) \ge \sum_{j=1}^{m} a_{ij}x_j(t) + \sum_{j=1}^{m} b_{ij} \min_{1\le k\le r}\{x_j(t-k)\}, \tag{18}$$

$$\begin{cases} \Delta x_i(t) \le a_{ii}x_i(t) + \sum_{j=1,j\ne i}^{m} a_{ij}|x_j(t)| + \sum_{j=1}^{m} b_{ij} \max_{1\le k\le r}\{|x_j(t-k)|\}, \\ \Delta x_i(t) \ge a_{ii}x_i(t) - \sum_{j=1,j\ne i}^{m} a_{ij}|x_j(t)| - \sum_{j=1}^{m} b_{ij} \max_{1\le k\le r}\{|x_j(t-k)|\}, \end{cases} \tag{19}$$

where $i \in \{1, 2, \cdots, m\}$. Denote $\bar{x}^{(r)} = \max_{1\le i\le m}\{\max\{|x_i(0)|, |x_i(-1)|, \cdots, |x_i(-r)|\}\}$, $\bar{A} = (a_{ij})_{m\times m}$, $\bar{B} = (b_{ij})_{m\times m}$.

Theorem 5. Let $r > 0$ be a natural number, and let $\{x(t)\}_{t\ge -r}$ be a sequence of real number vectors satisfying the inequality (17). If $\forall i, j \in \{1, 2, \cdots, m\}$, $a_{ii} \in [-1, 0); a_{ij} \ge 0, (i \ne j); b_{ij} \ge 0$, and $-\bar{A} - \bar{B}$ is a nonsingular M-matrix, then there exist positive constants β and $\lambda_0 \in (0, 1)$ such that for $\forall i \in \{1, 2, \cdots, m\}$,

$$x_i(t) \le \beta \bar{x}^{(r)} \lambda_0^t, \quad t \ge 1.$$

Proof. Since $-\bar{A} - \bar{B}$ is a nonsingular M-matrix, there exist positive constants $\gamma_1, \gamma_2, \cdots, \gamma_m$ such that for $\forall i \in \{1, 2, \cdots, m\}$,

$$\gamma_i a_{ii} + \sum_{j=1, j\ne i}^{m} \gamma_j a_{ij} + \sum_{j=1}^{m} \gamma_j b_{ij} < 0. \tag{20}$$

Let

$$\eta_i(\lambda) = \gamma_i \lambda^{r+1} - \gamma_i(1 + a_{ii})\lambda^r - \sum_{j=1, j\ne i}^{m} \gamma_j a_{ij}\lambda^r - \sum_{j=1}^{m} \gamma_j b_{ij},$$

then $\eta_i(0) = -\sum_{j=1}^{m} \gamma_j b_{ij} \le 0$, $\eta_i(1) > 0$. Hence, there exists $\lambda_{0i} \in (0, 1)$ such that $\eta_i(\lambda_{0i}) = 0$, and $\eta_i(\lambda) \ge 0,\ \lambda \in [\lambda_{0i}, 1)$.

In fact, if $\eta_i(0) \ne 0$, we can choose the largest value of $\lambda \in (0, 1)$ satisfying $\eta_i(\lambda_{0i}) = 0$, since $\eta_i(\lambda)$ is a polynomial and it has at most $r + 1$ real roots; if $\eta_i(0) = 0$, we can choose $\lambda_{0i} = 1 + a_{ii} + (\sum_{j=1, j\ne i}^{m} \gamma_j a_{ij})/\gamma_i$. (20) implies $1 + a_{ii} + (\sum_{j=1, j\ne i}^{m} \gamma_j a_{ij})/\gamma_i < 1$; $a_{ii} \in [-1, 0)$ implies $1 + a_{ii} + (\sum_{j=1, j\ne i}^{m} \gamma_j a_{ij})/\gamma_i \ge 0$.

Choose $\lambda_0 = \max_{1\le i\le m}\{\lambda_{0i}\}$, then for $\forall j \in \{1, 2, \cdots, m\}$,

$$\eta_j(\lambda_0) \ge 0. \tag{21}$$

Let $y_i(t) = x_i(t)/\gamma_i$, then according to (17),

$$\Delta y_i(t) \le [\sum_{j=1}^{m} a_{ij}\gamma_j y_j(t) + \sum_{j=1}^{m} b_{ij} \max_{1\le k\le r}\{\gamma_j y_j(t-k)\}]/\gamma_i. \tag{22}$$

Let $\bar{y}^{(r)} = \max_{1\le i\le m}\{\max\{0, y_i(0), y_i(-1), \cdots, y_i(-r)\}\}$, then for all natural number t, $y_i(t) \le \bar{y}^{(r)}\lambda_0^t$. Otherwise, there exist $p \in \{1, 2, \cdots, m\}$ and natural number $q \ge 1$ such that $y_p(q) > \bar{y}^{(r)}\lambda_0^q$, and for all $j \ne p, j \in \{1, 2, \cdots, m\}$,

$$y_j(s) \le \bar{y}^{(r)}\lambda_0^s, \ -r \le s \le q; \ \ y_p(s) \le \bar{y}^{(r)}\lambda_0^s, \ -r \le s < q.$$

Hence, since $1 + a_{pp} \ge 0; a_{pj} \ge 0, (p \ne j); b_{pj} \ge 0$, from (22),

$$\begin{aligned}\bar{y}^{(r)}\lambda_0^q < y_p(q) &\le (1+a_{pp})\bar{y}^{(r)}\lambda_0^{q-1} + [\sum_{j=1, j\ne p}^{m} \gamma_j a_{pj}\bar{y}^{(r)}\lambda_0^{q-1} \\ &\quad + \sum_{j=1}^{m} \gamma_j b_{pj}\bar{y}^{(r)}\lambda_0^{q-1-r}]/\gamma_p \\ &\le \bar{y}^{(r)}\lambda_0^{q-1}\{(1+a_{pp}) + [\sum_{j=1, j\ne p}^{m} \gamma_j a_{pj} + \sum_{j=1}^{m}\gamma_j b_{pj}\lambda_0^{-r}]/\gamma_p\},\end{aligned}$$

i.e., $\gamma_p\lambda_0^{r+1} < [\gamma_p(1+a_{pp}) + \sum_{j=1, j\ne p}^{m}\gamma_j a_{pj}]\lambda_0^r + \sum_{j=1}^{m}\gamma_j b_{pj}$, this contradicts (21). Hence for all natural number $t \ge 1$, $x_i(t) = \gamma_i y_i(t) \le \gamma_i\bar{y}^{(r)}\lambda_0^t$. Choose $\beta = \max_{1\le i\le m}\{\gamma_i\bar{y}^{(r)}\}/\bar{x}^{(r)}$, the result of Theorem 5 holds.

Theorem 6. Let $r > 0$ be a natural number, and let $\{x(t)\}_{t\ge -r}$ be a sequence of real number vectors satisfying the inequality (18). If $a_{ii} \in [-1, 0), a_{ij} \ge 0, (i \ne j), b_{ij} \ge 0, -\bar{A}-\bar{B}$ is a nonsingular M-matrix, then there exist positive constants β and $\lambda_0 \in (0, 1)$ such that for $\forall i \in \{1, 2, \cdots, m\}$,

$$x_i(t) \ge -\beta\bar{x}^{(r)}\lambda_0^t, \quad t \ge 1.$$

Proof. Since $-\bar{A} - \bar{B}$ is a nonsingular M-matrix, it is similar to the proof of Theorem 5 that there exist positive constants $\gamma_1, \gamma_2, \cdots, \gamma_m$ and $\lambda_0 \in (0, 1)$ such that for $\forall i \in \{1, 2, \cdots, m\}$,

$$\gamma_i\lambda_0^{r+1} - \gamma_i(1+a_{ii})\lambda_0^r - \sum_{j=1, j\ne i}^{m}\gamma_j a_{ij}\lambda_0^r - \sum_{j=1}^{m}\gamma_j b_{ij} \ge 0.$$

Let $z_i(t) = x_i(t)/\gamma_i$. Denote $\bar{z}^{(r)} = \min_{1\le i\le m}\{\min\{0, z_i(0), z_i(-1), \cdots, z_i(-r)\}\} < 0$. It is similar to the proof of Theorem 5 that $x_i(t) = \gamma_i z_i(t) \ge \gamma_i\bar{z}^{(r)}\lambda_0^t, \ t \ge 1$. Choose $\beta = \max_{1\le i\le m}\{-\gamma_i\bar{z}^{(r)}\}/\bar{x}^{(r)}$, then the result of Theorem 6 holds.

For $i, j \in \{1, 2, \cdots, m\}$, let

$$c_{ij} = \begin{cases} -a_{ii} - b_{ii}, \ i = j, \\ -a_{ij} - b_{ij}, \ i \ne j, \end{cases} \qquad \tilde{c}_{ij} = \begin{cases} 2 + a_{ii} - b_{ii}, \ i = j, \\ -a_{ij} - b_{ij}, \quad i \ne j. \end{cases}$$

Denote matrices $C_1 = (c_{ij})_{m\times m}, C_2 = (\tilde{c}_{ij})_{m\times m}$.

Theorem 7. Let $r > 0$ be a natural number, and let $\{x(t)\}_{t\geq -r}$ be a sequence of real number vectors satisfying the inequality (19). If one of the following two cases holds:

case (i) $a_{ii} \in [-1, 0); a_{ij} \geq 0, (i \neq j); b_{ij} \geq 0$, C_1 is a nonsingular M-matrix;

case (ii) $a_{ii} \in (-2, -1); a_{ij} \geq 0, (i \neq j); b_{ij} \geq 0$, C_2 is a nonsingular M-matrix, then there exist positive constants β and $\lambda_0 \in (0, 1)$ such that for $\forall i \in \{1, 2, \cdots, m\}$,

$$|x_i(t)| \leq \beta \bar{x}^{(r)} \lambda_0^t, \quad t \geq 1.$$

Proof. When $a_{ii} \in [-1, 0)$, $|1 + a_{ii}| = 1 + a_{ii}$; when $a_{ii} \in (-2, -1)$, $|1 + a_{ii}| = -1 - a_{ii}$. It is similar to the proof of Theorems 5 and 6 that there exist positive constants β and $\lambda_0 \in (0, 1)$ such that for $\forall i \in \{1, 2, \cdots, m\}$ and all natural number $t \geq 1$, $|x_i(t)| \leq \beta \bar{x}^{(r)} \lambda_0^t$.

3.2 Asymptotic Stability of Discrete-Time Delayed Neural Networks

In this section, we consider a class of discrete-time delayed neural networks described by the following difference equation: for $i = 1, 2, \cdots, m$,

$$\Delta x_i(t) = a_{ii} x_i(t) + \sum_{j=1, j\neq i}^{m} \tilde{a}_{ij} f_j(x_j(t)) + \sum_{j=1}^{m} \tilde{b}_{ij} g_j(x_j(t - \tau_{ij}(t))), \tag{23}$$

where $x = (x_1, \cdots, x_m)^T \in \Re^m$ is the state vector, $\Delta x_i(t) = x_i(t+1) - x_i(t)$, $\forall t \geq 0, \forall i \in \{1, 2, \cdots, n\})$ $\tau_{ij}(t)$ is the time-varying delay that satisfies $0 \leq \tau_{ij}(t) \leq r = \max_{1\leq i,j\leq n}\{\sup\{\tau_{ij}(t), t \in \mathcal{N}\}\}$, f_j and g_j are neuron activation functions.

Although for every initial string $\{x(-r), \cdots, x(0)\}$, the state $\{x(t)\}_{t\geq -r}$ of (23) can be explicitly calculated by a recurrence similar to (16), in general it is difficult to investigate the asymptotic behaviour of the states using that formula. The next result gives an asymptotic estimate by a simple use of the discrete Halanay inequality.

Theorem 8. If $a_{ii} \in (-2, 0)$ and there exist constants $a_{ij} \geq 0, (i \neq j), b_{ij} \geq 0$, such that $\forall i \in \{1, 2, \cdots, m\}$,

$$\begin{aligned} &|\sum_{j=1, j\neq i}^{m} \tilde{a}_{ij} f_j(x_j(t)) + \sum_{j=1}^{m} \tilde{b}_{ij} g_j(x_j(t - \tau_{ij}(t)))| \\ &\leq \sum_{j=1, j\neq i}^{m} a_{ij} |x_j(t)| + \sum_{j=1}^{m} b_{ij} \max_{1\leq k\leq r}\{|x_j(t-k)|\}, \end{aligned} \tag{24}$$

in addition, one of the following two cases holds:

case (i) $a_{ii} \in [-1, 0); a_{ij} \geq 0, (i \neq j); b_{ij} \geq 0$, C_1 is a nonsingular M-matrix;

case (ii) $a_{ii} \in (-2, -1); a_{ij} \geq 0, (i \neq j); b_{ij} \geq 0$, C_2 is a nonsingular M-matrix, then (23) is globally asymptotically stable.

Proof. From (23) and (24),

$$\begin{cases} \Delta x_i(t) \le a_{ii}x_i(t) + \sum_{j=1,j\neq i}^{m} a_{ij}|x_j(t)| + \sum_{j=1}^{m} b_{ij} \max_{1\le k\le r}\{|x_j(t-k)|\}, \\ \Delta x_i(t) \ge a_{ii}x_i(t) - \sum_{j=1,j\neq i}^{m} a_{ij}|x_j(t)| - \sum_{j=1}^{m} b_{ij} \max_{1\le k\le r}\{|x_j(t-k)|\}. \end{cases}$$

According to Theorem 7, Theorem 8 holds.

4 Concluding Remarks

In this paper, a slight generalization of the celebrated continuous Halanay inequality is developed, a discrete analogue of the continuous Halanay inequality is proved. Using continuous and discrete Halanay delayed inequalities, we derived some sufficient conditions for the global exponential (asymptotic) stability of the equilibrium of neural networks. And the estimates of the state of such neural networks are also obtained. Conditions of these results can be directly derived from the parameters of the inequalities and equations, are very easy to verified. Hence, it is very convenience in application.

References

1. Halanay, A.: Differential Equations: Stability, Oscillations, Time Lags. Academic Press, New York (1966)
2. Gopalsamy, K.: Stability and Oscillations in Delay Differential Equations of Population Dynamics. Springer, Cambridge (1992)
3. Liz, E., Trofimchuk, S.: Existence and Stability of Almost Periodic Solutions for Quasilinear Delay Systems and Halanay Inequality. J. Math. Anal. Appl. 248, 625–644 (2000)
4. Mohamad, S., Gopalsamy, K.: Continuous and Discrete Halanay-type Inequalities. Bull. Aus. Math. Sot. 61, 371–385 (2000)
5. Pinto, M., Trofimchu, S.: Stability and Existence of Multiple Periodic Solutions for A Quasilinear Differential Equation with Maxima. Proceedings of the Royal Society of Edinburgh: Section A Mathematics 130, 1103–1118 (2000)
6. Cooke, K.L., Ivanov, A.F.: On the Discretization of A Delay Differential Equation. J. Differ. Equations Appl. 6, 105–119 (2000)
7. Andreev, A.: On the Stability of Nonautonomous Functional Differential Equations. Nonlinear Anal. 30, 448–457 (1997)
8. Cermák, J.: The Asymptotic Bounds of Solutions of Linear Delay Systems. Journal of Mathematical Analysis and Applications 2, 373–388 (1998)
9. Hatvani, L.: On Lyapunov's Direct Method for Nonautonomous FDE's. Functional Differential equations 5, 315–323 (1998)
10. Hatvani, L.: On the Asymptotic Stability for Functional Differential Equations by Lyapunov Functionals. Nonlinear Anal. 40, 251–263 (2000)
11. Lipovan, O.: A retarded Gronwall-like Inequality and Its Applications. Joural of Mathematical Analysis and Applications 252, 389–401 (2000)
12. Zeng, Z.G., Wang, J., Liao, X.X.: Global Exponential Stability of A General Class of Recurrent Neural Networks with Time-varying Delays. IEEE Trans. Circuits and Systems 50, 1353–1358 (2003)

13. Liz, E., Ferreiro, J.B.: A Note on the Global Stability of Generalized Difference Equations. Applied Mathematics Letters 15, 655–659 (2002)
14. Zeng, Z.G., Wang, J., Liao, X.X.: Stability Analysis of Delayed Cellular Neural Networks Described Using Cloning Templates. IEEE Trans. Circuits and Syst. 51, 2313–2324 (2004)
15. Zeng, Z.G., Wang, J.: Global Exponential Stability of Recurrent Neural Networks with Time-varying Delays in the Presence of Strong External Stimuli. Neural Networks 19, 1528–1537 (2006)
16. Zeng, Z.G., Wang, J.: Multiperiodicity and Exponential Attractivity Evoked by Periodic External Inputs in Delayed Cellular Neural Networks. Neural Computation 18, 848–870 (2006)

A Discrete-Time Recurrent Neural Network with One Neuron for k-Winners-Take-All Operation

Qingshan Liu[1], Jinde Cao[2], and Jinling Liang[2]

[1] School of Automation, Southeast University, Nanjing 210096, China
qsliu@seu.edu.cn
[2] Department of Mathematics, Southeast University, Nanjing 210096, China
{jdcao,jinlliang}@seu.edu.cn

Abstract. In this paper, a discrete-time recurrent neural network with one neuron and global convergence is proposed for k-winners-take-all (kWTA) operation. Comparing with the existing kWTA networks, the proposed network has simpler structure with only one neuron. The global convergence of the network can be guaranteed for kWTA operation. Simulation results are provided to show that the outputs vector of the network is globally convergent to the solution of the kWTA operation.

Keywords: Discrete-time recurrent neural network, Global convergence, k-winners-take-all operation.

1 Introduction

The winner-take-all (WTA) networks have been widely used in various applications, such as signal processing [1], associative memories [2], and cooperative models of binocular stereo [3]. The WTA operation is to select the maximum from a collection of input signals. The k-winners-take-all (kWTA) operation selects the k largest inputs out of n inputs ($1 \leq k \leq n$), which can be considered as a generalized version of WTA operation. In literature, many WTA and kWTA networks have been proposed [4-12]. However, neural networks with simple structure for kWTA operation are desired in real applications. In this paper, we devote to present a discrete-time recurrent neural network with one neuron for kWTA operation.

Generally, the kWTA operation can be defined as the following function

$$u_i = f(v_i) = \begin{cases} 1, \text{ if } v_i \in \{k \text{ largest elements of } v\}, \\ 0, \text{ otherwise}, \end{cases} \tag{1}$$

where $v = (v_1, v_2, \ldots, v_n)^T$ is the input vector and $u = (u_1, u_2, \ldots, u_n)^T$ is the output vector. According to [10], the solution of (1) can be determined from the

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 272–278, 2009.

following zero-one integer quadratic programming problem

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2\eta}u^T u - v^T u, \\ &\text{subject to} \sum_{i=1}^{n} u_i = k, \\ &\qquad\qquad u_i \in \{0,1\}, \quad i = 1,2,\ldots,n, \end{aligned} \tag{2}$$

where η is a positive constant. Let $\bar{v}_k$ be the kth largest element and $\bar{v}_{k+1}$ be the $(k+1)$th largest element. If $1/(\bar{v}_k - \bar{v}_{k+1}) \leq \eta$, then the optimal solution of problem (2) is the same as that of the following continuous quadratic programming problem [10]

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2\eta}u^T u - v^T u, \\ &\text{subject to} \sum_{i=1}^{n} u_i = k, \\ &\qquad\qquad 0 \leq u_i \leq 1, \quad i = 1,2,\ldots,n. \end{aligned} \tag{3}$$

Based on the quadratic programming problem (3), a new kWTA network will be proposed in this paper.

2 Model Description

In this section, a discrete-time recurrent neural network is constructed for the kWTA operation based on the quadratic programming problem (3).

According to the Karush-Kuhn-Tucker (KKT) conditions [13], x^* is a optimal solution of (3) if and only if there exist $x^* \in \mathbb{R}$ and $y^* \in \mathbb{R}^n$ such that $(u^*, x^*, y^*)^T$ satisfies the following optimality conditions:

$$\frac{1}{\eta}u - v - \frac{1}{\eta}ex - \frac{1}{\eta}y = 0, \tag{4}$$

$$e^T u = k, \tag{5}$$

$$u = P_\Omega(u - y), \tag{6}$$

where $e = (1,1,\ldots,1)^T \in \mathbb{R}^n$ and the projection operator $P_\Omega(u) = (P_\Omega(u_1), \ldots, P_\Omega(u_n))^T$ with

$$P_\Omega(u_i) = \begin{cases} 1, & u_i > 1, \\ u_i, & 0 \leq u_i \leq 1, \\ 0, & u_i < 0. \end{cases} \tag{7}$$

From (4), we have

$$u - y = ex + \eta v. \tag{8}$$

Substituting (8) into (6), we have

$$u = P_\Omega(ex + \eta v). \tag{9}$$

Substituting (9) into (5), we have

$$e^T P_\Omega(ex + \eta v) = k. \tag{10}$$

Based on equations (9) and (10), the proposed discrete-time kWTA network model is described as follows:

- state equation

$$x(m+1) = x(m) - \alpha[e^T P_\Omega(ex(m) + \eta v) - k], \tag{11}$$

- output equation

$$u(m) = P_\Omega(ex(m) + \eta v), \tag{12}$$

where α is a positive constant and m is the iteration step.

The architecture of the kWTA network described in (11) and (12) is depicted in Fig. 1. From which we can see that the network proposed herein has only one neuron. The circuit realizing the network consists of $2n+6$ simple summers and n weighted connections. Comparing with the existing kWTA networks [6,11,14], the proposed discrete-time kWTA network has simpler structure with only one neuron which can be implemented by circuit easily.

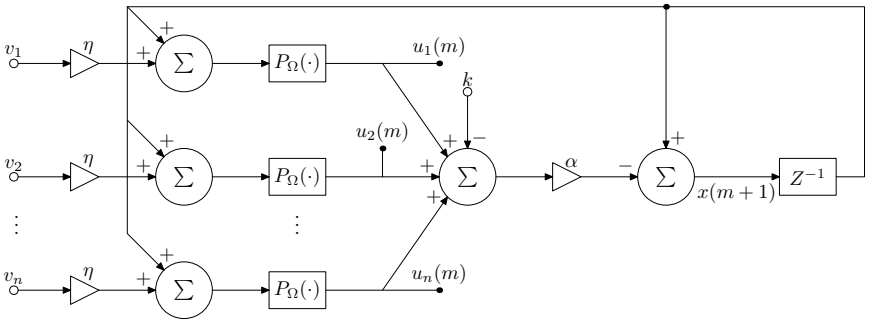


Fig. 1. Architecture of the proposed kWTA network

Definition 1. *x^* is said to be an equilibrium point of network (11) if x^* satisfies*

$$e^T P_\Omega(ex + \eta v) - k = 0. \tag{13}$$

From above analysis, the following lemma holds obviously.

Lemma 1. *u^* is an optimal solution of problem (3) if and only if there exists $x^* \in \mathbb{R}$ such that x^* is an equilibrium point of network (11) and $u^* = P_\Omega(ex^* + \eta v)$ holds.*

The following lemma gives an inequality with respect to the projection operate defined in (7) which is useful in next section.

Lemma 2. [15] *For any $x, y \in \mathbb{R}^n$, the projection operate P_Ω satisfies the following inequality*

$$(x-y)^T(P_\Omega(x) - P_\Omega(y)) \geq \|P_\Omega(x) - P_\Omega(y)\|^2. \tag{14}$$

3 Global Convergence

In this section, the global convergence of neural network (11) is analysis and proved.

Theorem 1. *For each α in the range of $0 < \alpha < 2/n$, the neural network (11) is globally convergent to an equilibrium point for any initial point $x(0) \in \mathbb{R}$.*

Proof. Assume x^* being an equilibrium point of network (11). Denote $\tilde{x}(m) = P_\Omega(ex(m) + \eta v)$ and $\tilde{x}^* = P_\Omega(ex^* + \eta v)$. According to (11) and Lemma 2, for any $x(0) \in \mathbb{R}$, we have

$$\begin{aligned}
&[x(m+1) - x^*]^2 \\
&= [x(m) - \alpha(e^T\tilde{x}(m) - k) - x^* + \alpha(e^T\tilde{x}^* - k)]^2 \\
&= [x(m) - x^* - \alpha e^T(\tilde{x}(m) - \tilde{x}^*)]^2 \\
&= [x(m) - x^*]^2 - 2\alpha(x(m) - x^*)e^T(\tilde{x}(m) - \tilde{x}^*) \\
&\quad +\alpha^2(\tilde{x}(m) - \tilde{x}^*)^T ee^T(\tilde{x}(m) - \tilde{x}^*) \\
&\le [x(m) - x^*]^2 - 2\alpha(\tilde{x}(m) - \tilde{x}^*)^T(\tilde{x}(m) - \tilde{x}^*) \\
&\quad +\alpha^2(\tilde{x}(m) - \tilde{x}^*)^T ee^T(\tilde{x}(m) - \tilde{x}^*) \\
&= [x(m) - x^*]^2 - (\tilde{x}(m) - \tilde{x}^*)^T(2\alpha I - \alpha^2 ee^T)(\tilde{x}(m) - \tilde{x}^*), \qquad (15)
\end{aligned}$$

where I is identity matrix.

Consider the following Lyapunov function as such

$$V(x) = (x - x^*)^2. \qquad (16)$$

Since the maximum eigenvalue of ee^T is n, from (15), we have

$$\begin{aligned}
V(x(m+1)) - V(x(m)) &\le -(\tilde{x}(m) - \tilde{x}^*)^T(2\alpha I - \alpha^2 ee^T)(\tilde{x}(m) - \tilde{x}^*) \\
&\le -(2\alpha - n\alpha^2)(\tilde{x}(m) - \tilde{x}^*)^T(\tilde{x}(m) - \tilde{x}^*).
\end{aligned}$$

For any initial point $x(0) \in \mathbb{R}$, if $0 < \alpha < 2/n$, $V(x(m))$ is non-increasing as $m \to \infty$ and $\{x(m)\}$ is bounded. Then there exists an increasing sequence $\{m_N\}$ with $\lim_{N\to\infty} m_N = \infty$ and a limit point $\bar{x}$ such that $\lim_{N\to\infty} x(m_N) = \bar{x}$. Thus $\bar{x}$ is a ω-limit point of $x(m)$.

According to the LaSalle invariance principle for discrete-time system [16], $x(m)$ will converge to M, the largest invariant subset of the following set:

$$E = \{x \in \mathbb{R} : V(x(m+1)) - V(x(m)) = 0\}.$$

Note that, if $V(x(m+1)) - V(x(m)) = 0$ and $0 < \alpha < 2/n$, we have $\tilde{x}(m) = \tilde{x}^*$. Thus $e^T P_\Omega(ex(m) + \eta v) - k = 0$. That is to say x is an equilibrium point of network (11).

Conversely, if x is an equilibrium point of network (11) (i.e., x is a constant), it follows that $V(x(m+1)) = V(x(m))$. So, $V(x(m+1)) - V(x(m)) = 0$ if and only if x is an equilibrium point of network (11). Then we have

$$E = \{x \in \mathbb{R} : e^T P_\Omega(ex + \eta v) - k = 0\}.$$

Thus, $\bar{x} \in M \subseteq E$.

Finally, let's define another Lyapunov function

$$V(x) = (x - \bar{x})^2. \tag{17}$$

Similarly to the above proof, if $0 < \alpha < 2/n$, there exists an increasing subsequence m_{N_l} of $\{m_N\}$ such that

$$|x(m_{N_{l+1}}) - \bar{x}| \leq |x(m+1) - \bar{x}| \leq |x(m) - \bar{x}| \leq |x(m_{N_l}) - \bar{x}|. \tag{18}$$

Since $\lim_{l\to\infty} x(m_{N_l}) = \bar{x}$, it follows that $\lim_{m\to\infty} x(m) = \bar{x}$.

Consequently, for any initial point $x(0) \in \mathbb{R}$, $x(m)$ is globally convergent to an equilibrium point of network (11).

Theorem 2. *For each α in the range of $0 < \alpha < 2/n$, the output of the neural network is convergent to the optimal solution of problem (3).*

Proof. For any initial point $x(0)$, there exist an equilibrium point x^* of network (11) such that $\lim_{m\to\infty} x(m) = x^*$. According to Theorem 1 and (12), we have $\lim_{m\to\infty} u(m) = \lim_{m\to\infty} P_\Omega(ex(m) + \eta v) = P_\Omega(ex^* + \eta v)$ which is the optimal solution of problem (3).

4 Simulation Results

In this section, two examples are given to demonstrate the effectiveness of the proposed kWTA network.

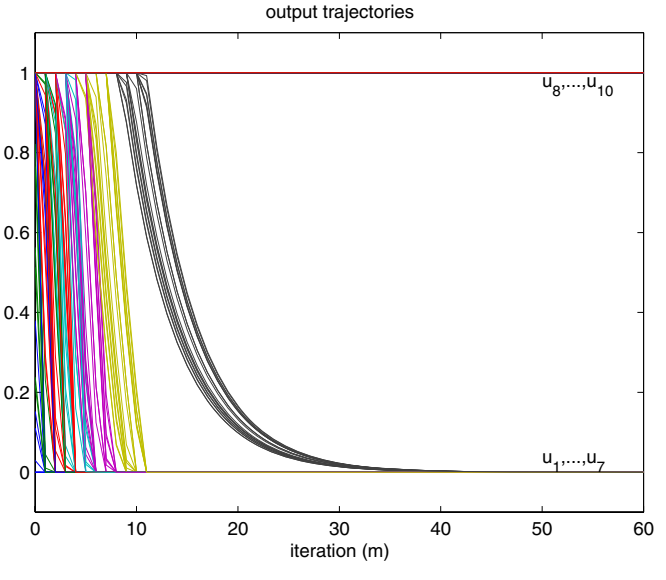


Fig. 2. Output behavior of the kWTA network in Example 1

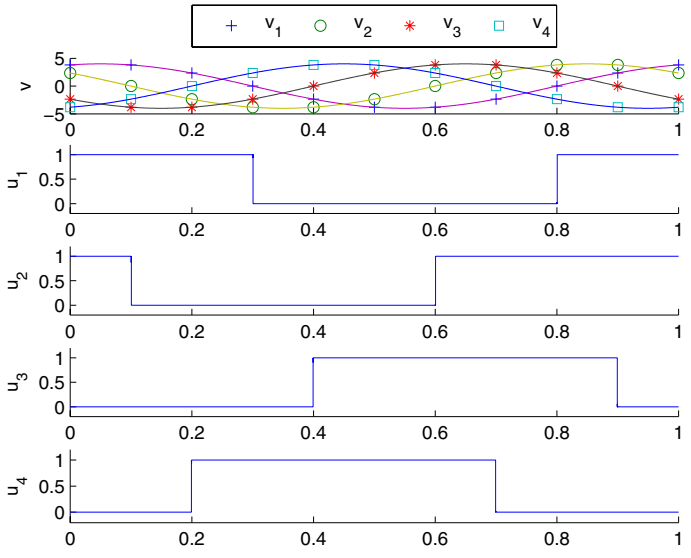


Fig. 3. Inputs and outputs of the kWTA network in Example 2

Example 1. Consider a kWTA problem with input vector $v_i = i$ $(i = 1, 2, \ldots, n)$, $k = 3$. The proposed discrete-time kWTA network is utilized to determine the three largest inputs.

According to Theorem 1, let $\eta = 1$ and $\alpha = 0.18$, the transient behaviors of output vector u of the kWTA network with 20 random initial values are depicted in Fig. 2 with $n = 10$. It shows that the output vector of the network is convergent to the unique optimal solution $u^* = (0, \ldots, 0, 1, 1, 1)^T$ which corresponds to the three largest inputs.

Example 2. Let's consider a set of four sinusoidal input signals with the following instantaneous values $v_p(t) = 4\sin[2\pi(t + 0.2p)]$ $(p = 1, 2, 3, 4)$ and $k = 2$. The four input signals and the transient outputs of the kWTA network with $\eta = 10^3$ and $\alpha = 0.4$ are depicted in Fig. 3. The simulation results show that the kWTA network can generate the two largest signals in real time.

5 Conclusions

In this paper, a discrete-time recurrent neural network with one neuron has been proposed for kWTA problems. The global convergence of the network is proved by LaSalle invariance principle. The theoretical results show that the network is efficient for kWTA problems if only the kth and $(k + 1)$th inputs are different. Simulation results illustrated the performance of the proposed kWTA network.

Acknowledgements. This work was supported by the Teaching and Research Fund for Excellent Young Teachers at Southeast University of China.

References

1. Andreou, A., Boahen, K., Pouliquen, P., Pavasovic, A., Jenkins, R., Strohbehn, K.: Current-mode subthreshold mos circuits for analog vlsi neural systems. IEEE Trans. Neural Networks 2, 205–213 (1991)
2. Hertz, J., Krogh, A., Palmer, R.: Introduction to the Theory of Neural Computing. Addison-Wesley, Massachusetts (1991)
3. Marr, D., Poggio, T.: Cooperative computation of stereo disparity. Science 194, 283–287 (1976)
4. Wolfe, W., Mathis, D., Anderson, C., Rothman, J., Gottler, M., Brady, G., Walker, R., Duane, G., Alaghband, G.: K-winner networks. IEEE Trans. Neural Networks 2, 310–315 (1991)
5. Wang, J.: Analogue winner-take-all neural networks for determining maximum and minimum signals. Int. J. Electron 77, 355–367 (1994)
6. Urahama, K., Nagao, T.: K-winners-take-all circuit with o(n) complexity. IEEE Trans. Neural Networks 6, 776–778 (1995)
7. Sekerkiran, B., Cilingiroglu, U.: A cmos k-winners-take-all circuit with o(n) complexity. IEEE Trans. Circuits and Systems-II 46, 1–5 (1999)
8. Maass, W.: Neural computation with winner-take-all as the only nonlinear operation. Advances in Neural Information Processing Systems 12, 293–299 (1999)
9. Marinov, C., Calvert, B.: Performance analysis for a k-winners-take-all analog neural network: basic theory. IEEE Trans. Neural Networks 14, 766–780 (2003)
10. Liu, S., Wang, J.: A simplified dual neural network for quadratic programming with its kwta application. IEEE Trans. Neural Networks 17, 1500–1510 (2006)
11. Liu, Q., Wang, J.: Two k-winners-take-all networks with discontinuous activation functions. Neural Networks 21, 406–413 (2008)
12. Hu, X., Wang, J.: An improved dual neural network for solving a class of quadratic programming problems and its k-winners-take-all application. IEEE Trans. Neural Networks 19, 2022–2031 (2008)
13. Bazaraa, M., Sherali, H., Shetty, C.: Nonlinear Programming: Theory and Algorithms, 2nd edn. John Wiley, New York (1993)
14. Marinov, C., Hopfield, J.: Stable computational dynamics for a class of circuits with o(n) interconnections capable of kwta and rank extractions. IEEE Trans. Circuits and Systems-I 52, 949–959 (2005)
15. Kinderlehrer, D., Stampacchia, G.: An Introduction to Variational Inequalities and Their Applications. Academic, New York (1982)
16. LaSalle, J.: The Stability of Dynamical Systems. Society for Industrial Mathematics (1976)

Stability of Stochastic Recurrent Neural Networks with Positive Linear Activation Functions

Wudai Liao[1], Xuezhao Yang[1], and Zhongsheng Wang[2]

[1] School of Electrical and Information Engineering,
Zhongyuan University of Technology, Zhengzhou 450007, China
[2] School of Information Engineering, Guangdong Polytechnic Normal University,
Guangzhou 510665, China
wdliao@zzti.edu.cn, yangxz@zzti.edu.cn, gszswang@126.com

Abstract. In view of the character of positive linearity of activation functions of neurons of the recurrent neural networks, the method decomposing the state space to sub-regions is adopted to study almost sure exponential stability on delayed cellular neural networks which are in the noised environment. When perturbed terms in the model of the neural network satisfy Lipschitz condition, some algebraic criteria are obtained. The results obtained in this paper show that if an equilibrium of the neural network is the interior point of a sub-region, and an appropriate matrix related to this equilibrium has some stable degree to stabilize the perturbation, then the equilibrium of the delayed cellular neural network can still remain the property of exponential stability. All results in the paper is only to compute eigenvalues of matrices. All results obtained in this paper include the deterministic neural network as special case.

Keywords: Stochastic recurrent neural networks, Positive linear activation, Almost sure stability.

1 Introduction

The stability problem on recurrent neural networks is widely studied from origin to now, including the original work [1], delayed cases [2] and other studies.

For reality, we should consider the case that the recurrent neural networks are in noised environment, this is because the network realization is through VLSI approach and the information transmission among real brain neuron cells ia a noised process. We call these neural networks stochastic neural networks. The stability analysis of these neural networks originates in [3], and after that, lots of results [4,5,6] are obtained by some scholars.

Because of the characteristic of positive linearity about the activation functions of neurons in recurrent neural networks, we are going to use this to study the stability problem on stochastic recurrent neural networks, it has the form

$$\mathrm{d}x(t) = [-Bx(t) + Af(x(t-\tau)) + I]\mathrm{d}t + \widetilde{\sigma}(x(t), x(t-\tau))\mathrm{d}w(t) \ . \qquad (1)$$

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 279–285, 2009.

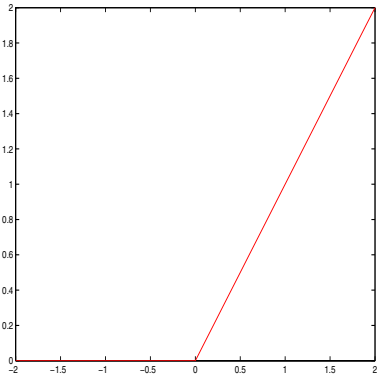


Fig. 1. the positive linear activation function

Where $x=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}\in\mathbb{R}^n$ is the state vector of the neural networks, $x(t-\tau)=(x_1(t-\tau_1),x_2(t-\tau_2),\cdots,x_n(t-\tau_n))^{\mathrm{T}}$, $\tau_i\geq 0$ is the time delay of the neuron i and $0\leq\tau_i\leq\tau$, $i=1,2,\cdots,n.f(\cdot)$ is the vector of the activation functions of the neurons, $f(x)=(f_1(x_1),f_2(x_2),\cdots,f_n(x_n))^{\mathrm{T}}$, $f_i(\cdot)$ is the positive linearity function as the following form:

$$f_i(x_i)=\mathrm{poslin}(x_i)=\begin{cases}0, & x_i<0\\ x_i, & x_i\geq 0\end{cases},\ i=1,2,\cdots,n\ , \tag{2}$$

and the figure of the activation is in Fig.1. $\widetilde{\sigma}(\cdot,\cdot)\in\mathbb{R}^{n\times m}$ is the perturbed matrix satisfied Lipschtz condition, that is there exists a positive constant L such that

$$\|\widetilde{\sigma}(x,y)-\widetilde{\sigma}(\bar{x},\bar{y})\|^2\leq L\cdot(\|x-\bar{x}\|^2+\|y-\bar{y}\|^2)\ . \tag{3}$$

$\|\cdot\|$ in this paper denotes the Frobenius norm of a matrix. $w(\cdot)$ is an m-dimension Brownian motion, $B=\mathrm{diag}.(b_1,b_2,\cdots,b_n)$ is a positive diagonal matrix, $A=(a_{ij})_{n\times n}$ is the weight matrix between neurons, I is the bias vector of the neurons.

2 Main Results

We will set up some sufficient criteria ensuring the neural network (1) almost sure exponential stability in this section.

According to $\mathbb{R}=(-\infty,\infty)$ being decomposed into two intervals $(-\infty,0)$ and $[0,\infty)$, the n-space $\mathbb{R}^n$ can be divided into 2^n sub-regions.

Suppose that $x^*=(x_1^*,x_2^*,\cdots,x_n^*)^{\mathrm{T}}$ is an equilibrium of the system (1), which is the interior point in one of the sub-regions, $N(x^*)$ is the greatest neighborhood of the point x^* which is in the same sub-region that the point x^* is in. Take the transformation $z=x-x^*$ and the unit step function

$$u(x_i^*)=\begin{cases}1, & x_i^*\geq 0\\ 0, & x_i^*<0\end{cases}\ .$$

From the assumptions above and Property (2) of the activation functions, the formula

$$f(z + x^*) - f(x^*) = U(x^*)z$$

holds for all points $z \in N(x^*)$, where $U(x^*) = \text{diag.}(u(x_1^*), u(x_2^*), \cdots, u(x_n^*))$ is a diagonal matrix which the elements are either 0 or 1. For example, assume that $x^* = (1.2, -0.6, 1.4)^{\mathrm{T}}$ is the equilibrium of a 3-order neural network, then,

$$U(x^*) = \text{diag.}(1, 0, 1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad .$$

Thus, in order to discuss the stability of the equilibrium x^* of System (1), we need only study the same property of the trivial equilibrium $z = 0$ of the following system

$$\mathrm{d}z(t) = [-Bz(t) + AU(x^*)z(t-\tau)]\mathrm{d}t + \sigma(z(t), z(t-\tau))\mathrm{d}w(t) \quad . \tag{4}$$

Where $\sigma(z(t), z(t-\tau)) = \widetilde{\sigma}(z(t) + x^*, z(t-\tau) + x^*) - \widetilde{\sigma}(x^*, x^*)$.

For $\forall z, y \in \mathbb{R}^n$ and a symmetric positive definite matrix Q, we have the following estimation on the matrix $\sigma(z, y)$ by using Condition (3):

$$\text{trace}(\sigma^{\mathrm{T}}(z, y)Q\sigma(z, y)) \leq \rho(Q) \cdot L(\|z\|^2 + \|y\|^2) \quad . \tag{5}$$

Where $\rho(Q)$ is the spectral radius of the matrix Q, that is, $\rho(Q) = \lambda_{\max}(Q)$, the maximal eigenvalue of the matrix Q.

We first give a definition and a lemma [4] which plays an important role in this paper.

Definition 1. *For a Lyapunov function $V \in C^2(\mathbb{R}^n; \mathbb{R}_+)$, that is, the function is continuously twice differentiable with respect to its variables, $\mathbb{R}_+ = [0, \infty)$, for any $z, y \in \mathbb{R}^n$, defines the operator L generated by the system (4) as following*

$$LV(z, y) = \dot{V}^T(z) \cdot [-Bz + AU(x^*)y] + \frac{1}{2}trace(\sigma^T(z, y)\ddot{V}(z)\sigma(z, y)) \quad .$$

Lemma 1. *For System (4), if there exist functions $V \in C^2(\mathbb{R}^n; \mathbb{R}_+)$, $\mu \in C(\mathbb{R}^n; \mathbb{R}_+)$, $\mu_i \in C(\mathbb{R}_+; \mathbb{R}_+)$, $i = 1, 2, \cdots, n$, and constants $\lambda_1 > \lambda_2 \geq 0$, such that*

$$(1)LV(z, y) \leq -\lambda_1\mu(z) + \lambda_2 \sum_{i=1}^{n} \mu_i(y_i),\ (2)V(z) \leq \mu(z),\ (3) \sum_{i=1}^{n} \mu_i(z_i) \leq \mu(z)$$

hold for any $z, y \in \mathbb{R}^n$, then, the trivial equilibrium $z = 0$ is almost surely exponentially stable.

In the following, we will get the main results.

Theorem 1. *Assume that x^* be the equilibrium of (1). For diagonal positive definite matrices $Q = diag.(q_1, q_2, \cdots, q_n)$ and $R = diag.(r_1, r_2, \cdots, r_n)$, if the matrix*

$$H = \begin{pmatrix} -2QB + \alpha E + R & QAU(x^*) \\ (QAU(x^*))^T & -R + \alpha E \end{pmatrix}_{2n \times 2n}$$

is negative definite, then the equilibrium x^* is almost surely exponentially stable, where $\alpha = \rho(Q) \cdot L, E \in \mathbb{R}^{n \times n}$ is the unit matrix.

Proof (of Theorem 1). For $\forall z, y \in \mathbb{R}^n$, choose the Lyapunov function $V(z) = z^{\mathrm{T}} Q z$, and then, the operator L generated by System (4) has the form

$$LV(z,y) = 2z^{\mathrm{T}} Q(-Bz + AU(x^*)y) + \mathrm{trace}(\sigma^{\mathrm{T}}(z,y) Q \sigma(z,y)) \ .$$

By using Condition (5) and denoting $-\lambda = \lambda_{\max}(H), \lambda > 0$, we have the following estimation:

$$\begin{aligned} LV(z,y) &\le -2z^{\mathrm{T}}(QB)z + 2z^{\mathrm{T}}[QAU(x^*)]y + \alpha(\|z\|^2 + \|y\|^2) \\ &= (z^{\mathrm{T}}, y^{\mathrm{T}}) H \begin{pmatrix} z \\ y \end{pmatrix} - \sum_{i=1}^{n} r_i z_i^2 + \sum_{i=1}^{n} r_i y_i^2 \\ &\le -\sum_{i=1}^{n} (\lambda + r_i) z_i^2 + \sum_{i=1}^{n} (r_i - \lambda) y_i^2 \ . \end{aligned}$$

From the construction of the matrix H, we can easily deduce that $r_i - \lambda \ge \alpha > 0$, $i = 1, 2, \cdots, n$. Denote that

$$\lambda_1 = \min_{1 \le i \le n} \{\frac{1}{q_i}(\lambda + r_i)\}, \ \lambda_2 = \max_{1 \le i \le n} \{\frac{r_i - \lambda}{\lambda + r_i}\} \ .$$

Obviously, $\lambda_1 > 0, \ 0 < \lambda_2 < 1$ (use $\lambda > 0$) and $(\lambda + r_i)/\lambda_1 \ge q_i, i = 1, 2, \cdots, n$. Let

$$\mu(z) = \frac{1}{\lambda_1} \sum_{i=1}^{n} (\lambda + r_i) z_i^2, \ \mu_i(y_i) = \frac{1}{\lambda_1} (\lambda + r_i) y_i^2 \ ,$$

then, we have

$$(1) LV(z,y) \le -\lambda_1 \mu(z) + \lambda_1 \lambda_2 \sum_{i=1}^{n} \mu_i(y_i), (2) V(z) \le \mu(z), (3) \sum_{i=1}^{n} \mu_i(y_i) = \mu(y) \ .$$

According to Lemma 1, the trivial equilibrium $z = 0$ of System (4), equivalently, the equilibrium x^* of (1) is almost sure exponentially stable. The proof is complete.

Remark 1. As a special case of Theorem 1, it includes the corresponding stability condition on the deterministic neural network. That is, for positive diagonal matrices Q, R, if the matrix

$$G = \begin{pmatrix} -2QB + R & QAU(x^*) \\ (QAU(x^*))^{\mathrm{T}} & -R \end{pmatrix}_{2n \times 2n}$$

is negative definite, then, the equilibrium x^* of the following deterministic recurrent neural network

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = -Bx(t) + Af(x(t-\tau)) + I \tag{6}$$

is exponential stability.

Corollary 1. *Let x^* be the equilibrium of (1). If there exist positive diagonal matrices $Q = diag.(q_1, q_2, \cdots, q_n)$ and $R = diag.(r_1, r_2, \cdots, r_n)$ such that the matrix*

$$H_1 = \begin{pmatrix} -2QB + R & QAU(x^*) \\ (QAU(x^*))^{T} & -R \end{pmatrix}_{2n \times 2n}$$

has the stable degree α, that is $\lambda_{\max}(H_1) < -\alpha$, then the equilibrium x^* is almost surely exponentially stable.

Proof (of Corollary 1). From Theorem 1, in order to prove Corollary 1, we need only to verify the matrix H in Theorem 1 to be negative definite. For $\forall z, y \in \mathbb{R}^n$,

$$\begin{aligned}(z^{\mathrm{T}}, y^{\mathrm{T}})H \begin{pmatrix} z \\ y \end{pmatrix} &= (z^{\mathrm{T}}, y^{\mathrm{T}})H_1 \begin{pmatrix} z \\ y \end{pmatrix} + \alpha\|z\|^2 + \alpha\|y\|^2 \\ &\leq (\lambda_{\max}(H_1) + \alpha)\|z\|^2 + (\lambda_{\max}(H_1) + \alpha)\|y\|^2 \ .\end{aligned}$$

Because $\lambda_{\max}(H_1) < -\alpha$, the matrix H in Theorem 1 is negative definite. The proof is complete.

In following, we give an approach to choose the optional matrix R and the matrix Q, so, it is easy to use in system synthesis.

Corollary 2. *Assume that x^* be the equilibrium of (1). For an appropriate positive number $m > 0$, if the matrix*

$$H_2 = \begin{pmatrix} -\frac{2m}{m+1}E & B^{-1}AU(x^*) \\ (B^{-1}AU(x^*))^{T} & -\frac{2}{m+1}E \end{pmatrix}_{2n \times 2n}$$

has the stable degree $\alpha = L \cdot \min\{b_i\}$, then the equilibrium x^* is almost surely exponentially stable.

Proof. Let $-2QB + R = -mR$ and $Q = B^{-1}$ in matrix H_1, this implies that $R = [2/(m+1)]E$ and $\rho(Q) = \max\{b_i^{-1}\} = \min\{b_i\}$, hence, the matrix H_1 becomes the matrix H_2, and the stable degree α becomes $L \cdot \min\{b_i\}$. The proof is complete.

Corollary 3. *For equilibrium* $x^* = (x_1^*, x_2^*, \cdots, x_n^*)^T$ *of System (1),* $x_i^* > 0,\ i = 1, 2, \cdots, n$. *If there exists a number* $m > 0$, *such that the matrix*

$$H_3 = \begin{pmatrix} -\frac{2m}{m+1}E & B^{-1}A \\ (B^{-1}A)^T & -\frac{2}{m+1}E \end{pmatrix}_{2n \times 2n}$$

has the stable degree $\alpha = L \cdot \min\{b_i\}$, *then the equilibrium* x^* *is almost sure exponentially stable.*

Proof. In this case, $U(x^*) = E$, by using Corollary 2, Corollary 3 holds. The proof is complete.

Corollary 4. *For equilibrium* $x^* = (x_1^*, x_2^*, \cdots, x_n^*)^T$ *of System (1),* $x_i^* < 0,\ i = 1, 2, \cdots, n$. *If there exists a number* $m > 0$, *such that the density of the perturbation satisfies*

$$0 \le L < \min\{\frac{2}{m+1}, \frac{2m}{m+1}\} \cdot \min\{b_i\}$$

then, the equilibrium x^* *can still remain exponential stability while* x^* *of the deterministic neural network (6) is exponentially stable.*

Proof. In this case, $U(x^*) = 0$, H_2 has the following form

$$\begin{pmatrix} -\frac{2m}{m+1}E & 0 \\ 0 & -\frac{2}{m+1}E \end{pmatrix}$$

the biggest eigenvalue is

$$\lambda_{\max}(H_2) = \max\left\{-\frac{2m}{m+1}, -\frac{2}{m+1}\right\} = -\min\left\{\frac{2m}{m+1}, \frac{2}{m+1}\right\}\ .$$

So, the condition of this corollary implies $\lambda_{\max}(H_2) < -\alpha = -L \cdot \max\{b_i^{-1}\}$, by using Corollary 2, this corollary holds. The proof is complete.

Corollary 5. *For equilibrium* $x^* = (x_1^*, x_2^*, \cdots, x_n^*)^T$ *of System (1),* $x_{j_0}^* > 0, x_i^* < 0,\ i = 1, \cdots, j_0 - 1, j_0 + 1, \cdots, n$. *If there exists positive number* m, *such that*

$$c_{j_0} := \sum_{i=1}^{n} \left(\frac{a_{i,j_0}}{b_i}\right)^2 < \frac{4m}{(m+1)^2}$$

and $0 \le L < \lambda(m)$ *hold, then, the equilibrium* x^* *is almost surely exponentially stable, where*

$$\lambda(m) = \min\left\{\frac{2}{m+1}, \frac{2m}{m+1}, 1 - \sqrt{c_{j_0} + (\frac{m-1}{m+1})^2)}\right\} \cdot \min\{b_i\}\ .$$

Proof. Without loss generality, let $x_1^* > 0, x_i^* < 0, i = 2, 3, \cdots, n$. Hence, $U(x^*) = \text{diag.}(1, 0, \cdots, 0)$, and then the character polynomial of matrix H_2 is as following

$$f(\lambda) = (\lambda + \frac{2}{m+1})^{n-1}(\lambda + \frac{2m}{m+1})^{n-1}[(\lambda + \frac{2}{m+1})(\lambda + \frac{2m}{m+1}) - c_1] \ .$$

Its all eigenvalues satisfy the following equations

$$\lambda + \frac{2}{m+1} = 0, \ \lambda + \frac{2m}{m+1} = 0, \ \lambda^2 + 2\lambda + \frac{4m}{(m+1)^2} - c_1 = 0 \ .$$

The condition $c_1 < 4m/(m+1)^2$ implies that the bigger root of the last equation above is negative, it has the form

$$\lambda = -1 + \sqrt{c_1 + (\frac{m-1}{m+1})^2} \ .$$

So, $\lambda_{\max}(H_2) = -\lambda(m) \cdot \max\{b_i^{-1}\} < -L \cdot \max\{b_i^{-1}\} = -\alpha$. Then, from the condition of this corollary, the condition of Corollary 2 holds. The proof is complete.

3 Conclusions

From discussion above, we conclude the results as following:

1. All results obtained in our paper hold for the deterministic case corresponding to neural network (1);
2. If the perturbed intensity is pre-estimated, then we can choose the parameter matrices B and A to design a deterministic neural network which has enough robustness to stabilize the perturbed intensity.

Acknowledgements. This work is supported by National Natural Science Foundation of China (60774051).

References

1. Chua, L., Yang, L.: Cellular Neural Networks: Theory. IEEE Trans. Circuits and Systems 35, 1257–1272 (1988)
2. Cao, J., Zhou, D.: Stability Analysis of Delayed Celluar Neural Networks. Neural Networks 11, 1601–1605 (1998)
3. Liao, X., Mao, X.: Stability of Stochastic Neural Networks. Neual, Parallel and Scientific Computations 14, 205–224 (1996)
4. Blythe, S., Mao, X.: Stability of Stochastic delay Neural Networks. Journal of The Franklin Institute 338, 481–495 (2001)
5. Shen, Y., Liao, X.: Robust Stability of Nonlinear Stochastic Delayed Systems. Acta Automatica Sinic. 25, 537–542 (1999)
6. Mao, X.: Stochastic Differential Equations and Their Applications, 1st edn. Horwood Pub., Chichester (1997)

Stability Conditions of Delayed Recurrent Neural Networks with Positive Linear Activation Functions

Dongyun Wang and Yan Wang

School of Electrical and Information Engineering, Zhongyuan University of Technology, Zhengzhou Henan 450007, China
{wdy,wangyan2003}@zzti.edu.cn

Abstract. By using the positive linearity of the activation functions of neurons in recurrent neural networks, and by adopting the method of decomposing the state space to sub-regions, the mathematical equations of delayed recurrent neural networks are rewritten to be the form of linear differential difference equations in the neighbourhood of each equilibrium, which is an interior point of some sub-region. Based on this linear form and by using the stability theory of linear differential difference equations and the tool of M-matrix, delay-dependent and delay-independent stability algebraic criteria are obtained. All results obtained in this paper need only to compute the eigenvalues of some matrices or to examine the matrices to be M-matrix or to verify some inequalities to be holden.

Keywords: Recurrent neural networks, Positive linear activation, Exponential stability.

1 Introduction

The stability problem of recurrent neural networks has been widely studied [1,2,3]. Now, many researchers focus their attention on the delayed cases [4,5,6,7], and the results obtained in these papers mainly use the Lyapunov direct method and the Razumikhin-type theorems.

In order to sufficiently use the characteristic of positive linearity of the output functions, we have introduced the method of decomposing state space to sub-regions to study the stability of CNN [3,8]. In this paper, we are going to use this method instead of Lyapunov direct method to study the stability problem of the delayed recurrent neural networks with positive linear activation functions.

Consider the system of delayed recurrent neural networks with positive linear activation functions:

$$\dot{x}(t) = -Bx(t) + Af(x(t-\tau)) + I \ , \tag{1}$$

where $x = (x_1, x_2, \cdots, x_n)^{\mathrm{T}} \in \mathbb{R}^n$ is the state vector of the neural networks, $x(t-\tau) = (x_1(t-\tau_1), x_2(t-\tau_2), \cdots, x_n(t-\tau_n))^{\mathrm{T}}$, $\tau_i \geq 0$ is the time delay of the neuron

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 286–294, 2009.

i and $0 \leq \tau_i \leq \tau$, $i = 1, 2, \cdots, n$. $f(\cdot)$ is the vector of the output functions of the neurons, $f(x) = (f_1(x_1), f_2(x_2), \cdots, f_n(x_n))^{\mathrm{T}}$, $f_i(\cdot)$ has the form

$$f_i(x_i) = \mathrm{poslin}(x_i) = \begin{cases} 0, & x_i < 0 \\ x_i, & x_i \geq 0 \end{cases}, \ i = 1, 2, \cdots, n \ , \tag{2}$$

and the figure of the activation is in Fig.1. $B = \mathrm{diag}(b_1, b_2, \cdots, b_n)$ denotes a diagonal matrix, $b_i > 0, i = 1, 2, \cdots, n$, $A = (a_{is})_{n \times n}$ is the weight matrix between neurons, I is the bias vector of the neurons in the delayed recurrent neural networks.

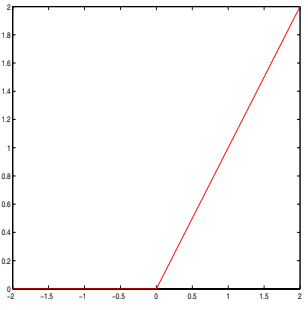


Fig. 1. the positive linear activation function

This paper is organized as follows: In Section 2, some definitions and lemmas are given used in this paper latter; In Section 3, Equation (1) is rewritten by decomposing the state space and we obtain the stability criteria which depend upon τ and are independent of τ respectively; And in Section 4, the results are concluded and the paper's feature is pointed out.

2 Definitions and Lemmas

In this section, we will give some preliminaries [9,10], including the stability criteria of linear differential difference equations and the definition of M-matrix and its equivalent conditions.

Consider the linear differential difference equations

$$\dot{x}(t) = Ax(t) + Bx(t - \tau) \ , \tag{3}$$

where $x \in \mathbb{R}^n, A, B \in \mathbb{R}^{n \times n}, x(t - \tau) = (x_1(t - \tau_1), \cdots, x_n(t - \tau_n))^T$, $0 \leq \tau_i \leq \tau, i = 1, 2, \cdots, n$ are constant delays.

The characteristic polynomial of Equation (3) is defined as

$$f(\lambda, \tau) = \det\big(\lambda E - A - B \cdot \mathrm{diag}(e^{-\lambda\tau_1}, e^{-\lambda\tau_2}, \cdots, e^{-\lambda\tau_n})\big) \ , \tag{4}$$

where E denotes $n \times n$ unit matrix.

Lemma 1. *(1) Suppose that each eigenvalue λ of the character equation $f(\lambda, 0) = 0$ has negative real part ($Re(\lambda) < 0$), then, there exists a positive number $\Delta = \Delta(A, B) > 0$ such that $x = 0$ of Equation (3) is asymptotical stability as $0 \leq \tau_i \leq \Delta$;*

(2) Assume that there is at least one eigenvalue λ of $f(\lambda, 0) = 0$ satisfying $Re(\lambda) > 0$, then, there exists a positive number $\Delta = \Delta(A, B) > 0$ such that $x = 0$ of Equation (3) is unstable as $0 \leq \tau_i \leq \Delta$.

Definition 1. *The equilibrium $x = 0$ of Equation (3) is called delay-independent asymptotic stable if it is asymptotic stable for any delays $\tau_i \geq 0, i = 1, 2, \cdots, n$.*

Lemma 2. *The equilibrium $x = 0$ of Equation (3) is delay-independent asymptotic stability if and only if*

1) Each eigenvalue λ of character equation $f(\lambda, 0) = 0$ has negative real part;

2) For any $\omega \in \mathbb{R}$ and any $\tau \geq 0$, $f(j\omega, \tau) \neq 0$, where j is the imaginary unit.

Definition 2. *A real matrix $A = (a_{is})_{n\times n}$ is called an M-matrix, if*

1) $a_{ii} > 0, i = 1, 2, \cdots, n, a_{is} \leq 0, i \neq s, i, s = 1, 2, \cdots, n$;

2) The determinants

$$\begin{vmatrix} a_{11} \cdots a_{1i} \\ \cdots\cdots\cdots \\ a_{i1} \cdots a_{ii} \end{vmatrix} > 0, i = 1, 2, \cdots, n.$$

Lemma 3. *For a matrix $A = (a_{is})_{n\times n}$, assume that $a_{ii} > 0, i = 1, 2, \cdots, n, a_{is} \leq 0, i \neq s, i, s = 1, 2, \cdots, n$. The matrix A is an M-matrix if and only if one of the following conditions holds.*

1) There exist positive constants $c_s, s = 1, 2, \cdots, n$, such that

$$\sum_{s=1}^{n} a_{is} c_s > 0, i = 1, 2, \cdots, n.$$

2) $-A$ is a stable matrix, that is, all eigenvalues of the matrix $-A$ have negative real parts.

Lemma 4. *Each eigenvalue of a matrix $A = (a_{is})_{n\times n}$ is in one of the following disk-fields:*

$$D_i = \{\lambda : |\lambda - a_{ii}| \leq \sum_{s=1, s\neq i}^{n} |a_{is}|\}, i = 1, 2, \cdots, n.$$

3 Main Results

In this section, we will set up some sufficient algebraic criteria ensuring the equilibrium of System (1) to be asymptotic stability or instability with respect to cases of delay-dependent and delay-independent.

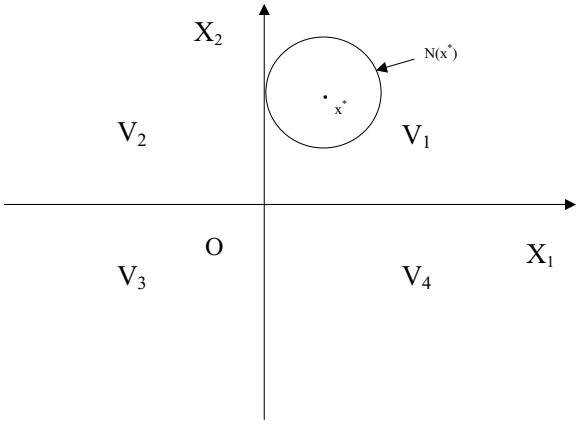


Fig. 2. Decompose $\mathbb{R}^2$ into 4 sub-domains

3.1 Rewrite System's Equations

According to decomposing $\mathbb{R} = (-\infty, \infty)$ into two intervals $(-\infty, 0)$ and $[0, \infty)$, the n-space $\mathbb{R}^n$ can be divided into 2^n sub-regions $V_k, k = 1, 2, \cdots, 2^n$, for the case $n = 2$, see Fig.2. Suppose that $x^* = (x_1^*, x_2^*, \cdots, x_n^*)^{\mathrm{T}}$ is an arbitrary equilibrium of System (1), which is an interior point in some sub-region V_{k_0}, $N(x^*) \subset V_{k_0}$ is the greatest neighborhood of the point x^*, see Fig.2. Take the transform $z = x - x^*$ and the unit step function

$$u(x_i^*) = \begin{cases} 1, & x_i^* \geq 0 \\ 0, & x_i^* < 0 \end{cases} .$$

By using the characteristic of the output functions (see formula (2)), for any $x \in N(x^*)$, we have

$$f_i(x_i) - f_i(x_i^*) = \begin{cases} x_i - x_i^*, & x_i^* > 0 \\ 0, & x_i^* < 0 \end{cases} = u(x_i^*)(x_i - x_i^*), \quad i = 1, 2, \cdots, n \ .$$

Furthermore, they can be rewritten as the vector form

$$f(x(t-\tau)) - f(x^*) = U(x^*)\big(x(t-\tau) - x^*\big) \ ,$$

where the matrix $U(x^*) = \mathrm{diag}(u(x_1^*), u(x_2^*), \cdots, u(x_n^*))$ is a diagonal matrix, the elements of which are either 0 or 1. For example, assume that $x^*=(1.2,-0.6,1.4)^{\mathrm{T}}$ is the equilibrium of a 3-order neural network, then,

$$U(x^*) = \mathrm{diag}.(1, 0, 1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \ .$$

Thus, in order to discuss the stability of the equilibrium x^* of System (1), we need only to study the same property of the trivial equilibrium $z = 0$ of the system

$$\dot{z}(t) = -Bz(t) + A^* z(t-\tau) \ , \tag{5}$$

where $A^* = AU(x^*) = \left(a_{is}u(x_s^*)\right)_{n\times n}$ related to the equilibrium x^*.

The character polynomial of System (5) is

$$g(\lambda,\tau) = \det\left(\lambda E + B - A^* \mathrm{diag}(e^{-\lambda\tau_1}, e^{-\lambda\tau_2}, \cdots, e^{-\lambda\tau_n})\right) \ .$$

Obviously, $g(\lambda, 0) = \det\left(\lambda E + B - A^*\right)$.

3.2 Case of Delay-Dependent

In reality, the delays of output of neurons in recurrent neural networks are very small. Based on Lemma 1, in order to test the stability of $z = 0$ of System (5), we need only to consider the eigenvalues of $g(\lambda, 0) = 0$.

It is easy to see that the following theorem holds (use Lemma 1).

Theorem 1. *1) If each eigenvalue of $g(\lambda, 0) = 0$ has negative real part, then, there exists a positive number $\Delta = \Delta(B, A^*)$, such that the equilibrium $z = 0$ of System (5) is asymptotic stability when time delays satisfy $0 \le \tau \le \Delta$;*

2) If $g(\lambda, 0) = 0$ has an eigenvalue λ with $Re(\lambda) > 0$, then, there exists a positive number $\Delta = \Delta(B, A^*)$, such that the equilibrium $z = 0$ of System (5) is unstable when time delays satisfy $0 \le \tau \le \Delta$.

For integer k, $1 \le k \le n$, denote

$$B_k = \mathrm{diag}(b_1, \cdots, b_k), A_k = \begin{pmatrix} a_{11} \cdots a_{1k} \\ \cdots\cdots\cdots \\ a_{k1} \cdots a_{kk} \end{pmatrix} .$$

We define $g_k(\lambda) = \det(\lambda E_k + B_k - A_k)$, here E_k is the $k \times k$ unit matrix.

Corollary 1. *For the equilibrium $x^* = (x_1^*, x_2^*, \cdots, x_n^*)^T$ of System (1), assume that $x_i^* > 0, i = 1, 2, \cdots, k$, and $x_i^* < 0, i = k+1, \cdots, n$.*

1) If each eigenvalue of $g_k(\lambda) = 0$ has negative real part, then the equilibrium x^* of System (1) is asymptotic stability so long as the time delays is sufficiently small;

2) If $g_k(\lambda) = 0$ has at least one eigenvalue with positive real part, then the equilibrium x^* of System (1) with small time lags is unstable.

Proof. By the assumption of this corollary, we see that

$$u(x_i^*) = 1, i = 1, 2, \cdots, k, u(x_i^*) = 0, i = k+1, \cdots, n \ .$$

This implies that

$$A^* = A\Phi(x^*) = \begin{vmatrix} a_{11} & \cdots & a_{1k} & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & & \vdots \\ a_{k1} & \cdots & a_{kk} & 0 & \cdots & 0 \\ a_{k+1,1} & \cdots & a_{k+1,k} & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & & \vdots \\ a_{n1} & \cdots & a_{nk} & 0 & \cdots & 0 \end{vmatrix},$$

hence,

$$\begin{aligned} g(\lambda, 0) &= \det\big(\lambda E + B - A^*\big) \\ &= \det(\lambda E_k + B_k - A_k)(\lambda + b_{k+1}) \cdots (\lambda + b_n) \\ &= g_k(\lambda)(\lambda + b_{k+1}) \cdots (\lambda + b_n) \ . \end{aligned}$$

If each eigenvalue of $g_k(\lambda) = 0$ has negative real part, then each eigenvalue of $g(\lambda, 0) = 0$ has negative real part. By Theorem 1, the corollary holds.

Remark 1. In particular, we can get simple conditions for several special cases:

1) If $k = 0$, then the equilibrium x^* of System (1) is asymptotic stability. In fact, all eigenvalues of $g(\lambda, 0) = 0$ are $-b_1, -b_2, \cdots, -b_n$;

2) If $k = 1$ and $b_1 > a_{11}$, then the equilibrium x^* of System (1) is asymptotic stable. In this case, $g_1(\lambda) = \lambda + b_1 - a_{11} = 0$ has root $\lambda = -(b_1 - a_{11}) < 0$;

3) If $k = 2$ and $b_1 + b_2 > a_{11} + a_{22}$, $(b_1 - a_{11})(b_2 - a_{22}) > a_{12}a_{21}$, then the equilibrium x^* of System (1) is asymptotic stable. In fact,

$$g_2(\lambda) = \lambda^2 + (b_1 + b_2 - a_{11} - a_{22})\lambda + (b_1 - a_{11})(b_2 - a_{22}) - a_{12}a_{21}$$

has two roots with negative real parts.

3.3 Case of Delay-Independent

Generally speaking, it is difficult to compute the boundary $\Delta(B, A^*)$ in the theorems stated above. In this sub-section, we consider delay-independent stability problems on equilibrium x^* of System (1) or, equivalently, the equilibrium $z = 0$ of System (5).

By Lemma 2, in order to study delay-independent stability problem of System (5), we not only verify that each eigenvalue λ of $g(\lambda, 0) = 0$ satisfies $\mathrm{Re}(\lambda) < 0$, which is discussed above, but also examine $g(j\omega, \tau) \neq 0$ for any $\omega \in \mathbb{R}$ and any $\tau \geq 0$, which is more difficult.

In the following, we will find out some sufficient conditions ensuring $g(j\omega, \tau) \neq 0$, which are related to M-matrix.

For matrices $B = \mathrm{diag}(b_1, b_2, \cdots, b_n)$ and $A^* = (a^*_{is})_{n\times n}$, we construct a matrix $L = (l_{is})_{n\times n}$ as follows:

$$l_{is} = \begin{cases} b_i - |a^*_{ii}|, & i = s = 1, 2, \cdots, n \\ -|a^*_{is}|, & i \neq s, i, s = 1, 2, \cdots, n \ . \end{cases}$$

In particular, if the weight matrix A is a non-negative matrix, then $L = B - A^*$.

Lemma 5. *If the matrix $L = (l_{is})_{n\times n}$ is an M-matrix, then, $g(j\omega,\tau) \neq 0$ for any $\omega \in \mathbb{R}$ and any $\tau \geq 0$.*

Proof. Let $G = (g_{is})_{n\times n} = j\omega E + B - A^* \mathrm{diag}(e^{-j\omega\tau_1}, e^{-j\omega\tau_2}, \cdots, e^{-j\omega\tau_n})$, then

$$g(j\omega,\tau) = \det(G) \ , g_{is} = \begin{cases} j\omega + b_i - a^*_{ii}e^{-j\omega\tau_i}, & i = s = 1,2,\cdots,n \\ -a^*_{is}e^{-j\omega\tau_s}, & i \neq s, i, s = 1,2,\cdots,n \ . \end{cases}$$

We estimate the term $|g_{ii}|$:

$$|g_{ii}| = |j\omega + b_i - a^*_{ii}e^{-j\omega\tau_i}| \geq |j\omega + b_i| - |a^*_{ii}e^{-j\omega\tau_i}| \geq b_i - |a^*_{ii}| \ .$$

By the assumption of this lemma that the matrix L is an M-matrix and by using 1) in Lemma 3, there exist positive numbers $c_i, i = 1,2,\cdots,n$, such that

$$c_i(b_i - |a^*_{ii}|) > \sum_{s\neq i} |a^*_{is}|c_s, i = 1,2,\cdots,n \ .$$

And this implies that

$$c_i|g_{ii}| \geq c_i(b_i - |a^*_{ii}|) > \sum_{s\neq i} |a^*_{is}|c_s = \sum_{s\neq i} |g_{is}|c_s, i = 1,2,\cdots,n \ .$$

That is,

$$|g_{ii}c_i| > \sum_{s\neq i} |g_{is}c_s|, i = 1,2,\cdots,n \ .$$

And by Lemma 4, we know that the matrix $G \cdot \mathrm{diag}(c_1, c_2, \cdots, c_n)$ has not zero eigenvalues, and this shows

$$\det\big(G \cdot \mathrm{diag}(c_1, c_2, \cdots, c_n)\big) = \det(G) \cdot \det(\mathrm{diag}(c_1, c_2, \cdots, c_n)) \neq 0 \ .$$

We get $g(j\omega,\tau) = \det(G) \neq 0$ for any $\omega \in \mathbb{R}$ and any $\tau \geq 0$.

According to Lemma 5 and Lemma 2, we immediately get the conditions ensuring the equilibrium x^* of System (1) to be delay-independent asymptotic stable.

Theorem 2. *Assume that the matrix L is an M-matrix and each eigenvalue of $g(\lambda, 0) = 0$ has negative real part, then the equilibrium x^* of System (1) is delay-independent asymptotic stable.*

Corollary 2. *Suppose that the matrix A is a non-negative matrix and $B - A^*$ is an M-matrix, then the equilibrium x^* of System (1) is delay-independent asymptotic stable.*

Proof. By the definition of the matrix L and the matrix A is non-negative, we have $L = B - A^*$, and by the assumption of the corollary, L is an M-matrix; In addition, by using 2) in Lemma 3, $-B + A^*$ is a stable matrix, that is, all roots of $g(\lambda, 0) = \det\big(\lambda E + B - A^*\big)$ have negative real parts. By Theorem 2, the corollary holds.

As the end of this section, we give three special cases of the equilibrium x^* to illustrate how to use the results obtained above, and we get some very simple inequality conditions.

Example 1. Assume that $x^* = (x_1^*, x_2^*, \cdots, x_n^*)^{\mathrm{T}}$ is an equilibrium of System (1) with $x_s^* < 0, s = 1, 2, \cdots, n$, then the equilibrium x^* is delay-independent asymptotic stability.

Proof. By using 1) in Remark 1, all eigenvalues of $g(\lambda, 0) = 0$ are real negative numbers $-b_1, -b_2, \cdots, -b_n$; And in this case, $a_{is}^* = a_{is}\phi(x_s^*) = 0, i, s = 1, 2, \cdots, n$, the matrix $L = B = \mathrm{diag}(b_1, b_2, \cdots, b_n)$ is an M-matrix. By Theorem 2, the proposition is proved.

Example 2. Assume that $x^* = (x_1^*, x_2^*, \cdots, x_n^*)^{\mathrm{T}}$ is an equilibrium of System (1) with $x_1^* > 0, x_s^* < 0, s = 2, \cdots, n$, and $b_1 > |a_{11}|$, then the equilibrium x^* is delay-independent asymptotic stable.

Proof. From $b_1 > |a_{11}| \geq a_{11}$, by using 2) in Remark 1, all eigenvalues of $g(\lambda, 0) = 0$ have negative real parts; And the matrix

$$L = \begin{pmatrix} b_1 - |a_{11}| & 0 & \cdots & 0 \\ -|a_{21}| & b_2 & \cdots & 0 \\ \vdots & & \ddots & \\ -|a_{n1}| & 0 & \cdots & b_n \end{pmatrix}$$

is an M-matrix. By Theorem 2, we get the proof.

Example 3. For the equilibrium $x^* = (x_1^*, x_2^*, \cdots, x_n^*)^{\mathrm{T}}$ of System (1) with $x_1^* > 0, x_2^* > 0, x_s^* < 0, s = 3, \cdots, n$, if the condition

$$b_1 > |a_{11}|, (b_1 - |a_{11}|)(b_2 - |a_{22}|) > |a_{12}a_{21}|$$

holds, then the equilibrium x^* of System (1) is delay-independent asymptotic stability.

Proof. First, we can easily deduce $b_2 > |a_{22}| \geq a_{22}$, so $b_1 + b_2 > a_{11} + a_{22}$; $a_{12}a_{21} \leq |a_{12}a_{21}| < (b_1 - |a_{11}|)(b_2 - |a_{22}|) \leq (b_1 - a_{11})(b_2 - a_{22})$, by using 3) in Remark 1, each eigenvalue of $g(\lambda, 0) = 0$ has negative real part; Obviously, the matrix L is an M-matrix. Here, the matrix L is

$$L = \begin{pmatrix} b_1 - |a_{11}| & -|a_{12}| & 0 & \cdots & 0 \\ -|a_{21}| & b_2 - |a_{22}| & 0 & \cdots & 0 \\ -|a_{32}| & -|a_{32}| & b_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \\ -|a_{n1}| & -|a_{n2}| & 0 & \cdots & b_n \end{pmatrix}.$$

By Theorem 2, the proposition is true.

4 Conclusions

1. For delayed recurrent neural networks(1) with positive linear activation functions, we first discuss the delay-dependent stability problem which can be determined by the roots of the character polynomial $g(\lambda, 0) = \det(\lambda E + B - A^*) = 0$. Besides this, we add the condition that the matrix L is an M-matrix to study delay-independent asymptotic stable;
2. If the interconnection matrix A is a non-negative matrix, and the matrix $B - A^*$ is an M-matrix, then the equilibrium x^* of the delayed recurrent neural network with positive linear activation functions is both delay-dependent asymptotic stable and delay-independent asymptotic stable;
3. All results obtained are algebraic criteria, it is convenient to test in system synthesis;
4. The key of this paper is that we sufficiently use the characteristic of positive linearity on the output functions of neurons. The results in this paper is true in more general conditions than existed results. In particular, the results of delay-dependent asymptotic stability are equivalent conditions.

Acknowledgments. This work was supported in part by the National Natural Science Foundation of China under grant No. 60774051.

References

1. Chua, L., Yang, L.: Cellular Neural Networks: Theory. IEEE Trans. Circuits and Systems 35, 1257–1272 (1988)
2. Slavova, A.: Stability Analysis of Cellular Neural Networks with Nonlinear Dynamics. Nonlinear Analysis: Real World Applications 2, 93–103 (2001)
3. Liao, W., Liao, X.: Stability Analysis of Cellular Neural Networks. Control Theory and Applications 20, 89–92 (2003)
4. Civalleri, P., Gill, M., Pandolfli, L.: On Stability of Cellular Neural Networks with Delay. IEEE Trans. Circuits and Systems 40, 157–165 (1993)
5. Cao, J., Zhou, D.: Stability Analysis of Delayed Celluar Neural Networks. Neural Networks 11, 1601–1605 (1998)
6. Takahashi, N.: A New Sufficient Condition for Complete Stability of Cellular Neural Networks with Delay. IEEE Trans. Circuits and Systems-1: Fundamental Theory and Applications 47, 793–799 (2000)
7. Liao, T., Wang, F.: Global Stability for Cellular Neural Networks with Time Delay. IEEE Trans. Neural Networks 11, 1481–1484 (2000)
8. Liao, X.: Mathematical Theory of Cellular Neural Networks 1. China Science 24, 902–910 (1994)
9. Hale, J.: Theory of Functional Differential Equations. Springer, New York (1977)
10. Qin, Y., Liu, Y., Wang, L.: Motion Stability of Dynamical Systems with Time-Delays, 2nd edn. Science Press, Beijing (1989)

The Dahlquist Constant Approach to Stability Analysis of the Static Neural Networks

Guanjun Li and Jin Xu

Department of Mathematics, Huainan Normal University, Anhui 232001, China
lichampion2004@126.com

Abstract. Avoiding the difficulty of constructing a proper Lyapunov function, the generalized Dahlquist constant approach is employed to investigate the exponential stability of the static neural networks. Without assuming the boundedness, monotonicity of the activations, a new sufficient conditions for existence of an unique equilibrium and the exponential stability of the neural networks are presented. An example is given to show the effectiveness of our results.

Keywords: Dahlquist constant, Static neural network, Exponential stability.

1 Introduction

The recurrently connected neural networks have been extensively studied in the past decade and successfully applied to different areas such as combinatorial optimization, pattern recognition, associative memory. These applications greatly rely on the dynamical behavior of the neural networks. Therefore, the dynamical analysis is fundamental step for practical design and applications of neural networks.

Depending upon whether neuron states (the external states of neurons) or local field states (the internal states of neurons) are taken as basic variables, neural networks can be classified as *static neural networks or local field neural networks* [1,2]. For instance, the recurrent back-propagation networks (ReBP-type NNs) [3,4], which are described by the following equations:

$$\frac{dx_i(t)}{dt} = -a_i x_i(t) + f_i(\sum_{j=1}^{n} w_{ij} x_j(t) + I_i), i = 1, 2, \cdots, n. \tag{1}$$

are *static neural networks*. Where x_i is the state variable of neuron i with $u_i = \sum_{j=1}^{n} w_{ij} x_j + I_i$ being its local field state; f_i is the activation function of neuron i; w_{ij} is the connection weight between neuron i and neuron j; n is the number of neurons in the neural networks. In contrast, Hopfield neural networks [5,6] firstly introduced by Hopfield are *local field neural networks*

$$\frac{dx_i(t)}{dt} = -a_i x_i(t) + \sum_{j=1}^{N} w_{ij} f_j(x_j(t)) + I_i, i = 1, 2, \cdots, n. \tag{2}$$

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 295–302, 2009.

where x_i is the local field state variable with $u_i = f_i(x_i)$ as the output of neuron i; I_i is the constant input from the outside of the system.

The static neural networks model (1) and the local field neural networks model (2) typically represent two fundamental modeling approaches in the current neural networks research [1,2,7]. However, they have been applied somehow in a separate manner and hardly been cross-fertilized. As a result, some types of networks such as the Hopfield-type NNs have been attracting considerable interests, and many deep theoretical results have been obtained for the models [8,9,10,11]. In contrast, other types of neural networks such as the ReBP-type NNs have not received so much attention [7,12].

To the best of authors' knowledge, the approaches extensively used in the existing investigation into stability of neural networks are mainly those based on Lyapunov direct method, that is, based on construction of Lyapunov functions. It is known, however, that no general rule can guide how a proper Lyapunov function should be constructed for a give system. Therefore, the construction of Lyapunov function becomes very skillful, there is little compatibility among the existing results.

The main difficulty for stability analysis of neural networks comes from the nonlinearity of the activation functions f_i. Almost all stability analysis is conducted under some special assumptions on f_i. There assumptions frequently include those such as differentiability, boundedness.

Motivated by the above discussion, in this paper, we consider the stability of model (1) and only assume the following properties of f_i:
(H) For each $i = 1, 2, \cdots, n$, f_i is Lipschitz continuous, that is, ther is a constant $L_i > 0$ such that $\quad |f_i(x) - f_i(y)| \leq L_i|x - y| \quad$ for any $x, y \in R$.

In the remainder of this paper, based on the generalized Dahlquist constant, a sufficient condition will be obtained to ensure the global exponential stability of (1). It should also be pointed out that here we only need the assumption (H) on the activation function. In addition, the monotonicity restriction of activation functions as well as boundedness of the activations are removed.

Throughout this paper, let $\mathbb{R}^n$ be the $n-$dimensional real vector space with norm $\|\cdot\|$ defined as $\|x\|_1 = \sum_{i=1}^{n} |x_i|$ for any $x \in \mathbb{R}^n$. E represent the identity matrix with appropriate dimensions.

2 Main Results

To obtain our main results, we still need the following definitions and lemmas. Let X be a Banach space endowed with the norm $\|\cdot\|$ and Ω be an open subset of X. Considering the following system

$$\frac{dx(t)}{dt} = F(x(t)), t \geq t_0. \tag{3}$$

where $F : \Omega \to X$ is a nonlinear operator, and $x(t) \in \Omega$.

Definition 1. *Suppose x^* is an equilibrium point of system (3), system (3) is called to be exponentially stability on a neighborhood Ω of x^*, if there exists $\alpha > 0$ and $\varepsilon > 0$ such that $\|x(t) - x^*\| \leq \varepsilon e^{-\alpha t}\|x(t_0) - x^*\|$ for $t \geq t_0$, where $x(t)$ is any solution of (3) initiated from $x(t_0)$.*

Definition 2. *[13] Suppose Ω be an open subset of Banach space X, $F : \Omega \to X$ is an operator, The constant*

$$\alpha(F) = \sup_{x,y\in\Omega, x\neq y} \frac{1}{\|x-y\|} \lim_{r\to+\infty} [\|(F+rE)x - (F+rE)y\| - r\|x-y\|] \quad (4)$$

is called to be the generalized Dahlquist constant of F on Ω.

Definition 3. *Suppose Ω be an open subset of Banach spaceX, $F : \Omega \to X$ is an operator, and x^0 is any fixed point in Ω. The constant*

$$\alpha(F, x^0) = \sup_{x\in\Omega, x\neq x^0} \frac{1}{\|x-x^0\|} \lim_{r\to+\infty} [\|(F+rE)x - (F+rE)x^0\| - r\|x-x^0\|] \quad (5)$$

is called to be the generalized relative Dahlquist constant of F at x^0.

Lemma 1. *[13] If $\alpha(F) < 0$, then F is an injective mapping on Ω. In addition, if $\Omega = \mathbb{R}^n$, then F is a homeomorphism of $\mathbb{R}^n$.*

Lemma 2. *Suppose $x^* \in \Omega$ is an equilibrium point of the system (3), if the operator F satisfies $\alpha(F, x^*) < 0$, then x^* is the unique equilibrium point of the system (3) in Ω, x^* is exponentially stable.*

Proof. Suppose the system (3) has an unique equilibrium point x^* in Ω. Otherwise, let $\widetilde{x} \in \Omega$ be any other equilibrium point of (3) different from x^*. Then $F(\widetilde{x}) = F(x^*) = 0, \widetilde{x} \neq x^*$. By Definition 3, we infer that

$$\begin{aligned}
\alpha(F, x^*) &= \sup_{x\in\Omega, x\neq x^*} \frac{1}{\|x-x^*\|} \lim_{r\to+\infty} [\|(F+rE)x - (F+rE)x^*\| - r\|x-x^*\|] \\
&\geq \frac{1}{\|\widetilde{x}-x^*\|} \lim_{r\to+\infty} [\|(F+rE)\widetilde{x} - (F+rE)x^*\| - r\|\widetilde{x}-x^*\|] \\
&= 0
\end{aligned}$$

in contradiction to $\alpha(F, x^*) < 0$. Thus $\widetilde{x} = x^*$, and the equilibrium point of (3) is unique in Ω.

If $x^* \in \Omega$ is the equilibrium point of system (3), then $F(x^*) = 0$. Suppose $x(t)$ is the solution of system (3) initiated from $x(0) = x_0 \in \Omega$. we have

$$(e^{rt}x(t))' = re^{rt}x(t) + e^{rt}Fx(t) = e^{rt}(F+rE)x(t)$$

for all $t > 0$ and $r > 0$.

$$e^{rt}[x(t) - x^*] = [x_0 - x^*] + \int_0^t e^{ru}[(F+rE)x(u) - (F+rE)x^*]du$$

then

$$e^{rt} \parallel x(t) - x^* \parallel - \parallel x_0 - x^* \parallel \leq \int_0^t e^{ru} \parallel (F + rE)x(u) - (F + rE)x^*] \parallel du$$

$$(e^{rt} \parallel x(t) - x^* \parallel)'_t \leq e^{rt} \parallel (F + rE)x(t) - (F + rE)x^*] \parallel .$$

Let $r \to +\infty$, we have

$$(\parallel x(t) - x^* \parallel)'_t \leq \alpha(F, x^*) \parallel x(t) - x^*] \parallel$$

integration from 0 to t yields

$$\parallel x(t) - x^* \parallel \leq e^{\alpha(F,x^*)t} \parallel x_0 - x^*] \parallel .$$

Since $\alpha(F, x^*) < 0$, we infer that the equilibrium point x^* is exponentially stable in Ω.

Theorem 1. *If* $L_i \sum_{j=1}^{n} |w_{ji}| < a_i, i = 1, 2, \cdots, n$, *then the neural networks (1) has an unique equilibrium point, which is globally exponentially stable.*

Proof. Step 1: For system (3), define an operator $F : \mathbb{R}^n \to \mathbb{R}^n$ by

$$F_i(x) = -a_i x_i + f_i(\sum_{j=1}^{n} w_{ij} x_j + I_i)$$

where $x = [x_1, x_2, \cdots, x_n]^T \in \mathbb{R}^n$, $F(x) = [F_1(x), F_2, \cdots, F_n(x)]^T$. we can conclude that system (1) and system (3) have the same equilibrium set. Then an equilibrium point $x^* = (x_1^*, \cdots, x_n^*)$ of neural networks (3) is a point in R^n such that $F(x^*) = 0$. If we can prove that F is a homeomorphism from $\mathbb{R}^n$ to $\mathbb{R}^n$, then the existence and uniqueness of the equilibrium are proved.
From Definition 2, we have

$$\alpha(F) = \sup_{x,y \in \mathbb{R}^n, x \neq y} \frac{1}{\|x - y\|_1} \lim_{r \to +\infty} [\|(F + rE)x - (F + rE)y\|_1 - r\|x - y\|_1]$$

where

$$\begin{aligned}
&\|(F + rE)x - (F + rE)y\|_1 - r\|x - y\|_1 \\
&= \sum_{i=1}^{n} \{| - (a_i x_i - a_i y_i) + f_i(\sum_{j=1}^{n} w_{ij} x_j + I_i) \\
&- f_i(\sum_{j=1}^{n} w_{ij} y_j + I_i) + r(x_i - y_i)| - r|x_i - y_i|\}.
\end{aligned}$$

For any $x, y \in \mathbb{R}^n$ and $r > 0$ sufficiently large,we have

$$\begin{aligned}
&\|(F+rE)x-(F+rE)y\|_1 - r\|x-y\|_1 \\
&\leq \sum_{i=1}^{n}\{-|(a_ix_i-a_iy_i)| + |f_i(\sum_{j=1}^{n} w_{ij}x_j + I_i) - f_i(\sum_{j=1}^{n} w_{ij}y_j + I_i)|\} \\
&\leq -\sum_{i=1}^{n} a_i|x_i-y_i| + \sum_{i=1}^{n} L_i|(\sum_{j=1}^{n} w_{ij}x_j + I_i) - (\sum_{j=1}^{n} w_{ij}y_j + I_i)| \\
&= -\sum_{i=1}^{n} a_i|x_i-y_i| + \sum_{i=1}^{n}\sum_{j=1}^{n} L_i|w_{ij}||x_j-y_j| \\
&= -\sum_{i=1}^{n}(a_i - L_i\sum_{j=1}^{n}|w_{ji}|)|x_i-y_i| \\
&\leq -\min_{1\leq i\leq n}(a_i - L_i\sum_{j=1}^{n}|w_{ji}|) \parallel x-y \parallel_1 .
\end{aligned}$$

Hence, $\alpha(F) \leq -\min\limits_{1\leq i\leq n}(a_i - L_i\sum\limits_{j=1}^{n}|w_{ji}|) < 0$, by Lemma 1, we conclude that system (3) or equivalently, system (1) has an unique equilibrium point x^*.

Step 2: Let $x = [x_1, x_2, \cdots, x_n]^T \in \mathbb{R}^n$, to simiplify the proof, we shift the equilibrium point x^* of (1) to the origin via the transformation $y_i(t) = x_i(t) - x_i^*$, then Eq.(1) can be transformed into the following form:

$$\frac{dy_i(t)}{dt} = -a_iy_i(t) + f_i[\sum_{j=1}^{n} w_{ij}(y_j(t)+x_j^*) + I_i] - f_i(\sum_{j=1}^{n} w_{ij}x_j^* + I_i), i = 1, 2, \cdots, n. \tag{6}$$

Define an operator $G : \mathbb{R}^n \to \mathbb{R}^n$ by

$$G_i(y) = -a_iy_i(t) + f_i[\sum_{j=1}^{n} w_{ij}(y_j(t)+x_j^*) + I_i] - f_i(\sum_{j=1}^{n} w_{ij}x_j^* + I_i), i = 1, 2, \cdots, n.$$

Where $y = [y_1, y_2, \cdots, y_n]^T \in \mathbb{R}^n$, $G(x) = [G_1(x), G_2, \cdots, G_n(x)]^T$, then system (6) can be described by

$$\frac{dy(t)}{dt} = G(y(t)), t \geq t_0. \tag{7}$$

We have

$$\alpha(G, 0) = \sup_{y\in\mathbb{R}^n, y\neq 0} \frac{1}{\|y\|_1} \lim_{r\to+\infty} \sum_{i=1}^{n}[|(G)_i(y) + ry_i| - r|y_i|].$$

For any $y \in \mathbb{R}^n$ and $r > 0$ sufficiently large, we have

$$\begin{aligned}
&\sum_{i=1}^{n}[|(G)_i(y) + ry_i| - r|y_i|] \\
&= \sum_{i=1}^{n} | - a_i y_i(t) + f_i[\sum_{j=1}^{n} w_{ij}(y_j(t) + x_j^*) + I_i] - f_i(\sum_{j=1}^{n} w_{ij}x_j^* + I_i) + ry_i| - r|y_i| \\
&\leq \sum_{i=1}^{n}\{-|(a_i y_i)| + |f_i[\sum_{j=1}^{n} w_{ij}(y_j + x_j^*) + I_i] - f_i(\sum_{j=1}^{n} w_{ij}x_j^* + I_i)|\} \\
&\leq -\sum_{i=1}^{n} a_i|y_i| + \sum_{i=1}^{n} L_i|[\sum_{j=1}^{n} w_{ij}(y_j + x_j^*) + I_i] - (\sum_{j=1}^{n} w_{ij}x_j^* + I_i)| \\
&= -\sum_{i=1}^{n} a_i|y_i| + \sum_{i=1}^{n}\sum_{j=1}^{n} L_i|w_{ij}||y_j| \\
&= -\sum_{i=1}^{n}(a_i - L_i\sum_{j=1}^{n}|w_{ji}|)|y_i| \\
&\leq -\min_{1\leq i\leq n}(a_i - L_i\sum_{j=1}^{n}|w_{ji}|) \parallel y \parallel_1 .
\end{aligned}$$

Hence, $\alpha(G, 0) \leq -\min\limits_{1\leq i\leq n}(a_i - L_i\sum\limits_{j=1}^{n}|w_{ji}|) < 0$, by Lemma 2, we conclude that the origin of (7) or equivalently the equilibrium point x^* is globally exponentially stable.

Corollary 1. *If $L_i\sum\limits_{j=1}^{n}|w_{ji}| < a_i$, $i = 1, 2, \cdots, n$, then the neural networks (2) has an unique equilibrium point, which is globally exponentially stable.*

Proof. Similar to the Proof of Theorem 1 and hence omitted.

3 An Illustrative Example

In this section, we will give a example to show the effectiveness of our results. Consider the following neural networks

$$\frac{dx(t)}{dt} = -Ax(t) + f(Wx) + I$$

where the parameters

$$A = \begin{bmatrix} 3 & 0 \\ 0 & 4 \end{bmatrix}, \quad W = \begin{bmatrix} -2.1 & 2 \\ -0.5 & 5 \end{bmatrix}$$

the activation $f_1(x) = \frac{3}{4}\sin x + \frac{1}{4}x, f_2(x) = \frac{1}{3}\sin x + \frac{1}{6}x$. Obviously, one can find that f_i satisfies the condition (H) with $L_1 = 1, L_2 = \frac{1}{2}$, on the other hand, it can be verified that f_i is not bounded and nonmonotonic. Thus, the condition in [14,15] fails to conclude whether this neural network is globally stable or not. Therefore, by Theorem 1, we know this neural network has and only has an equilibrium point which is globally exponentially stable.

4 Concluding Remarks

In this paper, by using the generalized Dahlquist constant approach, a new sufficient conditions have been derived to ensure the global exponential stability of the equilibrium point of static neural networks, which voiding the difficulty of constructing a proper Lyapunov function. Furthermore, the obtained results removed the boundedness, monotonicity of the activations.

Acknowledgments. This work was jointly supported by the Natural Science Foundation of Anhui Provincial Education Department under Grants KJ2008B251, the Foundation for Young of Anhui Provincial Education Department under Grants 2008jq1132.

References

1. Haykin, S.: Neural Networks: A Comprehensive Foundation, New York (1994)
2. Hertz, J., Krogh, A.: Introduction to Theory of Neural Computation. Addison-Wesley, Reading (1994)
3. Pineda, F.: Generalization of Back-propagation to Recurrent Neural Networks. Phys. Rev. Lett. 59, 2229–2232 (1987)
4. Rohwer, R., Forrest, B.: Training Time-dependence in Neural Networks. In: 1st IEEE Int. Conf. Neural Networks, San Diego, CA, pp. 701–708 (1987)
5. Hopfield, J.: Neural Networks and Physical Systems with Emergent Collective Computational Abilities. Proc. Nat. Acad. Sci. 2, 2554–2558 (1982)
6. Hopfield, J., Tank, D.: Computing with Neural Circuits:A Model. Scinence 233, 625–633 (1986)
7. Xu, Z., Qiao, H., Peng, J., Zhang, B.: A Comparative Study of Two Modeling Approaches in Neural Networks. Neural Networks 17, 73–85 (2004)
8. Arik, S.: Global Asymptotic Stability of A Class of Dynamical Neural Networks. IEEE Trans. Circuits Syst. I 47, 568–571 (2000)
9. Chen, T., Amari, S.: New Theorems on Global Convergence of Some Dynamical Systems. Neural Networks 14, 252–255 (2001)
10. Cohen, M., Grossberg, S.: Absolute Stability of Global Pattern Formation and Parallel Memory Storage by Competitive Neural Networks. IEEE Trans. Syst. Man Cybern. 13, 815–826 (1983)
11. Li, G., Xu, J.: Analysis of Global Asymptotic Stability of Delayed Recurrent Neural Networks. Dyna. of Cont. Disc. and Impu. Syst. Math. Anal. 13, 306–309 (2006)

12. Li, P., Cao, J.: Stability in Static Delayed Neural Networks: A Nonlinear Measure Approach. Neurocomputing 69, 1776–1781 (2006)
13. Wan, A.N., Peng, J., Wang, M.: Generalized Relative Dahlquis t Constant with Application in Stability Analysis of Nonlinear Systems. Mathematica Applicata 18, 328–332 (2005)
14. Xu, S., Lam, D.W.C.: Ho: Globally Robust Exponenttial Stability Analysis for Interval Recurrent Neural Networks. Phys. Lett. A 325, 124–133 (2004)
15. Liang, J., Cao, J.: A Based-on LMI Stability Criterion for Delayed Recurrent Neural networks. Chaos, Solutions & Fractals 28, 154–160 (2006)

Global Exponential Stability of Reaction-Diffusion Delayed BAM Neural Networks with Dirichlet Boundary Conditions

Chaojin Fu and Ailong Wu

Department of Mathematics, Hubei Normal University,
Huangshi 435002, China
fuchaojin@sina.com, alequ@126.com

Abstract. In this paper, the global exponential stability for a class of reaction-diffusion delayed bidirectional associate memory (BAM) neural networks with Dirichlet boundary conditions is addressed by using the method of variation parameter and inequality technique, the delay-independent sufficient conditions to guarantee the uniqueness and global exponential stability of the equilibrium point of such networks are established. Finally, an example is given to show the effectiveness of the obtained result.

Keywords: Bidirectional associate memory, Global exponential stability, Reaction-diffusion, Equilibrium point.

1 Introduction

The bidirectional associate memory (BAM) neural networks was first introduced by Kosto [1] in 1988. Recently, the stability of BAM neural networks has been extensively studied in both theory and applications (see [5-12]). Refs. [5,7-9,12] had derived some sufficient conditions for the global exponential stability of delayed BAM neural networks by constructing suitable Lyapunov functional, Refs. [6,10,11] used some analytic techniques and derived several simple criteria for the global exponential stability of delayed BAM neural networks.

However, strictly speaking, diffusion effects can not be avoided when electrons are moving in asymmetric electromagnetic fields, so we must consider that the activations vary in space as well as in time. Refs. [11-18] have considered the stability of neural networks with diffusion terms. It is also common to consider the diffusion effects in biological systems (such as immigration). The boundary conditions of the investigated reaction-diffusion neural networks [11-17]are all Neumann boundary conditions, the stability of reaction-diffusion neural networks with Dirichlet boundary conditions has not yet been full developed. So further investigation to these neural networks is significant.

To best of our knowledge, few authors have considered global exponential stability of reaction-diffusion delayed BAM neural networks with Dirichlet boundary conditions, which is very important in theory and applications and also is

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 303–312, 2009.

a very challenging problem. Motivated by the above discussion, in this paper, we will investigate the global exponential stability of a class of reaction-diffusion delayed BAM neural networks with Dirichlet boundary conditions. The work will have significance impact on the design and applications of neural circuits.

The rest of this paper is organized as follows. Preliminaries and model description are given in Section 2. In Section 3, we give main results and their proof. An example is given in Section 4. Finally, we give the conclusion.

2 Model Description and Preliminaries

Consider the reaction-diffusion delayed BAM neural networks with Dirichlet boundary conditions:

$$\begin{aligned}
\frac{\partial u_i(t,x)}{\partial t} &= \sum_{k=1}^{l} \frac{\partial}{\partial x_k}\big(D_{ik}\frac{\partial u_i}{\partial x_k}\big) - a_i u_i(t,x) + \sum_{j=1}^{m} b_{ji} f_j(v_j(t,x)) \\
&\quad + \sum_{j=1}^{m} \bar{b}_{ji} g_j(v_j(t-\sigma_{ji}(t),x)) + I_i, \\
\frac{\partial v_j(t,x)}{\partial t} &= \sum_{k=1}^{l} \frac{\partial}{\partial x_k}\big(\overline{D}_{jk}\frac{\partial v_j}{\partial x_k}\big) - c_j v_j(t,x) + \sum_{i=1}^{n} d_{ij} f_i(u_i(t,x)) \\
&\quad + \sum_{i=1}^{n} \bar{d}_{ij} g_i(u_i(t-\tau_{ij}(t),x)) + J_j, \\
u_i(t,x) &= 0, \quad v_j(t,x) = 0, \quad x \in \partial\Omega, \quad -\tau \le t < +\infty, \quad -\sigma \le t < +\infty, \\
u_i(s,x) &= \xi_i(s,x), \quad -\tau \le s \le 0, \quad 0 \le \tau_{ij}(t) \le \tau, \\
v_j(s,x) &= \eta_j(s,x), \quad -\sigma \le s \le 0, \quad 0 \le \sigma_{ji}(t) \le \sigma,
\end{aligned} \tag{1}$$

for $1 \le i \le n, 1 \le j \le m$ and $t \ge 0$. where $x = (x_1, \cdots, x_l)^T \in \Omega \subset R^l$ and $\Omega = \{x = (x_1, \cdots, x_l)^T : |x_k| < \omega_k, k = 1, 2, \cdots, l\}$ is a bounded compact set with smooth boundary $\partial\Omega$ and $mes\Omega > 0$ in space R^l; $u_i(t,x)$ and $v_j(t,x)$ denote the state variable of the ith neurons and jth neurons at time t and in space x, respectively; f_i, f_j, g_i, g_j are nonlinear activation functions; $\xi_i(s,x)$ and $\eta_j(s,x)$ denote the initial value; $\tau_{ij}(t)$ and $\sigma_{ji}(t)$ denote time delays required for neural processing and axonal transmission of signals; I_i and J_j denote external inputs to the neurons introduced from outside the network; $a_i > 0$ and $c_j > 0$ denote the rate with which the ith unit and jth unit will reset their potential to the resting state in isolation when disconnected from the networks and external inputs, respectively; $b_{ji}, \bar{b}_{ji}, d_{ij}, \bar{d}_{ij}$ denote synaptic connection weights; smooth functions $D_{ik} = D_{ik}(t,x,u) \ge 0$ and $\overline{D}_{jk} = D_{jk}(t,x,v) \ge 0$ denote transmission diffusion operators.

Let the solution of system (1) denote $(u(t,x;\xi), v(t,x;\eta))^T$, where $u(t,x;\xi) = (u_1(t,x;\xi_1), \cdots, u_n(t,x;\xi_n))^T$, $v(t,x;\eta) = (v_1(t,x;\eta_1), \cdots, v_m(t,x;\eta_m))^T$, or $(u(t), v(t))^T$, if no confusion occurs.

Let $L^2(\Omega)$ be the space of real Lebesgue measurable functions on Ω. It is a Banach space for L_2-norm $\|u\|_2 = (\int_\Omega |u(x)|^2 \,\mathrm{d}x)^{\frac{1}{2}}$, where $|u|$ denotes the Euclid norm of a vector $u \in R^n$ for any integer n. The norm $\|u\|$ is defined by $\|u\| = \sum_{i=1}^{n} \|u_i\|_2$.

Note that, $\xi = \{(\xi_1(s,x), \cdots, \xi_n(s,x))^T : -\tau \le s \le 0\}$ is $C([-\tau, 0] \times \Omega, R^n)$-valued function, $C([-\tau, 0] \times \Omega, R^n)$ is the space of all continuous R^n-valued functions defined on $[-\tau, 0] \times \Omega$ with the norm $\|\xi\| = \sup_{-\tau \le t \le 0} \{\sum_{i=1}^{n} \|\xi_i(t)\|_2\}$, $\|\xi_i(t)\|_2^2 = \int_\Omega |\xi_i(t,x)|^2 \,\mathrm{d}x$, $\eta = \{(\eta_1(s,x), \cdots, \eta_m(s,x))^T : -\sigma \le s \le 0\}$ is similar to ξ.

Throughout this paper, we have the following assumptions on system (1):

(A_1) $f_i, f_j, g_i, g_j : R \to R$ satisfy

$$
\begin{aligned}
&P_i = \sup_{u \ne v} \left| \frac{f_i(u) - f_i(v)}{u - v} \right|, \quad Q_j = \sup_{u \ne v} \left| \frac{f_j(u) - f_j(v)}{u - v} \right|, \\
&L_i = \sup_{u \ne v} \left| \frac{g_i(u) - g_i(v)}{u - v} \right|, \quad M_j = \sup_{u \ne v} \left| \frac{g_j(u) - g_j(v)}{u - v} \right|, \\
&|f_i(u)| \le A_i < +\infty, \quad |f_j(u)| \le B_j < +\infty, \\
&|g_i(u)| \le C_i < +\infty, \quad |g_j(u)| \le D_j < +\infty,
\end{aligned}
$$

for $1 \le i \le n, 1 \le j \le m, u, v \in R$, where $P_i, Q_j, L_i, M_j, A_i, B_j, C_i, D_j$ are positive constants.

(A_2)

$$
\rho\left(\begin{pmatrix} A & 0 \\ 0 & C \end{pmatrix}^{-1} \begin{pmatrix} 0 & B(0) \\ D(0) & 0 \end{pmatrix} \right) < 1,
$$

where

$$
\begin{aligned}
&A = diag\{a_1^*, \cdots, a_n^*\}, \quad a_i^* = a_i + \sum_{k=1}^{l} \frac{D_{ik}^*}{\omega_k^2}, \\
&D_{ik}^* = \inf_{x \in \Omega} D_{ik} \ge 0 \,\text{is nonnegative number}, \\
&C = diag\{c_1^*, \cdots, c_m^*\}, \quad c_j^* = c_j + \sum_{k=1}^{l} \frac{\overline{D}_{jk}^*}{\omega_k^2}, \\
&\overline{D}_{jk}^* = \inf_{x \in \Omega} \overline{D}_{jk} \ge 0 \,\text{is nonnegative number}, \\
&B(z) = (B_{ij}(z))_{n \times m}, \quad B_{ij}(z) = |b_{ji}| \, Q_j + \left|\bar{b}_{ji}\right| M_j e^{z\sigma}, \\
&D(z) = (D_{ji}(z))_{m \times n}, \quad D_{ji}(z) = |d_{ij}| \, P_i + \left|\bar{d}_{ij}\right| L_i e^{z\tau}.
\end{aligned}
$$

Definition 1. The equilibrium point $(u^*, v^*)^T$ of system (1) is said to be globally exponentially stable, if there exist constants $\lambda > 0$ and $M > 1$ such that for any ξ and η and all solutions $(u, v)^T$ of system (1)

$$
\|u(t;\xi) - u^*\| + \|v(t;\eta) - v^*\| \le M(\|\xi - u^*\| + \|\eta - v^*\|)e^{-\lambda t},
$$

for all $t \ge 0$.

Lemma 1. If $M \geq 0$ and $\rho(M) < 1$, then $(I - M)^{-1} \geq 0$, where I denotes the identity matrix and $\rho(M)$ denotes the spectral radius of a square matrix M, $M \geq 0$ means M is a nonnegative matrix.

Lemma 2 [18]. Let Ω be a cube $|x_k| < \omega_k (k = 1, 2, \cdots, l)$, and let $h(x)$ be a real-valued function belonging to $C^1(\Omega)$, which vanish on the boundary $\partial\Omega$ of Ω, i.e., $h(x)|_{\partial\Omega} = 0$. Then $\int_\Omega h^2(x)\mathrm{d}x \leq \omega_k^2 \int_\Omega \left|\frac{\partial h}{\partial x_k}\right|^2 \mathrm{d}x$.

3 Main Results

In this section, we will study the global exponential stability and uniqueness of equilibrium point of system (1). Let $D_{ik} = 0, \overline{D}_{jk} = 0, 1 \leq i \leq n, 1 \leq j \leq m, k = 1, 2, \cdots, l$, then system (1) turns to

$$\begin{aligned}
\frac{\partial u_i(t,x)}{\partial t} &= -a_i u_i(t,x) + \sum_{j=1}^{m} b_{ji} f_j(v_j(t,x)) + \sum_{j=1}^{m} \bar{b}_{ji} g_j(v_j(t-\sigma_{ji}(t),x)) + I_i, \\
\frac{\partial v_j(t,x)}{\partial t} &= -c_j v_j(t,x) + \sum_{i=1}^{n} d_{ij} f_i(u_i(t,x)) + \sum_{i=1}^{n} \bar{d}_{ij} g_i(u_i(t-\tau_{ij}(t),x)) + J_j,
\end{aligned} \tag{2}$$

It is known under the assumption (A_1), it can always guarantee that system (2) has an equilibrium point $(u^*, v^*)^T$, where $u^* = (u_1^*, \cdots, u_n^*)^T$, $v^* = (v_1^*, \cdots, v_m^*)^T$. Clearly, $(u^*, v^*)^T$ is also the equilibrium point of system (1). Therefore system (1) is equivalent to

$$\begin{aligned}
\frac{\partial(u_i(t) - u_i^*)}{\partial t} &= \sum_{k=1}^{l} \frac{\partial}{\partial x_k}\Big(D_{ik} \frac{\partial(u_i - u_i^*)}{\partial x_k}\Big) - a_i[u_i(t) - u_i^*] \\
&\quad + \sum_{j=1}^{m} b_{ji}[f_j(v_j(t) - f_j(v_j^*)] + \sum_{j=1}^{m} \bar{b}_{ji}[g_j(v_j(t-\sigma_{ji}(t))) - g_j(v_j^*)], \\
\frac{\partial(v_j(t) - v_j^*)}{\partial t} &= \sum_{k=1}^{l} \frac{\partial}{\partial x_k}\Big(\overline{D}_{jk} \frac{\partial(v_j - v_j^*)}{\partial x_k}\Big) - c_j[v_j(t) - v_j^*] \\
&\quad + \sum_{i=1}^{n} d_{ij}[f_i(u_i(t)) - f_i(u_i^*)] + \sum_{i=1}^{n} \bar{d}_{ij}[g_i(u_i(t-\tau_{ij}(t))) - g_i(u_i^*)].
\end{aligned} \tag{3}$$

Theorem 1. If system (1) satisfies assumptions (A_1) and (A_2), then the equilibrium point $(u^*, v^*)^T$ of system (1) is unique and globally exponentially stable.

Proof. From (A_2) and Lemma 1, we have

$$\left(I - \begin{pmatrix} A & 0 \\ 0 & C \end{pmatrix}^{-1} \begin{pmatrix} 0 & B(0) \\ D(0) & 0 \end{pmatrix}\right)^{-1} \geq 0.$$

Then there exists a positive constant $\lambda < \min\{a_1^*, \cdots, a_n^*, c_1^*, \cdots, c_m^*\}$ such that

$$E(\lambda) := \left(I - \begin{pmatrix} A - \lambda I & 0 \\ 0 & C - \lambda I \end{pmatrix}^{-1} \begin{pmatrix} 0 & B(\lambda) \\ D(\lambda) & 0 \end{pmatrix}\right)^{-1} \geq 0.$$

Let $\bar{u}_i = u_i - u_i^*, \bar{v}_j = v_j - v_j^*, i = 1, \cdots, n, j = 1, \cdots, m$. Multiplying the former of (3) by $\bar{u}_i$ and integrating with respect to x, it is easy to obtain

$$\begin{aligned}\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\left\|\bar{u}_i(t)\right\|_2^2 &= \int_\Omega \bar{u}_i(t)\sum_{k=1}^{l}\frac{\partial}{\partial x_k}\big(D_{ik}\frac{\partial \bar{u}_i}{\partial x_k}\big)\mathrm{d}x - a_i\left\|\bar{u}_i(t)\right\|_2^2 \\ &\quad + \int_\Omega \{\sum_{j=1}^{m} b_{ji}\bar{u}_i(t)[f_j(v_j(t) - f_j(v_j^*)] \\ &\quad + \sum_{j=1}^{m}\bar{b}_{ji}\bar{u}_i(t)[g_j(v_j(t-\sigma_{ji}(t))) - g_j(v_j^*)]\}\mathrm{d}x\end{aligned}$$

From Green formula, the boundary condition and Lemma 2, we have

$$\begin{aligned}&\int_\Omega \bar{u}_i(t)\sum_{k=1}^{l}\frac{\partial}{\partial x_k}\big(D_{ik}\frac{\partial \bar{u}_i}{\partial x_k}\big)\mathrm{d}x = \int_\Omega \bar{u}_i(t)\nabla\cdot\big(D_{ik}\frac{\partial \bar{u}_i}{\partial x_k}\big)_{k=1}^{l}\mathrm{d}x \\ &= \int_{\partial\Omega}\bar{u}_i(t)\big(D_{ik}\frac{\partial \bar{u}_i}{\partial x_k}\big)_{k=1}^{l}\mathrm{d}x - \int_\Omega \big(D_{ik}\frac{\partial \bar{u}_i}{\partial x_k}\big)_{k=1}^{l}\nabla\cdot(\bar{u}_i)\mathrm{d}x \\ &= -\int_\Omega \big(D_{ik}\frac{\partial \bar{u}_i)}{\partial x_k}\big)_{k=1}^{l}\nabla\cdot\big(\bar{u}_i\big)\mathrm{d}x = -\sum_{k=1}^{l}\int_\Omega D_{ik}\big(\frac{\partial \bar{u}_i}{\partial x_k}\big)^2\mathrm{d}x \\ &\leq -\sum_{k=1}^{l}\int_\Omega \frac{D_{ik}}{\omega_k^2}\bar{u}_i^2\mathrm{d}x \leq -\sum_{k=1}^{l}\frac{D_{ik}^*}{\omega_k^2}\int_\Omega \bar{u}_i^2\mathrm{d}x = -\sum_{k=1}^{l}\frac{D_{ik}^*}{\omega_k^2}\left\|\bar{u}_i(t)\right\|_2^2\end{aligned}$$

where $\bigtriangledown = (\dfrac{\partial}{\partial x_1}, \cdots, \dfrac{\partial}{\partial x_l})^T$ is the gradient operator.
Then we can get

$$\begin{aligned}\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\left\|\bar{u}_i(t)\right\|_2^2 &\leq -(a_i + \sum_{k=1}^{l}\frac{D_{ik}^*}{\omega_k^2})\left\|\bar{u}_i(t)\right\|_2^2 + \int_\Omega\{\sum_{j=1}^{m} b_{ji}\bar{u}_i(t)[f_j(v_j(t) - f_j(v_j^*)] \\ &\quad + \sum_{j=1}^{m}\bar{b}_{ji}\bar{u}_i(t)[g_j(v_j(t-\sigma_{ji}(t))) - g_j(v_j^*)]\}\mathrm{d}x \\ &= -a_i^*\left\|\bar{u}_i(t)\right\|_2^2 + \int_\Omega\{\sum_{j=1}^{m} b_{ji}\bar{u}_i(t)[f_j(v_j(t) - f_j(v_j^*)] \\ &\quad + \sum_{j=1}^{m}\bar{b}_{ji}\bar{u}_i(t)[g_j(v_j(t-\sigma_{ji}(t))) - g_j(v_j^*)]\}\mathrm{d}x\end{aligned}$$

So we can easily obtain that

$$\frac{\mathrm{d}}{\mathrm{d}t}\left\|\bar{u}_i(t)\right\|_2 \leq -a_i^*\left\|\bar{u}_i(t)\right\|_2 + \frac{1}{\left\|\bar{u}_i(t)\right\|_2}\int_\Omega\{\sum_{j=1}^m b_{ji}\bar{u}_i(t)[f_j(v_j(t)-f_j(v_j^*)] + \sum_{j=1}^m \bar{b}_{ji}\bar{u}_i(t)[g_j(v_j(t-\sigma_{ji}(t)))-g_j(v_j^*)]\}\mathrm{d}x \tag{4}$$

By using the the method of variation parameter and Holder inequality, we obtain

$$\begin{aligned}\left\|\bar{u}_i(t)\right\|_2 \leq & e^{-a_i^* t}\left\|\bar{u}_i(0)\right\|_2 + \int_0^t e^{-a_i^*(t-s)}\frac{1}{\left\|\bar{u}_i(s)\right\|_2}\int_\Omega\{\sum_{j=1}^m b_{ji}\bar{u}_i(s)[f_j(v_j(s)-f_j(v_j^*)] \\ & + \sum_{j=1}^m \bar{b}_{ji}\bar{u}_i(s)[g_j(v_j(s-\sigma_{ji}(s)))-g_j(v_j^*)]\}\mathrm{d}x\mathrm{d}s \\ \leq & e^{-a_i^* t}\left\|\bar{u}_i(0)\right\|_2 + \int_0^t e^{-a_i^*(t-s)}\frac{1}{\left\|\bar{u}_i(s)\right\|_2}\{\sum_{j=1}^m |b_{ji}|\, Q_j\left\|\bar{u}_i(s)\right\|_2\left\|\bar{v}_j(s)\right\|_2 \\ & + \sum_{j=1}^m \left|\bar{b}_{ji}\right| M_j\left\|\bar{u}_i(s)\right\|_2\left\|\bar{v}_j(s-\sigma_{ji}(s))\right\|_2\}\mathrm{d}s \\ = & e^{-a_i^* t}\left\|\bar{u}_i(0)\right\|_2 + \int_0^t e^{-a_i^*(t-s)}\{\sum_{j=1}^m |b_{ji}|\, Q_j\left\|\bar{v}_j(s)\right\|_2 \\ & + \sum_{j=1}^m \left|\bar{b}_{ji}\right| M_j\left\|\bar{v}_j(s-\sigma_{ji}(s))\right\|_2\}\mathrm{d}s\end{aligned}$$

Therefore, we have

$$\begin{aligned}\sup_{0\leq\theta\leq t}\left(\left\|\bar{u}_i(\theta)\right\|_2 e^{\lambda\theta}\right) \leq & \sup_{0\leq\theta\leq t} e^{(\lambda-a_i^*)\theta}\left\|\bar{u}_i(0)\right\|_2 + \sup_{0\leq\theta\leq t}\int_0^\theta e^{\lambda\theta-a_i^*(\theta-s)} \\ & \{\sum_{j=1}^m |b_{ji}|\, Q_j\left\|\bar{v}_j(s)\right\|_2 + \sum_{j=1}^m \left|\bar{b}_{ji}\right| M_j\left\|\bar{v}_j(s-\sigma_{ji}(s))\right\|_2\}\mathrm{d}s \\ \leq & \sup_{-\tau\leq\theta\leq 0}\left\|\bar{u}_i(\theta)\right\|_2 \\ & + \sup_{0\leq\theta\leq t}\int_0^\theta e^{\lambda\theta-a_i^*(\theta-s)}\sum_{j=1}^m |b_{ji}|\, Q_j\left\|\bar{v}_j(s)\right\|_2\mathrm{d}s \\ & + \sup_{0\leq\theta\leq t}\int_0^\theta e^{\lambda\theta-a_i^*(\theta-s)}\sum_{j=1}^m \left|\bar{b}_{ji}\right| M_j\left\|\bar{v}_j(s-\sigma_{ji}(s))\right\|_2\mathrm{d}s\end{aligned}$$

Compute that

$$\begin{aligned}
&\sup_{0\le\theta\le t}\int_0^\theta e^{\lambda\theta-a_i^*(\theta-s)}\sum_{j=1}^m |b_{ji}|\,Q_j\,\|\bar v_j(s)\|_2\,\mathrm{d}s\\
&=\sup_{0\le\theta\le t}\int_0^\theta e^{(\lambda-a_i^*)(\theta-s)}\sum_{j=1}^m |b_{ji}|\,Q_j\,\|\bar v_j(s)\|_2\,e^{\lambda s}\mathrm{d}s\\
&\le\sup_{0\le\theta\le t}\{\int_0^\theta e^{(\lambda-a_i^*)(\theta-s)}\mathrm{d}s\sum_{j=1}^m |b_{ji}|\,Q_j\sup_{0\le s\le\theta}(\|\bar v_j(s)\|_2\,e^{\lambda s})\}\\
&\le(a_i^*-\lambda)^{-1}\sum_{j=1}^m |b_{ji}|\,Q_j\sup_{0\le\theta\le t}(\|\bar v_j(\theta)\|_2\,e^{\lambda\theta})\\
&\le(a_i^*-\lambda)^{-1}\sum_{j=1}^m |b_{ji}|\,Q_j\sup_{-\sigma\le\theta\le t}(\|\bar v_j(\theta)\|_2\,e^{\lambda\theta})
\end{aligned}$$

and

$$\begin{aligned}
&\sup_{0\le\theta\le t}\int_0^\theta e^{\lambda\theta-a_i^*(\theta-s)}\sum_{j=1}^m \left|\bar b_{ji}\right|M_j\,\|\bar v_j(s-\sigma_{ji}(s))\|_2\,\mathrm{d}s\\
&=\sup_{0\le\theta\le t}\int_0^\theta e^{(\lambda-a_i^*)(\theta-s)}\sum_{j=1}^m \left|\bar b_{ji}\right|M_j e^{\lambda\sigma_{ji}(s)}\,\|\bar v_j(s-\sigma_{ji}(s))\|_2\,e^{\lambda(s-\sigma_{ji}(s))}\mathrm{d}s\\
&\le(a_i^*-\lambda)^{-1}\sum_{j=1}^m \left|\bar b_{ji}\right|M_j e^{\lambda\sigma}\sup_{-\sigma\le\theta\le t}(\|\bar v_j(\theta)\|_2\,e^{\lambda\theta})
\end{aligned}$$

Hence we have

$$\begin{aligned}
\sup_{0\le\theta\le t}(\|\bar u_i(\theta)\|_2\,e^{\lambda\theta})\le&\sup_{-\tau\le\theta\le 0}\|\bar u_i(\theta)\|_2+(a_i^*-\lambda)^{-1}\sum_{j=1}^m(|b_{ji}|\,Q_j\\
&+\left|\bar b_{ji}\right|M_j e^{\lambda\sigma})\sup_{-\sigma\le\theta\le t}(\|\bar v_j(\theta)\|_2\,e^{\lambda\theta})
\end{aligned}$$

Similarly, we can obtain

$$\begin{aligned}
\sup_{0\le\theta\le t}(\|\bar v_j(\theta)\|_2\,e^{\lambda\theta})\le&\sup_{-\sigma\le\theta\le 0}\|\bar v_j(\theta)\|_2+(c_j^*-\lambda)^{-1}\sum_{i=1}^n(|d_{ij}|\,P_i\\
&+\left|\bar d_{ij}\right|L_i e^{\lambda\tau})\sup_{-\tau\le\theta\le t}(\|\bar u_i(\theta)\|_2\,e^{\lambda\theta})
\end{aligned}$$

Thus, we have

$$\begin{aligned}
\sup_{-\tau\le\theta\le t}(\|\bar u_i(\theta)\|_2\,e^{\lambda\theta})\le 2&\sup_{-\tau\le\theta\le 0}\|\bar u_i(\theta)\|_2+(a_i^*-\lambda)^{-1}\sum_{j=1}^m(|b_{ji}|\,Q_j\\
&+\left|\bar b_{ji}\right|M_j e^{\lambda\sigma})\sup_{-\sigma\le\theta\le t}(\|\bar v_j(\theta)\|_2\,e^{\lambda\theta})
\end{aligned}$$

$$\sup_{-\sigma\le\theta\le t}(\|\bar{v}_j(\theta)\|_2 e^{\lambda\theta}) \le 2\sup_{-\sigma\le\theta\le 0}\|\bar{v}_j(\theta)\|_2 + (c_j^*-\lambda)^{-1}\sum_{i=1}^{n}(|d_{ij}|P_i + |\bar{d}_{ij}|L_i e^{\lambda\tau})\sup_{-\tau\le\theta\le t}(\|\bar{u}_i(\theta)\|_2 e^{\lambda\theta})$$

Let set

$$U_i(t) = \sup_{-\tau\le\theta\le t}(\|\bar{u}_i(\theta)\|_2 e^{\lambda\theta}), \quad U_{n+j}(t) = \sup_{-\sigma\le\theta\le t}(\|\bar{v}_j(\theta)\|_2 e^{\lambda\theta}),$$
$$U(t) = (U_1(t),\cdots,U_n(t),U_{n+1}(t),U_{n+m}(t))^T.$$

Then we have

$$U(t) = 2U(0) + \begin{pmatrix} A-\lambda I & 0 \\ 0 & C-\lambda I \end{pmatrix}^{-1}\begin{pmatrix} 0 & B(\lambda) \\ D(\lambda) & 0 \end{pmatrix}U(t)$$

and

$$U(t) = 2\left(I - \begin{pmatrix} A-\lambda I & 0 \\ 0 & C-\lambda I \end{pmatrix}^{-1}\begin{pmatrix} 0 & B(\lambda) \\ D(\lambda) & 0 \end{pmatrix}\right)^{-1}U(0)$$

We further obtain

$$\begin{aligned}\|u(t;\xi)-u^*\| + \|v(t;\eta)-v^*\| &\le e^{-\lambda t}\sum_{k=1}^{m+n}U_k(t)\\ &\le 2e^{-\lambda t}\left(\sum_{l=1}^{m+n}\sum_{k=1}^{m+n}E_{lk}(\lambda)\right)\left(\sum_{k=1}^{m+n}U_k(0)\right)\\ &\le Me^{-\lambda t}(\|\xi-u^*\| + \|\eta-v^*\|)\end{aligned}$$

where $M>1$ is some constant. From the above inequality, we can conclude that the equilibrium point of system (1) is unique. The proof is completed.

Remark. Our methods only require the time-varying delays $\tau_{ij}(t), \sigma_{ji}(t)(1 \le i \le n, 1 \le j \le m)$ are nonnegative and bounded. However, in some recent literatures, the time delays $\tau_{ij}(t), \sigma_{ji}(t)$ are often supposed to be differentiable and their derivatives $\dot{\tau}_{ij}(t), \dot{\eta}_{ji}(t)$ be bounded by 1 in order to construct suitable Lyapunov function.

For system (2), we have the following result

Corollary 1. If system (2) satisfies assumptions (A_1) and

(A_3)
$$\rho\left(\begin{pmatrix} A & 0 \\ 0 & C \end{pmatrix}^{-1}\begin{pmatrix} 0 & B(0) \\ D(0) & 0 \end{pmatrix}\right) < 1,$$

where $A = diag\{a_1,\cdots,a_n\}$, $C = diag\{c_1,\cdots,c_m\}$, $B(z) = (B_{ij}(z))_{n\times m}$, $B_{ij}(z) = |b_{ji}|Q_j + |\bar{b}_{ji}|M_j e^{z\sigma},, D(z) = (D_{ji}(z))_{m\times n}$, $D_{ji}(z) = |d_{ij}|P_i + |\bar{d}_{ij}|L_i e^{z\tau}$. Then system (2) has a unique equilibrium point $(u^*,v^*)^T$, which is globally exponentially stable.

4 Illustrative Example

Consider the following reaction-diffusion delayed BAM neural networks with Dirichlet boundary conditions:

$$\begin{aligned}
\frac{\partial u(t,x)}{\partial t} &= \sum_{k=1}^{2} \frac{\partial}{\partial x_k}\big(D_k \frac{\partial u}{\partial x_k}\big) - \frac{1}{2}u(t,x) + \frac{1}{4}\sin\left(v(t,x)\right) \\
&\quad + \frac{1}{4}\tanh\left(v(t-\sigma(t),x)\right) + I, \\
\frac{\partial v(t,x)}{\partial t} &= \sum_{k=1}^{2} \frac{\partial}{\partial x_k}\big(\overline{D}_k \frac{\partial v}{\partial x_k}\big) - \frac{1}{3}v(t,x) + \frac{1}{8}\cos\left(u(t,x)\right) \\
&\quad + \frac{1}{8}\tanh\left(u(t-\tau(t),x)\right) + J, \\
u(t,x) &= 0, \quad v(t,x) = 0, \quad x \in \partial\Omega, \quad -\tau \le t < +\infty, \quad -\sigma \le t < +\infty, \\
u(s,x) &= \xi(s,x), \quad -\tau \le s \le 0, \quad 0 \le \tau(t) \le \tau, \\
v(s,x) &= \eta(s,x), \quad -\sigma \le s \le 0, \quad 0 \le \sigma(t) \le \sigma,
\end{aligned} \tag{5}$$

where $D_1 = \frac{1}{4}$, $D_2 = \frac{1}{4}$, $\overline{D}_1 = \frac{1}{3}$, $\overline{D}_2 = \frac{1}{3}$, $\Omega = \{x \mid |x_k| < 1, k = 1, 2\}$ is a bounded compact set. It is clear that $P = Q = L = M = 1$, $A = 1$, $C = 1$, $B(0) = \frac{1}{2}$, $D(0) = \frac{1}{4}$.

Compute that $\rho\left(\begin{pmatrix} A & 0 \\ 0 & C \end{pmatrix}^{-1} \begin{pmatrix} 0 & B(0) \\ D(0) & 0 \end{pmatrix}\right) = \frac{1}{2\sqrt{2}} < 1$. Hence, from Theorem 1, we know that the equilibrium point $(u^*, v^*)^T$ of system (5) is unique and globally exponentially stable.

5 Conclusion

In this paper, some novel sufficient conditions have been presented for the global exponential stability of reaction-diffusion delayed BAM neural networks with Dirichlet boundary conditions by employing the method of variation parameter and inequality technique. The conditions possess highly important significance in some applied fields, and can be easily checked in practice by simple algebraic methods. These play an important role in design and applications of BAM neural networks.

Acknowledgements. The work is supported by Key Science Foundation of Educational Department of Hubei Province under Grant D20082201 and Innovation Teams of Hubei Normal University.

References

1. Kosto, B.: Bi-Directional Associative Memories. IEEE Trans. Syst. Man Cybernet 18, 49–60 (1988)

2. Liao, X., Liu, G., Yu, J.: Neural Networks of Bi-Directional Associative Memory with Axonal Signal Transmission Delays. J. Electron. 19, 439–444 (1997)
3. Liao, X., Yu, J.: Qualitative Analysis of Bi-Directional Associative Memory with Time-Varying Delay. J. Circ. Theory Appl. 26, 219–229 (1998)
4. Cao, J., Wang, L.: Periodic Oscillatory Solution of Bidirectional Associative Memory Networks with Delays. Phys. Rev. E 61, 1825–1828 (2000)
5. Wang, B., Jian, J., Guo, C.: Global Exponential Stability of a Class of BAM Networks with Time-Varying Delays and Continuously Distributed Delays. Neurocomputing 71, 495–501 (2008)
6. Li, K.: Delay-Dependent Stability Analysis for Impulsive BAM Neural Networks with Time-Varying Delays. Computers and Mathematics with Applications 56, 2088–2099 (2008)
7. Huang, Z., Xia, Y.: Global Exponential Stability of BAM Neural Networks with Transmission Delays and Nonlinear Impulses. Chaos, Solitons and Fractals 38, 489–498 (2008)
8. Li, Y., Wang, J.: An Analysis on the Global Exponential Stability and the Existence of Periodic Solutions for Non-Autonomous Hybrid BAM Neural Networks with Distributed Delays and Impulses. Computers and Mathematics with Applications 56, 2256–2267 (2008)
9. Zhou, Q.: Global Exponential Stability of BAM Neural Networks with Distributed Delays and Impulses. Nonlinear Analysis 10, 144–153 (2009)
10. Chen, A., Du, D.: Global Exponential Stability of Delayed BAM Network on Time Scale. Neurocomputing 71, 3582–3588 (2008)
11. Wang, L., Zhou, Q.: Global Exponential Stability of BAM Neural Networks with Time-Varying Delays and Reaction-Diffusion Terms. Phys. Lett. A 371, 83–89 (2007)
12. Lou, X., Cui, B., Wu, W.: On Global Exponential Stability and Existence of Periodic Solutions for BAM Neural Networks with Distributed Delays and Reaction-Diffusion Terms. Chaos, Solitons and Fractals 36, 1044–1054 (2008)
13. Wang, L., Xu, D.: Global Exponential Stability of Hopfield Reaction-Diffusion Neural Networks with Time-Varying Delays. Science in China(Series F) 46, 466–474 (2003)
14. Wang, L., Zhang, Z., Wang, Y.: Stochastic Exponential Stability of the Delayed Reaction-Diffusion Recurrent Neural Networks with Markovian Jumping Parameter. Physics Letter A 372, 3201–3209 (2008)
15. Kao, Y., Gao, C., Wang, D.: Global Exponential Stability of Reaction-Diffusion Hopfield Neural Networks with Continuously Distributed Delays. Mathematica Application 21, 457–462 (2008)
16. Wang, J., Lu, J.: Global Exponential Stability of Fuzzy Celluar Neural Networks with Delays and Reaction-Diffusion Terms. Chaos, Solitons and Fractals 38, 878–885 (2008)
17. Yang, Z., Xu, D.: Global Dynamics for Non-Autonomous Reaction-Diffusion Neural Networks with Time-Varying Delays. Theoretical Computer Science 403, 3–10 (2008)
18. Lu, J.: Global Exponential Stability and Periodic Solutions of Reaction-Diffusion Delayed Recurrent Neural Networks with Dirichlet Boundary Conditions. Chaos, Solitons and Fractals 35, 116–125 (2008)

Global Robustly Asymptotically Stability of Cohen-Grossberg Neural Networks with Nonnegative Amplification Function

Yongsu Kim, Huaguang Zhang, Lili Cui, and Xin Zhang

Information Science and Engineering, Northeastern University, Shenyang 110004, China
Kim_ys@yahoo.cn, hgzhang@ieee.org, cuilili8396@163.com,
jackie_zx@yahoo.com.cn

Abstract. The global robust asymptotic stability problem of Cohen-Grossberg neural networks with nonnegative amplification function is considered in this paper. The amplification function condition is assumed to be nonnegative. In the terms of linear matrix inequalities (LMIs), sufficient conditions are obtained by using Lyapunov- Krasovskii method which guarantee the existence and global robustly asymptotic stability of the equilibrium point of the Cohen-Grossberg neural networks with nonnegative amplification function. Finally, a numerical example is provided to verify the effectiveness of the proposed results.

Keywords: Cohen-Grossberg neural networks, Nonlinear complementary problem(NCP), Robust asymptotic stability, Linear matrix inequalities (LMIs).

1 Introduction

Cohen-Grossberg neural networks, proposed by Cohen and Grossberg in 1983[1], have been applied to various signal processing problems such as optimization, image processing, and associative memory design[2]. Recently, much attention is being focused on the analysis of stability of Cohen-Grossberg neural networks.[3]-[7]. In [8] and [9], the methods to analyze the global stability by the proper Lyapunov function are proposed. Problems of exponential stability of neural networks with multiple time delays were considered in [10]. In electronic implementation of neural networks, there also exist inevitably some uncertainties due to the existence of modeling errors and parameter fluctuations, which lead to complex dynamical behaviors. This fact paves the way for introducing the theory of interval matrices and interval dynamics to investigate the global stability of interval neural networks. In the design of neural networks, it is important to ensure that system be stable with respect to these uncertainties. In [11], [6]and [7], the robust stability of interval neural networks with time delay is investigated. In most of papers, the stability of Cohen-Grossberg neural networks are considered under the positive amplification function condition. To the best of our knowledge, there are few results about the stability problem of Cohen-Grossberg neural networks with nonnegative amplification function[12], especially considering uncertainties.

In this paper, not only nonnegative amplification function but also uncertainties are taken into consideration. The existence of unique nonnegative equilibrium of Cohen-Grossberg neural networks with nonnegative amplification function and the sufficient

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 313–322, 2009.

condition for its global robustly asymptotically stability criteria are obtained. Finally, two numerical examples are given to show the effectiveness of the obtained results.

2 Preliminaries

Consider the following Cohen–Grossberg neural networks with uncertainties:

$$\begin{aligned}\frac{dx_i(t)}{dt} =& a_i(x_i(t))[-d_i(x_i(t)) + \sum_{j=1}^{n} \left(w_{ij}^0 + \Delta w_{ij}^0\right) g_j(x_j(t)) \\ &+ \sum_{j=1}^{n} (w_{ij} + \Delta w_{ij})\, g_j(x_j(t-\tau)) + I_i], \quad i = 1, 2, ..., n \end{aligned} \tag{1}$$

where $x_i(t)$ denotes the state variable of the ith neuron at time t; $a_i(x_i(t))$ represents an amplification function at time t; $d_i(x_i(t))$ is an appropriately the behaved function at time t; g_i denotes the activation function; I_i is the external constant input; $W^0 + \Delta W^0 = (w_{ij}^0)_{n\times n} + \left(\Delta w_{ij}^0\right)_{n\times n}$ and $W + \Delta W = (w_{ij})_{n\times n} + (\Delta w_{ij})_{n\times n}$ denote the connection weight matrix and delayed connection weight matrix which contain parameter uncertainties, respectively; τ is the time delay.

The network parameters are intervalized as follows:

$$\begin{aligned} &\underline{W}^0 \le W^0 + \Delta W^0 \le \overline{W}^0, \quad i.e., \underline{w}_{ij}^0 \le w_{ij}^0 + \Delta w_{ij}^0 \le \overline{w}_{ij}^0, \quad i, j = 1, 2, ..., n \\ &\underline{W} \le W + \Delta W \le \overline{W}, \quad i.e., \underline{w}_{ij} \le w_{ij}^{+} \Delta w_{ij} \le \overline{w}_{ij}, \quad i, j = 1, 2, .., n \end{aligned} \tag{2}$$

let

$$\begin{aligned} &x(t) = (x_1(t), x_2(t), ..., x_n\ (t)\ )^T, \quad d(x) = (d_1(x_1(t)), d_2(x_2(t)), ..., d_n(x_n(t)))^T \\ &g(x) = (g_1(x_1(t)), g_2(x_2(t)), ..., g_n(x_n(t)))^T, I = (I_1, I_2, ..., I_n)^T \end{aligned}$$

then the system (1) can be rewritten as

$$\frac{dx(t)}{dt} = a(x)\left[-d(x) + (W^0 + \Delta W^0)g(x(t)) + (W + \Delta W)g(x(t-\tau)) + I\right]. \tag{3}$$

Throughout the paper, we make the following assumptions for Cohen–Grossberg neural networks (3).

Assumption 1. *For the amplification function, we have the following assumptions.*

1) $a(x) \in A_1$*:* $a(x)$ *is continuous and* $a_i(0) = 0$, $a_i(x) \ge 0 \quad (x > 0)$;

2) $a(x) \in A_2$*:* $a(x) \in A_1$ *and for any* $\varepsilon > 0$, $\int_0^{\varepsilon} \frac{dx}{a_i(x)} = +\infty \quad (i = 1, 2, ...n)$;

3) $a(x) \in A_3$*:* $a(x) \in A_1$ *and for any* $\varepsilon > 0$, $\int_{\varepsilon}^{\infty} \frac{dx}{a_i(x)} = +\infty \quad (i = 1, 2, ...n)$;

4) $a(x) \in A_4$*:* $a(x) \in A_1$ *and for any* $\varepsilon > 0$, $\int_0^{\varepsilon} \frac{xdx}{a_i(x)} < +\infty \quad (i = 1, 2, ...n)$.

Assumption 2. *There exists a positive diagonal matrix* $D = diag(D_1, D_2, ..., D_n)$ *such that* $0 < D_i \le \frac{d_i(\xi) - d_i(\zeta)}{\xi - \zeta}, \xi \ne \zeta, i = 1, 2, ..., n$.

Assumption 3. *For each activation function $g_i(i = 1, 2, \ldots, n)$, there exist positive finite constants G_i such that $0 \leq \frac{g_i(\xi)-g_i(\zeta)}{\xi-\zeta} \leq G_i$ for all $\xi, \zeta \in R$, $\xi \neq \zeta$ and $i = 1, 2, ..., n$.*

Assumption 4. *The uncertainties in the connection weight matrices are assumed to satisfy the following assumption:*

$$\left[\Delta W^0 \ \Delta W\right] = HF\left[A \ B\right]$$

where F is an unknown matrix representing parametric uncertainty which satisfies $F^TF \leq I$ and A, B and H are known real constant matrices with appropriate dimensions.

Notations: $R_+^n = \left\{x = (x_1, x_2, ..., x_n)^T : x_i \geq 0, i = 1, 2...., n\right\}$ denotes the first orthant, $\lambda_m(D)$ and $\lambda_M(D)$ denote the minimum eigenvalue and the maximum eigenvalue of a square matrix D.

If $a(x) \in A_1$, then any equilibrium in R_+^n of system (3) is a solution of the following equations:

$$x_i\left[f_i(x) - I_i\right] = 0 \quad i = 1, 2, ..., n \tag{4}$$

$$f_i(x) = d_i(x_i) - \sum_{j=1}^{n}(w_{ij}^0 + \Delta w_{ij}^0 + w_{ij} + \Delta w_{ij})g_j(x_j), i = 1, 2, .., n \tag{5}$$

Definition 1. *(nonlinear complementary problem(NCP)) A NCP is to find x^*, $i = 1, 2, ...n$ satisfying*

$$x_i^* \geq 0, \quad f_i(x^*) - I_i \geq 0$$
$$x_i^*(f_i(x^*) - I_i) = 0, i = 1, 2, ...n$$

where $f = (f_1(x), f_2(x), ..., f_n(x))^T : R_+^n \to R^n$ is continuous. $I_i \in R, i = 1, 2, ..., n$.

Definition 2. *If x^* is the solution of the NCP when $f(x)$ is expressed as equation (5) , then $x^*$is said to be a nonnegative equilibrium of the system (3) in the NCP sense . Especially if $x^* > 0$, then $x^*$is said to be a positive equilibrium of the system (3).*

Definition 3. *The neural network defined by (1) or (3) with the parameter ranges defined by (2) is R_+^n- globally asymptotically robust stable if the unique equilibrium point $x^* = (x_1^*, x_2^*, ..., x_n^*)^T$of the neural system is globally asymptotically stable for any positive initial condition $\phi_i(t) > 0$ holds for all $t \in [-\tau, 0]$ and $i = 1, 2, ..., n$.*

Lemma 1. [12] *The NCP (10) has a unique solution for every $I \in R^n$ if and only if $F(x)$ is norm-coercive, i.e. $\lim_{\|x\|\to\infty} \|F(x)\| = \infty$ and locally univalent, where $F(x) : R^n \to R^n$ is defined as follows:*

$$F(x) = f(x^+) + x^-, \ x^+ = (x_1^+, x_2^+, ..., x_n^+)^T, \ x^- = (x_1^-, x_2^-, ..., x_n^-)^T$$
$$x_i^+ = \begin{cases} x_i & x_i \geq 0 \\ 0 & x_i < 0 \end{cases}, \qquad x_i^- = \begin{cases} x_i & x_i \leq 0 \\ 0 & x_i > 0 \end{cases}, \qquad i = 1, 2, ...n.$$

Lemma 2. [12] *For given positive diagonal matrices $D \in R^{n\times n}$ and $G \in R^{n\times n}$ and given matrices $A \in R^{n\times n}$ and $B \in R^{n\times n}$, if there exist a positive definite diagonal matrix $P \in R^{n\times n}$ and a positive definite symmetric matrix $Q \in R^{n\times n}$such that*

$$\begin{bmatrix} 2PDG^{-1} - PA - A^TP - Q & -PB \\ -B^TP & Q \end{bmatrix} > 0$$

holds, then there exists a constant $\beta > 0$ such that

$$\begin{bmatrix} 2\beta D & -\beta A & -\beta B \\ -\beta A^T & 2PDG^{-1} - PA - A^TP - Q & -PB \\ -\beta B^T & -B^TP & Q \end{bmatrix} > 0$$

Lemma 3. [13] *Let U, V, W and M be real matrices of appropriate dimensions with M satisfying $M = M^T$, then we have*

$$M + UVW + W^TV^TU^T < 0$$

for all $VV^T \le I$, if and only if there exists a scalar $\varepsilon > 0$ such that

$$M + \varepsilon^{-1}UU^T + \varepsilon W^TW < 0$$

3 Main Results

Theorem 1. *Under assumptions 1, 2 and 3, if there exists a positive definite diagonal matrix $P = diag(p_1, p_2, ..., p_n)$ such that*

$$\Omega = \{\ rI - (\overline{S}1 + \overline{S}2)\ \} > 0 \tag{6}$$

holds, then there exists a unique nonnegative equilibrium of system (3), for any $I \in R^n$. where

$$\begin{aligned} r =& \min(\frac{p_i}{\underline{D}\overline{G}}), \quad \underline{D} = \min(D_1, D_2, ..., D_n), \quad \overline{G} = \max(G_1, G_2, ..., G_n), \\ \overline{S}1 =& (\bar{s}1_{ij})_{n\times n}, \quad \bar{s}1_{ii} = p_i\bar{w}^0_{ii}, \quad \bar{s}1_{ij} = \max(\left|p_i\underline{w}^0_{ij}\right|, \left|p_i\bar{w}^0_{ij}\right|), (i \neq j) \\ \overline{S}2 =& (\bar{s}2_{ij})_{n\times n}, \quad \bar{s}2_{ii} = p_i\bar{w}_{ii}, \quad \bar{s}2_{ij} = \max(\left|p_i\underline{w}_{ij}\right|, \left|p_i\bar{w}_{ij}\right|), (i \neq j). \end{aligned}$$

Proof. According to (5) and Lemma 1, if $F(x)$ is locally univalent and norm-coercive, then for every $I \in R^n$, there exists a unique equilibrium of the system (3). At first, we prove $F(x)$ is locally univalent.

For any $x = (x_1, x_2, ..., x_n)^T \in R^n$, without loss of generality, by some rearrangement of x_i, we can assume $x_i > 0$, if $i = 1, 2, \ldots, p$; $x_i < 0$, if $i = p+1, p+2, ..., m$; $x_i = 0$, if $i = m+1, m+2, ..., n$, for integers $p \le m \le n$. Moreover, if $y \in R^n$is sufficiently close to $x \in R^n$, without loss of generality, we can also assume $y_i > 0$ if $i = 1, 2, ..., p$, $y_i < 0$ if $i = p+1, p+2, ..., m$, $y_i > 0$ if $i = m+1, m+2, ..., m_1$,

$y_i < 0$ if $i = m_1 + 1, m_1 + 2, ..., m_2$, $y_i = 0$ if $i = m_2 + 1, m_2 + 2, ..., n$, for some integers $p \le m \le m_1 \le m_2 \le n$. It can be seen that

$$(x_i^+ - y_i^+)(x_i^- - y_i^-) = 0, \quad i = 1, 2, ..., n \tag{7}$$

$$\begin{aligned} F(x) - F(y) = {} & d(x^+) - d(y^+) - (W^0 + \Delta W^0)(g(x^+) - g(y^+)) \\ & - (W + \Delta W)(g(x^+) - g(y^+)) + (x^- - y^-) \end{aligned} \tag{8}$$

For $x \neq y$, if $x^+ = y^+$, then $x^- \neq y^-$. Therefore $F(x) \neq F(y)$.
If $x^+ \neq y^+$, then there exit two cases.
Case 1: $x^+ \neq y^+$ and $g(x^+) = g(y^+)$.

In this case, $F(x) - F(y) = d(x^+) - d(y^+) + (x^- - y^-)$. According to assumption 2, we can know if $x^+ \neq y^+$, then $d(x^+) \neq d(y^+)$. By equation (8), $F(x) - F(y) = d(x^+) - d(y^+) \neq 0$, so we have $F(x) \neq F(y)$.
Case 2: $x^+ \neq y^+$and $g(x^+) \neq g(y^+)$.

Multiplying both sides of (8) by $(g(x^+) - g(y^+))^T P$ results in

$$\begin{aligned} (g(x^+) - g(y^+))^T P(F(x) - F(y)) = {} & (g(x^+) - g(y^+))^T P(d(x^+) - d(y^+)) \\ & - (g(x^+) - g(y^+))^T P\tilde{W}^0(g(x^+) - g(y^+)) - (g(x^+) - g(y^+))^T P\tilde{W}(g(x^+) \\ & - g(y^+)) + (g(x^+) - g(y^+))^T P(x^- - y^-) \end{aligned} \tag{9}$$

where $\tilde{W}^0 = (\tilde{w}_{ij}^0)_{n\times n} = W^0 + \Delta W^0$, $\tilde{W} = (\tilde{w}_{ij})_{n\times n} = W + \Delta W$, $P = diag(p_i > 0)$. According to Assumption 2,

$$\begin{aligned} (g(x^+) - g(y^+))^T P(d(x^+) - d(y^+)) &= \sum_{i=1}^{n} p_i((g_i(x^+) - g_i(y^+))(d_i(x^+) - d_i(y^+)) \\ &\ge \sum_{i=1}^{n} \frac{p_i}{\underline{D}}((g_i(x^+) - g_i(y^+))(x^+ - y^+) \ge \sum_{i=1}^{n} \frac{p_i}{\underline{D}\overline{G}}(g_i(x^+) - g_i(y^+))^2 \\ &\ge r(g(x^+) - g(y^+))^2 \end{aligned} \tag{10}$$

$$\begin{aligned} (g(x^+) - g(y^+))^T P\tilde{W}^0(g(x^+) - g(y^+)) &= \sum_{i=1}^{n} p_i\tilde{w}_{ii}^0(g_i(x^+) - g_i(y^+))^2 \\ &\quad + \sum_{i=1}^{n}\sum_{j=1, j\neq i}^{n} (p_i\tilde{w}_{ij}^0(g_i(x^+) - g_i(y^+))(g_j(x^+) - g_j(y^+)) \\ &\le \sum_{i=1}^{n} p_i\bar{w}_{ii}^0(g_i(x^+) - g_i(y^+))^2 \end{aligned}$$

$$+\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n}\left|p_i\bar{w}_{ij}^{0}\right|\left|(g_i(x^+)-g_i(y^+))\right|\left|(g_j(x^+)-g_j(y^+)\right|$$

$$\leq\sum_{i=1}^{n}\bar{s}1_{ii}(g_i(x^+)-g_i(y^+))^2$$

$$+\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n}\bar{s}1_{ij}\left|(g_i(x^+)-g_i(y^+))\right|\left|(g_j(x^+)-g_j(y^+)\right|$$

$$=\left|(g_i(x^+)-g_i(y^+))^T\right|\bar{S}1\left|(g_i(x^+)-g_i(y^+)\right| \tag{11}$$

Similarly,

$$(g(x^+)-g(y^+))^T P\tilde{W}(g(x^+)-g(y^+))\leq$$
$$\left|(g_i(x^+)-g_i(y^+))^T\right|\overline{S}2\left|(g_i(x^+)-g_i(y^+))\right| \tag{12}$$

by (7), if $x^+\neq y^+$, then $(x^- - y^-)=0$

$$(g(x^+)-g(y^+))^T P(x^- - y^-)=0 \tag{13}$$

From (9)$\sim$(13), we have

$$(g(x^+)-g(y^+))^T P(F(x)-F(y))\geq\left|(g(x^+)-g(y^+))^T\right|\Omega\left|(d(x^+)-d(y^+))\right| \tag{14}$$

If $\Omega>0$, then $(g(x^+)-g(y^+))^T P(F(x)-F(y))>0$. $g(x^+)\neq g(y^+)$ implies that $F(x)\neq F(y)$, it is proved that $F(x)$ is locally univalent.

In (14), let $y=0$, we have

$$(g(x^+)-g(0^+))^T P(F(x)-F(0))\geq\left|(g(x^+)-g(0^+))^T\right|\Omega\left|(d(x^+)-d(0^+))\right|$$
$$\left|(g(x^+)-g(0^+))^T P(F(x)-F(0))\right|\geq\left|(g(x^+)-g(0^+))^T\right|\Omega\left|(d(x^+)-d(0^+))\right|$$
$$\|P\|_2\left\|(g(x^+)-g(0))\right\|_2\|(F(x)-F(0))\|_1\geq\lambda_m(\Omega)\left\|(d(x^+)-d(0))\right\|_2^2$$
$$\left\|(g(x^+)-g(0))\right\|_2\geq\left\|g(x^+)\right\|_2-\|g(0)\|_2,\ \|(F(x)-F(0))\|_1\leq\|F(x)\|_1+\|F(0)\|_1$$
$$\|F(x)\|_1\geq\|P\|_2^{-1}(\lambda_m(\Omega)\left\|g(x^+)\right\|_2)-\|P\|_2^{-1}(\lambda_m(\Omega)\|g(0)\|_2)-\|F(0)\|_1$$

Since $\|P\|_2$, $\|g(0)\|_2$and $\|F(0)\|_2$ are finite, we have

$$\lim_{\|x\|\to\infty}\|F(x)\|=\infty \tag{15}$$

$F(x)$ is norm-coercive.

Let x^* be the nonnegative equilibrium point of system (3) and $\tilde{x}=x-x^*$, then the system (1) can be transformed as

$$\frac{d\tilde{x}_i(t)}{dt}=\tilde{a}_i(\tilde{x}_i(t))\left[-\tilde{d}_i(\tilde{x}_i(t))+\sum_{j=1}^{n}\left(w_{ij}^{0}+\Delta w_{ij}^{0}\right)\tilde{g}_j(\tilde{x}_j(t))\right.$$
$$\left.+\sum_{j=1}^{n}(w_{ij}+\Delta w_{ij})\,\tilde{g}_j(\tilde{x}_j(t-\tau))+J_i\right] \tag{16}$$

where

$$\tilde{a}_i(\tilde{x}_i(t))=a_i(\tilde{x}_i+x_i^*),\tilde{d}_i(\tilde{x}_i(t))=d_i(\tilde{x}_i+x_i^*)-d_i(x_i^*),$$
$$\tilde{g}_i(\tilde{x}_i(t))=g_i(\tilde{x}_i+x_i^*)-g_i(x_i^*)$$
$$J_i=\begin{cases}-\tilde{d}_i(x_i^*)+\sum\limits_{j=1}^{n}\left(w_{ij}^0+\Delta w_{ij}^0+w_{ij}+\Delta w_{ij}\right)\tilde{g}_j(x_j^*)+I_i & x_i^*=0\\ 0 & x_i^*>0\end{cases}$$

Since x^* is the nonnegative equilibrium point of system (3) , thus from (7) we know that $J_i \le 0$ holds for all $i=1,2,...,n$, which implies that $\tilde{g}_i(\tilde{x}_i(t))J_i \le 0$ holds for all $i=1,2,...,n$ and $t \ge 0$.

Theorem 2. *Under assumptions 1, 2 ,3 and 4, if there exist a positive definite diagonal matrix P, a positive definite symmetric matrix Q and a scalar $\varepsilon > 0$ such that*

$$\begin{bmatrix}\Psi_{11} & PW+\varepsilon A^T B & PH\\ * & \Psi_{22} & 0\\ * & * & \Psi_{33}\end{bmatrix}<0 \tag{17}$$

holds, then the unique nonnegative equilibrium $x^*$of the system (3) is R_+^n- globally robustly asymptotically stable. where

$$\Psi_{11}=-2PDG^{-1}+PW^{0^T}+W^0P+\varepsilon A^TA,\qquad \Psi_{22}=-Q+\varepsilon B^TB$$
$$\Psi_{33}=-\varepsilon I,\qquad D=diag(D_1,D_2,...,D_n),\qquad G=diag(G_1,G_2,...,G_n)$$

Proof. Construct the following Lyapunov-Krasovskii functional:

$$V(\tilde{x}(t))=2\beta\sum_{i=1}^{n}\int_0^{\tilde{x}(t)}\frac{s}{\tilde{a}_i(s)}ds+2\sum_{i=1}^{n}p_i\int_0^{\tilde{x}(t)}\frac{\tilde{g}_i(s)}{\tilde{a}_i(s)}ds+\int_{t-\tau}^{t}\tilde{g}^T(\tilde{x}(s))Q\tilde{g}(\tilde{x}(s))ds \tag{18}$$

where $P=diag(p_i)$ with $p_i>0$, $Q=Q^T>0$, and $i=1,2,\ldots,n$.

According to Assumption 1 and 3, we have

$$\int_0^{\tilde{x}(t)}\frac{s}{\tilde{a}_i(s)}ds<+\infty,\qquad \int_0^{\tilde{x}(t)}\frac{\tilde{g}_i(s)}{\tilde{a}_i(s)}ds<+\infty \tag{19}$$

Obviously, the $V(\tilde{x}(t))$ is the non-negative and radically unbounded function. Calculating the time derivatives of $V(\tilde{x}(t))$ along the trajectories of system (3) yield

$$\begin{aligned}\dot{V}(\tilde{x}(t))=&2\beta\sum_{i=1}^{n}\tilde{x}_i(t)[-\tilde{d}_i(\tilde{x}_i(t))+\sum_{j=1}^{n}(w_{ij}^0+\Delta w_{ij}^0)\tilde{g}_j(\tilde{x}_j(t))\\ &+\sum_{j=1}^{n}(w_{ij}+\Delta w_{ij})\tilde{g}_j(\tilde{x}_j(t-\tau))+J_i]+2\sum_{i=1}^{n}p_i\tilde{g}(\tilde{x}_i(t))[-\tilde{d}_i(\tilde{x}_i(t))\\ &+\sum_{j=1}^{n}(w_{ij}^0+\Delta w_{ij}^0)\tilde{g}_j(\tilde{x}_j(t))+\sum_{j=1}^{n}(w_{ij}+\Delta w_{ij})\tilde{g}_j(\tilde{x}_i(t-\tau))+J_i]\\ &+\tilde{g}^T(\tilde{x}(t))Q\tilde{g}(\tilde{x}(t))-\tilde{g}^T(\tilde{x}(t-\tau))Q\tilde{g}(\tilde{x}(t-\tau))\end{aligned} \tag{20}$$

x^* is the nonnegative equilibrium point of system (3), therefore $\tilde{g}(\tilde{x}(s))J_i \leq 0$. From (19)-(20), we have

$$\begin{aligned}\dot{V}(\tilde{x}(t)) \leq & 2\beta\,[-\,\tilde{x}^T(t)D\tilde{x}(t) + \tilde{x}^T(t)(W^0+\Delta W^0)g(\tilde{x}^T(t)) \\ & + \tilde{x}^T(t)(W+\Delta W)\tilde{g}((\tilde{x}(t-\tau))] + 2[-\tilde{g}^T(\tilde{x}(t))PDG^{-1}\tilde{g}(\tilde{x}(t)) \\ & + \tilde{g}^T(\tilde{g}(t))P(W^0+\Delta W^0)\tilde{g}(\tilde{x}(t)) + \tilde{g}^T(\tilde{x}(t))P(W+\Delta W)\tilde{g}((\tilde{x}(t-\tau))] \\ & + \tilde{g}(\tilde{x}^T(t)Q\tilde{g}(\tilde{x}(t)) - \tilde{g}(\tilde{x}^T(t-\tau)Q\tilde{g}(\tilde{x}(t-\tau)) = -\zeta^T(t)\Psi_1\zeta(t) \qquad (21)\end{aligned}$$

where

$$\begin{aligned}&\zeta(t) = [\tilde{x}^T(t)\ \tilde{g}^T(\tilde{x}(t))\ \tilde{g}^T((\tilde{x}(t-\tau))]^T, \\ &\Psi_1 = \begin{bmatrix} \psi_{11} & -\beta(W^0+\Delta W^0) & -\beta(W+\Delta W) \\ * & \varphi_{22} & -P(W+\Delta W) \\ * & * & \varphi_{33} \end{bmatrix}, \\ &\psi_{11} = 2\beta D,\ \psi_{22} = 2PDG^{-1} - P(W^0+\Delta W^0)^T - (W^0+\Delta W)P - Q,\ \psi_{33} = Q.\end{aligned}$$

Obviously, if $\Phi_1 > 0$, it implies the $\dot{V}(x) < 0$ for any $\zeta(t) \neq 0$.
According to Lemma 2, for a proper β, if

$$\Psi_2 = \begin{bmatrix} \psi_{22} & -P(W+\Delta W) \\ * & \psi_{33} \end{bmatrix} > 0 \qquad (22)$$

where ψ_{22}, ψ_{33} are same as those defined above, it implies the $\Phi_1 > 0$.

By the Assumption 4, inequality (22) can be rewritten in the following form:

$$\Xi + \begin{bmatrix} PH \\ 0 \end{bmatrix} F \begin{bmatrix} A\ B \end{bmatrix} + \begin{bmatrix} A^T \\ B^T \end{bmatrix} F^T \begin{bmatrix} H^T P\ 0 \end{bmatrix} < 0 \qquad (23)$$

where $\Xi = \begin{bmatrix} -2PDG^{-1} + PW^0 + {W^0}^T P + Q & PW \\ W^T P & -Q \end{bmatrix}$ is a definite diagonal matrix.
According to Lemma 3 , inequality (23) can be rewritten as follow

$$\Xi + \varepsilon^{-1} \begin{bmatrix} PHH^T P & 0 \\ 0 & 0 \end{bmatrix} + \varepsilon \begin{bmatrix} A^T A & A^T B \\ B^T A & B^T B \end{bmatrix} < 0 \qquad (24)$$

According to Schur complement [5], (24) is equivalent to (17). Therefore, if inequality (17) is satisfied, then $\dot{V}(\tilde{x}) < 0$ and $\dot{V}(\tilde{x}) \leq -\lambda_m(\Phi_1)\,\|\tilde{x}(t)\|_2^2$, $\lim_{t\to\infty} \|\tilde{x}(t)\|_2 = 0$. x^* is R_+^n - globally robustly asymptotically stable.

4 Numerical Example

In this section , we present a numerical example to verify the theoretical results obtained above. Consider the uncertain delayed Cohen-Grossberg neural network (3), where $g(x(t)) = [\tanh(0.2x_1), \tanh(0.4x_2)]^T$, $a_i(x_i) = x_i(t)$, $d_1(x_1) = x_1(t)$, $d_2(x_2) = 6x_2(t)$ and

$$D = \begin{bmatrix} 1 & 0 \\ 0 & 6 \end{bmatrix}, \quad G = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.4 \end{bmatrix}, \quad W^0 = \begin{bmatrix} -2 & 1 \\ -1 & -1 \end{bmatrix}, \quad W = \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix},$$
$$A = \begin{bmatrix} 0.1 & 0.4 \\ 0 & 0.5 \end{bmatrix}, \quad B = \begin{bmatrix} -0.2 & 0.4 \\ 0.6 & 0.3 \end{bmatrix}, \quad H = \begin{bmatrix} 0.2 & -0.1 \\ 0 & 0.4 \end{bmatrix}, \quad F = \begin{bmatrix} \sin t & 0 \\ 0 & \sin t \end{bmatrix}.$$

By using the Matlab LMI Toolbox, a positive definite diagonal matrix Psatisfies inequality (6) can be given as follow

$$P = \begin{bmatrix} 6.5295 & 0 \\ 0 & 6.5295 \end{bmatrix} > 0.$$

According to Theorem 1, there exists a unique nonnegative equilibrium.

A positive definite diagonal matrix P, a positive definite symmetric matrix Q and a scalar ε satisfies inequality (17) by using the Matlab LMI Toolbox can be given as follows.

$$P = \begin{bmatrix} 0.1189 & 0 \\ 0 & 0.1189 \end{bmatrix} > 0, \quad Q = \begin{bmatrix} 1.1413 & 0.0226 \\ 0.0226 & 1.6505 \end{bmatrix} > 0, \quad \varepsilon = 1.1289.$$

According to Theorem 2, the nonnegative equilibrium point of neural networks (3) is globally asymptotically stable.

When $\Delta W^0 = 0$, $\Delta W = 0$ and $I = (1, 0.1)^T$ the solution of system (3) are $(0, 0)^T$, $(0.1072, 0)^T (0.1569, 0)^T$ and $(0.1072, 0.1569)^T$. Among them, $(0.1072, 0.1569)^T$ is the unique positive equilibrium point. When the initial condition is $(0.7, 0.4)^T$, the state response curves for $\tau = 2$ and $\tau = 5$ are shown in Fig.1. Fig.2 show state response curves, which initial conditions are $(0.4, 0.5)^T$ and $(1.5, 0.3)^T$.

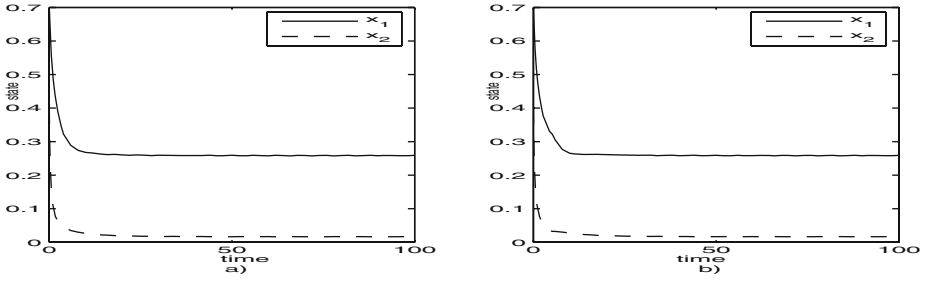


Fig. 1. State response for $\tau = 2$(a) and $\tau = 5$(b)

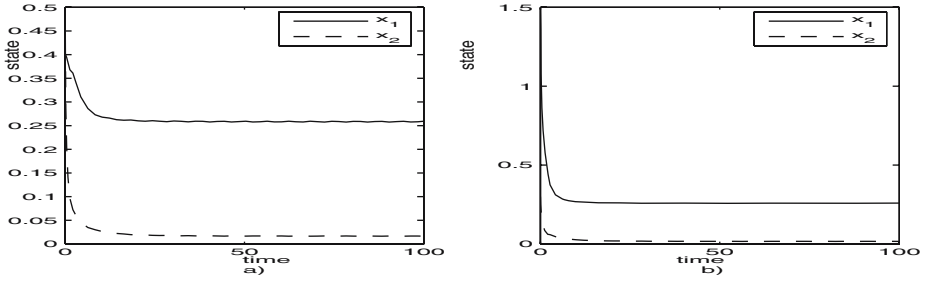


Fig. 2. State response with initial conditions $(0.4, 0.5)^T$(a) and $(1.5, 0.3)^T$(b)

5 Conclusion

This paper has the problem of global robust asymptotic stability analysis of a class of Cohen-Grossberg neural networks with nonnegative amplification function. Some sufficient conditions ensuring the existence and global robustly asymptotic stability of the equilibrium point of the Cohen-Grossberg neural networks with nonnegative amplification function have been obtained in terms of LMIs, respectively. Numerical examples verify the validity of the method proposed in the paper.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (60534010, 60572070, 60774048, and 60728307), the Program for Changjiang Scholar, the Funds for Creative Research Groups of China (Grant No. 60521003), the Research Fund for the Doctoral Program of China's Higher Education (20070145015), and the National High Technology Research and Development Program of China (2006AA04Z183).

References

1. Cohen, M.A., Grossberg, S.: Stability and Global Pattern Formulation and Memory Storage by Competitive Neural Networks. IEEE Trans Syst. Man Cybern. 13, 15–826 (1983)
2. Takahashi, Y.: Solving Optimization Problems with Variable-Constraint by an Extended Cohen-Grossberg Model. The Oretical Computer Science 158, 279–341 (1996)
3. Shao, H.: Delay-Dependent Stability for Recurrent Neural Networks with Time-Varying Delays. IEEE trans. Neural Networks 19, 1647–1651 (2008)
4. Zhang, H.G., Wang, Z.S., Liu, D.R.: Global Asymptotic Stability of Recurrent Neural Networks with Multiple Time-Varying Delays. IEEE trans Neural networks 19, 855–873 (2008)
5. Boyd, S., Ghaoui, L.E., Feron, E., Balakrishnan, V.: Linear Matrix Inequalities in System and Control Theory. SIAM, Philadelphia (1994)
6. Zhang, H.G., Wang, Z.S., Liu, D.R.: Robust Stability Analysis for Interval Cohen–Grossberg Neural Networks With Unknown Time-Varying Delays. IEEE trans Neural Networks 19, 1942–1955 (2008)
7. Guo, S.J., Huang, L.H.: Stability Analysis of Cohen-Grossberg Neural Networks. IEEE trans. Neural Networks 17, 106–117 (2006)
8. Wang, L., Zou, X.F.: Harmless Delays in Cohen-Grossberg Neural Networks. Physica D 170, 162–173 (2002)
9. Chen, T.P., Rong, L.B.: Delay Independent Stability Analysis of Cohen-Grossberg Neural Networks. Physics Letters A 317, 436–449 (2003)
10. Hwang, C.C., Cheng, C.J., Liao, T.L.: Globally Exponential Stability of Generalized Cohen-Grossberg neural networks with delays. Physics Letters A 319, 157–166 (2003)
11. Cao, J., Wang, J.: Global Asymptotic and Robust Stability of Recurrent Neural Networks with Time Delays. IEEE Trans. Circuits Syst. I 52, 417–426 (2005)
12. Lu, W.L., Chen, T.P.: R^n_+-Global Stability of a Cohen-Grossberg Neural Network System with Nonnegative Equilibria. Neural Networks 20, 714–722 (2007)
13. Singh, V.: Robust Stability of Cellular Neural Networks with Delay: Linear Matrix Inequality Approach. In: IEE Proceedings of Control Thory and Applications, vol. 151, pp. 125–129 (2004)
14. Orman, Z.: Global Stability Analysis of Cohen–Grossberg Neural Networks with Time Varying Delay. Physics Letters A 341, 410–421 (2005)
15. Ensari, T., Arik, S.: Global Stability of a Class of Neural Networks with Time Varying Delay. IEEE Trans. Circuits Syst. II 52, 126–130 (2005)

Multistability of Neural Networks with a Class of Activation Functions*

Lili Wang, Wenlian Lu, and Tianping Chen

Key Laboratory of Nonlinear Science of Chinese Ministry of Education, School of Mathematical Sciences, Fudan University, Shanghai, 200433, P.R.China
{042018047,wenlian,tchen}@fudan.edu.cn

Abstract. In this paper, we investigate the multistability of neural networks with a class of activation functions, which are nondecreasing piecewise linear with $2r(r \geq 1)$ corner points. It shows that the n-neuron neural networks can have and only have $(2r+1)^n$ equilibria under some conditions, $(r+1)^n$ of which are locally exponentially stable and others are unstable. In addition, we discuss the attraction basins of the stable equilibria for the two-dimensional case and found out that under several conditions, the stable manifolds of the unstable equilibria precisely comprise of the bounds of each attractor.

1 Introduction

Recently, the neural networks have been extensively studied due to their potential applications in classification, associative memory, parallel computation and other fields. They can be modeled by the following differential equation

$$\frac{du_i(t)}{dt} = -d_i u_i(t) + \sum_{j=1}^{n} w_{ij} f_j(u_j(t)) + I_i, \quad i = 1, \cdots, n, \tag{1}$$

where $u_i(t)$ represents the state of the i-th unit at time t; $d_i > 0$ denotes the rate with which the i-th unit will reset its potential to the resting state in isolation when disconnected from the network and external inputs; w_{ij} corresponds to the connection weight of the j-th unit on the i-th unit; $f_j(\cdot)$ is the activation function; and I_i stands for the external input.

There have been a great number of literatures concerned with the dynamical behaviors of delayed neural networks. Most of them focus on the uniqueness and global stability of the attractor, which is called as monostability, see [1] - [3]. However, it is worth noting that there may exist more than one attractors. It is called as multistability and it has many applications. For example, in pattern recognition, multiple attractors correspond to the possible patterns and the converging to a certain attractor means that the system recognizes the given pattern. For more references, see [4] - [6].

* This work was supported by Graduate Innovation Foundation of Fudan University under Grant EYH1411028, the National Natural Sciences Foundation of China under Grant No. 60774074 and 60804044, and sponsored by Shanghai Pujiang Program No. 08PJ14019.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 323–332, 2009.

In the pioneering paper [2], the authors have pointed out that the one neuron model $\frac{du(t)}{dt} = -u(t)+(1+\epsilon)g(u(t))$, where ϵ is a small positive number and $g(u) = tanh(u)$, has three equilibrium points and two of them are locally stable, one is unstable. Recent studies have generalized it to the n-neuron neural networks, and many results have been reported in the literature, see [7]-[13]. In [9], by decomposition of state space into 3^n subsets, the authors gave some conditions on the multiperiodicity of delayed cellular neural networks with saturated activation function $f(x) = \frac{|x+1|-|x-1|}{2}$, which shows that the n-neural networks can have 2^n periodic orbits located in 2^n subsets of R^n and they are attractive in the corresponding subsets, respectively. The multistability of Cohen-Grossberg neural networks with a general class of activation functions was also discussed in [10]. It showed that the n-neuron networks can have 2^n locally exponentially stable equilibrium points located in saturation regions, and some sufficient conditions are given meanwhile.

However, to the best of our knowledge, there are few papers discussing the existence of attractors or repellers in other $3^n - 2^n$ subsets. Are there any equilibrium points? Are they stable or unstable? In [11], the authors investigated the multiple stationary equilibria of n-neuron neural networks. By defining 3^n regions, it showed that the system can have 3^n equilibrium points, 2^n of them are stable and they are attractive in a class of subsets with positive invariance. Motivated by these works, we are to investigate the multistability of neural networks (1) in this paper, to address the dynamics in $3^n - 2^n$ subsets, to figure out the number of equilibria located in R^n in all, and to see the attraction basins of these stationary equilibrium points.

Consider the neural networks (1) with a class of activation functions:

$$f_j(x) = \begin{cases} m_j^1 & -\infty < x < p_j^1, \\ \frac{m_j^2 - m_j^1}{q_j^1 - p_j^1}(x - p_j^1) + m_j^1 & p_j^1 \le x \le q_j^1, \\ m_j^2 & q_j^1 < x < p_j^2, \\ \frac{m_j^3 - m_j^2}{q_j^2 - p_j^2}(x - p_j^2) + m_j^2 & p_j^2 \le x \le q_j^2, \\ m_j^3 & q_j^2 < x < p_j^3, \\ \cdots \quad \cdots & \cdots \quad \cdots \\ \cdots \quad \cdots & \cdots \quad \cdots \\ \frac{m_j^{r+1} - m_j^r}{q_j^r - p_j^r}(x - p_j^r) + m_j^r & p_j^r \le x \le q_j^r, \\ m_j^{r+1} & q_j^r < x < +\infty, \end{cases} \tag{2}$$

where $r \ge 1, \{m_j^k\}_{k=1}^{r+1}$ is an increasing constants series, p_j^k, q_j^k, $k = 1, 2, \cdots, r$ are constants with $-\infty < p_j^1 < q_j^1 < p_j^2 < q_j^2 < \cdots \quad \cdots < p_j^r < q_j^r < +\infty$, $j = 1, 2, \cdots, n$. That is, f_j is a nondecreasing piecewise linear function with $2r$ corner points.

In the following, the multistability of the system (1) is studied and a new viewpoint on their attraction basins is presented.

2 Main Results

2.1 Multistability on Neural Networks

Case 1: $r = 1,\ n = 2$**.**

For simplicity, we start with the neuron neural networks with $r = 1$ and $n = 2$. In this case, the dynamical system (1) reduces to

$$\begin{cases} \frac{du_1(t)}{dt} = -d_1u_1(t) + w_{11}f_1(u_1(t)) + w_{12}f_2(u_2(t)) + I_1, \\ \frac{du_2(t)}{dt} = -d_2u_2(t) + w_{21}f_1(u_1(t)) + w_{22}f_2(u_2(t)) + I_2, \end{cases} \tag{3}$$

with

$$f_j(x) = \begin{cases} m_j & -\infty < x < p_j, \\ \frac{M_j - m_j}{q_j - p_j}(x - p_j) + m_j & p_j \le x \le q_j, \\ M_j & q_j < x < +\infty, \end{cases} \tag{4}$$

where m_j, M_j, p_j, q_j are constants with $m_j < M_j,\ p_j < q_j, j = 1, 2, \cdots, n$.

Then, R^2 can be divided into 9 subsets: $(-\infty, p_1) \times (-\infty, p_2)$, $(-\infty, p_1) \times [p_2, q_2]$, $(-\infty, p_1) \times (q_2, \infty)$, $[p_1, q_1] \times (-\infty, p_2)$, $[p_1, q_1] \times [p_2, q_2]$, $[p_1, q_1] \times (q_2, \infty)$, $(q_1, \infty) \times (-\infty, p_2)$, $(q_1, \infty) \times [p_2, q_2]$, $(q_1, \infty) \times (q_2, \infty)$. And we have

Theorem 1. *Suppose that*

$$\begin{cases} -d_ip_i + w_{ii}m_i + \max\{w_{ij}m_j, w_{ij}M_j\} + I_i < 0, \\ -d_iq_i + w_{ii}M_i + \min\{w_{ij}m_j, w_{ij}M_j\} + I_i > 0, \end{cases} \tag{5}$$

for $i, j = 1, 2, i \neq j$. Then, system (3) with activation (4) has 9 equilibrium points, in which 4 are locally exponentially stable and others are unstable. □

Case 2: $r = 1,\ n \ge 2$**.**

Similarly, for the neural networks with n-neuron, if we denote

$$\begin{aligned} (-\infty, p_k) &= (-\infty, p_k)^1 \times [p_k, q_k]^0 \times (q_k, +\infty)^0; \\ [p_k, q_k] &= (-\infty, p_k)^0 \times [p_k, q_k]^1 \times (q_k, +\infty)^0; \\ (q_k, +\infty) &= (-\infty, p_k)^0 \times [p_k, q_k]^0 \times (q_k, +\infty)^1, \end{aligned}$$

then R^n can be divided into 3^n subsets:

$$\prod_{k=1}^{n} (-\infty, p_k)^{\delta_1^k} \times [p_k, q_k]^{\delta_2^k} \times (q_k, +\infty)^{\delta_3^k} :$$
$$(\delta_1^k, \delta_2^k, \delta_3^k) = (1, 0, 0), or (0, 1, 0), or (0, 0, 1), k = 1, \cdots, n.$$

Denote

$$\Phi_1 = \bigcup_{\substack{\eta^k \in \{0,1\}, \\ k=1,\cdots,n}} \Big(\prod_{k=1}^{n} (-\infty, p_k)^{\eta^k} \times (q_k, +\infty)^{1-\eta^k} \Big),$$

$$\Phi_2 = \prod_{k=1}^{n} [p_k, q_k], \quad \Phi_3 = R^n - \Phi_1 - \Phi_2.$$

And we get

Theorem 2. *Suppose that*

$$\begin{cases} -d_i p_i + w_{ii} m_i + \sum\limits_{\substack{j=1,\\ j\neq i}}^{n} \max\{w_{ij} m_j, w_{ij} M_j\} + I_i < 0, \\ -d_i q_i + w_{ii} M_i + \sum\limits_{\substack{j=1,\\ j\neq i}}^{n} \min\{w_{ij} m_j, w_{ij} M_j\} + I_i > 0, \end{cases} \tag{6}$$

for $i, j = 1, 2, \cdots, n$. Then, the dynamical system (1) with activation (4) has 3^n equilibrium points in all, 2^n of which are locally exponentially stable and others are unstable. □

Case 3: $r \geq 1$, $n \geq 2$**.**

Inspired by the discussions above, it is easy to see that R can be divided into $(2r+1)$ subsets, so that R^n can be divided into $(2r+1)^n$ subsets. Then, we have

Theorem 3. *Suppose that*

$$\begin{cases} -d_i p_i^k + w_{ii} m_i^k + \sum\limits_{\substack{j=1,\\ j\neq i}} \max\{w_{ij} m_j^1, w_{ij} m_j^{r+1}\} + I_i < 0, \\ -d_i q_i^k + w_{ii} m_i^{k+1} + \sum\limits_{\substack{j=1,\\ j\neq i}} \min\{w_{ij} m_j^1, w_{ij} m_j^{r+1}\} + I_i > 0, \end{cases} \tag{7}$$

for $i, j = 1, 2, \cdots, n, k = 1, 2, \cdots, r$. Then, the dynamical system (1) with activation (2) has $(2r+1)^n$ equilibrium points in all, and $(r+1)^n$ of them are locally stable and others are unstable. □

Remark 1. In fact, the method can be applied to the neural networks with multilevel sigmoid functions, such as,

$$g_j(x) = \begin{cases} -2r\ tanh(\lambda_j) + tanh(\lambda_j(x+2r)), \\ \qquad\qquad -\infty < x \leq -2r-1, \\ 2k\ tanh(\lambda_j) + tanh(\lambda_j(x-2k)), \\ \qquad\qquad 2k-1 \leq x \leq 2k+1, \\ 2r\ tanh(\lambda_j) + tanh(\lambda_j(x-2r)), \\ \qquad\qquad 2r+1 < x < \infty, \end{cases} \tag{8}$$

where $\lambda_j > 0$, $j = 1, \cdots, n$, $k = -r, \cdots, -1, 0, 1, \cdots, r$, and we can get similar conclusions. Of course, the more levels the response function has, the more tight conditions it should have for the input I_i.

2.2 Attraction Basins of Stationary Equilibria

To clarify our main points, we begin with the case of 2-neural networks with $r = 1$ for simplicity. According to Theorem 1, under conditions (5), there are 4 stable equilibrium points $u^{S_1}, u^{S_2}, u^{S_3}, u^{S_4}$, in subsets S_1, S_2, S_3, S_4, respectively, and 5 unstable equilibrium points $u^{\Xi_1}, u^{\Xi_2}, u^{\Xi_3}, u^{\Xi_4}, u^{\Lambda}$ in $\Xi_1, \Xi_2, \Xi_3, \Xi_4, \Lambda$, respectively (See Fig. 1).

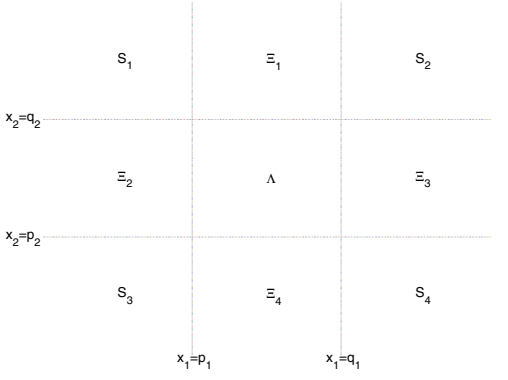


Fig. 1. Sketched map on subsets of R^2

Take a further look on the dynamics in subsets $\Xi_1, \Xi_2, \Xi_3, \Xi_4$, for example, in $\Xi_1 = (p_1, q_1) \times (q_2, +\infty)$, system (1) reduces to the following equation

$$\begin{cases} \frac{du_1(t)}{dt} = (-d_1 + w_{11}l_1)u_1(t) + w_{12}M_2 + w_{11}c_1 + I_1, \\ \frac{du_2(t)}{dt} = -d_2u_2(t) + w_{21}l_1u_1(t) + w_{22}M_2 + w_{11}c_1 + I_2, \end{cases} \tag{9}$$

where

$$l_j = \frac{M_j - m_j}{q_j - p_j},\ c_j = \frac{m_jq_j - M_jp_j}{q_j - p_j},\ i, j = 1, 2.$$

Denote $x(t) = u(t) - u^{\Xi_1}$, and we have

$$\begin{cases} \frac{dx_1(t)}{dt} = (-d_1 + w_{11}l_1)x_1(t), \\ \frac{dx_2(t)}{dt} = -d_2x_2(t) + w_{21}l_1x_1(t). \end{cases} \tag{10}$$

Then, it can be seen that, u^{Ξ_1} is attractive along the direction $x_1 = 0$, i.e., $u_1 = u_1^{\Xi_1}$. While for directions with $u_1 < u_1^{\Xi_1}$ (similarly, $u_1 > u_1^{\Xi_1}$), we see that $u_1(t)$ will exceed $p_1(q_1)$ when t is big enough, and then it is attracted to the equilibrium in $S_1(S_2)$ correspondingly. Similar conclusions can be derived for $u^{\Xi_2}, u^{\Xi_3}, u^{\Xi_4}$, and attraction basins of $u^{S_1}, \cdots, u^{S_4}$ can be expanded to subsets $\Xi_1, \cdots, \Xi_4$.

Then, consider the dynamics in Λ, which can be described as

$$\begin{cases} \frac{du_1(t)}{dt} = (-d_1 + w_{11}l_1)u_1(t) \\ \qquad\qquad + w_{12}l_2u_2(t) + w_{11}c_1 + w_{12}c_2 + I_1, \\ \frac{du_2(t)}{dt} = (-d_2 + w_{22}l_2)u_2(t) \\ \qquad\qquad + w_{21}l_1u_1(t) + w_{11}c_2 + w_{22}c_2 + I_2, \end{cases}$$

which can also be rewritten as

$$\frac{du^T(t)}{dt} = Au^T(t) + \alpha^T, \tag{11}$$

where $A = \begin{pmatrix} -d_1 + w_{11}l_1 & w_{12}l_2 \\ w_{21}l_1 & -d_2 + w_{22}l_2 \end{pmatrix}$, and $\alpha = [w_{11}c_1 + w_{12}c_2 + I_1, w_{11}c_1 + w_{12}c_2 + I_2]$.

Suppose that $det(A) > 0$. Then, all eigenvalues of A have positive real-parts, which implies that u^{Λ} is unstable and the solutions with initial states in the neighborhood of u^{Λ} diverge.

On the other hand, by translation $y^T(t) = u^T(t) + A^{-1}\alpha^T$, system (11) can be considered as homogeneous. Suppose its fundamental matrix of solution is $\hat{U}(t)$, then, its solution can be described as $y^T(t) = \hat{U}(t)y^T(0)$. Similarly to time reversal, define $\tilde{y}^T(t) = \hat{U}^{-1}(t)\tilde{y}^T(0)$, $\tilde{u}^T(t) = \tilde{y}^T(t) - A^{-1}\alpha^T$. Then, we have

$$\frac{d\tilde{y}^T(t)}{dt} = -A\tilde{y}^T(t), \tag{12}$$

and

$$\frac{d\tilde{u}^T(t)}{dt} = -A\tilde{u}^T(t) - \alpha^T, \tag{13}$$

which is an asymptotical stable system and all orbits converge to u^{Λ}.

Denote Γ_1 as the orbit of system (13) with initial state $(u_1^{\Xi_1}, q_2)$, that is,

$$\Gamma_1: \quad \tilde{u}(t) = [(u_1^{\Xi_1}, q_2) + \alpha(A^T)^{-1}]e^{-A^T t} - \alpha(A^T)^{-1}. \tag{14}$$

Similarly, there are other 3 orbits $\Gamma_2, \Gamma_3, \Gamma_4$, of system (13) with initial states $(p_1, u_2^{\Xi_2})$, $(q_1, u_2^{\Xi_3})$, $(u_1^{\Xi_4}, p_2)$, respectively, and all of them converge to u^{Λ}. Denote as

$$\Gamma_2: \quad \tilde{u}(t) = [(p_1, u_2^{\Xi_2}) + \alpha(A^T)^{-1}]e^{-A^T t} - \alpha(A^T)^{-1}, \tag{15}$$

$$\Gamma_3: \quad \tilde{u}(t) = [(q_1, u_2^{\Xi_3}) + \alpha(A^T)^{-1}]e^{-A^T t} - \alpha(A^T)^{-1}, \tag{16}$$

$$\Gamma_4: \quad \tilde{u}(t) = [(u_1^{\Xi_4}, p_2) + \alpha(A^T)^{-1}]e^{-A^T t} - \alpha(A^T)^{-1}. \tag{17}$$

Denote

$\Delta_1 \subset \Lambda$ as the region bounded by Γ_1 and Γ_2,
$\Delta_2 \subset \Lambda$ as the region bounded by Γ_1 and Γ_3,
$\Delta_3 \subset \Lambda$ as the region bounded by Γ_2 and Γ_4,
$\Delta_4 \subset \Lambda$ as the region bounded by Γ_3 and Γ_4.

Then, for

$$S_1 \bigcup (\Xi_1 \bigcap \{u : u_1 < u_1^{\Xi_1}\}) \bigcup (\Xi_2 \bigcap \{u : u_2 > u_2^{\Xi_2}\}) \bigcup \Delta_1,$$

its interior is in attraction basin of u^{S_1}, while its bound is stable manifolds of unstable equilibria u^{Ξ_1} and u^{Ξ_2}. Similar conclusions can be derived for $u^{S_2}, u^{S_3}, u^{S_4}$, and other equilibrium points.

To sum up, we have

Theorem 4. *Suppose that (5) is satisfied. If there also holds $det(A) > 0$, then, the whole state space R^2 can be divided into 4 parts, the interior of which are the very attraction basins of equilibria $u^{S_1}, u^{S_2}, u^{S_3}, u^{S_4}$, respectively.*

Furthermore, for the case $r \geq 1, n = 2$, denote

$$A^{(k,j)} = \begin{pmatrix} -d_1 + w_{11} l_1^k & w_{12} l_2^j \\ w_{21} l_1^k & -d_2 + w_{22} l_2^j \end{pmatrix},$$

where $l_\theta^k = \frac{m_\theta^{k+1} - m_\theta^k}{q_\theta^k - p_\theta^k}$, $\theta = 1, 2, k, j = 1, \cdots, r$. And we have

Theorem 5. *Suppose that (7) (with $n = 2$) is satisfied. If there also holds $det(A^{(k,j)}) > 0$ for all $k, j = 1, \cdots, r$, then, the whole state space R^2 can be divided into $(r+1)^2$ parts, the interior of which are the very attraction basins of equilibria $u^{S_1}, \cdots, u^{S_{r+1}}$, respectively.*

The details of proof and the further investigation to the general case of $n > 2$ will be presented anywhere else in our future papers.

3 Simulation

Example 1. Consider the following 2-neuron neural networks:

$$\begin{cases} \frac{du_1(t)}{dt} = -u_1(t) + 3f_1(u_1(t)) + f_2(u_2(t)) + \frac{1}{2}, \\ \frac{du_2(t)}{dt} = -2u_2(t) + f_1(u_1(t)) + 4f_2(u_2(t)) - \frac{1}{2}, \end{cases} \tag{18}$$

where we take the activation function as $f_i(x) = \frac{|x+1| - |x-1|}{2}$, $i = 1, 2$.

It is easy to see that the conditions (5) are satisfied. Therefore, according to the Theorem 1, there must be 3^2 equilibria of (18), and 4 of them are locally stable while others are unstable.

In fact, by simulations, the dynamics of system (18) are illustrated in Figs. 2 and 3, where evolutions of more than 220 initial states have been tracked. It shows that there are 4 stable equilibrium points, as confirmed by our theory.

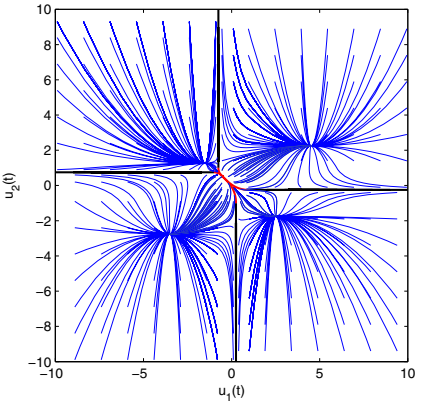


Fig. 2. The dynamics of (18) with different initial states, in which solutions with 180 initial states converging to 4 stable equilibria are depicted in blue; and other 40 solutions with initial states $u_1 = -3/4$, *or* $u_1 = 1/4$, *or* $u_2 = -1/4$, *or* $u_2 = 3/4$, respectively, are depicted in black

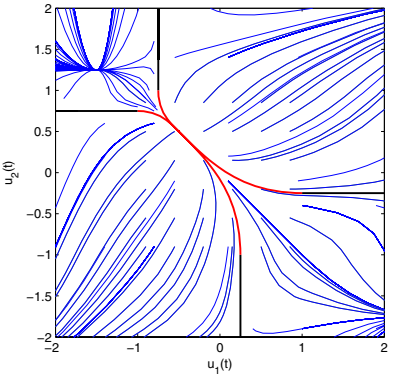


Fig. 3. Details of the inset. The 4 curves in red represent $\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4$, respectively.

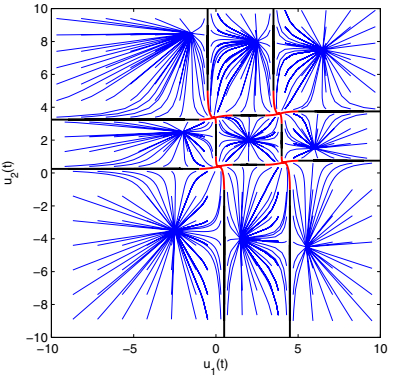


Fig. 4. The dynamics of (19) with different initial states, in which solutions with more than 220 initial states converging to 9 stable equilibria are depicted in blue; and solutions with initial states $u_1 = -1/2, 7/2, 0, 4, 1/2, 9/2,$ *or* $u_2 = 13/4, 7/2, 15/4, 1/4, 1/2, 3/4$, respectively, are depicted in black

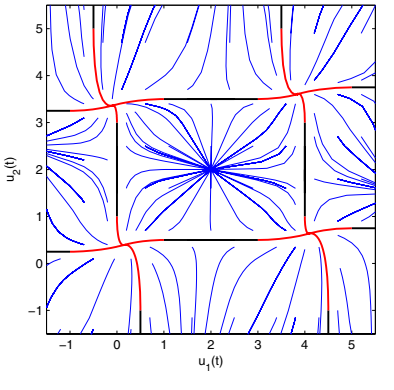


Fig. 5. Details of the inset. The 16 curves in red represent $\Gamma_1, \cdots, \Gamma_{16}$, respectively.

Example 2. Consider the following 2-neuron neural networks:

$$\begin{cases} \frac{du_1(t)}{dt} = -2u_1(t) + 4f(u_1(t)) + \frac{1}{2}f(u_2(t)) - \frac{1}{2}, \\ \frac{du_2(t)}{dt} = -2u_2(t) - \frac{1}{2}f(u_1(t)) + 6f(u_2(t)) - \frac{3}{2}, \end{cases} \tag{19}$$

where we take the activation function as

$$f(x) = \begin{cases} -1 & x \le -1, \\ x & -1 \le x \le 1, \\ 1 & 1 < x \le 3, \\ x-2 & 3 < x \le 5, \\ 3 & x > 5. \end{cases} \tag{20}$$

It is easy to see that the conditions (7) are satisfied. Therefore, according to the Theorem 3, there must be 5^2 equilibria of (19), and 3^2 of them are locally stable while others are unstable.

In fact, by simulations, the dynamics of system (19) are illustrated in Figs. 4 and 5, where evolutions of more than 300 initial states have been tracked. It shows that there are 9 stable equilibrium points, as confirmed by our theory.

References

1. Chen, T.: Global Exponential Stability of Delayed Hopfield Neural Networks. Neural Networks 14(8), 977–980 (2001)
2. Chen, T., Amari, S.: New Theorems on Global Convergence of Some Dynamical Systems. Neural Networks 14, 251–255 (2001)
3. Chen, T.: Global Convergence of Delayed Dynamical Systems. IEEE Transanctions on Neural Networks 12(6), 1532–1536 (2001)
4. Douglas, R., Koch, C., Mahowald, M., Martin, K., Suarez, H.: Recurrent Excitation in Neocortical Circuits. Science 269, 981–985 (1995)
5. Hahnloser, R.: On the Piecewise Analysis of Networks of Linear Threshold Neurons. Neural Networks 11, 691–697 (1998)
6. Wersing, H., Beyn, W.J., Ritter, H.: Dynamical Stability Conditions for Recurrent Neural Networks with Unsaturating Piecewise Linear Transfer Functions. Neural Computation 13(8), 1811–1825 (2001)
7. Forti, M.: A Note on Neural Networks with Multiple Equilibrium Points. IEEE Transactions on Circuits And Systems-I: Fundamental Theory And Applications 43(6), 487–491 (1996)
8. Zhang, Y., Tan, K.: Multistability of Discrete-Time Recurrent Neural Networks with Unsaturating Piecewise Linear Activation Functions. IEEE Transactions on Neural Networks 15(2), 329–336 (2004)
9. Zeng, Z., Wang, J.: Multiperiodicity and Exponential Attractivity Evoked by Periodic External Inputs in Delayed Cellular Neural Networks. Neural Computation 18, 848–870 (2006)
10. Cao, J., Feng, G., Wang, Y.: Multistability And Multiperiodicity of Delayed Cohen-Grossberg Neural Networks with A General Class of Activation Functions. Physica D: Nonlinear Phenomena 237(13), 1734–1749 (2008)

11. Cheng, C., Lin, K., Shih, C.: Multistability in Recurrent Neural Networks. SIAM Journal on Applied Mathematics 66(4), 1301–1320 (2006)
12. Zhang, L., Yi, Z., Yu, J.: Multiperiodicity and Attractivity of Delayed Recurrent Neural Networks With Unsaturating Piecewise Linear Transfer Functions. IEEE Transactions On Neural Networks 19(1), 158–167 (2008)
13. Ye, M.: Multistability of competitive neural networks with different time scales. In: 2005 International Conference on Communications, Circuits and Systems, Proceedings, vol. 1-2, pp. 939–943 (2005)

Robust Stability of Control Systems with One Form of Uncertain Parameters

Faming Zhang

College of Science of Hunan University of Technology , 412008 Hunan,China

Abstract. By using Lyapunov stability method and linear matrix inequalities and through establishing proper Lyapunov function, the paper studies uncertain parameter time-delay Lurie control systems with structured parameter perturbations and norm bound parameter perturbations, and thus obtains delay-dependent sufficient conditions for robust absolute stability of the systems. The research shows: When parameter is uncertain and has no norm bound restriction, these above conditions can be presented in terms of the positive definite characteristic of diagonal matrix and linear matrix, which is very directly perceived and easy to operate and can be easily solved by using Matlab Toolbox.

Keywords: Lurie control systems, Robust absolute stability, Linear matrix inequalities, Lyapunov function.

1 Introduction

Lurie control system is a very important non-linear control system. The issue of absolute stability of Lurie control system receives much attention and extensive research, which has obtained good results[1-5]. These results are all based on system model when parameter is certain. In practice, there always exist errors in model building and uncertainties in model itself, thus the parameter of system model is uncertain, which shows the importance of research on robust absolute stability of Lurie control system. Nowadays, the research of robust absolute stability of uncertain Lurie control system has gained fruitful results[6-10]. Because the introduction of linear matrix in equalities makes research results about stability less conservative than the results expressed by the traditional method of norm estimation, this paper gives sufficient condition of robust absolute stability of the system expressed by linear matrix in equalities.

Consider the system

$$\begin{cases} \dot{x}(t) = (A + \Delta A(\theta))x(t) + (B + \Delta B(\theta))x(t-\tau) + (D + \Delta D(\theta))f(\sigma(t)) \\ \sigma(t) = C^T x(t) \\ x(t) = \varphi(t), t \in [-T, 0] \end{cases} \tag{1}$$

Where $x(t) \in R^n$ is the state vector, A.B.C.D is constant matrix: $A, B \in R^{n\times n}$, $D \in R^{n\times m}$, $C_i \in R^n, i = 1,2,\cdots,m$, $C = [C_1 \quad C_2 \quad \cdots \quad C_m] \in R^{n\times m}$, $\sigma(t) = [\sigma_1(t) \quad \sigma_2(t) \quad \cdots \quad \sigma_m(t)]^T \in R^m$, $\tau > 0$ is a constant time delay, unknown

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 333–339, 2009.

parameter vector $\theta \in \Theta, \Theta$ is a set with bound restriction in R, $\varphi(t)$ is continuous vector initial function, $f(\sigma(t)) = [f_1(\sigma_1(t)) \quad f_2(\sigma_2(t)) \quad \cdots \quad f_m(\sigma_m(t))]^T$,

$f_i(\cdot) \in K_{[0,k_i]} = \{f_i(\cdot) \mid f_i(0) = 0, 0 < \sigma_i f_i(\sigma_i) \le k_i \sigma_i^2, \sigma_i \ne 0\}$,

$k_i > 0, (i = 1,2,\cdots,m)$, $\Delta A(\theta), \Delta B(\theta), \Delta D(\theta), \Delta E(\theta)$ is uncertain parameter of the system and satisfy:

$\Delta A(\theta) = G_1 F_1(\theta) H_1$, $\Delta B(\theta) = G_2 F_2(\theta) H_2$, $\Delta D(\theta) = G_3 F_3(\theta) H_3$,

Where G_i, H_i $(i = 1,2,3)$ is known as constant matrix, $F_i(\theta)$ is uncertain and satisfy: $F_i^T(\theta) F_i(\theta) \le I, (i = 1,2,3)$ .Simply, $K = diag(k_1 \quad k_2 \quad \cdots \quad k_m)$, $\overline{A} = A + \Delta A(\theta), \overline{B} = B + \Delta B(\theta), \ \overline{D} = D + \Delta D(\theta)$.

2 Main Result

Lemma 1 [11]. For arbitrary vector or matrix u, v and arbitrary symmetric positive definite matrix, the following inequality holds:

$$-u^T v - v^T u \le u^T P u + v^T P^{-1} v, \ u^T v + v^T u \le u^T P u + v^T P^{-1} v.$$

Lemma 2 [12]. For arbitrary vector or matrix u, v and arbitrary symmetric positive definite matrix, the following in equality holds:

$$(\int_0^\gamma u(\xi) d\xi)^T M \int_0^\gamma u(\xi) d\xi \le \gamma \int_0^\gamma u^T(\xi) M u(\xi) d\xi$$

Lemma 3 [13]. If real symmetric matrix u is $U = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix}$, where $A \in R^{m \times m}$, $B \in R^{m \times p}$, $C \in R^{p \times p}$, then $U < 0$ is sufficient condition of $C < 0, A - BC^{-1}B^T < 0$

Theorem 1. If there exist matrixes $P > 0, R_i > 0, (i = 1,2,3)$, $S = diag(s_1, s_2, \cdots, s_m) \ge 0$ and constant $\mu_i > 0, \varepsilon_i > 0, \eta_i > 0, (i = 1,2,3)$ satisfy the following linear matrix in equality

$$\Theta = \begin{bmatrix} \Theta_{11} & \Theta_{12} & \Theta_{13} & 0 & \Theta_{15} & \Theta_{16} \\ * & \Theta_{22} & 0 & \Theta_{24} & 0 & 0 \\ * & * & -\Theta_{33} & 0 & 0 & 0 \\ * & * & * & -\Theta_{44} & 0 & 0 \\ * & * & * & * & -\Theta_{55} & 0 \\ * & * & * & * & * & -\Theta_{66} \end{bmatrix} < 0,$$

System (1) has robust absolute stability, where

$$\Theta_{11} = \tau\sum_{i=1}^{2}\varepsilon_i H_i^T H_i + \tau(A^T R_1 A + B^T R_2 B)$$
$$+ \mu_1 H_1^T H_1 + \mu_2 H_2^T H_2 + P(A+B) + (A+B)^T P,$$
$$\Theta_{12} = PD + CSK,\ \ \Theta_{22} = \tau D^T R_3 D + (\mu_3 + \tau\varepsilon_3) H_3^T H_3 - 2S$$
$$\Theta_{13} = \left[\sqrt{\tau}A^T R_1 G_1 \quad \sqrt{\tau}B^T R_2 G_2\right],\ \Theta_{15} = \left[\sqrt{\tau}PB \quad \sqrt{\tau}PB \quad \sqrt{\tau}PB\right],$$
$$\Theta_{16} = \left[\sqrt{\tau}PG_2 \quad \sqrt{\tau}PG_2 \quad \sqrt{\tau}PG_2\right],\ \Theta_{24} = \sqrt{\tau}D^T R_3 G_3,$$
$$\Theta_{33} = diag(\varepsilon_1 I - G_1^T R_1 G_1, \varepsilon_2 I - G_2^T R_2 G_2),\ \Theta_{44} = \varepsilon_3 I - G_3^T R_3 G_3,.$$
$$\Theta_{55} = diag(R_1 - \eta_1 H_2^T H_2, R_2 - \eta_2 H_2^T H_2, R_3 - \eta_3 H_2^T H_2),$$
$$\Theta_{66} = diag(\eta_1 I, \eta_2 I, \eta_3 I),.$$

Proof: suppose $\varphi(t) = \varphi(-T), t \in [-T-\tau, -T]$, when $t \geq \tau$, then (1) can be written as:

$$\begin{cases} \dot{x}(t) = (A + B + \Delta B(\theta) + \Delta A(\theta))x(t) - (B + \Delta B(\theta))\int_{t-\tau}^{t}\dot{x}(\xi)d\xi \\ \qquad\qquad + (D + \Delta D(\theta))f(\sigma(t)) \\ \sigma(t) = C^T x(t) \\ x(t) = \varphi(t), t \in [-T-\tau, 0] \end{cases} \tag{2}$$

If $V_1(t) = x^T(t)Px(t)$, then according to the solution of (2) consider derivative, we have:

$$\dot{V}_1(t) = 2x^T(t)P\dot{x}(t) = 2x^T(t)P(\overline{A} + \overline{B})x(t) - 2x^T(t)P\overline{B}\int_{t-\tau}^{t}\dot{x}(\xi)d\xi$$
$$+ 2x^T(t)P\overline{D}f(\sigma(t))$$

From Lemma 2, we have:

$$-2x^T(t)P\overline{B}\int_{t-\tau}^{t}\dot{x}(\xi)d\xi = -2x^T(t)P\overline{B}\int_{t-\tau}^{t}[\overline{A}x(\xi) + \overline{B}x(\xi-\tau) + \overline{D}f(\sigma(\xi))]d\xi$$
$$\leq \tau x^T(t)P\overline{B}(R_1^{-1} + R_2^{-1} + R_3^{-1})\overline{B}^T Px(t) + \int_{t-\tau}^{t}x^T(\xi)\overline{A}^T R_1\overline{A}x(\xi)d\xi$$
$$+ \int_{t-\tau}^{t}x^T(\xi-\tau)\overline{B}^T R_2\overline{B}x(\xi-\tau)d\xi + \int_{t-\tau}^{t}f^T(\sigma(\xi))\overline{D}^T R_3\overline{D}f(\sigma(\xi))d\xi$$
$$2x^T(t)P\Delta A(\theta)x(t) \leq \mu_1^{-1}x^T(t)PG_1G_1^T Px(t) + \mu_1 x^T(t)H_1^T F_1^T(\theta)F_1(\theta)H_1 x(t)$$
$$\leq x^T(t)\left(\mu_1^{-1}PG_1G_1^T P + \mu_1 H_1^T H_1\right)x(t),$$
$$2x^T(t)P\Delta B(\theta)x(t) \leq x^T(t)\left(\mu_2^{-1}PG_2G_2^T P + \mu_2 H_2^T H_2\right)x(t),$$
$$2x^T(t)P\Delta D(\theta)f(\sigma(t)) \leq \mu_3^{-1}x^T(t)PG_3G_3^T Px(t) + \mu_3 f^T(\sigma(t))H_3^T H_3 f(\sigma(t))$$

According to $\Theta<0$, we have: $R_i-\eta_i H_2^T H_2>0,(i=1,2,3)$ and

$(\eta_i^{-1}I-H_2R_i^{-1}H_2^T)^{-1}=\eta_i I+\eta_i^2H_2(R_i-\eta_iH_2^TH_2)^{-1}H_2^T>0$,

thus $\eta_i^{-1}I-H_2R_i^{-1}H_2^T>0$, From Lemma 1, we have:

$\overline{B}R_i^{-1}\overline{B}^T=BR_i^{-1}B^T+G_2F_2(\theta)H_2R_i^{-1}B^T+BR_i^{-1}H_2^TF_2^T(\theta)G_2^T+G_2F_2(\theta)H_2R_i^{-1}H_2^TF_2^T(\theta)G_2^T$

$\le BR_i^{-1}B^T+G_2F_2(\theta)(\eta_i^{-1}I-H_2R_i^{-1}H_2^T)F_2^T(\theta)G_2^T+G_2F_2(\theta)H_2R_i^{-1}H_2^TF_2^T(\theta)G_2^T$

$+BR_i^{-1}H_2^T(\eta_i^{-1}I-H_2R_i^{-1}H_2^T)^{-1}H_2R_i^{-1}B^T$

$\le BR_i^{-1}B^T+\eta_i^{-1}G_2G_2^T+BR_i^{-1}H_2^T(\eta_i^{-1}I-H_2R_i^{-1}H_2^T)^{-1}H_2R_i^{-1}B^T$

$=BR_i^{-1}B^T+\eta_i^{-1}G_2G_2^T+B[-R_i^{-1}-(\eta_iH_2^TH_2-R_i)^{-1}]B^T$

$=\eta_i^{-1}G_2G_2^T+B(R_i-\eta_iH_2^TH_2)^{-1}B^T$, $(i=1,2,3)$.

$$\dot{V}_1(t)\le x^T(t)P\left(\sum_{i=1}^{3}\mu_i^{-1}G_iG_i^T+\tau\sum_{i=1}^{3}\eta_i^{-1}G_2G_2^T+\tau B\sum_{i=1}^{3}(R_i-\eta_iH_2^TH_2)^{-1}B^T\right)Px(t)$$

$$+x^T(t)[\mu_1H_1^TH_1+\mu_2H_2^TH_2+P(A+B)+(A+B)^TP]x(t)$$

$$+2x^T(t)PDf(\sigma(t))+\mu_3f^T(\sigma(t))H_3^TH_3f(\sigma(t))+I_1+I_2+I_3,$$

Where o, $I_2=\int_{t-\tau}^{t}x^T(\xi-\tau)\overline{B}^TR_2\overline{B}x(\xi-\tau)d\xi$,

$I_3=\int_{t-\tau}^{t}f^T(\sigma(\xi))\overline{D}^TR_3\overline{D}f(\sigma(\xi))d\xi$,

If $V_2(t)=\int_{-\tau}^{0}\int_{t+\eta}^{t}x^T(\xi)\overline{A}^TR_1\overline{A}x(\xi)d\xi d\eta$, $V_3(t)=\int_{-\tau}^{0}\int_{t+\eta-\tau}^{t}x^T(\xi)\overline{B}^TR_2\overline{B}x(\xi)d\xi d\eta$,

$V_4(t)=\int_{-\tau}^{0}\int_{t+\eta}^{t}f^T(\sigma(\xi))\overline{D}^TR_3\overline{D}f(\sigma(\xi))d\xi d\eta$,

Then we have:

$\dot{V}_2(t)=\tau x^T(t)\overline{A}^TR_1\overline{A}x(t)-\int_{-\tau}^{0}x^T(t+\eta)\overline{A}^TR_1\overline{A}x(t+\eta)d\eta$

$=\tau x^T(t)\overline{A}^TR_1\overline{A}x(t)-I_1$,

$\dot{V}_3(t)=\tau x^T(t)\overline{B}^TR_2\overline{B}x(t)-\int_{-\tau}^{0}x^T(t+\eta-\tau)\overline{B}^TR_2\overline{B}x(t+\eta-\tau)d\eta$

$=\tau x^T(t)\overline{B}^TR_2\overline{B}x(t)-I_2$,

$\dot{V}_4(t)=\tau f^T(\sigma(t))\overline{D}^TR_3\overline{D}f(\sigma(t))-\int_{-\tau}^{0}f^T(\sigma(t+\eta))\overline{D}^TR_3\overline{D}f(\sigma(t+\eta))d\eta$

$=\tau f^T(\sigma(t))\overline{D}^TR_3\overline{D}f(\sigma(t))-I_3$,

According to $\Theta<0$ we have: $\varepsilon_iI-G_i^TR_iG_i>0,(i=1,2,3)$, From lemma 1, we have:

$$\overline{A}^T R_1 \overline{A} = A^T R_1 A + 2A^T R_1 G_1 F_1(\theta) H_1 + H_1^T F_1^T(\theta) G_1^T R_1 G_1 F_1(\theta) H_1$$
$$\leq A^T R_1 A + A^T R_1 G_1 (\varepsilon_1 I - G_1^T R_1 G_1)^{-1} G_1^T R_1 A + \varepsilon_1 H_1^T H_1,$$
$$\overline{B}^T R_2 \overline{B} \leq B^T R_2 B + B^T R_2 G_2 (\varepsilon_2 I - G_2^T R_2 G_2)^{-1} G_2^T R_2 B + \varepsilon_2 H_2^T H_2,$$
$$\overline{D}^T R_3 \overline{D} \leq D^T R_3 D + D^T R_3 G_3 (\varepsilon_3 I - G_3^T R_3 G_3)^{-1} G_3^T R_3 D + \varepsilon_3 H_3^T H_3,$$

According to $f_i(\cdot) \in K_{[0,k_i]}$, we have:

$$f_i(\sigma_i(t))[k_i C_i^T x(t) - f_i(\sigma_i(t))] \geq 0, i = 1,2,\cdots,m.$$

Establish Lyapunov norm function: $V(t) = V_1(t) + V_2(t) + V_3(t) + V_4(t)$, then

$$\dot{V}(t) = \sum_{i=1}^{4} \dot{V}_i(t) \leq \sum_{i=1}^{4} \dot{V}_i(t) + 2\sum_{i=1}^{m} s_i f_i(\sigma_i(t))[k_i C_i^T x(t) - f_i(\sigma_i(t))]$$

$$= \begin{bmatrix} x(t) \\ f(\sigma(t)) \end{bmatrix}^T \Xi \begin{bmatrix} x(t) \\ f(\sigma(t)) \end{bmatrix}, \text{ where } \Xi = \begin{bmatrix} \Xi_{11} & PD + CSK \\ KSC^T + D^T P & \Xi_{22} \end{bmatrix},$$

$$\Xi_{11} = \Theta_{11} + P\left(\sum_{i=1}^{3} \mu_i^{-1} G_i G_i^T + \tau \sum_{i=1}^{3} \eta_i^{-1} G_2 G_2^T + \tau B \sum_{i=1}^{3} (R_i - \eta_i H_2^T H_2)^{-1} B^T \right) P$$
$$+ \tau [A^T R_1 G_1 (\varepsilon_1 I - G_1^T R_1 G_1)^{-1} G_1^T R_1 A + B^T R_2 G_2 (\varepsilon_2 I - G_2^T R_2 G_2)^{-1} G_2^T R_2 B],$$
$$\Xi_{22} = \Theta_{22} + \tau D^T R_3 G_3 (\varepsilon_3 I - G_3^T R_3 G_3)^{-1} G_3^T R_3 D,$$

From Lemma 3, $\Theta < 0$ is the sufficient condition of $\Xi < 0$ thus from $\Theta < 0$, the system has robust absolute stability.

If parameter is uncertain and has norm bound restriction

$\|\Delta A(\theta)\| \leq \alpha, \|\Delta B(\theta)\| \leq \beta, \ \|\Delta D(\theta)\| \leq \gamma,$

then the uncertainty of corresponding system (1) can be assumed:

$G_1 = H_1 = \sqrt{\alpha} I_{n \times n}, G_2 = H_2 = \sqrt{\beta} I_{n \times n}, G_3 = \sqrt{\gamma} I_{n \times m}, H_3 = \sqrt{\gamma} I_{m \times m}$, from which, time-delay dependant rules can be shown as follow:

Theorem 2. If there exist matrixes $P > 0, R_i > 0, (i = 1,2,3)$, diagonal matrix $S = diag(s_1, s_2, \cdots, s_m) \geq 0$ and constant $\mu_i > 0, \varepsilon_i > 0, \eta_i > 0, (i = 1,2,3)$ satisfy the following linear matrix in equality:

$$\Theta = \begin{bmatrix} T_{11} & T_{12} & T_{13} & 0 & T_{15} & T_{16} \\ * & T_{22} & 0 & T_{24} & 0 & 0 \\ * & * & -T_{33} & 0 & 0 & 0 \\ * & * & * & -T_{44} & 0 & 0 \\ * & * & * & * & -T_{55} & 0 \\ * & * & * & * & * & -T_{66} \end{bmatrix} < 0,$$

Then system (1) has robust absolute stability, where:

$$\mathrm{T}_{11} = [(\tau\varepsilon_1 + \mu_1)\alpha + (\tau\varepsilon_2 + \mu_2)\beta] I + \tau(A^T R_1 A + B^T R_2 B)$$
$$+ P(A + B) + (A + B)^T P$$
$$\mathrm{T}_{12} = PD + CSK, \mathrm{T}_{13} = \left[\sqrt{\tau\alpha} A^T R_1 \quad \sqrt{\tau\beta} B^T R_2 \right],$$

$$T_{15} = \left[\sqrt{\tau}PB \quad \sqrt{\tau}PB \quad \sqrt{\tau}PB\right], T_{16} = \left[\sqrt{\tau\beta}P \quad \sqrt{\tau\beta}P \quad \sqrt{\tau\beta}P\right],$$

$$T_{22} = \tau D^T R_3 D + (\mu_3 + \tau\varepsilon_3)\gamma I - 2S \ , \ T_{24} = \sqrt{\tau\gamma}D^T R_3 \ ,$$

$$T_{33} = diag(\varepsilon_1 I - \alpha R_1, \varepsilon_2 I - \beta R_2), T_{44} = \varepsilon_3 I - \gamma R_3,$$

$$T_{55} = diag(R_1 - \eta_1\beta I, R_2 - \eta_2\beta I, R_3 - \eta_3\beta I), \ T_{66} = diag(\eta_1 I, \eta_2 I, \eta_3 I)$$

3 Result Discussion

(1). About time-delay Luries control system of uncertain parameter with structured parameter perturbations and norm bound parameter perturbations, this paper gives the sufficient condition of time-delay robust absolute stability through constructing Lyapunov function, which is expressed in terms of linear matrix inequality and is easy to operate and can be easily solved by using Matlab Toolbox through computer.

(2). Reference[3]obtains the time-delay dependant sufficient condition of robust absolute stability by means of norm estimation, using Razumikhin technique. In practice, the need of calculating the maximum eigenvalue and the minimum eigenvalue of many matrixes and the norm of vectors makes calculation very complicated, however, the said results only involves the negative definite judgement of a linear matrix by using computer, which is very directly perceived and easy to operate and less conservative.

(3). Reference[7] discussed robust absolute stability of uncertain Lurie control system while this paper analyzes it when uncertain perturbations is unknown and has norm bound restriction. The said Lemma 2 is the extension of the result in Reference[7].

(4). Compared with Reference[8], this paper considers the fact that "Activation function" also has time-delay effects and improves the establishment of Lyapunov function. So the results of robust absolute stability is more general and the time-delay dependant sufficient condition is less conservative, that is, the said results are more effective than those in Reference[8].

References

1. Liao, X.X.: Necessary and Sufficient Conditions for Absolute Stability of Lurie Indirect Control Systems. J. Science in China Series A. 32, 104–1061 (1989)
2. Wang, P.G.: Absolute Stability of Lurie Indirect Control Systems with Delay-time. J. Advances in Modelling & Simulation 29, 43–49 (1992)
3. Xu, B.J., Liao, X.X.: Absolute Stability Criteria of delay-dependent for Lurie Control System. J. ACTA Automatica Sinica 28, 317–320 (2002)
4. Zhang, F.M.: Robust Absolute Stability of Control Systems with Uncertain Parameters. J. Journal of Huazhong University of Science and Technology 33, 118–122 (2005)
5. Ju, H.P., Kwon, O.M., Lee, S.M.: LMI Optimization Approach on Stability for Delayed Neural Networks of Neutral-type. J. Applied Mathematics and Computation 196, 236–244 (2008)

6. Park, J.H.: An Analysis of Global Robust Stability of Uncertain Cellular Neural Networks with Discrete and Distributed Delays. J. Chaos,Solitons,&Fractals 32, 800–807 (2007)
7. Zhang, F.M., Zhang, C.C.: Robust Stability of Lurie Control Systems with Time-delay and Uncertain Parameters. J. Journal of Hunan University of Technology 21, 28–33 (2007)
8. Feng, J.T., Nian, X.H.: Robust Absolute Stability of Uncertain Time-delay Control Systems-LMI Approach. J. Journal of Changsha Railway University 20, 37–42 (2001)
9. Liu, D.Y., Zhang, J.H., Guan, X.P.: Generalized LMI-Based Approach to the Global Asymptotic Stability of Cellular Neural Networks with Delay. J. Applied Mathematics and Mechanics 29, 735–740 (2008)
10. Liu, X.X., Xu, B.J.: Absolute Stability for Impulsive Lurie Control Systems with Multi-delay. J. Systems Engineering and Electronics 29, 946–949 (2007)
11. Xu, L.D., Cheng, C.W., Tang, B.Y.: A Linear Matrix Inequality Approach for Robust Control of Systems with Delayed States. J. European Journal of Opertational Research 124, 332–341 (2000)
12. Gu, K.Q.: An Integral Inequality in the Stability Problem of Time-delay Systems. C. In: 39th IEEE Conference on Decision and Control, pp. 2805–2810. IEEE Press, New York (2000)
13. Boyd, S., Ghaoui, L.E., Feron, E.: Linear Matrix Inequalities in System and Control Theory. SIAM, Philadelphia (1994)

Stability Analysis of a General Class of Continuous-Time Recurrent Neural Networks

Chaojin Fu[1] and Zhongsheng Wang[2]

[1] School of Mathematics and Statistics, Hubei Normal University, Huangshi, Hubei, 435002, China
chaojinfu@126.com
[2] School of Automation, Guangdong Polytechnic Normal University, Guangzhou, Guangdong, 510665, China
gszswang@126.com

Abstract. This paper presents the results of stability analysis of a general class of continuous-time recurrent neural networks. The new stability results includes sufficient conditions for global asymptotic stability. With weaker conditions and less restrictive activation functions, the new stability results improve and extend existing ones. Discussions and examples are given to illustrate and compare the new results with the existing ones.

Keywords: Neural networks, Stability, M-matrix, Positive half trajectory.

1 Introduction

Neural networks have been extensively investigated in the past two decades and have found many applications in a variety of areas, such as signal processing, pattern recognition, static image processing, optimization, associative memory [1], [2]. Some of these applications require that the equilibrium points of the designed networks be stable. So, it is important to study the stability of recurrent neural networks.

Stability is one of the important properties for dynamic systems. From a systems-theoretic point of view, the global stability of recurrent neural networks is a very interesting issue for research because of the special nonlinear structure of recurrent neural networks. From a practical point of view, the global stability of neural networks is also very important because it is a prerequisite in many neural network applications such as optimization, control, and signal processing.

Global stability of various recurrent neural networks (such as Hopfield neural networks, cellular neural networks) has been investigated extensively [3]-[8]. In stability analysis of neural networks, the qualitative properties primarily concerned are uniqueness, global asymptotic stability, and global exponential stability of their equilibria. Among the numerous results, the stability of recurrent neural networks are characterized using symmetry of weight matrices, diagonal domination of matrices, positive definiteness of matrices, M-matrix characteristics, LMI technique and Lyapunov diagonal stability [9]-[12]. Despite the existence of many

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 340–346, 2009.

reported results in the literature, there are still needs for more in-depth and comprehensive investigations. For example, in almost all the existing results, the activation functions of the neural networks are limited to be sigmoid functions, piecewise linear monotone nondecreasing functions with bounded ranges.

The purpose of this paper is to provide more comprehensive and less conservative stability results for a class of continuous-time recurrent neural networks with a general activation function. The main contributions of this paper include the derivations of new sufficient conditions for global asymptotic stability. It is shown in this paper that the new stability results in the present paper are different from and less conservative than many existing ones by using illustrative examples.

2 Model Description

Consider a general class of continuous-time recurrent neural networks described by the following dynamic equation:

$$\frac{dx(t)}{dt} = -x(t) + Wg(x(t)) + I, \;\; x(t_0) = x_0 \tag{1}$$

where $x = (x_1, x_2, \cdots, x_n)^T \in \Re^n$ is the state vector, the superscript T is the transpose operator, $W = [w_{ij}] \in \Re^{n\times n}$ is the connection weight matrix which is not necessarily symmetric, $g(x(t)) = (g_1(x(t)_1), g_2(x(t)_2), \cdots, g_n(x_n(t)))^T : \Re^n \to \Re^n$ is the vector-valued activation function, and $I = (I_1, I_2, \cdots, I_n)^T \in \Re^n$ is the external input vector (or bias).

Let x^* be an equilibrium of (1), $z := x(t) - x^*$ and $f(z) := g(z + x^*) - g(x^*)$, the dynamic equation (1) can be rewritten as

$$\frac{dz(t)}{dt} = -z(t) + Wf(z(t)), \;\; z(t_0) = z_0. \tag{2}$$

Note that an equilibrium of (2) is 0 since $f(0) = 0$.

The basic assumptions on an activation function $g_i(x(t))$ are continuous and monotone nondecreasing; i.e., its upper-right Dini-derivative $D^+g_i \geq 0$. Let this class of activation function be denoted as

$$\mathcal{G} := \{g(x(t))|g_i(x_i(t)) \in C[\Re, \Re], D^+g_i(x_i(t)) \geq 0, \;\; i = 1, 2, \ldots, n\}.$$

Note that the derivative of the activation function in $\mathcal{G}$ is allowed to be unbounded. In this paper, we will study mainly the global stability of recurrent neural networks with activation functions in $\mathcal{G}$.

3 Preliminaries

Consider a general autonomous dynamic system defined by the following ordinary differential equation:

$$\frac{dz(t)}{dt} = s(z(t)), \;\; z(t_0) = z_0, \tag{3}$$

where $z \in \Re^n$, $s(0) = 0$, $s(z(t))$ is locally Lipschitz continuous.

Definition 1. The set $Z_+ = \{z|z(t,\ t_0,\ z_0),\ t \geq t_0\}$ is called a positive half trajectory with initial condition (t_0, z_0). If $z_0 \neq 0$, then Z_+ is called a nontrivial positive half trajectory.

Lemma 1. (Barbashin-Krasovski asymptotic stability theorem): If there exists a positive definite and radially unbounded Lyapunov function $V(z(t)) \in C[\Re^n, \Re]$ such that $dV(z(t))/dt|_{(3)} \leq 0$ and the set $M := \{z|dV(z(t))/dt|_{(3)} = 0\}$ does not include the whole positive half trajectory except at 0, then the zero solution of the dynamic system (3) is globally asymptotically stable.

4 Global Asymptotic Stability

In this section, we investigate the global asymptotic stability of equilibria of the neural network (1).

Theorem 1. Let $g_i \in \mathcal{G}$. Suppose that $\forall x_i(t) \neq x_i^*, g_i(x_i(t)) \neq g_i(x_i^*); L_i := \sup_{r\in\Re} D^+g_i(r) = D^+g_i(x_i^*) \not\equiv D^+g_i(x_i(t))$ $(i = 1, 2, \ldots, n)$. If there exists a positive diagonal matrix P such that $Q := P(W - L^{-1}) + (W - L^{-1})^T P$ is negative semi-definite, where $L = diag\{L_i\}$, then the equilibrium point x^* of the neural network (1) is globally asymptotically stable.

Proof: Let's rewrite (1) as (2). We construct a positive definite and radially unbounded Lyapunov function

$$V(z(t)) = \sum_{i=1}^{n} p_i \int_0^{z_i(t)} f_i(z_i(t))dz_i(t). \tag{4}$$

Calculating the time derivative $dV(z(t)/dt$ along the positive half trajectory of (2), we obtain

$$\begin{aligned}
\frac{dV(z(t))}{dt}|_{(2)} &= \frac{1}{2}f(z(t))^T(PW + W^TP)f(z(t)) - z(t)^TPf(z(t)) \\
&\leq \frac{1}{2}f(z(t))^T(PW + W^TP)f(z(t)) - f(z(t))^TPL^{-1}f(z(t)) \\
&= \frac{1}{2}f(z(t))^T[P(W - L^{-1}) + (W - L^{-1})^TP]f(z(t)) \\
&\leq 0.
\end{aligned}$$

Now let's prove that the set $\{z \mid dV(z(t))/dt|_{(2)} = 0\}$ does not include the positive half trajectory of (2) except at 0. It is very obvious that $\forall z_i \neq 0$, $L_i = \sup_{r\in\Re} D^+f_i(r) \not\equiv D^+f_i(z_i)$. Without loss of generality, let $z(t) = (0, \ldots, 0, z_\ell(t), \ldots, z_n(t)) \neq 0$ be any non-equilibrium state in the positive half trajectory of (2), $z_j(t) \neq 0$ for $\ell \leq j \leq n$ and $\delta_j := D_+f_j(z_j(t)) < \sup_{r\in\Re} D^+f_j(r) = D^+f_j(0) = L_j$ $(j = \ell, \ell + 1, \ldots, n)$. Because $D^+f_i(z_i(t)) \not\equiv L_i$,

$$
\begin{aligned}
\frac{dV(z(t))}{dt}|_{(2)} &= \frac{1}{2}f(z(t))^T\left[P\left(W-\mathrm{diag}(\frac{1}{L_1},\cdots,\frac{1}{L_{\ell-1}},\frac{1}{\delta_\ell},\cdots,\frac{1}{\delta_n})\right)\right.\\
&\quad\left.+\left(W-\mathrm{diag}(\frac{1}{L_1},\cdots,\frac{1}{L_{\ell-1}},\frac{1}{\delta_\ell},\cdots,\frac{1}{\delta_n})\right)^T P\right]f(z(t))\\
&= \frac{1}{2}f(z(t))^T\left[P\left(W-\mathrm{diag}(\frac{1}{L_1},\cdots,\frac{1}{L_{\ell-1}},\ \frac{1}{\delta_\ell},\cdots,\frac{1}{\delta_n})\right)\right.\\
&\quad\left.+\left(W-\mathrm{diag}(\frac{1}{L_1},\cdots,\frac{1}{L_{\ell-1}},\frac{1}{\delta_\ell},\cdots,\frac{1}{\delta_n})\right)^T P\right]f(z(t))\\
&\quad+\frac{1}{2}f(z(t))^T\left[P(W-L^{-1})+(W-L^{-1})^T P\right]f(z(t))\\
&\quad-\frac{1}{2}f(z(t))^T\left[P(W-L^{-1})+(W-L^{-1})^T P\right]f(z(t))\\
&= \frac{1}{2}f(z(t))^T\left[P\left(W-\mathrm{diag}(\frac{1}{L_1},\cdots,\frac{1}{L_{\ell-1}},\ \frac{1}{\delta_\ell},\cdots,\frac{1}{\delta_n})\right)\right.\\
&\quad\left.+\left(W-\mathrm{diag}(\frac{1}{L_1},\cdots,\ \frac{1}{L_{\ell-1}},\ \frac{1}{\delta_\ell},\cdots,\frac{1}{\delta_n})\right)^T P\right]f(z(t))\\
&\quad+\frac{1}{2}f(z(t))^T\left[P(W-\mathrm{diag}(\frac{1}{L_1},\cdots,\frac{1}{L_n})\right.\\
&\quad-P(W-\mathrm{diag}(\frac{1}{L_1},\cdots,\frac{1}{L_{\ell-1}},\frac{1}{\delta_\ell},\cdots,\frac{1}{\delta_n})\\
&\quad+(W-\mathrm{diag}(\frac{1}{L_1},\cdots,\frac{1}{L_n})^T P\\
&\quad\left.-(W-\mathrm{diag}(\frac{1}{L_1},\cdots,\frac{1}{L_{\ell-1}},\frac{1}{\delta_\ell},\cdots,\frac{1}{\delta_n})^T P\right]f(x(z))\\
&\quad+\frac{1}{2}f(z(t))^T\left[P\left(\mathrm{diag}(0,\ \cdots,\ 0,\frac{1}{L_\ell}-\frac{1}{\delta_\ell},\cdots,\frac{1}{L_n}-\frac{1}{\delta_n})\right)\right.\\
&\quad\left.+\left(\mathrm{diag}(0,\cdots,0,\frac{1}{L_\ell}-\frac{1}{\delta_\ell},\cdots,\frac{1}{L_n}-\frac{1}{\delta_n})\right)^T P\right]f(z(t))\\
&\le \frac{1}{2}f(z(t))^T\left[P\left(\mathrm{diag}(0,\ \cdots,\ 0,\frac{1}{L_\ell}-\frac{1}{\delta_\ell},\cdots,\frac{1}{L_n}-\frac{1}{\delta_n})\right)\right.\\
&\quad\left.+\left(\mathrm{diag}(0,\cdots,0,\frac{1}{L_\ell}-\frac{1}{\delta_\ell},\cdots,\frac{1}{L_n}-\frac{1}{\delta_n})\right)^T P\right]f(z(t))\\
&= \frac{1}{2}\sum_{j=\ell}^{n}p_j(\frac{1}{L_j}-\frac{1}{\delta_j})(f_j(z_j(t)))^2<0,\ \forall z(t)\neq 0.
\end{aligned}
$$

Therefore, the positive half trajectory of (2) is not included in $\{x|dV(z(t))/dt = 0\}$ except at 0. According to Lemma 1, we know that the equilibrium point x^* of (1) is globally asymptotically stable. As a result, the equilibrium point is unique.

Remark 1. Theorem 1 is an extension and improvement of the existing ones.

Definition 2. An $n \times n$ matrix $A = [a_{ij}]$ is a nonsingular M-matrix if for $i, j = 1, 2, \ldots, n; i \neq j$; $a_{ii} > 0, a_{ij} \leq 0$, and

$$\begin{pmatrix} a_{11} & \ldots & a_{1i} \\ \vdots & \ddots & \vdots \\ a_{i1} & \cdots & a_{ii} \end{pmatrix} > 0.$$

Remark 2. For $i, j = 1, 2, \ldots, n$, let $\tilde{W} = [\tilde{w}_{ij}]_{n\times n}$ and

$$\tilde{w}_{ij} := \begin{cases} |w_{ii}| + \frac{1}{L_i}, & \text{if } \ i = j; \\ -|w_{ij}| & \text{if } \ i \neq j \end{cases}$$

It is known that the so-called M-criterion is a very useful method on the stability of neural networks. The M-criterion ensure that if $\tilde{W}$ is a nonsingular M-matrix, then the equilibrium point x^* of (1) is globally asymptotically stable.

Example 1. Given a two-state neural network as follows:

$$\begin{aligned} \frac{dx_1(t)}{dt} &= -x_1(t) - 2g_1(x_1(t)) + 9g_2(x_2(t)) + I_1, \\ \frac{dx_2(t)}{dt} &= -x_2(t) + g_1(x_1(t)) - 2g_2(x_2(t)) + I_2, \end{aligned} \tag{5}$$

where $L_i = \sup_{r\in\Re} D^+g_i(r) = D^+g_i(x_i^*) \not\equiv D^+g_i(x_i(t))$ at finite points of $x_i(t) \neq x_i^*$ only $D^+g_i(x_i^*) = L_i$, $L_i = 1$ for $i = 1, 2$.

In this case,

$$\tilde{W} = \begin{pmatrix} |w_{11}| + \frac{1}{L_1}, & -|w_{12}| \\ -|w_{21}|, & |w_{22}| + \frac{1}{L_2} \end{pmatrix} = \begin{pmatrix} 3 & -9 \\ -1 & 3 \end{pmatrix}, \tag{6}$$

which is not an M matrix. Hence, the M-criterion is no longer has any effect on neural network (5). In addition,

$$A := W - L^{-1} = \begin{pmatrix} -3 & 9 \\ 1 & -3 \end{pmatrix}$$

is not Lyapunov diagonally stable because A is Lyapunov diagonally stable if and only if $a_{11} = -3 < 0$, $a_{22} = -3 < 0$, and $a_{11}a_{22} > a_{12}a_{21}$. However, $a_{11}a_{22} = a_{12}a_{21} = 9$.

Now we take $P = \mathrm{diag}(1/3,\ 3)$, then

$$\begin{aligned} PA + A^TP &= \begin{pmatrix} \frac{1}{3} & 0 \\ 0 & 3 \end{pmatrix} \begin{pmatrix} -3 & 9 \\ 1 & -3 \end{pmatrix} + \begin{pmatrix} -3 & 1 \\ 9 & -3 \end{pmatrix} \begin{pmatrix} \frac{1}{3} & 0 \\ 0 & 3 \end{pmatrix} \\ &= \begin{pmatrix} -1 & 3 \\ 3 & -9 \end{pmatrix} + \begin{pmatrix} -1 & 3 \\ 3 & -9 \end{pmatrix} = \begin{pmatrix} -2 & 6 \\ 6 & -18 \end{pmatrix} \leq 0. \end{aligned}$$

So the condition of Theorem 1 is satisfied. Hence, the equilibrium point of (5) is globally asymptotically stable.

Corollary 1. Let $L_i = \sup_{r\in\Re} D^+g_i(r) = D^+g_i(x_i^*) \not\equiv D^+g_i(x_i), i = 1, 2, \cdots, n$, $\forall x_i \neq x_i^*$. If there exist n positive constants $p_i > 0$ $(i = 1, 2, \ldots, n)$ such that

$$p_i(w_{ii} - \frac{1}{L_i}) + \sum_{i=1, i\neq j}^{n} p_i|w_{ij}| \leq 0, \tag{7}$$

then the equilibrium point of x^* of (1) is globally asymptotically stable.

Proof: Let the positive definite and radially unbounded Lyapunov function be

$$V(z(t)) = \sum_{i=1}^{n} p_i|z_i(t)|.$$

Calculating the right-upper Dini-derivative of (2), we obtain

$$\begin{aligned} D^+V(z(t))|_{(2)} &= \sum_{i=1}^{n} p_i \frac{dz_i(t)}{dt} sgn|z_i(t)| \\ &\leq -\sum_{j=1}^{n} p_j|z_j(t)| + \sum_{j=1}^{n} \left[p_j w_{jj} + \sum_{i=1, i\neq j}^{n} p_i|w_{ij}| \right] |f_j(z_j(t))| \\ &\leq -\sum_{j=1}^{n} \frac{p_j}{L_j}|f_j(z_j(t))| + \sum_{j=1}^{n} \left[p_j w_{jj} + \sum_{i=1, i\neq j}^{n} p_i|w_{ij}| \right] |f_j(z_j(t))| \\ &= \sum_{j=1}^{n} \left[p_j(w_{jj} - \frac{1}{L_j}) + \sum_{i=1, i\neq j}^{n} p_i|w_{ij}| \right] |f_j(z_j(t))| \ \leq 0. \end{aligned}$$

Similar to the proof of the final part of Theorem 1, we can prove that $\{z \ |D^+V(z(t))|_{(2)} = 0\}$ does not include the whole positive half trajectory of (2) except at 0. According to Lemma 1, we know the equilibrium point x^* of (1) is globally asymptotically stable.

Example 2. Consider a neural network as follows:

$$\begin{aligned} \frac{dx_1(t)}{dt} &= -x_1(t) - 3g_1(x_1(t)) + 2g_2(x_2(t)) - 2g_3(x_3(t)) + I_1, \\ \frac{dx_2(t)}{dt} &= -x_2(t) + 3g_1(x_1(t)) - 3g_2(x_2(t)) + g_3(x_3(t)) + I_2, \\ \frac{dx_3(t)}{dt} &= -x_3(t) + g_1(x_1(t)) + 2g_2(x_2(t)) - 2g_3(x_3(t)) + I_3, \end{aligned}$$

where $g_i(x_i(t)) \in \mathcal{G}, L_i = 1, i = 1, 2, 3$.

Similar to (6), it is well known that

$$\tilde{W} = \begin{pmatrix} 4 & -2 & -2 \\ -3 & 4 & -1 \\ -1 & -2 & 3 \end{pmatrix}.$$

Because $\det(\tilde{W}) = 0$, $\tilde{W}$ is not an M matrix. Therefore, the conditions of the M-criterion are not satisfied. Now we take $p_1 = p_2 = p_3 = 1$, then the condition of Corollary 1 is satisfied.

5 Concluding Remarks

In this paper, several sufficient conditions are presented for checking the global asymptotic stability of a general class of continuous-time recurrent neural networks. The new stability conditions have shown to have improved results compared with existing ones.

Acknowledgments. This work was supported by the Innovative Research Team of Hubei Normal University and the Key Program of Educational Department of Hubei Province of China under Grant D20082201.

References

1. Hopfield, J.J.: Neurons with Graded Response Have Collective Computational Properties Like Those of Two State Neurons. Proceedings of the National Academy of Sciences, USA 81, 3088–3092 (1984)
2. Gopalsamy, K., He, X.Z.: Stability in Asymmetric Hopfield Nets with Transmission Delays. Physica D 76, 344–358 (1994)
3. Zhang, Y., Zhong, S.M., Li, Z.L.: Periodic Solutions and Stability of Hopfield Neural Networks with Variable Delays. International Journal of Systems Sciences 27, 895–901 (1996)
4. Zeng, Z.G., Wang, J.: Complete Stability of Cellular Neural Networks with Time-varying Delays. IEEE Trans. on Circuits and Systems-I: Regular Papers 53, 944–955 (2006)
5. Zeng, Z.G., Huang, D.S., Wang, Z.F.: Global Stability of a General Class of Discrete-time Recurrent Neural Networks. Neural Processing Letters 22, 33–47 (2005)
6. Van, D.D., Zou, X.: Global Aattractivity in Delayed Hopfield Neural Networks Model. SIAM Journal of Applied Mathematics 58, 1878–1890 (1998)
7. Chu, T.G.: An Exponential Convergence Estimate for Analog Neural Networks with Delay. Physics Letters A 283, 113–118 (2001)
8. Li, Y.K., Lu, L.H.: Global Exponential Stability and Existence of Periodic Solution of Hopfield-Type Neural Networks with Impulses. Physics Letters A 333, 62–71 (2004)
9. Zeng, Z.G., Wang, J., Liao, X.X.: Global Asymptotic Stability and Global Exponential Stability of Neural Networks with Unbounded Time-varying Delays. IEEE Trans. on Circuits and Systems II, Express Briefs 52, 168–173 (2005)
10. Zeng, Z.G., Wang, J.: Multiperiodicity and Exponential Attractivity Evoked by Periodic External Inputs in Delayed Cellular Neural Networks. Neural Computation 18, 848–870 (2006)
11. Xu, D.Y., Yang, Z.C.: Impulsive Delay Differential Inequality and Stability of Neural Networks. Journal of Mathematical Analysis and Applications 305, 107–120 (2005)
12. Zhang, Y., Sun, J.T.: Stability of Impulsive Neural Networks with Time Delays. Physics Letters A 348, 44–50 (2005)

A New LMI-Based Stability Criteria for Delayed Cellular Neural Networks

Yanjun Shen[1], Linguo Zhang[2], and Yong Zhang[1]

[1] College of Science, China Three Gorges University,
Yichang, Hubei 443002, China
shenyj@ctgu.edu.cn
[2] Hubei Three Gorges Vocation and Technology College,
Yichang, Hubei 443000, China
linguozhang@126.com

Abstract. In this paper, the global asymptotic stability for delayed cellular neural networks is addressed with a new Lyapunov-Krasovskii function. New delay-independent LMI-based conditions for global asymptotic stability are derived. A key feature of the new approach is the introduction an integral of term of neuron activation functions in the Lyapunov-Krasovskill function, which can provide useful and less conservative results. Finally, two numerical examples show the effectiveness of the proposed method.

Keywords: Delay cellular neural networks; Global asymptotic stability; Linear matrix inequality (LMI); Lyapunov-Krasovskii function.

1 Introduction

Cellular neural networks (CNNs), which were introduced in [1], have many applications in image processing, pattern recognition, optimal computation, association *etc.* Some of these applications require that the equilibrium points of the designed network be stable. Thus, it is important to study the stability of CNNs. In biological and artificial neural networks, time delays often arise in the processing of information storage and transmission. The existence of time delays may lead to oscillation, divergence or instability in neural networks due to finite speed of information processing. Fundamental results have been established on uniqueness, global stability of the equilibrium point for delayed cellular neural networks (DCNNs) [2,3,4,5,6,7,8,9]. Furthermore, in the design of neural networks, one is not only interested in global stability, but also interested in some other performance such as global exponential stability. Therefore, exponential stability for DCNNs has been investigated by many researchers [10,11]. The same idea of free-weighting matrices has been used in [12] as well to provide better exponential stability test for DCNNs.

Recently, an LMI-based asymptotical stability criterion was derived for DCNNs by employing S-proceduce [13] to deal with the nonlinearities [14]. However, the constraints on the nonlinearities in [14] are very strict, *i.e.*, the upper

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 347–356, 2009.

bounds of the sectors are set to 1. Even though they are relaxed to k in [15], all the bounds are set to the same value. For different value of upper bounds, the global asymptotic stability for DCNNs was studied by Cao and Ho in [16]. They obtained a generalized and improved results. Later, Singh [17] presented a simplified version of the respective LMI based criterion given by Cao and Ho [16]. He *et al* [18] presented a new Lyapunov-Krasovskii functional containing an integral term of state for DCNNs. A global asymptotical stability criterion was derived. However, these results are all sufficient conditions and have conservatism to some degree, which leaves space for further improvement.

In this paper, by utilizing a new Lyapunov-Krasovskii functional, we present an improved global asymptotic stability conditions for DCNNs. These derived conditions are expressed in terms of LMIs, which can be checked numerically very efficiently. A key feature of our approach is the introduction an integral of term of neuron activation functions in the Lyapunov-Krasovskill function, which is helpful to reduce conservatism in applying the criteria in the analysis. This feature also enables us to cast a series of previous results such as in [3,4,5,6,7,14,16,17,18,19] into even more general framework. Two numerical examples show the effectiveness of the proposed method.

The organization of this paper is as follows. The problem statement is given in Section II. Section III presents the main results. Numerical examples are given in Section IV. Finally, the paper is concluded in Section V.

2 Problem Statement

Consider the following cellular neural networks with a time-varying delay, which is described by a nonlinear delay differential equation of the form:

$$\dot{x}(t) = -Cx(t) + Ag(x(t)) + Bg(x(t - d(t))) + u, \tag{1}$$

where $x(t) = [x_1(t), x_2(t), \cdots, x_n(t)]^T$ is the state vector associated with n neurons, $C = \text{diag}\{c_1, c_2, \cdots, c_n\} > 0$, A is a feedback matrix, and B is a delayed feedback matrix. $u = [u_1, u_2, \cdots, u_n]^T$ is a constant external input vector. The scalar $d(t)$ represents the time-varying delay which satisfies that

$$0 \leq \dot{d}(t) \leq \mu. \tag{2}$$

$g(x(t)) = [g_1(x_1(t)), g_2(x_2(t)), \cdots, g_n(x_n(t))]^T$ is the neuron activation function. It is assumed that each neuron activation function $g_i(x_i)$ is bounded, monotonically nondecreasing and satisfies the following condition:

$$|g_i(x) - g_i(y)| \leq k_i|x - y|, \quad i = 1, 2, \cdots, n, \tag{3}$$

for any $x, y \in \mathcal{R}$, where $k_i > 0$ for $i = 1, 2 \cdots, n$.

Let $x^* = [x_1^*, x_2^*, \cdots, x_n^*]^T$ be the equilibrium point of (1). Then, by letting $z(t) = x(t) - x^*$, the system (1) can be transformed into:

$$\dot{z}(t) = -Cz(t) + Af(z(t)) + Bf(z(t - d(t))), \tag{4}$$

where $z(t) = [z_1(t), z_2(t), \cdots, z_n(t)]^T$ is the state, $f(z(t)) = [f_1(z_1(t)), f_2(z_2(t)), \cdots, f_n(z_n(t))]^T$ and $f_i(z_i) = g_i(z_i + x_i^*) - g_i(x_i^*),\ \ i = 1, 2, \cdots, n$. Note that the functions $f_i(\cdot)$, $i = 1, 2, \cdots, n$ is monotonically nondecreasing with the relation:

$$|f_i(z_i)| \leq k_i|z_i|,\ i = 1, \cdots, n,$$

i.e.,

$$0 \leq \frac{f_i(z_i)}{z_i} \leq k_i,\ \forall z_i \neq 0,\ f_i(0) = 0, i = 1, \cdots, n, \tag{5}$$

which are equivalent to

$$f_j(z_j)[f_j(z_j) - k_j z_j] \leq 0,\ f_j(0) = 0,\ j = 1, 2, \cdots, n. \tag{6}$$

The following Lemma is useful to derive the main result.

Lemma 1. *[13] (S-procedure) Let $T_i \in \mathcal{R}^{n\times n}$ $(i = 0, 1, \cdots, p)$ be symmetric matrices. The conditions on T_i $(i = 0, 1, \cdots, p)$*

$$\zeta^T T_0 \zeta > 0,\ \forall \zeta \neq 0\ s.t.\ \zeta^T T_i \zeta \geq 0\ (i = 1, 2, \cdots, p), \tag{7}$$

hold if there exist $\tau_i \geq 0$ $(i = 1, 2, \cdots, p)$ such that

$$T_0 - \sum_{i=1}^{p} \tau_i T_i > 0. \tag{8}$$

3 Main Results

In this section, we construct a new Lyapunov-Krasovskii functional and use the S-procedure to handle the nonlinearities. Then, new asymptotic stability criteria are obtained.

Theorem 1. *Under the conditions (2) and (5), the system (4) has a unique equilibrium point and that is global asymptotically stable if there exist four positive definite symmetric matrices P, R, Q, V and three definite diagonal matrices $T = diag\{t_1, t_2, \cdots, t_n\}$, $S = diag\{s_1, s_2, \cdots, s_n\}$, $D = diag\{d_1, d_2, \cdots, d_n\}$, $W = diag\{w_1, w_2, \cdots, w_n\}$ and four matrices U, Y_1, Y_2, Y_3 such that the following LMIs hold:*

$$\Phi = \begin{bmatrix} \Phi_{11} & -Y_1^T + Y_2 - V & \Phi_{13} \\ -Y_1 + Y_2^T - V & \Phi_{22} & 0 \\ \Phi_{13}^T & 0 & \Phi_{33} \\ B^T P & SK - U^T & B^T D \\ -Y_1 + Y_3^T & -Y_2 - Y_3^T & 0 \end{bmatrix}$$

$$\begin{bmatrix} PB & -Y_1^T + Y_3 \\ KS - U & -Y_2^T - Y_3 \\ DB & 0 \\ -(1-\mu)(Q - K^T W K) - 2S & 0 \\ 0 & -V - Y_3 - Y_3^T \end{bmatrix} < 0, \tag{9}$$

$$\begin{bmatrix} R & U \\ U^T & Q \end{bmatrix} > 0, \tag{10}$$

where $\Phi_{11} = -CP - PC + R + Y_1 + Y_1^T + V + K^T W K$, $\Phi_{22} = -(1-\mu)R - Y_2 - Y_2^T + V - (1-\mu)K^T W K$, $\Phi_{13} = PA - CD + KT + U$, $\Phi_{33} = A^T D + DA + Q - 2T - K^T W K$, $K = diag\{k_1, k_2, \cdots, k_n\}$.

Proof. First, we prove the uniqueness of the equilibrium point by contradiction. Assume that $\bar{z}$ is the equilibrium point of the delayed DCNNs in (4). Then, we have

$$-C\bar{z} + (A+B)f(\bar{z}) = 0, \tag{11}$$

Suppose $f(\bar{z}) \neq 0$. By (11), we can obtain

$$2\bar{z}^T P[-C\bar{z} + (A+B)f(\bar{z})] = 0, \tag{12}$$

and

$$2f^T(\bar{z})D[-C\bar{z} + (A+B)f(\bar{z})] = 0, \tag{13}$$

Note that

$$\begin{aligned} &2f^T(\bar{z})D[-C\bar{z} + (A+B)f(\bar{z})] \\ &= 2f^T(\bar{z})[D(A+B) - S - T]f(\bar{z}) + 2f^T(\bar{z})(S+T)f(\bar{z}) - 2f^T(\bar{z})DC\bar{z} \\ &\leq 2f^T(\bar{z})[D(A+B) - S - T]f(\bar{z}) + 2f^T(\bar{z})(SK + TK - DC)\bar{z}. \end{aligned} \tag{14}$$

and

$$\mu z^T(t)K^T W K z(t) - \mu f^T(z(t))K^T W K f(z(t)) \geq 0. \tag{15}$$

Then, form (12)-(15) it follows that

$$\begin{aligned} &2\bar{z}^T P[-C\bar{z} + (A+B)f(\bar{z})] + 2f^T(\bar{z})(SK + TK - DC)\bar{z} \\ &+2f^T(\bar{z})[D(A+B) - S - T - \mu K^T W K]f(\bar{z}) + \mu z^T(t)K^T W K z(t) \geq 0, \end{aligned}$$

i.e.,

$$\begin{bmatrix} \bar{z} \\ f(\bar{z}) \end{bmatrix}^T \begin{bmatrix} -CP - PC + \mu K^T W K & \Omega \\ \Omega^T & \Lambda - \mu K^T W K \end{bmatrix} \begin{bmatrix} \bar{z} \\ f(\bar{z}) \end{bmatrix} \geq 0, \tag{16}$$

where $\Omega = P(A+B) + K(T+S) - CD$, $\Lambda = D(A+B) + (A+B)^T D - 2S - 2T$. On the other hand, pre- and post-multiplying (9) by

$$\begin{bmatrix} I & I & 0 & 0 & 0 \\ 0 & 0 & I & I & 0 \end{bmatrix}$$

and its transpose, respectively, we have

$$\begin{bmatrix} -CP - PC + \mu(R + K^T W K) & \Omega \\ \Omega & \Lambda + \mu(R - K^T W K) \end{bmatrix} < 0. \tag{17}$$

(9) and (17) imply that

$$\begin{bmatrix} -CP - PC + \mu K^T W K & \Omega \\ \Omega & \Lambda - \mu K^T W K \end{bmatrix} < 0, \tag{18}$$

which contradicts with (16). Therefore, we can conclude that (4) has a unique equilibrium point.

Next, we prove that the unique equilibrium point of (4) is asymptotically stable. Consider the following Lyapunov-Krasovskii functional:

$$V(z(t)) = z^T(t)Pz(t) + 2\sum_{j=1}^{n}\int_0^{z_j(t)} d_j f_j(s)ds + \int_{t-d(t)}^{t}[x^T(s)KWKx(s) - f^T(x(s))Wf(x(s))]ds + \int_{t-d(t)}^{t}\begin{bmatrix} z(s) \\ f(z(s)) \end{bmatrix}^T \begin{bmatrix} R & U \\ U^T & Q \end{bmatrix}\begin{bmatrix} z(s) \\ f(z(s)) \end{bmatrix} ds. \tag{19}$$

Calculating the derivative of $V(z(t))$ along the solution of system (4), we can obtain:

$$\begin{aligned}\dot{V}(z(t)) &= 2z^T(t)P\dot{z}(t) + 2\sum_{j=1}^{n} d_j f_j(z_j(t))\dot{z}_j(t) + x^T(t)K^TWKx(t) \\ &-f^T(x(t))K^TWKf(x(t)) - (1-\dot{d}(t))[x^T(t-d(t))K^TWKx(t-d(t)) \\ &-f^T(x(t-d(t)))K^TWKf(t-d(t))] + \begin{bmatrix} z(t) \\ f(z(t)) \end{bmatrix}^T \begin{bmatrix} R & U \\ U^T & Q \end{bmatrix}\begin{bmatrix} z(t) \\ f(z(t)) \end{bmatrix} \\ &-(1-\dot{d}(t))\begin{bmatrix} z(t-d(t)) \\ f(z(t-d(t))) \end{bmatrix}^T \begin{bmatrix} R & U \\ U^T & Q \end{bmatrix}\begin{bmatrix} z(t-d(t)) \\ f(z(t-d(t))) \end{bmatrix}.\end{aligned}$$

Form (2) it follows that

$$\begin{aligned}\dot{V}(z(t)) &\le 2z^T(t)P\dot{z}(t) + 2\sum_{j=1}^{n} d_i f_j(z_j(t))\dot{z}_j(t) + x^T(t)K^TWKx(t) \\ &-f^T(x(t))K^TWKf(x(t)) - (1-\mu)[x^T(t-d(t))K^TWKx(t-d(t)) \\ &-f^T(x(t-d(t)))K^TWKf(t-d(t))] + \begin{bmatrix} z(t) \\ f(z(t)) \end{bmatrix}^T \begin{bmatrix} R & U \\ U^T & Q \end{bmatrix}\begin{bmatrix} z(t) \\ f(z(t)) \end{bmatrix} \\ &-(1-\mu)\begin{bmatrix} z(t-d(t)) \\ f(z(t-d(t))) \end{bmatrix}^T \begin{bmatrix} R & U \\ U^T & Q \end{bmatrix}\begin{bmatrix} z(t-d(t)) \\ f(z(t-d(t))) \end{bmatrix}.\end{aligned} \tag{20}$$

Note that

$$\begin{aligned} 2z^T(t)Y_1^T[z(t) - z(t-d(t)) - \textstyle\int_{t-d(t)}^{t}\dot{z}(s)ds] = 0, \\ 2z^T(t-d(t))Y_2^T[z(t) - z(t-d(t)) - \textstyle\int_{t-d(t)}^{t}\dot{z}(s)ds] = 0, \\ (2\textstyle\int_{t-d(t)}^{t}\dot{z}^T(s)ds)Y_3^T[z(t) - z(t-d(t)) - \textstyle\int_{t-d(t)}^{t}\dot{z}(s)ds] = 0, \end{aligned} \tag{21}$$

and

$$\begin{aligned} &-\left(\textstyle\int_{t-d(t)}^{t}\dot{z}(s)ds\right)^T V\left(\textstyle\int_{t-d(t)}^{t}\dot{z}(s)ds\right) \\ &+\begin{bmatrix} z(t) \\ z(t-d(t)) \end{bmatrix}^T \begin{bmatrix} V & -V \\ -V & V \end{bmatrix}\begin{bmatrix} z(t) \\ z(t-d(t)) \end{bmatrix} = 0. \end{aligned} \tag{22}$$

From (20)-(22), we have

$$\begin{aligned}\dot{V}(z(t)) \le\ & 2z^T(t)P\dot{z}(t) + 2\sum_{j=1}^{n} d_i f_j(z_j(t))\dot{z}_j(t) + \begin{bmatrix} z(t) \\ f(z(t)) \end{bmatrix}^T \begin{bmatrix} R & U \\ U^T & Q \end{bmatrix} \begin{bmatrix} z(t) \\ f(z(t)) \end{bmatrix} \\ & + x^T(t)K^T W K x(t) - f^T(x(t))K^T W K f(x(t)) \\ & -(1-\mu)[x^T(t-d(t))K^T W K x(t-d(t)) - f^T(x(t-d(t)))K^T W K f(t-d(t))] \\ & -(1-\mu)\begin{bmatrix} z(t-d(t)) \\ f(z(t-d(t))) \end{bmatrix}^T \begin{bmatrix} R & U \\ U^T & Q \end{bmatrix} \begin{bmatrix} z(t-d(t)) \\ f(z(t-d(t))) \end{bmatrix} \\ & + 2z^T(t)Y_1^T[z(t) - z(t-d(t)) - \textstyle\int_{t-d(t)}^{t} \dot{z}(s)ds] \\ & + 2z^T(t-d(t))Y_2^T[z(t) - z(t-d(t)) - \textstyle\int_{t-d(t)}^{t} \dot{z}(s)ds] \\ & + (2\textstyle\int_{t-d(t)}^{t} \dot{z}^T(s)ds)Y_3^T[z(t) - z(t-d(t)) - \textstyle\int_{t-d(t)}^{t} \dot{z}(s)ds] \\ & - \left(\textstyle\int_{t-d(t)}^{t} \dot{z}(s)ds\right)^T V \left(\textstyle\int_{t-d(t)}^{t} \dot{z}(s)ds\right) + \begin{bmatrix} z(t) \\ z(t-d(t)) \end{bmatrix}^T \begin{bmatrix} V & -V \\ -V & V \end{bmatrix} \begin{bmatrix} z(t) \\ z(t-d(t)) \end{bmatrix}. \end{aligned} \tag{23}$$

Now, considering the relationship in (6), we have

$$f_j(z_j)[f_j(z_j) - k_j z_j] \le 0,\ j = 1, 2, \cdots, n, \tag{24}$$

and

$$\begin{aligned} f_j(z_j(t-d(t)))[f_j(z_j(t-d(t))) - k_j z_j(t-d(t))] \le 0, \\ j = 1, 2, \cdots, n. \end{aligned} \tag{25}$$

Then, applying the S-procedure, we find that system (4) is asymptotically stable if there exist $T = \mathrm{diag}\{t_2, t_2, \cdots, t_n\} \ge 0$ and $S = \mathrm{diag}\{s_1, s_2, \cdots, s_n\} \ge 0$ such that

$$\begin{aligned} & \dot{V}(z(t)) - 2\sum_{j=1}^{n} t_j f_j(z_j(t))[f_j(z_j(t)) - k_j z_j(t)] \\ & -2\sum_{j=1}^{n} s_j f_j(z_j(t-d(t)))[f_j(z_j(t-d(t))) - k_j z_j(t-d(t))] \\ & = \xi^T(t)\Phi\xi(t) < 0, \end{aligned} \tag{26}$$

for all $\xi(t) \neq 0$, where

$$\xi(t) = [z^T(t),\ z^T(t-d(t)),\ f^T(z(t)),\ f^T(t-d(t)),\ \textstyle\int_{t-d(t)}^{t} \dot{z}^T(s)ds].$$

This completes the proof.

Remark 1. Unlike the Lyapunov-Krasovskii functional in [19], the one given above contains not only the integrals of term of states and of a cross product term, but also a integral of term of the neuron activation function. The advantage of this is that Theorem 1 can lead to better stability results for DCNNs.

Now, let

$$C = I, d(t) = d, \tag{27}$$

and the activation functions

$$g_i(x_i) = \frac{1}{2}(|x_i + 1| - |x_i - 1|), \tag{28}$$

where d is a given constant. Clearly, when invoking Theorem 1, we set $k_i = 1$, $i = 1, 2, \cdots, n$ and $\mu = 0$. The DCNNs of such case have been discussed by many researchers [3,4,5,6,7,16,19]. For this model, we have the following corollary.

Corollary 1. *Under conditions (27) and (28), the system (4) has a unique equilibrium point and that is global asymptotically stable if there exist four positive definite symmetric matrices P, R, Q, V and three definite diagonal matrices $T = diag\{t_1, t_2, \cdots, t_n\}$, $S = diag\{s_1, s_2, \cdots, s_n\}$, $D = diag\{d_1, d_2, \cdots, d_n\}$, $W = diag\{w_1, w_2, \cdots, w_n\}$ and four matrices U, Y_1, Y_2, Y_3 such that (10) and*

$$\Phi = \begin{bmatrix} \Phi_{11} & -Y_1^T + Y_2 - V & \Phi_{13} & PB & -Y_1^T + Y_3 \\ -Y_1 + Y_2^T - V & -R - Y_2 - Y_2^T + V & 0 & S - U & -Y_2^T - Y_3 \\ \Phi_{13}^T & 0 & \Phi_{33} & DB & 0 \\ B^T P & S - U^T & B^T D & -Q - 2S & 0 \\ -Y_1 + Y_3^T & -Y_2 - Y_3^T & 0 & 0 & -V - Y_3 - Y_3^T \end{bmatrix} < 0, \tag{29}$$

hold, where $\Phi_{11} = -2P + R + Y_1 + Y_1^T + V + \mu K^T W K$, $\Phi_{13} = PA - D + T + U$, $\Phi_{33} = A^T D + DA + Q - 2T - \mu K^T W K$.

Now, we show that the conditions given in [16], [18] and [19] can be considered as the special cases of the results obtained in Theorem 1 and Corollary 5.

Theorem 2 ([16]). *Under conditions (2) and (5), the system (4) has a unique equilibrium point and that is global asymptotically stable if there exist $P = P^T > 0$, $Q = Q^T > 0$ and diagonal matrix $D > 0$ such that the following condition holds:*

$$\begin{bmatrix} -PC - CP & PA & PB \\ A^T P & \Psi & DB \\ B^T P & B^T D & -(1-\mu)Q \end{bmatrix} < 0, \tag{30}$$

where $\Psi = DA + A^T D + Q - 2DK^{-1}C$, $K = \mathrm{diag}\{k_1, k_2, \cdots, k_n\}$.

In fact, letting $W = U = S = Y_1 = Y_2 = Y_3 = 0$, $T = K^{-1}CD$ and $R = V = \varepsilon I$ with sufficient small scalar $\varepsilon > 0$ in (9) and (10) yields the condition (30).

Theorem 3 ([16]). *Under conditions (27) and (28), the system (4) has a unique equilibrium point and that is global asymptotically stable if there exist $P = P^T > 0$, $Q = Q^T > 0$ and diagonal matrix $D > 0$ such that the following condition holds:*

$$\begin{bmatrix} -2P & PA & PB \\ A^T P & DA + A^T D + Q - 2D & DB \\ B^T P & B^T D & -Q \end{bmatrix} < 0, \tag{31}$$

Using the same method, from Corollary 5 we can obtain the condition (31).

Theorem 4 ([18]). *Under conditions (2) and (5),the system (4) ($C = I$) is global asymptotically stable if there exist three positive definite symmetric matrices P, R, Q and three definite diagonal matrices $T = diag\{t_1, t_2, \cdots, t_n\}$, $S = diag\{s_1, s_2, \cdots, s_n\}$, $D = diag\{d_1, d_2, \cdots, d_n\}$ such that the following LMIs hold:*

$$\begin{bmatrix} \Phi_{11} & 0 & \Phi_{13} & PB \\ 0 & -(1-\mu)R & 0 & KS \\ \Phi_{13}^T & 0 & \Phi_{33} & DB \\ B^TP & SK & B^TD & -(1-\mu)Q-2S \end{bmatrix} < 0, \tag{32}$$

where $\Phi_{11} = -2P + R$, $\Phi_{13} = PA - D + KT$, $\Phi_{33} = A^TD + DA + Q - 2T$, $K = diag\{k_1, k_2, \cdots, k_n\}$.

It is obviously that the condition of Theorem 4 is recovered by setting $U = Y_1 = Y_2 = Y_3 = 0$ and $V = \varepsilon I$ with sufficient small scalar $\varepsilon > 0$ in Theorem 1 ($C = I$).

Theorem 5. *Under conditions (27) and (28), the system (4) has a unique equilibrium point and that is global asymptotically stable if there exist four positive definite symmetric matrices P, R, Q, V and three definite diagonal matrices $T = diag\{t_1, t_2, \cdots, t_n\}$, $S = diag\{s_1, s_2, \cdots, s_n\}$, $D = diag\{d_1, d_2, \cdots, d_n\}$ and four matrices U, Y_1, Y_2, Y_3 such that (10) and*

$$\Phi = \begin{bmatrix} \Phi_{11} & -Y_1^T + Y_2 - V & \Phi_{13} & PB & -Y_1^T + Y_3 \\ -Y_1 + Y_2^T - V & -R - Y_2 - Y_2^T + V & 0 & S - U & -Y_2^T - Y_3 \\ \Phi_{13}^T & 0 & \Phi_{33} & DB & 0 \\ B^TP & S - U^T & B^TD & -Q - 2S & 0 \\ -Y_1 + Y_3^T & -Y_2 - Y_3^T & 0 & 0 & -V - Y_3 - Y_3^T \end{bmatrix} < 0, \tag{33}$$

hold, where $\Phi_{11} = -2P + R + Y_1 + Y_1^T + V$, $\Phi_{13} = PA - D + T + U$, $\Phi_{33} = A^TD + DA + Q - 2T$.

Theorem 5 is recovered by setting $W = 0$ in Corollary 1.

Remark 2. The results obtained by Shen [19] generalized and improved those given in [3,4,5,6,7,14,16,17]. Based on the above analysis, these results can be considered as a special case of our results. Therefore, our results is more general and less conservative.

4 Numerical Examples

In this section, the validity of the new criteria is demonstrated by two examples which are borrowed from [18,19] for the purpose of comparison.

Example 1. Consider the second-order DCNNs (4) with the following parameters:

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, A = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}, \ B = \begin{bmatrix} 0.5 & 0.5 \\ 1 & 0 \end{bmatrix}.$$

For a time-varying delay, we calculate the upper bounds on k_2 with $k_1 = 1.2$. The results are listed in Table 1 along with the results obtained by the methods in this paper, in [16,17,18] and in [19], respectively.

Table 1. Upper bounds on k_2 for $k_1 = 1.2$ and various μ

μ	0.8	0.9	1.0
[16] [17] [18]	infeasible	infeasible	infeasible
[19]	0.46	0.33	infeasible
Theorem 1	0.46	0.33	0.11

Example 2. Consider the second-order DCNN (4) with the following parameters:

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, A = \begin{bmatrix} 0.5 & 0.5 \\ -1 & -0.5 \end{bmatrix}, \ B = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0 \end{bmatrix}.$$

For a time-varying delay, we calculate the upper bounds on k_2 with $k_1 = 0.12$ by the methods in [18], [16], [17], [19] and our method, respectively. The results are listed in Table 2.

Table 2. Upper bounds on k_2 for $k_1 = 0.12$ with $\mu = 1$

	[16,17,18]	Theorem 1 in [19]	Theorem 1
$\max k_2$	infeasible	infeasible	0.06

Clearly, our method produces much less conservative results, thus demonstrating its validity.

5 Conclusion

This paper has investigated the problem of global asymptotic stability for DCNNs with time-varying delays. By constructing a new Lyapunov-Krasovskii functional with a integral of term of neuron activation function, new LMI-based sufficient conditions for global asymptotic stability are derived. Examples have been provided to demonstrate the validity of the proposed results.

Acknowledgements. This work was supported by the National Science Foundation of China (No. 60773190), the National Science Foundation of Hubei Province (No. 2008CDZ046), the Scientific Innovation Team Project of Hubei Provincial Department of Education (T200809).

References

1. Chua, L., Yang, L.: Cellular Neural Networks: Theory. IEEE Trans. Circuits Syst. 35, 1257–1272 (1988)
2. Cao, J., Zhou, D.: Stability Analysis of Delayed Cellular Neural Nwtworks. Neural Netw. 11, 1601–1605 (1998)

3. Arik, S., Tavsanoglu, V.: On the Global Asymptotic Stability of Delayed Cellular Neural Networks. IEEE Trans. Circuits Syst. I. 47, 571–574 (2000)
4. Liao, T., Wang, F.: Global Stability of Cellular Neural Networks with Time Delay. IEEE Trans. on Neural Netw. 11, 1481–1484 (2000)
5. Cao, J.: Global Stability Conditions for Delayed CNNs. IEEE Trans Circuits Syst. I. 48, 1330–1333 (2001)
6. Arik, S.: An Improved Global Stability Result for Delayed Cellular Neural Networks. IEEE Trans. Circuits Syst. I 49, 1211–1214 (2002)
7. Arik, S.: An Analysis of Global Asymptotic Stability of Delayed Cellular Neural Networks. IEEE Trans. Neural Netw. 13, 1239–1242 (2002)
8. Ensari, T., Arik, S.: Global Stability Analysis of Neural Networks with Multiple Time Varying Delays. IEEE Trans. Automat. Control 50, 1781–1785 (2005)
9. Xu, S., Lam, J., Ho, D., Zou, Y.: Novel Global Asymptotic Stability Criteria for Delayed Cellular Neural Networks. IEEE Trans. Circuits Syst. II 52, 349–353 (2005)
10. Liao, X., Chen, G., Sanchez, E.N.: Delay-Dependent Exponential Stability Analysis of Delayed Neural Networks: an LMI Approach. Neural Netw. 15, 855–866 (2002)
11. Yucel, E., Arik, S.: New Exponential Stability Results for Delayed Neural Networks with Time Varying Delays. Phys. D: Nonlinear Phenomena 191, 314–322 (2004)
12. He, Y., Wu, M., She, J.H.: Delay-Dependent Exponential Stability of Delayed Neural Networks with Time-Varying Delay. IEEE Trans. Circuit Syst. II 53, 553–557 (2006)
13. Boyd, S., El Ghaoui, L., Feron, E., Balakrishnan, V.: Linear Matrix Inequalities in System and Control Theory. SIAM, Philadlephia (1994)
14. Singh, V.: A Generalized Lmi-Based Approach to the Global Asymptotic Stability of Delayed Cellular Neural Networks. IEEE Trans. Neural Netw. 15, 223–225 (2004)
15. Zhang, H., Li, C., Liao, X.: A Note on the Robust Stability of Neural Networks with Time Delay. Chaos, Solitons and Fractals 25, 357–360 (2005)
16. Cao, J., Ho, D.: A General Framework for Global Asymptotic Stability of Delayed Neural Networks Based on LMI Approach. Chaos, Solitons and Fractals 24, 1317–1329 (2005)
17. Singh, V.: Simplified LMI Condition for Global Asymptotic Stability of Delayed Neural Networks. Chaos, Solitons and Fractals 29, 470–473 (2006)
18. He, Y., Wu, M., She, J.: An Improved Global Asymptotic Stability Criterion for Delayed Cellular Neural Networks. IEEE Trans. Neural Netw. 17, 250–252 (2006)
19. Shen, Y.: LMI-based Stability Criteria with Auxiliary Matrices for Delayed Recurrent Neural Networks. IEEE Transaction on Circuits System II 55, 811–815 (2008)

Multi-sensor Optimal H_∞ Fusion Filters for a Class of Nonlinear Intelligent Systems with Time Delays

Meiqin Liu[1,2], Meikang Qiu[2], and Senlin Zhang[1]

[1] College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China
[2] Department of Electrical Engineering, University of New Orleans, New Orleans, LA 70148, USA
liumeiqin@zju.edu.cn

Abstract. This paper proposes a nonlinear system model, which is composed of a linear time-delay dynamic system and a bounded static nonlinear operator. Base on the H_∞ performance analysis of this nonlinear model, H_∞ fusion filter is designed for this model with multiple sensors to guarantee the asymptotic stability of the fusion error system and reduce the effect of the noise signals on the filtering error to a lowest level. The parameters of the filter are obtained by solving the eigenvalue problem (EVP). Some delayed (or non-delayed) intelligent systems composed of neural networks or Takagi and Sugeno (T-S) fuzzy models can be transformed into this nonlinear model, then the multi-sensor optimal H_∞ fusion filters for them are designed.

Keywords: Nonlinear intelligent system; H_∞ fusion filter; Time-delay.

1 Introduction

The H_∞ filtering problem is to design an estimator to estimate the unknown state combination via output measurement, which guarantees the L_2 gain (from the external disturbance to the estimation error) less than a prescribed level [1-6]. In contrast with the well-known Kalman filter, one of the main advantages of H_∞ filtering is that, it is not necessary to know exactly the statistical properties of the external disturbance, but only assumes the external disturbance to have bounded energy. The H_∞ filtering technique has been found useful in certain applications. One of such applications is reported in [6] for seismic signal deconvolution. Various approaches, such as the algebraic Riccati equations, interpolation, linear matrix inequality (LMI) and so on, to H_∞ filter design have been successfully proposed and many results on this topic have been reported in the literature [1-6].

As modern industrial systems are more and more complex, the observation systems based on a single sensor are often unable to meet the needs of the information acquisition. In contrast, multi-sensor system has inherent redundancy, may obtain more observation information, and will continue running in low performance while some part of sensors can not work. It expands the space and time domains of information coverage, collects more data in a unit time, and can be achieved performance requirement in a short time. Therefore, the study of multi-sensor fusion filtering

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 357–365, 2009.

method for the nonlinear systems with uncertainty models and non-Gaussian noise inputs, is very important in the area of data fusion. In recent years, there are a lot of research results about the design of multi-sensor robust filter for uncertain linear systems, such as H_2 filtering [7], H_∞ filtering [8] and guaranteed cost filtering [9]. The study on estimating the states of the complex nonlinear dynamic systems (e.g. networked control system with a variety of models and controllers), is a prerequisite of guaranteeing safety and economy of the system, and is also the basis of many applications (such as process monitoring, fault diagnosis and process optimization, etc.). However, the research on the multi-sensor filtering of nonlinear systems is little, since complicated nonlinear terms make it difficult to find a fusion method. If the nonlinear terms of some systems, such as neural network systems, fuzzy control systems and hybrid intelligent systems consisting of them, have certain characteristics (e.g. sector bound), some useful fusion methods can be developed. There does not seem to be much (if any) study on this.

In this paper, we propose a dynamic system model with unified nonlinear operators. This model can represent not only the nonlinear systems which consist of neural networks or Takagi and Sugeno (T-S) fuzzy models, but also linear systems; not only time-delayed systems, but also non-time-delayed systems. By virtue of Lyapunov stability theory and the dissipative theory (mainly L_2 gain analysis), we can use the LMI approach to design multi-sensor H_∞ fusion filters for the dynamic systems with unified nonlinear operators.

Notation. The superscript "T" stands for matrix transposition. $l_2[0, \infty)$ is the space of square integrable vectors. I denotes identity matrix of appropriate order. $\|x\|$ denotes the Euclid norm of the vector x. $*$ denotes the symmetric parts. diag$\{\cdots\}$ denotes the block diagonal matrix. The notations $X>Y$ and $X\geq Y$, where X and Y are matrices of same dimensions, mean that the matrix $X-Y$ is positive definite and positive semi-definite, respectively. If $X\in\Re^p$ and $Y\in\Re^q$, $C(X; Y)$ denotes the space of all continuous functions mapping $\Re^p \to \Re^q$.

2 Problem Formulation

We consider the following stochastic nonlinear system with time delays:

$$\begin{cases} \dot{x}(t) = Ax(t) + A_d x(t-\tau) + B_p \phi(\xi(t)) + B_w w(t), \\ \xi(t) = C_q x(t) + C_{qd} x(t-\tau) + D_p \phi(\xi(t)) + D_{qw} w(t), \end{cases} \tag{1}$$

with the initial condition function $x(t)=\varpi(t)$, $\forall t\in[-\tau, 0]$, where $x(t)\in\Re^n$ is the system state, $A\in\Re^{n\times n}$, $A_d\in\Re^{n\times n}$, $B_p\in\Re^{n\times L}$, $B_w\in\Re^{n\times m}$, $C_q\in\Re^{L\times n}$, $C_{qd}\in\Re^{L\times n}$, $D_p\in\Re^{L\times L}$, and $D_{qw}\in\Re^{L\times m}$ are the corresponding state-space matrices, $\xi\in\Re^L$ is the input of nonlinear function ϕ, $\phi\in C(\Re^L; \Re^L)$ is nonlinear continuous function satisfying $\phi(0)=0$, $w(t)\in\Re^m$ is stochastic process noise which belongs to $l_2[0, \infty)$, $L\in N$ is the number of nonlinear functions, $\tau\in\Re$ is the time delay, $\varpi(t)$ is the given continuous function on $[-\tau, 0]$.

Remark 1. While $w(t)=0$, the nonlinear model (1) unifies linear systems, several well-known intelligent systems including dynamic neural networks or fuzzy models with

or without time delays, and Lur'e systems. Ref. [10]-[12] illustrate that some intelligent systems are special examples of (1).

Here, we assume the system (1) has h sensors:

$$y_i(t) = C_i x(t) + D_i v_i(t), \qquad i = 1, 2, \cdots, h, \tag{2}$$

where $y_1(t) \in \Re^{p_1}$, $y_2(t) \in \Re^{p_2}$, …, and $y_h(t) \in \Re^{p_h}$ are the measurement outputs, $v_1(t) \in \Re^{s_1}$, $v_2(t) \in \Re^{s_2}$, …, and $v_h(t) \in \Re^{s_h}$, are the measurement noise which belong to $l_2[0, \infty)$. $C_1 \in \Re^{p_1 \times n}$, $C_2 \in \Re^{p_2 \times n}$, …, $C_h \in \Re^{p_h \times n}$, $D_1 \in \Re^{p_1 \times s_1}$, $D_2 \in \Re^{p_2 \times s_2}$, …, and $D_h \in \Re^{p_h \times s_h}$ are constant matrices.

The estimated signals are the combination of the system states described as follows:

$$z(t) = C_z x(t), \tag{3}$$

where $z(t) \in \Re^r$ is the unmeasurable estimated signal, $C_z \in \Re^{r \times n}$ is constant matrix.

For convenience, the measurement model (2) can be denoted as the following augmented measurement equation:

$$y(t) = Cx(t) + Dv(t), \tag{4}$$

where $y(t)=[y_1^{\mathrm{T}}(t), y_2^{\mathrm{T}}(t), \ldots, y_h^{\mathrm{T}}(t)]^{\mathrm{T}}$, $v(t)=[v_1^{\mathrm{T}}(t), v_2^{\mathrm{T}}(t), \ldots, v_h^{\mathrm{T}}(t)]^{\mathrm{T}}$, $C=[C_1^{\mathrm{T}}, C_2^{\mathrm{T}}, \ldots, C_h^{\mathrm{T}}]^{\mathrm{T}}$, $D=\mathrm{diag}\{D_1, D_2, \ldots, D_h\}$. Combining Eq.(1) with Eqs. (3), (4), we have

$$\begin{cases} \dot{x}(t) = Ax(t) + A_d x(t-\tau) + B_p \phi(\xi(t)) + B_w w(t), \\ \xi(t) = C_q x(t) + C_{qd} x(t-\tau) + D_p \phi(\xi(t)) + D_{qw} w(t), \\ y(t) = Cx(t) + Dv(t), \\ z(t) = C_z x(t). \end{cases} \tag{5}$$

For Eq. (5), we construct the following Luenberger-like estimator for $z(t)$:

$$\begin{cases} \dot{\hat{x}}(t) = A\hat{x}(t) + A_d \hat{x}(t-\tau) + B_p \phi(\hat{\xi}(t)) + K[y(t) - C\hat{x}(t)], \\ \hat{\xi}(t) = C_q \hat{x}(t) + C_{qd} \hat{x}(t-\tau) + D_p \phi(\hat{\xi}(t)), \\ \hat{z}(t) = C_z \hat{x}(t). \end{cases} \tag{6}$$

where $K \in \Re^{n \times \sum_{j=1}^{h} p_j}$ is filter gain to be determined to meet certain performance criteria, $\hat{x}(t)$ and $\hat{z}(t)$ denote the estimates of $x(t)$ and $z(t)$, respectively. Defining the error vector $e(t)=x(t)-\hat{x}(t)$, we have the following dynamic equations which $e(t)$ satisfies.

$$\begin{cases} \dot{e}(t) = (A - KC)e(t) + A_d e(t-\tau) + B_p f(\tilde{\xi}(t)) + B_w w(t) - KDv(t), \\ \tilde{\xi}(t) = \xi(t) - \hat{\xi}(t) = C_q e(t) + C_{qd} e(t-\tau) + D_p f(\tilde{\xi}(t)) + D_{qw} w(t), \\ \tilde{z}(t) = z(t) - \hat{z}(t) = C_z e(t), \end{cases} \tag{7}$$

where $\tilde{z}(t)$ is fusion error signal, and $f(\tilde{\xi}(t)) = \phi(\xi(t)) - \phi(\hat{\xi}(t))$.

Definition 1. If there exists a positive scalar γ such that

$$J(w(t),v(t)) = \int_0^\infty [\tilde{z}^{\mathrm{T}}(t)\tilde{z}(t) - \gamma^2(w^{\mathrm{T}}(t)w(t) + v^{\mathrm{T}}(t)v(t))] < 0\,, \tag{8}$$

for any nonzero $w(t)\in l_2[0, \infty)$, $v(t)\in l_2[0, \infty)$, with the initial state $x(t)=\varpi(t)=0$, $\forall t\in[-\tau, 0]$, and the system (7) is asymptotically stable when $w(t)=0$ and $v(t)=0$, then the L_2 gain of the system (7) does not exceed γ, that is, the system (6) is an H_∞ fusion estimator for $z(t)$. If we find a minimal positive γ to satisfy the above conditions, the system (6) is an optimal H_∞ fusion estimator for $z(t)$.

3 H_∞ Fusion Filter Design

In this paper, we assume that the nonlinear functions in (1) are monotonically non-decreasing and globally Lipschitz. That is, there exist non-negative scalar q_i and positive scalar u_i such that

$$q_i \le \frac{\phi_i(\sigma)-\phi_i(\delta)}{\sigma-\delta} \le u_i\,,\ \forall\sigma,\ \delta\in\Re,\ u_i>q_i\ge 0,\ i=1,\ \ldots,\ L. \tag{9}$$

Then the nonlinear functions $f_i(\tilde{\xi}_i(t))$ (i=1, …, L) satisfy the following sector-bounded conditions:

$$q_i \le \frac{f_i(\tilde{\xi}_i(t))}{\tilde{\xi}_i(t)} \le u_i\,,\ \text{i.e. } [f_i(\tilde{\xi}_i(t)) - q_i\tilde{\xi}_i(t)]\cdot[f_i(\tilde{\xi}_i(t)) - u_i\tilde{\xi}_i(t)] \le 0\,. \tag{10}$$

Theorem 1. There exists a multi-sensor optimal H_∞ fusion filter (6) such that the system (7) is globally asymptotically stable when $w(t)=0$, $v(t)=0$, and the upper bound on the L_2 gain of the system (7) is minimal provided that there exist symmetric positive definite matrices P and Γ, diagonal semi-positive definite matrix Λ, a matrix S, and a positive scalar γ that satisfy the following EVP:

Minimize γ^2, (11)

Subject to

$$\begin{bmatrix}
\begin{pmatrix}(PA-SC)^T+PA\\ -SC+C_z^{\mathrm{T}}C_z+\Gamma\end{pmatrix} & PA_d & PB_p+C_q^T(Q+U)\Lambda & PB_w & -SD\\
* & -\Gamma & C_{qd}^T(Q+U)\Lambda & 0 & 0\\
* & * & \begin{pmatrix}D_p^T(Q+U)\Lambda\\ +\Lambda(Q+U)D_p\\ -2\Lambda\end{pmatrix} & \Lambda(Q+U)D_{qw} & 0\\
* & * & * & -\gamma^2 I & 0\\
* & * & * & * & -\gamma^2 I
\end{bmatrix} < 0, \tag{12}$$

where Q=diag(q_1, q_2, …, q_L), U=diag(u_1, u_2, …, u_L). Furthermore, the gain of desired H_∞ fusion filter (6) can be determined by:

$$K = P^{-1}S\ . \tag{13}$$

Proof. Considering (12) (13), we have

$$G=\begin{bmatrix} \begin{pmatrix}(A-KC)^T P\\ +P(A-KC)\\ +C_z^{\mathrm{T}}C_z+\Gamma\end{pmatrix} & PA_d & PB_p+C_q^T(Q+U)\Lambda & PB_w & -PKD \\ * & -\Gamma & C_{qd}^T(Q+U)\Lambda & 0 & 0 \\ * & * & \begin{pmatrix}D_p^T(Q+U)\Lambda\\ +\Lambda(Q+U)D_p\\ -2\Lambda\end{pmatrix} & \Lambda(Q+U)D_{qw} & 0 \\ * & * & * & -\gamma^2 I & 0 \\ * & * & * & * & -\gamma^2 I \end{bmatrix} < 0. \tag{14}$$

For simplicity, we denote $e(t)$ as e, $e(t-\tau)$ as e_τ, $\tilde{z}(t)$ as $\tilde{z}$, $w(t)$ as w, $v(t)$ as v, $\tilde{\xi}_i(t)$ as $\tilde{\xi}_i$, $f_i(\tilde{\xi}_i(t))$ as f_i, and $f(\tilde{\xi}(t))$ as f. Firstly, we consider the system (7) with $w(t)$=0 and $v(t)$=0; that is

$$\begin{cases}\dot{e}(t) = (A-KC)e(t) + A_d e(t-\tau) + B_p f(\tilde{\xi}(t)),\\ \tilde{\xi}(t) = C_q e(t) + C_{qd} e(t-\tau) + D_p f(\tilde{\xi}(t)).\end{cases} \tag{15}$$

Since $e(t)$=0 and $\tilde{\xi}(t)$=0 are solutions to (15), there exists at least one equilibrium point located at the origin, i.e. e_{eq}=0, $\tilde{\xi}_{eq}$=0. For the system (15), we adopt the following Lyapunov-Krasovskii functional:

$$V(t) = e^{\mathrm{T}}Pe + \int_{-\tau}^{0} e^{\mathrm{T}}(t+\theta)\Gamma e(t+\theta)d\theta\ , \tag{16}$$

where P>0, Γ>0. Thus, $\forall e\neq 0$, $V(t)$>0, and $V(t)$=0 iff e=0. The derivative of $V(t)$ along the solution of system (15) is:

$$\begin{aligned}\dot{V}(t) &= \dot{e}^{\mathrm{T}}Pe + e^{\mathrm{T}}P\dot{e} + e^{\mathrm{T}}\Gamma e - e_\tau^{\mathrm{T}}\Gamma e_\tau \\ &= \left[(A-KC)e + A_d e_\tau + B_p f\right]^{\mathrm{T}} Pe + e^{\mathrm{T}}P\left[(A-KC)e + A_d e_\tau + B_p f\right] + e^{\mathrm{T}}\Gamma e - e_\tau^{\mathrm{T}}\Gamma e_\tau \\ &= \begin{bmatrix} e\\ e_\tau\\ f\end{bmatrix}^{\mathrm{T}} \underbrace{\begin{bmatrix}(A-KC)^{\mathrm{T}}P + P(A-KC) + \Gamma & PA_d & PB_p\\ A_d^{\mathrm{T}}P & -\Gamma & 0\\ B_p^{\mathrm{T}}P & 0 & 0\end{bmatrix}}_{R_0} \begin{bmatrix} e\\ e_\tau\\ f\end{bmatrix}.\end{aligned} \tag{17}$$

For the system (7) under zero initial condition, $J(w, v)$ in Eq. (8) is equivalent to

$$\begin{aligned}
J(w,v) &= \int_0^\infty [\tilde{z}^{\mathrm{T}}\tilde{z} - \gamma^2(w^{\mathrm{T}}w + v^{\mathrm{T}}v)]dt \\
&= \int_0^\infty [\tilde{z}^{\mathrm{T}}\tilde{z} - \gamma^2(w^{\mathrm{T}}w + v^{\mathrm{T}}v) + \dot{V}(t)]dt - V(\infty) \\
&\le \int_0^\infty [\tilde{z}^{\mathrm{T}}\tilde{z} - \gamma^2(w^{\mathrm{T}}w + v^{\mathrm{T}}v) + \dot{V}(t)]dt \\
&= \int_0^\infty \Big[\big((A-KC)e + A_d e_\tau + B_p f + B_w w - KDv\big)^{\mathrm{T}} Pe \\
&\qquad + e^{\mathrm{T}} P\big((A-KC)e + A_d e_\tau + B_p f + B_w w - KDv\big) \\
&\qquad + e^{\mathrm{T}}\Gamma e - e_\tau^{\mathrm{T}}\Gamma e_\tau + e^{\mathrm{T}}C_z^{\mathrm{T}}C_z e - \gamma^2(w^{\mathrm{T}}w + v^{\mathrm{T}}v)\Big] dt \\
&= \int_0^\infty \begin{bmatrix} e \\ e_\tau \\ f \\ w \\ v \end{bmatrix}^{\mathrm{T}} \underbrace{\begin{bmatrix} \begin{pmatrix} (A-KC)^{\mathrm{T}}P \\ +P(A-KC) \\ +C_z^{\mathrm{T}}C_z + \Gamma \end{pmatrix} & PA_d & PB_p & PB_w & -PKD \\ A_d^{\mathrm{T}}P & -\Gamma & 0 & 0 & 0 \\ B_p^{\mathrm{T}}P & 0 & 0 & 0 & 0 \\ B_w^{\mathrm{T}}P & 0 & 0 & -\gamma^2 I & 0 \\ -(KD)^{\mathrm{T}}P & 0 & 0 & 0 & -\gamma^2 I \end{bmatrix}}_{\bar{R}_0} \begin{bmatrix} e \\ e_\tau \\ f \\ w \\ v \end{bmatrix} dt .
\end{aligned} \tag{18}$$

The sector-bounded conditions (10) can be rewritten as follows:

$$f_i^2 - (q_i + u_i) f_i \tilde{\xi}_i \le -q_i u_i \tilde{\xi}_i^2 \le 0 ,$$

which is equivalent to:

$$2\phi_i^2 - 2(q_i + u_i) f_i (C_{q,i} e + C_{qd,i} e_\tau + D_{p,i} f + D_{qw,i} w) \le 0 , \tag{19}$$

where $C_{q,i}$ is the i^{th} row of matrix C_q, $C_{qd,i}$ is the i^{th} row of matrix C_{qd}, $D_{p,i}$ is the i^{th} row of matrix D_p, $D_{qw,i}$ is the i^{th} row of D_{qw}. We rewrite (19) in matrix notation as follows:

$$\begin{bmatrix} e \\ e_\tau \\ f_1 \\ \vdots \\ f_{i-1} \\ f_i \\ f_{i+1} \\ \vdots \\ f_L \\ w \\ v \end{bmatrix}^{\mathrm{T}} \underbrace{\begin{bmatrix} 0 & 0 & -C_{q,i}^{\mathrm{T}}(q_i + u_i) & 0 & 0 \\ 0 & 0 & -C_{qd,i}^{\mathrm{T}}(q_i + u_i) & 0 & 0 \\ -(q_i + u_i)C_{q,i} & -(q_i + u_i)C_{qd,i} & M & -(q_i + u_i)D_{qw,i} & 0 \\ 0 & 0 & -D_{qw,i}^{\mathrm{T}}(q_i + u_i) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\bar{R}_i} \begin{bmatrix} e \\ e_\tau \\ f_1 \\ \vdots \\ f_{i-1} \\ f_i \\ f_{i+1} \\ \vdots \\ f_L \\ w \\ v \end{bmatrix}, \tag{20}$$

where

$$M=\begin{bmatrix} 0 & \cdots & 0 & -d_{p,(i,1)}(q_i+u_i) & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & -d_{p,(i,i-1)}(q_i+u_i) & 0 & \cdots & 0 \\ -(q_i+u_i)d_{p,(i,1)} & \cdots & -(q_i+u_i)d_{p,(i,i-1)} & 2-2(q_i+u_i)d_{p,(i,i)} & -(q_i+u_i)d_{p,(i,i+1)} & \cdots & -(q_i+u_i)d_{p,(i,L)} \\ 0 & \cdots & 0 & -d_{p,(i,i+1)}(q_i+u_i) & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & -d_{p,(i,L)}(q_i+u_i) & 0 & \cdots & 0 \end{bmatrix},$$

and $d_{p,(i,j)}$ is the entry of the matrix D_p at the i^{th} row and j^{th} column. By virtue of S-procedure [13], if there exist $\lambda_i \geq 0$ (i=1, …, L), such that the following inequality holds

$$\bar{R}_0 - \sum_{i=1}^{L} \lambda_i R_i = G < 0, \tag{21}$$

then $\bar{R}_0 < 0$, where Λ=diag$\{\lambda_1, \lambda_2, \ldots, \lambda_L\}$, and $\Lambda \geq 0$. We can obtain that for any $[e^{\mathrm{T}}\ e_\tau^{\mathrm{T}}\ f^{\mathrm{T}}\ w^{\mathrm{T}}\ v^{\mathrm{T}}]^{\mathrm{T}} \neq 0$, $J(w,v)<0$, for any nonzero $w \in l_2[0, \infty)$ and $v \in l_2[0, \infty)$. By the well-known Schur complement [13], $\bar{R}_0 < 0$ is equivalent to

$$\begin{bmatrix} (A-KC)^{\mathrm{T}}P + P(A-KC) + \Gamma & PA_d & PB_p & PB_w & -PKD & C_z^{\mathrm{T}} \\ A_d^{\mathrm{T}}P & -\Gamma & 0 & 0 & 0 & 0 \\ B_p^{\mathrm{T}}P & 0 & 0 & 0 & 0 & 0 \\ B_w^{\mathrm{T}}P & 0 & 0 & -\gamma^2 I & 0 & 0 \\ -(KD)^{\mathrm{T}}P & 0 & 0 & 0 & -\gamma^2 I & 0 \\ C_z & 0 & 0 & 0 & 0 & -I \end{bmatrix} < 0. \tag{22}$$

From (22), we have

$$R_0 = \begin{bmatrix} (A-KC)^{\mathrm{T}}P + P(A-KC) + \Gamma & PA_d & PB_p \\ A_d^{\mathrm{T}}P & -\Gamma & 0 \\ B_p^{\mathrm{T}}P & 0 & 0 \end{bmatrix} < 0. \tag{23}$$

So the system (7) with w=0 and v=0 [i.e. the system (15)] is globally asymptotically stable.

We hope that γ is minimal such that the system (7) can reject the external disturbance as strong as possible. It requires solving the eigenvalue problem (EVP) (11)-(12), which is a convex optimization problem and can be solved by using the MATLAB LMI Control Toolbox [14]. This completes the proof.

Remark 2. For the convenience of the application of designed filters in the engineering practice, it is necessary to limit the magnitude of filter gain K, which is equivalent to restricting the norm of P and S in some ranges, that is,

$$\begin{cases} \left\| P^{-1} \right\| < \delta, \\ \left\| S \right\| < \sigma, \end{cases} \tag{24}$$

where δ and σ are positive scalars, which are well chosen according to the design requirement in practical systems. By virtue of the well-known Schur complement formula [13], the constraints (24) is equivalent to

$$\begin{bmatrix} -\delta P & I \\ I & -\delta P \end{bmatrix} < 0, \tag{25}$$

$$\begin{bmatrix} -\sigma I & S^{\mathrm{T}} \\ S & -\sigma I \end{bmatrix} < 0. \tag{26}$$

Remark 3. Most literature [15] [16], which investigating H_∞ filter problem, assumes that the estimated system (1) is asymptomatically stable in the absence of process noise w and measurement noise v. However, in most case, the information about the systems' stability can not be known. Here, we need not this assumption. On the other hand, while constructing filter structure, we take full advantage of the information provided by the system (1) and measurement from sensors (2) (or (4)), such as A, A_d, B_p, C_q, C_{qd}, D_p, C , D and C_z, therefore, only one parameter K in the H_∞ Filter (6) should be determined.

4 Conclusion

In this paper, we have studied an H_∞ fusion filtering algorithm for a class of time-delayed systems with unified nonlinear operators, such as recurrent neural networks, fuzzy dynamic systems, and Lur'e systems. Central to our design are the introduction of the unified model, which interconnects a linear time-delayed dynamic system with static nonlinear operators, and the transformation of the nonlinear system to this unified model. An optimal H_∞ fusion filter has been designed for this unified nonlinear model such that L_2 gain of the system (7) is minimized. The filter structure is based on the system (1), and the only one filter parameter should be determined. The resulting design equations are a set of LMIs which can be solved by the MATLAB LMI Control Toolbox [14]. The fusion method is applicable to not only stable nonlinear systems, but also unstable complex systems.

Acknowledgment. This work was supported in part by the National Natural Science Foundation of China under Grant 60504024 and 60874050, in part by the Zhejiang Provincial Natural Science Foundation of China under Grant Y106010, and in part by the Specialized Research Fund for the Doctoral Program of Higher Education (SRFDP), China under Grant 20060335022. This work was also supported by the "151 Talent Project " of Zhejiang Province (No. 05-3-1013 and No. 06-2-034).

References

1. Tseng, C.S.: Robust Fuzzy Filter Design for a Class of Nonlinear Stochastic Systems. IEEE Trans. on Fuzzy systems 15, 261–274 (2007)
2. Yung, C.F., Li, Y.F., Sheu, H.T.: H_∞Filtering and Solution Bound for Non-linear Systems. Int. J. Control 74, 565–570 (2001)
3. Burl, J.B.: H_∞ Estimation for Nonlinear Systems. IEEE Signal Processing Letters 5, 199–202 (1998)
4. Xu, S.: Robust Filtering for a Class of Discrete-Time Uncertain Nonlinear Systems with State Delay. IEEE Trans. on Circuits and Systems-I 49, 1853–1859 (2002)
5. Gao, H., Meng, X., Chen, T.: New Design of Robust $H_($ Filter for 2-D System. IEEE Signal Processing Letters 18, 217–220 (2008)
6. Shaked, U.: H_∞ Minimum Error State Estimation of Linear Stationary Processes. IEEE Trans. Automat. Control 35, 554–558 (1990)
7. Sun, H., Liu, R., Wen, C.: Design and Performance Analysis of Robust Fusion Filters for a Class of Continuous Uncertain Dynamic Systems. In: Proc. Of the 26th Chinese Control Conference, pp. 301–305. Beihang University Press, Beijing (2007)
8. Li, Q., Wang, H., Zhang, W., Liu, X.: H_∞ Fusion Filter Design in Multi-sensor Fusion System with State Time-Delays. In: IEEEInternational Conference on Automation and Logistics, pp. 2784–2789. IEEE Press, New York (2007)
9. Wang, J., Xue, A.: Robust Guaranteed Cost Data Fusion Approach for Uncertain Systems. In: Fifth World Congress on Intelligent Control and Automation (WCICA), pp. 3064–3066. Zhejiang University Press, Hangzhou (2004)
10. Liu, M.: Delayed Standard Neural Network Models for Control Systems. IEEE Trans. on Neural Networks 18, 1376–1391 (2007)
11. Liu, M., Wang, H.F.: A Novel Stabilizing Control for Neural Nonlinear Systems with Time Delays by State and Dynamic Output Feedback. International Journal of Control, Automation, and Systems 6, 24–34 (2008)
12. Liu, M., Zhang, J.: Exponential Synchronization of General Chaotic Delayed Neural Networks via Hybrid Feedback. Journal of Zhejiang University SCIENCE A 9, 262–270 (2008)
13. Boyd, S.P., Ghaoui, L.E., Feron, E., Balakrishnan, V.: Linear Matrix Inequalities in System and Control Theory. SIAM, Philadelphia (1994)
14. Gahinet, P., Nemirovski, A., Laub, A.J., Chilali, M.: LMI Control Toolbox- for Use with Matlab. The MATH Works, Inc., Natick (1995)
15. Gao, H., Wang, C.: A Delay-Dependent Approach to Robust H_∞ Filtering for Uncertain Discrete-Time State-Delayed Systems. IEEE Trans. on Signal Processing 52, 1631–1640 (2004)
16. Wang, Z., Yang, F., Ho, D.W.C., Liu, X.: Robust H_∞ Filtering for Stochastic Time-Delay Systems with Missing Measurements. IEEE Trans. on Signal Processing 54, 2579–2587 (2006)

H_∞ Synchronization of General Discrete-Time Chaotic Neural Networks with Time Delays

Meiqin Liu[1,2], Senlin Zhang[1], and Meikang Qiu[2]

[1] College of Electrical Engineering, Zhejiang University,
Hangzhou, 310027, China
[2] Department of Electrical Engineering, University of New Orleans,
New Orleans, LA 70148, USA
{liumeiqin,slzhang}@zju.edu.cn
mqiu@uno.edu

Abstract. This paper investigates robust H_∞ synchronization problem of general discrete-time time-delayed chaotic neural networks with external disturbance. Based on Lyapunov stability theory and H_∞ control concept, time-delayed state feedback controllers are established to not only guarantee exponential stable synchronization between two general chaotic neural networks with time delays, but also reduce the effect of external disturbance on synchronization error to a minimal H_∞ norm constraint. The control design problem is shown to be a linear matrix inequality (LMI) standard problem which can be easily solved by various convex optimization algorithms to determine the optimal H_∞ synchronization control law.

1 Introduction

Since Aihara firstly introduced chaotic neural network model to simulate the chaotic behavior of biological neurons in 1990 [1], chaotic neural networks have been successfully applied in combinational optimization [2], associative memory [3], secure communication [4], chemical biology [5] and so on. Based on the drive-response (master-slave) concept proposed by Pecora and Carroll [6], research on the synchronization of chaotic neural networks has broadened considerably in the last few years. A wide variety of approaches have been proposed for the synchronization and control of chaotic neural networks with or without delays, which include linear and nonlinear feedback control, adaptive design control, impulsive control method, and invariant manifold method, among many others (see [7-10] and references cited therein).

In real physical systems, some noise or disturbances always exist that may cause instability and poor performance. Therefore, the effect of the noises or disturbances must be also reduced in synchronization process for chaotic systems. In this regards, Suykens et al. [11] firstly adopted the H_∞ control concept to reduce the effect of the disturbance for chaotic synchronization problem of chaotic Lur'e systems. Refs. [12] and [13] investigated the H_∞ synchronization problem for a general class of chaotic systems with external disturbance via dynamic feedback approach. On the other hand, since Mackey and Glass [14] first found chaos in time-delay system, there has been

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 366–374, 2009.

increasing interest in time-delay chaotic systems. The H_∞ synchronization problem for time-delayed chaotic systems is also investigated by some researchers [15].

However, to our best knowledge, the above aforementioned methods and many other existing synchronization methods are only applied to the continuous-time chaotic systems. There does not seem to be much (if any) study on the H_∞ synchronization for discrete-time delayed chaotic systems with external disturbance. It is well known that discrete-time systems play a very important role in digital signal analysis and processing. Especially, the discrete-time neural networks have been found intensive application, such as bidirectional associative memory, nonlinear output regulation and adaptive tracking etc. Therefore, here we will combine the H_∞ control concept and Lyapunov stability theory to investigate the optimal H_∞ synchronization problem for a general class of discrete-time time-delayed chaotic neural networks with external disturbance. We are inspired by the standard neural network model (SNNM) in [16] and put forward this general discrete-time chaotic neural network, which is the interconnection of a linear delayed dynamic system and a bounded static nonlinear operator. Most chaotic systems with time delays, such as Hopfield neural networks, cellular neural networks (CNNs), bidirectional associative memory (BAM) networks, recurrent multilayer perceptrons (RMLPs), and Cohen-Grossberg neural networks (CGNNs) etc, can be transformed into this general chaotic neural network to be H_∞ synchronization controller designed in a unified way. Time-delayed state feedback controller for the synchronization between two general discrete-time time-delayed chaotic neural networks is proposed. By the state feedback control scheme, the closed-loop error system is exponentially stable and the H_∞-norm from the disturbance to controlled output is reduced to a lowest level.

Notation. $\Re^n$ denotes n dimensional Euclidean space, and $\Re^{n\times m}$ is the set of all $n\times m$ real matrices. $l_2[0, \infty)$ is the space of square integrable vectors. I denotes the identity matrix of appropriate order. A^{T} means the transpose of the matrix A. diag$\{\cdots\}$denotes the block diagonal matrix. $*$ denotes the symmetric parts. $\lambda_M(A)$ and $\lambda_m(A)$ denote the maximal and minimal eigenvalue of a square matrix A, respectively. $\|x\|$ denotes the Euclid norm of the vector x. The notations $X>Y$ and $X\geq Y$, where X and Y are matrices of same dimensions, mean that the matrix $X-Y$ is positive definite and positive semi-definite, respectively. If $X\in\Re^p$ and $Y\in\Re^q$, $C(X; Y)$ denotes the space of all continuous functions mapping $\Re^p\rightarrow\Re^q$.

2 Problem Formulation

In this paper, we consider the following chaotic delayed neural network model [16]:

$$\begin{cases} x(k+1) = Ax(k) + A_d x(k-\tau) + B_p\phi(\xi(k)), \\ \xi(k) = C_q x(k) + C_{qd} x(k-\tau) + D_p\phi(\xi(k)), \\ z_x(k) = Cx(k), \end{cases} \tag{1}$$

with the initial condition function $x(k)=\varpi(k)$, $\forall k\in[-\tau, 0]$, where $x(k)\in\Re^n$ is the state vector associated with the neurons, $A\in\Re^{n\times n}$, $A_d\in\Re^{n\times n}$, $B_p\in\Re^{n\times L}$, $C_q\in\Re^{L\times n}$, $C_{qd}\in\Re^{L\times n}$,

$D_p \in \Re^{L\times L}$, and $C \in \Re^{l\times n}$ are the corresponding state-space matrices, $\phi \in C(\Re^L; \Re^L)$ is nonlinear activation function satisfying $\phi(0)=0$, $\xi \in \Re^L$ is the input vector of ϕ, $\tau \geq 1$ is the transmission delay, $\varpi(\cdot)$ is the given continuous function on $[-\tau, 0]$, $z_x(k) \in \Re^l$ is the output vector, and $L \in \Re$ is the total number of neurons in the hidden layers and output layer of the neural network.

In this paper, we assume that the activation functions in (1) are monotonically non-decreasing and globally Lipschitz. That is, there exist non-negative scalar q_i and positive scalar h_i such that

$$q_i \leq \frac{\phi_i(\mu)-\phi_i(\upsilon)}{\mu-\upsilon} \leq h_i,\ h_i > q_i \geq 0\ i=1, \ldots, L, \tag{2}$$

for all arbitrary $\mu,\ \upsilon \in \Re$.

The synchronization problem of system (1) is considered using the drive-response configuration [6]. This is, if the system (1) is regarded as the drive system, a suitable response system with control input should be constructed to synchronize the drive system. According to the above drive-response concept, unidirectionally coupled chaotic systems can be described by the following equations:

$$\begin{cases} y(k+1) = Ay(k) + A_d y(k-\tau) + B_p \phi(\zeta(k)) + u(k) + Dw(k), \\ \zeta(k) = C_q y(k) + C_{qd} y(k-\tau) + D_p \phi(\zeta(k)), \\ z_y(k) = Cy(k), \end{cases} \tag{3}$$

with the initial condition function $y(k)=\sigma(k)$, $\forall k \in [-\tau, 0]$, where $y(k) \in \Re^n$ is the state vector of response system, $D \in \Re^{n\times s}$ *is* constant matrices, $\sigma(\cdot)$ is the given continuous function on $[-\tau, 0]$, $w(k) \in \Re^s$ is external disturbance which belongs to $l_2[0, \infty)$, $z_y(k) \in \Re^l$ is the output of the response system, and $u(k) \in \Re^n$ is a unidirectionally coupled term, which is regarded as the control input and will be appropriately designed such that the specific control objective is achieved.

Now, we define the synchronization error signal $e(k)=y(k)-x(k)$, where $x(k)$ and $y(k)$ are the state variables of drive system (1) and response system (3), respectively. Therefore, the error dynamical system between (1) and (3) is given as follows:

$$\begin{cases} e(k+1) = Ae(k) + A_d e(k-\tau) + B_p \psi(\eta(k)) + u(k) + Dw(k), \\ \eta(k) = C_q e(k) + C_{qd} e(k-\tau) + D_p \psi(\eta(k)), \\ z_e(k) = Ce(k), \end{cases} \tag{4}$$

where $e(k) \in \Re^n$, $z_e(k)=z_y(k)-z_x(k)$, $\eta(k)=\zeta(k)-\xi(k)$, and $\psi(\eta(k))=\phi(\zeta(k))-\phi(\xi(k))=\phi(\eta(k)+\xi(k))-\phi(\xi(k))$, therefore $\psi(0)=0$. Since all the $\phi_i(\cdot)$ are globally Lipschitz, $\psi_i(\cdot)$ satisfy the sector conditions, i.e. , for each $i=1, \ldots, L$,

$$q_i \leq \psi_i(\eta_i(k))/\eta_i(k) \leq h_i \text{ or } [\psi_i(\eta_i(k))-q_i\eta_i(k)]\cdot[\psi_i(\eta_i(k))-h_i\eta_i(k)] \leq 0. \tag{5}$$

Next, in order to synchronize between drive system (1) and response one (3) in the sense of H_∞ theory, let us consider the following state and time-delay state feedback controller:

$$u(k) = K_1 e(k) + K_2 e(k-\tau), \tag{6}$$

where $K_1 \in \Re^{n\times n}$ and $K_2 \in \Re^{n\times n}$ are feedback gains to be scheduled. With the control law (6), the error dynamics can be expressed by the following form:

$$\begin{cases} e(k+1) = (A+K_1)e(k) + (A_d+K_2)e(k-\tau) + B_p\psi(\eta(k)) + Dw(k), \\ \eta(k) = C_q e(k) + C_{qd} e(k-\tau) + D_p\psi(\eta(k)), \\ z_e(k) = Ce(k). \end{cases} \tag{7}$$

Since $\psi(0)=0$, the system (7) admits a trivial solution $e(k)\equiv 0$ in the absence of external disturbance $w(k)$.

Before stating the main results, we first need the following definition.

Definition 1 (Exponential H_∞ synchronization). [12][13][15]**.** The drive system (1) and the response system (3) are said to be exponentially synchronized in H_∞ sense if

($\mathbf{H}_1$) there exist constants $\lambda(\alpha)\geq 1$ and $\alpha>0$ under $w(k)\equiv 0$ such that $\|x(k)-y(k)\| \leq \lambda(\alpha) \sup_{-\tau\leq i\leq 0} \|y(i)-x(i)\| \exp(-\alpha k)$, for any $k\geq 0$. Moreover, the constant α is defined as the exponential synchronization rate;

($\mathbf{H}_2$) the following condition holds under zero initial condition with a given positive constant γ:

$$J = \sum_{k=0}^{\infty} [z_e^T(k) z_e(k) - \gamma^2 w^{\mathrm{T}}(k) w(k)] < 0 \ (\text{i.e.} \sup_{w(k)\neq 0, w(k)\in l_2[0,\infty)} \frac{\|z_e(k)\|}{\|w(k)\|} < \gamma). \tag{8}$$

Then, the controller $u(k)$ is said to be the H_∞ synchronization controller with the disturbance attenuation γ. The parameter γ is called the H_∞-norm bound of the controller. If we find a minimal positive γ to satisfy the above conditions, the controller (6) is an optimal H_∞ synchronizer.

3 Main Results

Theorem 1. For given $\alpha>0$, if there exist positive definite matrices R and Γ, diagonal semi-positive definite matrix Σ, a positive scalar γ, and nonzero matrices Y_1 and Y_2, that satisfy the following eigenvalue problem (EVP):

Minimize γ^2, (9)

Subject to

$$\begin{bmatrix} -R & RA+Y_1 & RA_d+Y_2 & RB_p & RD \\ * & -\exp(-2\alpha)R + \Gamma + C^{\mathrm{T}}C & 0 & C_q^{\mathrm{T}}(Q+H)\Sigma & 0 \\ * & * & -\exp(-2\alpha\tau)\Gamma & C_{qd}^{\mathrm{T}}(Q+H)\Sigma & 0 \\ * & * & * & \begin{pmatrix} D_p^{\mathrm{T}}(Q+H)\Sigma - 2\Sigma \\ +\Sigma(Q+H)D_p \end{pmatrix} & 0 \\ * & * & * & * & -\gamma^2 I \end{bmatrix} < 0, \tag{10}$$

where Q=diag(q_1, q_2, …, q_L), H=diag(h_1, h_2, …, h_L), then the drive system (1) and the response system (3) can be synchronized with a prescribed exponential synchronization rate α, and H_∞-norm bound of the controller (6) does not exceed γ. Moreover, the feedback gains of optimal H_∞ controller (6) are obtained as $K_1=R^{-1}Y_1$ and $K_2=R^{-1}Y_2$.

Proof. Substituting $Y_1=RK_1$ and $Y_2=RK_2$ into (10), and using the well-known Schur complement [17], Ieq. (10) is equivalent to:

$$M=\begin{bmatrix} \begin{pmatrix}(A+K_1)^{\mathrm{T}}R(A+K_1)\\ -\exp(-2\alpha)R+\Gamma\\ +C^{\mathrm{T}}C\end{pmatrix} & (A+K_1)^{\mathrm{T}}R(A_d+K_2) & \begin{pmatrix}(A+K_1)^{\mathrm{T}}RB_p\\ +C_q^{\mathrm{T}}(Q+H)\Sigma\end{pmatrix} & (A+K_1)^{\mathrm{T}}RD \\ * & \begin{pmatrix}(A_d+K_2)^{\mathrm{T}}R(A_d+K_2)\\ -\exp(-2\alpha\tau)\Gamma\end{pmatrix} & \begin{pmatrix}(A_d+K_2)^{\mathrm{T}}RB_p\\ +C_{qd}^{\mathrm{T}}(Q+H)\Sigma\end{pmatrix} & (A_d+K_2)^{\mathrm{T}}RD \\ * & * & \begin{pmatrix}B_p^{\mathrm{T}}RB_p+D_p^{\mathrm{T}}(Q+H)\Sigma\\ +\Sigma(Q+H)D_p-2\Sigma\end{pmatrix} & B_p^{\mathrm{T}}RD \\ * & * & * & D^{\mathrm{T}}RD-\gamma^2 I \end{bmatrix}<0. \tag{11}$$

Firstly, we consider the system (7) with $w(k)$=0; that is

$$\begin{cases} e(k+1)=(A+K_1)e(k)+(A_d+K_2)e(k-\tau)+B_p\psi(\eta(k)),\\ \eta(k)=C_q e(k)+C_{qd}e(k-\tau)+D_p\psi(\eta(k)), \end{cases} \tag{12}$$

For the error dynamical system (12), we define a positive definite Lyapunov-Krasovskii functional as:

$$V(e(k))=\exp(2\alpha k)e^{\mathrm{T}}(k)Pe(k)+\sum_{i=k-\tau}^{k-1}\exp(2\alpha i)e^{\mathrm{T}}(i)\Gamma e(i)\ , \tag{13}$$

where $P=P^{\mathrm{T}}>0$, $\Gamma=\Gamma^{\mathrm{T}}>0$, and $\alpha>0$. Thus, $\forall e(k)\neq 0$, $V(e(k))>0$, and $V(e(k))=0$ iff $e(k)=0$. The difference of $V(e(k))$ along the solution to (12) is

$$\begin{aligned} \Delta V(e(k)) &= V(e(k+1))-V(e(k))\\ &= \exp(2\alpha k)\{\exp(2\alpha)[(A+K_1)e(k)+(A_d+K_2)e(k-\tau)+B_p\psi(\eta(k))]^{\mathrm{T}}\\ &\quad \times P[(A+K_1)e(k)+(A_d+K_2)e(k-\tau)+B_p\psi(\eta(k))]\\ &\quad -e^{\mathrm{T}}(k)Pe(k)+e^{\mathrm{T}}(k)\Gamma e(k)-\exp(-2\alpha\tau)e^{\mathrm{T}}(k-\tau)\Gamma e(k-\tau)\}\ . \end{aligned} \tag{14}$$

Let $\exp(2\alpha)P=R$, we can get $P=\exp(-2\alpha)R$. We rewrite the above formulation of $\Delta V(e(k))$ as:

$$\Delta V(e(k))=\exp(2\alpha k)\begin{bmatrix} e(k)\\ e(k-\tau)\\ \psi(\eta(k))\end{bmatrix}^{T} G_0 \begin{bmatrix} e(k)\\ e(k-\tau)\\ \psi(\eta(k))\end{bmatrix} \tag{15}$$

where

$$G_0 = \begin{bmatrix} \begin{pmatrix} (A+K_1)^{\mathrm{T}} R(A+K_1) \\ -\exp(-2\alpha)R + \Gamma \end{pmatrix} & (A+K_1)^{\mathrm{T}} R(A_d+K_2) & (A+K_1)^{\mathrm{T}} RB_p \\ * & (A_d+K_2)^{\mathrm{T}} R(A_d+K_2) - \exp(-2\alpha\tau)\Gamma & (A_d+K_2)^{\mathrm{T}} RB_p \\ * & * & B_p^{\mathrm{T}} RB_p \end{bmatrix}.$$

Next, for the system (7) under zero initial condition, J in Eq. (8) is equivalent to

$$\begin{aligned} J &= \sum_{k=0}^{\infty} [z_e^{\mathrm{T}}(k) z_e(k) - \gamma^2 w^{\mathrm{T}}(k) w(k)] \\ &= \sum_{k=0}^{\infty} [e^{\mathrm{T}}(k) C^{\mathrm{T}} Ce(k) - \gamma^2 w^{\mathrm{T}}(k) w(k) + \Delta V(e(k))] - V(e(\infty)) + V(e(0)) \\ &\le \sum_{k=0}^{\infty} [e^{\mathrm{T}}(k) C^{\mathrm{T}} Ce(k) - \gamma^2 w^{\mathrm{T}}(k) w(k) + \Delta V(e(k))] \\ &= \sum_{k=0}^{\infty} \exp(2\alpha k) \begin{bmatrix} e(k) \\ e(k-\tau) \\ \psi(\eta(k)) \\ w(k) \end{bmatrix}^T \bar{G}_0 \begin{bmatrix} e(k) \\ e(k-\tau) \\ \psi(\eta(k)) \\ w(k) \end{bmatrix}, \end{aligned} \tag{16}$$

where

$$\bar{G}_0 = \begin{bmatrix} \begin{pmatrix} (A+K_1)^{\mathrm{T}} R(A+K_1) \\ -\exp(-2\alpha)R + \Gamma \\ +C^{\mathrm{T}} C \end{pmatrix} & (A+K_1)^{\mathrm{T}} R(A_d+K_2) & (A+K_1)^{\mathrm{T}} RB_p & (A+K_1)^{\mathrm{T}} RD \\ * & \begin{pmatrix} (A_d+K_2)^{\mathrm{T}} R(A_d+K_2) \\ -\exp(-2\alpha\tau)\Gamma \end{pmatrix} & (A_d+K_2)^{\mathrm{T}} R\bar{B}_p & (A_d+K_2)^{\mathrm{T}} RD \\ * & * & B_p^{\mathrm{T}} RB_p & B_p^{\mathrm{T}} RD \\ * & * & * & D^{\mathrm{T}} RD - \gamma^2 I \end{bmatrix}.$$

The sector conditions (5) can be rewritten as follows:

$$\psi_i^2(\eta_i(k)) - \psi_i(\eta_i(k))(q_i+h_i)\eta_i(k) + q_i h_i \eta_i^2(k) \le 0, \tag{17}$$

Since $q_i h_i \eta_i^2(k) \ge 0$, we can obtain

$$\psi_i^2(\eta_i(k)) - \psi_i(\eta_i(k))(q_i+h_i)\eta_i(k) \le 0, \tag{18}$$

which is equivalent to:

$$\begin{aligned} &2\psi_i^2(\eta_i(k)) - 2\psi_i(\eta_i(k))(q_i+h_i)C_{q,i}e(k) - 2\psi_i(\eta_i(k))(q_i+h_i)C_{qd,i}e(k-\tau) \\ &\quad -2\psi_i(\eta_i(k))(q_i+h_i)D_{p,i}\psi(\eta(k)) \le 0, \end{aligned} \tag{19}$$

where $C_{q,i}$ denotes the i^{th} row of C_q, $C_{qd,i}$ denotes the i^{th} row of C_{qd}, $D_{p,i}$ denotes the i^{th} row of D_p. We rewrite (19) in matrix notation as follows:

$$\begin{aligned}&\left[e^{\mathrm{T}}(k)\quad e^{\mathrm{T}}(k-\tau)\quad \psi_1(\eta_1(k))\quad\cdots\quad \psi_{i-1}(\eta_{i-1}(k))\quad \psi_i(\eta_i(k))\quad \psi_{i+1}(\eta_{i+1}(k))\right.\\&\cdots\quad \psi_L(\eta_L(k))\quad w(k)\left]G_i\right[e^{\mathrm{T}}(k)\quad e^{\mathrm{T}}(k-\tau)\quad \psi_1(\eta_1(k))\quad\cdots\quad \psi_{i-1}(\eta_{i-1}(k))\\&\left.\psi_i(\eta_i(k))\quad \psi_{i+1}(\eta_{i+1}(k))\quad\cdots\quad \psi_L(\eta_L(k))\quad w(k)\right]^{\mathrm{T}}\le 0,\end{aligned}\tag{20}$$

where

$$G_i=\begin{bmatrix}0&0&0&\cdots&0&-C_{q,i}^{\mathrm{T}}s_i&0&\cdots&0&0\\0&0&0&\cdots&0&-C_{qd,i}^{\mathrm{T}}s_i&0&\cdots&0&0\\0&0&0&\cdots&0&-d_{p,i,1}s_i&0&\cdots&0&0\\\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots\\0&0&0&\cdots&0&-d_{p,i,i-1}s_i&0&\cdots&0&0\\-s_iC_{q,i}&-s_iC_{qd,i}&-s_id_{p,i,1}&\cdots&-s_id_{p,i,i-1}&2-2s_id_{p,i,i}&s_id_{p,i,i+1}&\cdots&s_id_{p,i,L}&0\\0&0&0&\cdots&0&d_{p,i,i+1}s_i&0&\cdots&0&0\\\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots\\0&0&0&\cdots&0&d_{p,i,L}s_i&0&\cdots&0&0\\0&0&0&\cdots&0&0&0&\cdots&0&0\end{bmatrix},$$

where $s_i=q_i+h_i$, $d_{p,i,j}$ is the entry of the matrix D_p at the i^{th} row and j^{th} column. By the S-procedure [17] and (11), if there exist $\varepsilon_i\ge 0$ ($i=1, \ldots, L$), such that the following inequality holds:

$$\begin{aligned}&\bar{G}_0-\sum_{i=1}^{L}\varepsilon_iG_i=\bar{G}_0\\&-\begin{bmatrix}0&0&-C_q^{\mathrm{T}}(Q+H)\Sigma&0\\0&0&-C_{qd}^{\mathrm{T}}(Q+H)\Sigma&0\\-\Sigma(Q+H)C_q&-\Sigma(Q+H)C_{qd}&2\Sigma-D_p^{\mathrm{T}}(Q+H)\Sigma-\Sigma(Q+H)D_p&0\\0&0&0&0\end{bmatrix}\\&=M<0,\end{aligned}\tag{21}$$

where $\Sigma=\mathrm{diag}\{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_L\}$, and $\Sigma\ge 0$, then $\bar{G}_0<0$. we can obtain that for any $[e^{\mathrm{T}}(k)\ e^{\mathrm{T}}(k-\tau)\ \psi^{\mathrm{T}}(\eta(k))\ w^{\mathrm{T}}(k)]^{\mathrm{T}}\ne 0$, $J<0$, for any nonzero $w\in l_2[0, \infty)$. By the well-known Schur complement [17], $\bar{G}_0<0$ is equivalent to

$$\begin{bmatrix}\begin{pmatrix}(A+K_1)^{\mathrm{T}}R(A+K_1)\\-\exp(-2\alpha)R+\Gamma\end{pmatrix}&(A+K_1)^{\mathrm{T}}R(A_d+K_2)&(A+K_1)^{\mathrm{T}}RB_p&(A+K_1)^{\mathrm{T}}RD&C^{\mathrm{T}}\\*&\begin{pmatrix}(A_d+K_2)^{\mathrm{T}}R(A_d+K_2)\\-\exp(-2\alpha\tau)\Gamma\end{pmatrix}&(A_d+K_2)^{\mathrm{T}}RB_p&(A_d+K_2)^{\mathrm{T}}RD&0\\*&*&B_p^{\mathrm{T}}RB_p&B_p^{\mathrm{T}}RD&0\\*&*&*&D^{\mathrm{T}}RD-\gamma^2I&0\\*&*&*&*&-I\end{bmatrix}<0,\tag{22}$$

From (22), we have $G_0<0$, that is, $\Delta V(e(k))\le 0$, therefore, $V(e(k))\le V(e(0))$. However,

$$\begin{aligned} V(e(0)) &= e(0)^{\mathrm{T}} Pe(0) + \sum_{i=-\tau}^{-1} \exp(2\alpha i) e^{\mathrm{T}}(i) \Gamma e(i) \\ &\le \lambda_M(P) \|e(0)\|^2 + \lambda_M(\Gamma) \|\Theta\|^2 \sum_{i=-\tau}^{-1} \exp(2\alpha i) \\ &\le \lambda_M(P) \|\Theta\|^2 + \lambda_M(\Gamma) \|\Theta\|^2 \frac{\exp(-2\alpha) - \exp(-2\alpha(\tau+1))}{1-\exp(-2\alpha)} \\ &= \left[\lambda_M(P) + \lambda_M(\Gamma) \frac{\exp(-2\alpha) - \exp(-2\alpha(\tau+1))}{1-\exp(-2\alpha)} \right] \|\Theta\|^2, \end{aligned} \tag{23}$$

where $\|\Theta\| = \sup_{-\tau \le i \le 0} \|e(i)\|$, and $V(e(k)) \ge \exp(2\alpha k) e^{\mathrm{T}}(k) P e(k) \ge \exp(2\alpha k) \lambda_m(P) \|e(k)\|^2$, therefore the convergence rates of $e(k)$ are

$$\|e(k)\| \le \sqrt{\frac{\lambda_M(P)}{\lambda_m(P)} + \frac{\lambda_M(\Gamma)}{\lambda_m(P)} \cdot \frac{\exp(-2\alpha) - \exp(-2\alpha(\tau+1))}{1-\exp(-2\alpha)}} \|\Theta\| \exp(-\alpha k). \tag{24}$$

From **Definition 1**, it concludes that the drive system (1) and the response system (3) are exponentially synchronized with an exponential synchronization rate α.

We hope that γ is minimal such that the system (7) can reject the external disturbance as strong as possible. It requires solving the eigenvalue problem (EVP) (9) (10), which is a convex optimization problem and can be solved by using MATLAB's LMI Control Toolbox [18]. We thus complete the proof.

4 Conclusion

In this paper, we provide a general discrete-time chaotic neural network model to unify several well-known dynamic neural networks with delays. Utilizing time-delay feedback control and LMI techniques, we have proposed a criterion to design optimal H_∞ synchronization controller of this general chaotic neural network model. Solving the EVP by using MATLAB's LMI Control Toolbox [18], we have obtained the optimal H_∞-norm bound of the controller and the feedback gain matrices of optimal controller in the response network, with which the drive system and the response system can be exponentially synchronized in the prescribed convergent rate, and reject the external disturbance as strong as possible. In addition, the design approach can be easily extended to synthesize synchronization controllers for any discrete-time chaotic systems as long as their equations can be transformed into the general models (1).

Acknowledgment. This work was supported in part by the National Natural Science Foundation of China under Grant 60504024 and 60874050, in part by the Zhejiang Provincial Natural Science Foundation of China under Grant Y106010, and in part by the Specialized Research Fund for the Doctoral Program of Higher Education (SRFDP), China under Grant 20060335022. This work was also supported by the "151 Talent Project " of Zhejiang Province (No. 05-3-1013 and No. 06-2-034).

References

1. Aihara, K., Takabe, T., Toyoda, M.: Chaotic Neural Networks. Physics Letters A 144, 333–340 (1990)
2. Kwok, T., Smith, K.A.: Experimental Analysis of Chaotic Neural Network Models for Combinatorial Optimization under a Unifying Framework. Neural Network 13, 731–744 (2000)
3. Tan, Z., Ali, M.K.: Associative Memory Using Synchronization in a Chaotic Neural Network. International Journal of Modern Physics C 12, 19–29 (2001)
4. Milanovic, V., Zaghloul, M.E.: Synchronization of Chaotic Neural Networks and Applications to Communications. International Journal of Bifurcation and Chaos 6, 2571–2585 (1996)
5. Han, S.K., Kurrer, C., Kuramoto, Y.: Dephasing and Bursting in Coupled Neural Oscillators. Physical Review Letters 75, 3190–3193 (1995)
6. Pecora, L.M., Carroll, T.L.: Synchronization in Chaotic Systems. Physical Review Letters 64, 821–824 (1990)
7. Liu, M., Zhang, J.: Exponential Synchronization of General Chaotic Delayed Neural Networks via Hybrid Feedback. Journal of Zhejiang University Science A 9, 262–270 (2008)
8. Lu, H., Leeuwen, C.V.: Synchronization of Chaotic Neural Networks via Output or State Coupling. Chaos, Solitons and Fractals 30, 166–176 (2006)
9. Zhou, J., Chen, T., Xiang, L.: Robust Synchronization of Delayed Neural Networks Based on Adaptive Control and Parameters Identification. Chaos, Solitons and Fractals 27, 905–913 (2006)
10. Lu, J., Cao, J.: Synchronization-Based Approach for Parameters Identification in Delayed Chaotic Neural Network. Physica A 382, 672–682 (2007)
11. Suykens, J.A.K., Curran, P.F., Vandewalle, J., Chua, L.O.: Robust Nonlinear $H_{(}$ Synchronization of Chaotic Lur'e Systems. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications 44, 891–904 (1997)
12. Hou, Y.Y., Liao, T.-L., Yan, J.-J.: H_{∞} Synchronization of Chaotic Systems Using Output Feedback Control Design. Physica A 379, 81–89 (2007)
13. Lee, S.M., Ji, D.H., Park, J.H., Won, S.C.: H_{∞} Synchronization of Chaotic Systems via Dynamic Feedback Approach. Physics letters A 372, 4012–4905 (2008)
14. Mackey, M., Glass, L.: Oscillation and Chaos in Physiological Control Systems. Science 197, 287–289 (1977)
15. Park, J.H., Ji, D.H., Won, S.C., Lee, S.M.: H_{∞} Synchronization of Time-Delayed Chaotic Systems. Applied Mathematics and Computation 204, 170–177 (2008)
16. Liu, M.: Discrete-Time Delayed Standard Neural Network Model and Its Application. Science in China: Series F Information Sciences 49, 137–154 (2006)
17. Boyd, S.P., Ghaoui, L.E., Feron, E., Balakrishnan, V.: Linear Matrix Inequalities in System and Control Theory. SIAM, Philadelphia (1994)
18. Gahinet, P., Nemirovski, A., Laub, A.J., Chilali, M.: LMI Control Toolbox- for Use with Matlab. The MATH Works, Inc., Natick (1995)

Stability of Hopfield Neural Networks with Time-Varying Delay

Huimin Xiao

Information School, Henan University of Finance and Economics,
Zhengzhou 450002, P.R. China
xiaohm@hnufe.edu.cn

Abstract. In this paper, the stability of Hopfield Neural Networks with Time-Varying Delay is investigated by means of Lyapunov functions and the generalized Halanay delay differential inequality. The obtained results are the algebraic criteria entirely and are obviously practical

Keywords: Stability; time-delay; neural networks; Lyapunov functions.

1 Introduction

It is known that the stability of Hopfield continuous neural networks

$$C_i \frac{du_i}{dt} = -\frac{1}{R_i} u_i + \sum_{j=1}^{n} T_{ij} g_j(u_j(t-\tau_j(t))) + I_i, \quad i = 1,2,...,n$$

Hopfield continuous neural networks with constant time-delay

$$C_i \frac{du_i}{dt} = -\frac{1}{R_i} u_i + \sum_{j=1}^{n} T_{ij} g_j(u_j(t-\tau)) + I_i, \quad i = 1,2,...,n$$

is investigated by a lot of people, many results of stability are obtained. But the stability of Hopfield continuous neural networks with time-varying delay is rarely studied. In reality，the neural networks with time-varying delay is universal. In many cases, we only known the time-delay bound, but don't know the real value of time-delay. In dynamical process, the absolute constant time-delay is rarely existed, and is only ideal approximation. Therefore, the investigation of stability of the Hopfield neural networks with time-varying delay is of the theoretical meaning and realistic meaning.

Consider the Hopfield neural networks with time-varying delay

$$C_i \frac{du_i}{dt} = -\frac{1}{R_i} u_i + \sum_{j=1}^{n} T_{ij} g_j(u_j(t-\tau_j(t))) + I_i \quad , \quad i = 1,2,...,n \tag{1}$$

where C_i is capacitor, R_i is resistor, and $1/R_i$ represents the time constant of the change rate for the *ith* neuron potential; I_i is current, and is the constant external input to the network; T_{ij} is the synaptic efficacy of the *jth* neuron potential transmitted to the *ith* neuron. $g_j \in C^1$ is monotonous increasing function satisfying

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 375–382, 2009.

$g_j(0)=0$, and $g_j(u_j)u_j>0$ for $u_j\neq 0$, $j=1,2,\cdots,n$. $g_j'(u_j)$ is bounded, time-delay $\tau_j(t)$ is the t function satisfying $0\le\tau_j(t)\le\tau$ (τ is constant).

Suppose that $u^*=col(u_1^*,u_2^*,...,u_n^*)$ is the equilibrium of system (1). Let

$$x=col(x_1,x_2,...,x_n)=col(u_1-u_1^*,u_2-u_2^*,...,u_n-u_n^*),$$

$$f_i(x_i)=g_i(x_i+u_i^*)-g_i(u_i^*),$$

then the system (1) may be changed into

$$C_i\frac{dx_i}{dt}=\sum_{j=1}^{n}T_{ij}f_j(x_j(t-\tau_j(t)))-\frac{1}{R_i}x_i(t),\ i=1,2,...,n. \tag{2}$$

Therefore, the equilibrium $x=0$ of system (2) is equal to the equilibrium $u=u^*$ of system (1) in stability..

2 Main Results

Lemma 1. Generalized Halanay Time-delay Differential Inequality
Suppose that $a>0,b>0$ are constants, and $a>b$. The function $x(t)$ is one variable continuous function, and nonnegative. As $t\ge t_0$, the following inequality is satisfied

$$D^+x(t)\le -ax(t)+b\bar{x}(t) \tag{3}$$

where $\bar{x}(t)\overset{\Delta}{=}\sup_{t-\tau\le s\le t}\{x(s)\}$, $\tau\ge 0$ is a constant, then we have

$$x(t)\le\bar{x}(t_0)\exp[-\lambda(t-t_0)] \qquad \text{as } t\ge t_0$$

where λ is only one positive root of the following transcendental equation

$$\lambda=a-be^{\lambda\tau} \tag{4}$$

Proof:
① We proof that the following transcendental equation (4) has only one positive solution.
② Let $y(t)=\bar{x}(t_0)\exp[-\lambda(t-t_0)]$, then

$$\frac{dy(t)}{dt}\le -ay(t)+by(t-\tau)\quad,\quad t\in[t_0,+\infty) \tag{5}$$

Suppose that k is a any constant and $k>1$, then

$$x(t)<ky(t),\quad t\in[t_0-\tau,t_0] \tag{6}$$

Now suppose that there is a $t\in(t_0,+\infty)$, such that $x(t)=ky(t)$. then from continuous of $x(t),y(t)$, we have that there exists $t_1\in(t_0,+\infty)$ satisfying the following

$$x(t)<ky(t)\quad,\quad t\in[t_0-\tau,t_1)\quad,\quad x(t_1)=ky(t_1) \tag{7}$$

On the other hand, from (3), we have

$$D^+x(t_1)\le -ax(t_1)+b\bar{x}(t_1)<-aky(t_1)+bky(t_1-\tau)=k\frac{dy(t_1)}{dt} \tag{8}$$

so we have

$$D^+x(t_1) < k\frac{dy(t_1)}{dt} \tag{9}$$

(7) is contradiction to (9). Then for any $t \in [t_0,+\infty)$, we have

$$x(t) < ky(t)$$

Again, letting $k \to 1$,then

$$x(t) < y(t) = \bar{x}(t_0)\exp[-\lambda(t-t_0)] \tag{10}$$

Theorem 1. Suppose that $u = u^*$ is the equilibrium state of system (1) , and satisfies the following conditions

(1)$0 \le g_j'(u_j) \le L_j < \infty$

(2) $a \overset{\Delta}{=} \min\limits_{1\le j\le n}\left(\dfrac{1}{R_jC_j}\right) > \max\limits_{1\le j\le n}\sum\limits_{i=1}^{n}\left|\dfrac{T_{ij}L_j}{C_i}\right| \overset{\Delta}{=} b$

Then the equilibrium state $u = u^*$ of system (1) is global exponentially stable, and the unique positive solution λ of characteristic equation (4) is regard as Lyapunov exponent of system (1).

Proof: For system (1), we choose Lyapunov function $W(x) = \sum\limits_{i=1}^{n}|x_i|$. The Dini derivative of $W(x)$ is obtained along the system (2),

$$\begin{aligned} D^+W(x)\big|_{(2)} &= \frac{dx_i}{dt}\operatorname{sgn} x_i \le \sum_{i=1}^{n}\frac{-1}{C_iR_i}|x_i| + \sum_{i=1}^{n}\sum_{j=1}^{n}\left|\frac{T_{ij}L_j}{C_i}\right|\cdot|x_j(t-\tau_j(t))| \\ &\le -\min_{1\le i\le n}\frac{1}{C_iR_i}\sum_{j=1}^{n}|x_j| + \sum_{j=1}^{n}\sum_{i=1}^{n}\left|\frac{T_{ij}L_i}{C_i}\right|\cdot|\bar{x}_j(t)| \\ &\le -\min_{1\le i\le n}\frac{1}{C_iR_i}\sum_{j=1}^{n}|x_j| + \left(\max_{1\le j\le n}\sum_{i=1}^{n}\left|\frac{T_{ij}L_i}{C_i}\right|\right)\cdot\sum_{j=1}^{n}|\bar{x}_j(t)| \\ &\overset{\Delta}{=} -aW(x) + bW(\bar{x}) = -aW(x) + b\overline{W}(x) \end{aligned} \tag{11}$$

By Lemma 1,we have：

$$\sum_{i=t}^{n}|x_i(t,t_0,x_0)| = W(t) \le \overline{W}(t_0)e^{-\lambda(t-t_0)} \tag{12}$$

where λ is unique positive root of the transcendental equation (4). From (12), the equilibrium state $u = u^*$ of system (1) is global exponentially stable.

Theorem 2. Suppose that $u = u^*$ is the equilibrium state of system (1) , and satisfies the following conditions

(1) $0 \le g_j'(u_j) \le L_j < \infty$;

(2) $a \overset{\Delta}{=} \min_{1\le i\le n}(\frac{1}{R_i C_i}) > \lambda$.

Let $H = \begin{pmatrix} 0 & B \\ B^{\mathrm{T}} & 0 \end{pmatrix}_{2\pi\times 2n}$, $B = \left(|\frac{T_{ij}L_j}{C_i}|\right)_{\pi\times n}$, λ is maximum eigenvalue of matrix H, B^{T} is the transition matrix of B, then the equilibrium state $u = u^*$ of system (1) is global exponentially stable.

Let the unique positive solution of $\mu = a - \frac{\lambda}{2} + \frac{\lambda}{2}e^{\mu\tau}$ is μ, then $\frac{\mu}{2}$ is regarded as Lyapunov exponent of system (1).

Proof: For system (1), we choose Lyapunov function $W(x) = \frac{1}{2}\sum_{i=1}^{n} x_i^2$, then

$$\begin{aligned}
\frac{dW(t)}{dt}\Big|_{(1)} &= -\sum_{i=1}^{n}\frac{1}{C_i R_i}x_i^2 + \sum_{j=1}^{n}\sum_{i=1}^{n} x_i T_{ij} f_j(x_j(t-\tau_j(t))) \\
&\le -\left(\min_{1\le i\le n}\frac{1}{C_i R_i}\right)\sum_{j=1}^{n} x_j^2 + \frac{1}{2}\begin{pmatrix} |x| \\ |\bar{x}| \end{pmatrix}^T \begin{pmatrix} 0 & B \\ B^T & 0 \end{pmatrix} \\
&\le -aW(x) + \frac{\lambda}{2}W(x) + \frac{\lambda}{2}\overline{W}(x) = (-a+\frac{\lambda}{2})W(x) + \frac{\lambda}{2}\overline{W}(x) \\
&= -\bar{a}W(x) + b\overline{W}(x) \qquad (13)
\end{aligned}$$

where $\bar{a} = a - \frac{\lambda}{2}$, $b = \frac{\lambda}{2}$, $\overline{W}(x) = \sum_{i=1}^{n} |\bar{x}_i(t)|^2$. From $a > \lambda$, we have $\bar{a} > b$. By Lemma 1, we have

$$W(x) \le W(x(0))\ e^{-\mu(t-t^0)} \qquad (14)$$

where μ is unique positive solution of the transcendental equation: $\mu = \bar{a} + be^{\mu\tau}$, i.e.

$$\sqrt{\sum_{i=1}^{n} x_i^2(t, t^0, x^0)} \le \sqrt{\sum_{i=1}^{n} x_{0i}^2}\, e^{-\frac{\mu}{2}(t-t_0)} \qquad (15)$$

The proof of Theorem 2 is completed.

Definition 1. $G(t, x, y)$ is said to be H_n class function, if the following conditions are satisfied:

(1) $\forall t \in I, \forall x \in R^n, \forall y^{(1)}, y^{(2)} \in C^n$, as $y^{(1)} \le y^{(2)}$ (i.e. $y_i^{(1)} \le y_i^{(2)}$, $i = 1,2,\cdots,n$), we have

$$G(t, x, y^{(1)}) \le G(t, x, y^{(2)}) \qquad (16)$$

(2) $\forall t \in I, \forall y \in C^n, \forall x^{(1)}, x^{(2)} \in R^n$. As $x^{(1)} \le x^{(2)}$, for some I, we have $x_i^{(1)} = x_i^{(2)}$, and for these i, we have

$$g_i(t, x^{(1)}, y) \le g_i(t, x^{(2)}, y) \tag{17}$$

where $f \in C^n = C[[t-\tau, t], R^n]$, $G(t, x, y) \in C[R_+ \times R^n \times C^n, R^n]$.

Lemma 2. 设 $x(t), y(t)$ are n-dimension vector functions, and $\bar{x}(t) = \sup_{\xi \in [t-\tau, t]} x(\xi)$, $\bar{y}(t) = \sup_{\xi \in [t-\tau, t]} y(\xi)$, $G(t, x, y) \in H_n$, the following conditions are satisfied

(1) $x(\theta) < y(\theta), \quad \theta \in [-\tau, 0]$

(2) $D^+ y_i(t) > g_i(t, y(t), \bar{y}(t)), \quad i = 1,2,\cdots,n, \quad t \ge 0$

$D^+ x_i(t) \le g_i(t, x(t), \bar{x}(t)), \quad i = 1,2,\cdots,n, \quad t \ge 0$

then $x(t) < y(t)$, as $t > 0$.

where $G(t, x(t), \bar{x}(t)) = col(g_1(t, x(t), \bar{x}(t)), \cdots, g_n(t, x(t), \bar{x}(t)))$

Proof(Proof by contradiction): Suppose that there exists a constant $\eta > 0$ and some i, such that $x_i(\eta) = y_i(\eta)$.

Let $Z = \{\eta | x_i(\eta) = y_i(\eta)\}$, it is obvious that $Z \ne \phi$, therefore there exists a η_0, such that $\eta_0 = \inf_{\eta \in Z} \eta$. From condition (1), we have $x(\theta) < y(\theta), as\ \theta \in [-\tau, 0]$. So we know that $\eta_0 > 0$, $x(\eta_0) \le y(\eta_0)$, and $\bar{x}(\eta_0) \le \bar{y}(\eta_0)$. Therefore, from given condition (2), there exists a integer j, $1 \le j \le n$, such that

$$D^+ x_j(\eta_0) \le g_j(\eta_0, x(\eta_0), \bar{x}(\eta_0)) \le g_j(\eta_0, y(\eta_0), \bar{y}(\eta_0)) < D^+ y_j(\eta_0) \tag{18}$$

But as $t \in (0, \eta_0)$, we have $x(t) < y(t)$, *i.e.* $x_j(t) < y_j(t)$, but $x_j(\eta_0) = y_j(\eta_0)$. So we have

$$D^+ x_j(\eta_0) \ge D^+ y_j(\eta_0) \tag{19}$$

This is contradictive to (18). The proof of Lemma 2 is completed.

Lemma 3. Suppose that the following conditions are satisfied

$$(1)\ D^+ x_j(t) \le (\sum_{j=1}^{n} a_{ij} x_j(t) + \sum_{j=1}^{n} b_{ij} \bar{x}_j(t)), \quad i = 1,2,\cdots,n \tag{20}$$

where $a_{ij} \ge 0, \quad i \ne j, b_{ij} \ge 0, \quad i, j = 1,2,\cdots,n, \quad \sum_{i=1}^{n} \bar{x}_i(t_0) > 0$

(2) $M = -(a_{ij} + b_{ij})_{n \times n}$ is M-matrix.

Then there exists constants $\gamma_i > 0, \alpha > 0$, such that the solution of differential inequality (20) satisfies estimation formula.

$$x_i(t) \le \gamma_i [\sum_{j=1}^{n} \bar{x}_j(t_0)] e^{-\alpha(t-t_0)} \tag{21}$$

Proof: Let $g_i(t,x(t),\bar{x}(t)) = \sum_{j=1}^{n} a_{ij} x_j(\delta)) + \sum_{j=1}^{m} b_{ij} \bar{x}_j(t)$

$$G(t,x(t),\bar{x}(t)) = col(g_1(t,x(t),\bar{x}(t)),\cdots,g_n(t,x(t),\bar{x}(t))) \in H_n$$

From Condition (2), there exist $\delta > 0$ *and* $d_j > 0$ $(j = 1,2\cdots,n)$, such that

$$\sum_{j=1}^{n} (a_{ij} + b_{ij}) d_j < -\delta, \qquad i = 1,2,\cdots,n$$

Choose α, $0 < \alpha < 1$,such that

$$\alpha d_i + \sum_{j=1}^{n} [a_{ij} d_j + b_{ij} d_j e^{\alpha\tau}] < 0 \tag{22}$$

As $t \in [t_0 - \tau, t_0]$, we choose $R, R > 1$, such that $R d_i e^{\alpha\tau} > 1$. For any given $\varepsilon > 0$, let $q_i(t) = R d_i [\sum_{j=1}^{n} \bar{x}_j(t_0) + \varepsilon] e^{-\alpha(t-t_0)}$. By (22), we have

$$D^+ q_i(t) = -\alpha R d_i [\sum_{j=1}^{n} \bar{x}_j(t_0) + \varepsilon] e^{-\alpha(t-t_0)}$$

$$\geq \sum_{j=1}^{n} [a_{ij} d_j + b_{ij} d_j e^{\alpha\tau}] R [\sum_{j=1}^{n} \bar{x}_j(t_0) + \varepsilon] e^{-\alpha(t-t_0)}$$

$$= \sum_{j=1}^{n} a_{ij} d_j R (\sum_{j=1}^{n} \bar{x}_j(t_0) + \varepsilon) e^{-\alpha(t-t_0)} + \sum_{j=1}^{n} b_{ij} d_j R (\sum_{j=1}^{n} \bar{x}_j(t_0) + \varepsilon) e^{-\alpha(t-t_0)} e^{\alpha\tau}$$

$$\geq \sum_{j=1}^{n} a_{ij} q_j(t) + \sum_{j=1}^{n} b_{ij} \bar{q}_j(t) = g_i(t,q(t),\bar{q}(t)) \text{ 。}$$

Therefore $D^+ q_i(t) \geq g_i(t,q(t),\bar{q}(t))$ 。 But as $t \in [t_0 - \tau, t_0]$, we have

$$q_i(\delta) = R d_i [\sum_{j=1}^{n} \bar{x}_j(t_0) + \varepsilon)] e^{-\alpha(t-t_0)} > \sum_{j=1}^{n} \bar{x}_j(t_0) + \varepsilon$$

Let $x_i(t) \leq \sum_{j=1}^{n} \bar{x}_j(t_0) + \varepsilon$, $t \in [t_0 - \tau, t_0]$. By Lemma 2, we have

$$x_i(t) < q_i(t) = R d_i [\sum_{j=1}^{n} \bar{x}_j(t_0) + \varepsilon)] e^{-\alpha(t-t_0)}$$

Let $\varepsilon \to 0^+$, $R d_i = \gamma_i$, then we obtain that

$$x_i(t) \leq \gamma_i [\sum_{j=1}^{n} x_j(t_0)] e^{-\alpha(t-t_0)}, \qquad as \quad t \geq t_0 \text{ , } i = 1,2,\cdots,n \text{ .}$$

The proof of Lemma3 is completed.

Theorem 3. If $|f_j| \le L_j$, $j = 1,2\cdots,n,$ and $diag(\frac{1}{R_1C_1},\cdots,\frac{1}{R_nC_n}) - (\frac{|T_{ij}|}{C_i}L_j)_{n\times n}$ is M-matrix, then universal solution of system (2) is global exponentially stable. i.e. the equilibrium state $u = u^*$ of system (1) is global exponentially stable.

Proof: The system (2) is written as

$$\frac{dx_i}{dt} = -\frac{x_j(t)}{R_iC_i}|x_i(t)| - \le (\frac{1}{R_iC_i})|x_i(t)| + \sum_{j=i}^{n}\frac{|t_{ij}|}{C_i}L_j\bar{x}_j(t)\,, \quad i = 1,2,\cdots,n \qquad (23)$$

Because $diag(\frac{1}{R_1C_1},\cdots,\frac{1}{R_nC_n}) - (\frac{|T_{ij}|}{C_i}L_j)_{n\times n}$ is *M*-matrix, by lemma3, we know that the solution of differential inequality (23) satisfies estimation formula：

$$|x_i(t)| \le \gamma_i[\sum_{j=1}^{n}|x_j(t_0)|]e^{-\alpha(t-t_0)}, \qquad t \ge t_0 \qquad (24)$$

where γ_i, α are positive constants. So Theorem3 comes into existence.

Theorem 4. If the conditions of Theorem3 are satisfied, and there exists a constant $\gamma > 0$, such that $diag(\frac{1}{R_1C_1},-\lambda\cdots,\frac{1}{R_nC_n}-\lambda) - (\frac{|T_{ij}|}{C_i}L_j)_{n\times n}$ is also M-matrix, then the solution of differential inequality (24) at least, satisfies estimation formula (24)

$$|x_i(t)| \le \gamma_i[\sum_{j=1}^{n}|x_j(t_0)|]e^{-\lambda[t-t_0]}, \qquad i = 1,\cdots,n \qquad (25)$$

Therefore, λ is Lyapunov exponent of system (1)..

Proof: Let $|y_i(t)| = |x_i(t)|e^{\lambda t}$, $i = 1,\cdots,n$, then

$$\begin{aligned} D^+|y_i(t)| &\le e^{\lambda t}D^+|x_i(t)| + \lambda e^{\lambda t}|x_i(t)| \\ &\le e^{\lambda t}[\frac{-1}{R_iC_i}|x_i(t)| + \lambda|x_i(t)| + \sum_{j=1}^{n}\frac{|T_{ij}|}{C_i}L_j|\bar{x}_j(t)|] \\ &\le \left(-\frac{1}{R_iC_i}+\lambda\right)\bullet|y_i(t)| + \sum_{j=1}^{n}\frac{|T_{ij}|}{C_i}L_j|\bar{y}_j(t)|\ , \quad i = 1,\cdots,n \end{aligned}$$

By Theorem3, $|y_i(t)| \le \gamma_i[\sum_{j=1}^{n}|y_j(t_0)|e^{-\alpha[t-t_0]}], i = 1,\cdots,n$. So we have

$$|x_i(t)| \le \bar{\gamma}_i[\sum_{j=1}^{n}|x_j(t_0)|]e^{(-\alpha-\lambda)(t-t_0)} \le \bar{\gamma}_i[\sum_{j=1}^{n}|x_j(t_0)|]e^{-\lambda(t-t_0)}, \qquad i = 1,\cdots,n,$$

If $\min\limits_{1\le i\le n}\left[\frac{1}{R_iC_i}-\sum\limits_{j=1}^{n}\frac{|T_{ij}|}{C_i}L_j\right]=\bar{\lambda}>0$, then the conditions of Theorem 4 are satisfied. $\bar{\lambda}$ is Lyaunprov exponent.

Acknowledgements

This project is supported by Henan Province Natural Science Foundation (No.072300410110).

References

1. Jiao, L.C.: Neural Network System Theory. The Press of Xidian University, Xian (1992)
2. Xu, B.Z., Zhang, B.L., Wei, G.: Neural Network Theory and Application. The Press of South China university of Technology, Guangzhou (1994)
3. Halanory, A.: Differential Equations: Stability Oscillations Time-lags. Academic, New York (1996)
4. Liao, X.X., Liao, Y.: Stability of Hopfield-type Neural Networks (II). Science in China 40(8), 813–816 (1997)
5. Liao, X.X.: Stability of Hopfield-type Neural Networks (I). Science in China 38(4), 407–418 (1995)
6. Sudnarsanan, S.I., Sundaroshan, M.K.: Exponential Stability and Systematic Synthesis of a Neural for Quadratic Migration Neural Networks 4(5), 599–613 (1991)

Global Exponential Stability of FCNNs with Bounded Uncertain Delays

Guozheng Wang[1], Qianhong Zhang[1,2], and Zhenguo Luo[2]

[1]Basic Science Department, Hunan Institute of Technology,
Hengyang, Hunan 421002, P.R. China
[2]School of Mathematical Science and Computing Technology,
Central South University, Changsha, Hunan 410083, P.R.China
gzhwang123@163.com,zqianhong68@163.com,luozhenguo0701@yahoo.com.cn

Abstract. In this paper, fuzzy cellular neural networks with bounded uncertain delays are investigated. By constructing Lyapunov functional method, applying inequality technique and the homeomorphism theory, we derive some new sufficient conditions to guarantee existence, uniqueness of the equilibrium point and its global exponential stability for fuzzy cellular neural networks. The results of this paper are new and they complement previously known results.

Keywords: Fuzzy cellular neural networks; Global exponential stability; Homeomorphism theory; Lyapunov functional; Equilibrium point.

1 Introduction

So far, there are two basic cellular neural networks being proposed. The first one is traditional cellular neural networks (CNNs)which was first introduced by Chua and Yang in 1988(see [1] [2]). The dynamical behaviors of CNNs and with delays (DCNNs) have received much attention due to their potential application in associated memory, parallel computing, pattern recognition, signal processing and optimization problems (see [3]-[11]). When a neural circuit is employed as associated memory, the existence of many equilibrium points is necessary feature. However, in application to solve optimization problems, the networks must possess a unique globally asymptotically stable (GAS) or globally exponentially stable(GES) equilibrium point for every input vector. GAS or GES of cellular neural networks without delay and with delay have been extensively investigated(see[3]-[11]). Based on traditional CNNs , T.Yang and L.B.Yang [12]-[13] proposed another type-fuzzy cellular neural networks (FCNNs), which integrates fuzzy logic into the structure of cellular neural networks. Unlike CNNs structure, FCNNs has fuzzy logic between its template input and/or output besides the sum of product operations. Studies have shown that FCNNs has its potential in image processing and pattern recognition. Like the traditional CNNs, the stability of the system is very important in the design of the FCNNs. In recent years some results on stability for FCNNs have been derived(see [14]-[17]). To the best of our knowledge. FCNNs with delays are seldom considered.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 383–394, 2009.

Authors in reference [16] give some conditions to guarantee the global stability of FCNNs with constant delays and time-varying delays under the assumption $(A3)$, i.e. delay $\tau(t)$ is differentiable satisfied with $\tau_j^{\cdot}(t) \leq \mu_j, \tau(t) \leq \tau$. However, in most of the practical applications, it is difficult to know the delays exactly. Hence, it is not easy to estimate the bounds of the derivatives of delay in advance. In this paper we consider the following FCNNs with bounded uncertain delays.

$$\dot{x}_i(t) = -d_i x_i(t) + \sum_{j=1}^{n} a_{ij} f_j(x_j(t)) + \sum_{j=i}^{n} b_{ij} u_j + I_i + \bigwedge_{j=1}^{n} \alpha_{ij} g_j(x_j(t-\tau_{ij})) + \bigwedge_{j=1}^{n} T_{ij} u_j + \bigvee_{j=1}^{n} \beta_{ij} g_j(x_j(t-\tau_{ij})) + \bigvee_{j=1}^{n} H_{ij} u_j \tag{1}$$

$i = 1, 2, \cdots, n$ where $\alpha_{ij}, \beta_{ij}, T_{ij}$ and H_{ij} are elements of fuzzy feedback MIN template and fuzzy feedback MAX template, fuzzy feed forward MIN template and fuzzy feed forward MAX template, respectively. a_{ij} and b_{ij} are elements of feedback template and feed forward template. $\bigwedge$ and $\bigvee$ denote the fuzzy AND and fuzzy OR operation, respectively. x_i, u_j and I_i denote state, input and bias of the ith neurons, respectively. τ_{ij} is the transmission delay with $0 \leq \tau_{ij} \leq \tau$, f_j and g_j are the activation functions.

Suppose that system (1) has the initial conditions with $x_i(t) = \phi_i(t), -\tau \leq t \leq 0$. A continuous solution denoted by $x(t, 0, \phi)$ or $x(t)$ if no confusion should occur, where $x(t) = (x_1(t), x_2(t), \ldots, x_n(t))^T$ (T denote transpose). For $x \in R^n$,we define the vector norm $\|x\| = (\sum_{i=i}^{n} \mid x_i \mid^2)^{\frac{1}{2}}, \|x\|_\infty = \max_{1 \leq i \leq n} \mid x_i \mid$. For any $\phi = (\phi_1, \phi_2, \ldots, \phi_n)^T \in C$(where $C = C([-\tau, 0], R^n)$), we define a norm in C by $\|\phi\|_\tau = \sup_{-\tau \leq \theta \leq 0} \|\phi(\theta)\|_\infty$

Definition 1. *The equilibrium point $x^* = (x_1^*, x_2^*, \ldots, x_n^*)^T$ of system (1) is said to be GES, if there are constants $\lambda > 0$ and $M \geq 1$ such that, for any $t \geq 0$*

$$\|x(t) - x^*\|_\infty \leq M \|\phi - x^*\|_\tau e^{-\lambda t}$$

Definition 2. *If $f(t) : R \to R$ is a continuous function, then the upper right derivative of f is defined as*

$$D^+ f(t) = \lim_{l \to 0^+} \sup \frac{1}{l}(f(t+l) - f(t))$$

Definition 3. *[18] A map $H : R^n \to R^n$ is a homeomorphism of R^n onto itself if H is continuous and one-to-one and its inverse map H^{-1} is also continuous.*

Lemma 1. *[18] Let $H : R^n \to R^n$ be continuous. If H satisfies the following conditions*
(1) $H(x)$ is injective on R^n
(2) $\|H(x)\| \to \infty$ as $\|x\| \to \infty$.
Then H is homeomorphism.

Lemma 2. *(Mitrinovic and Vasic,1970) For $a_k \geq 0, b_k \geq 0, (k = 1, 2, \ldots, m)$ the following inequality holds*

$$a \prod_{k=1}^{n} b_k^{q_k} \leq \frac{1}{r} \sum_{k=1}^{m} q_k b_k^r + \frac{1}{r} a^r, \tag{2}$$

where $q_k > 0, (k = 1, 2, \ldots, m)$ is some constant , $\sum_{k=1}^{m} q_k = r - 1$ and $r > 1$.

To obtain our results, we make the following assumptions.
$(A1)$: f_j and $g_j (j = 1, 2, \ldots, n)$ are globally Lipschitz continuous, i.e., there exist positive constant u_j and σ_j such that

$$|f_j(x) - f_j(y)| \leq u_j |x - y|, \quad |g_j(x) - g_j(y)| \leq \sigma_j |x - y|, \tag{3}$$

and $f_j(0) = g_j(0) = 0$ for any $x, y \in R$ and $j = 1, 2, \ldots, n$.
$(A2)$: There are constants $\delta_{kj}, \gamma_{kj}, \xi_{kj} \in R, q_k > 0$ and $c_i > 0, i, j = 1, 2, \ldots, n; k = 1, 2, \ldots, m$ such that

$$\begin{aligned} rd_i > & \sum_{j=1}^{n} \sum_{k=1}^{m} q_k |a_{ij}|^{\frac{r\delta_{kj}}{q_k}} u_j + \sum_{j=1}^{n} \sum_{k=1}^{m} q_k |\alpha_{ij}|^{\frac{r\gamma_{kj}}{q_k}} \sigma_j + \sum_{j=1}^{n} \sum_{k=1}^{m} q_k |\beta_{ij}|^{\frac{r\xi_{kj}}{q_k}} \sigma_j \\ & + \frac{1}{c_i} \left[\sum_{j=1}^{n} |a_{ij}|^{r\delta_{m+1,j}} c_j u_i + \sum_{j=1}^{n} |\alpha_{ij}|^{r\gamma_{m+1,j}} c_j \sigma_i + \sum_{j=1}^{n} |\beta_{ij}|^{r\xi_{m+1,j}} c_j \sigma_i \right] \end{aligned} \tag{4}$$

where

$$\sum_{k=1}^{m+1} \delta_{kj} = 1, \sum_{k=1}^{m+1} \gamma_{kj} = 1, \sum_{k=1}^{m+1} \xi_{kj} = 1, \sum_{k=1}^{m} q_k = r - 1, r \geq 1, i, j = 1, 2, \ldots, n$$

The rest of this paper is organized as follows. in Section 2, we will give the existence and uniqueness of equilibrium point for fuzzy cellular neural networks. Results for global exponential stability of fuzzy cellular neural networks with bounded unknown delays will be given and proved in Section 3. Conclusion will be given in Section 4.

2 Existence and Uniqueness of the Equilibrium Point

In this section, we will discuss the existence and uniqueness of equilibrium point for fuzzy cellular neural networks (1). In order to prove our results some lemma is given as follows.

Lemma 3. *[12] Suppose x and y are two states of system (1), then we have*

$$\left| \bigwedge_{j=1}^{n} \alpha_{ij} f_j(x_j) - \bigwedge_{j=1}^{n} \alpha_{ij} f_j(y_j) \right| \leq \sum_{j=1}^{n} |\alpha_{ij}| |f_j(x_j) - f_j(y_j)|, \tag{5}$$

and

$$\left|\bigvee_{j=1}^{n}\beta_{ij}f_j(x_j)-\bigvee_{j=1}^{n}\beta_{ij}f_j(y_j)\right|\leq\sum_{j=1}^{n}|\beta_{ij}||f_j(x_j)-f_j(y_j)| \tag{6}$$

Firstly, we study the existence and uniqueness of the equilibrium point, considering the following equations associated with system (1)

$$-d_ix_i(t)+\sum_{j=1}^{n}a_{ij}f_j(x_j(t))+\sum_{j=i}^{n}b_{ij}u_j+I_i+\bigwedge_{j=1}^{n}\alpha_{ij}g_j(x_j(t-\tau_{ij}))+\bigwedge_{j=1}^{n}T_{ij}u_j$$
$$+\bigvee_{j=1}^{n}\beta_{ij}g_j(x_j(t-\tau_{ij}))+\bigvee_{j=1}^{n}H_{ij}u_j=0.$$

Define the map H as follows

$$H(x)=(h_1(x_1),h_2(x_2),\ldots,h_n(x_n))^T \tag{7}$$

in which

$$h_i(x_i)=-d_ix_i(t)+\sum_{j=1}^{n}a_{ij}f_j(x_j(t))+\sum_{j=i}^{n}b_{ij}u_j+I_i+\bigwedge_{j=1}^{n}\alpha_{ij}g_j(x_j(t-\tau_{ij}))$$
$$+\bigwedge_{j=1}^{n}T_{ij}u_j+\bigvee_{j=1}^{n}\beta_{ij}g_j(x_j(t-\tau_{ij}))+\bigvee_{j=1}^{n}H_{ij}u_j \tag{8}$$

Theorem 1. *Assume that (A1) and (A2) hold , then system (1) has a unique equilibrium point x^*.*

Proof. We define map H as (7) and (8), we only need to show that H satisfies two conditions of Lemma 1. We will show that If $x\neq\overline{x}$ then $H(x)\neq H(\overline{x})$ holds for any $x,\overline{x}\in R^n$. The component $h_i(x_i)-h_i(\overline{x}_i)$ of the vector $H(x)-H(\overline{x})$ is as follows: for $i=1,2,\ldots,n$.

$$h_i(x_i)-h_i(\overline{x}_i)=-d_i(x_i-\overline{x}_i)+\sum_{j=1}^{n}a_{ij}(f_j(x_j))-f_j(\overline{x}_j))+\bigwedge_{j=1}^{n}\alpha_{ij}g_j(x_j)$$
$$-\bigwedge_{j=1}^{n}\alpha_{ij}g_j(\overline{x}_j)+\bigvee_{j=1}^{n}\beta_{ij}g_j(x_j)-\bigvee_{j=1}^{n}\beta_{ij}g_j(\overline{x}_j) \tag{9}$$

By (A1) and Lemma 2, we obtain
$\sum_{i=1}^{n}\mathrm{sgn}(x_i-\overline{x}_i)rc_i[h_i(x_i)-h_i(\overline{x}_i)]|x_i-\overline{x}_i|^{r-1}$

$$=\sum_{i=1}^{n}\mathrm{sgn}(x_i-\overline{x}_i)rc_i|x_i-\overline{x}_i|^{r-1}\left\{-d_i(x_i-\overline{x}_i)+\sum_{j=1}^{n}a_{ij}(f_j(x_j)-f_j(\overline{x}_j))\right.$$

$$
\begin{aligned}
&+\bigwedge_{j=1}^{n}\alpha_{ij}g_j(x_j)-\bigwedge_{j=1}^{n}\alpha_{ij}g_j(\overline{x}_j)+\bigvee_{j=1}^{n}\beta_{ij}g_j(x_j)-\bigvee_{j=1}^{n}\beta_{ij}g_j(\overline{x}_j)\Bigg\}\\
&\le\sum_{i=1}^{n}rc_i\Bigg\{-d_i|x_i-\overline{x}_i|^r+\sum_{j=1}^{n}|a_{ij}|u_j|x_j-\overline{x}_j||x_i-\overline{x}_i|^{r-1}\\
&\quad+\sum_{j=1}^{n}|\alpha_{ij}|\sigma_j|x_j-\overline{x}_j||x_i-\overline{x}_i|^{r-1}+\sum_{j=1}^{n}|\beta_{ij}|\sigma_j|x_j-\overline{x}_j||x_i-\overline{x}_i|^{r-1}\Bigg\}\\
&=\sum_{i=1}^{n}rc_i\Bigg\{-d_i|x_i-\overline{x}_i|^r+\sum_{j=1}^{n}u_j|a_{ij}^{\delta_{m+1,j}}(x_j-\overline{x}_j)|\prod_{k=1}^{m}|a_{ij}^{\frac{\delta_{kj}}{q_k}}(x_i-\overline{x}_i)|^{q_k}\\
&\quad+\sum_{j=1}^{n}\sigma_j|\alpha_{ij}^{\gamma_{m+1,j}}(x_j-\overline{x}_j)|\prod_{k=1}^{m}|\alpha_{ij}^{\frac{\gamma_{kj}}{q_k}}(x_i-\overline{x}_i)|^{q_k}\\
&\quad+\sum_{j=1}^{n}\sigma_j|\beta_{ij}^{\xi_{m+1,j}}(x_j-\overline{x}_j)|\prod_{k=1}^{m}|\beta_{ij}^{\frac{\xi_{kj}}{q_k}}(x_i-\overline{x}_i)|^{q_k}\Bigg\}\\
&\le\sum_{i=1}^{n}rc_i\Bigg\{-d_i|x_i-\overline{x}_i|^r+\sum_{j=1}^{n}\frac{1}{r}\left[\sum_{k=1}^{m}q_k|a_{ij}|^{\frac{r\delta_{kj}}{q_k}}|x_i-\overline{x}_i|^r+|a_{ij}|^{r\delta_{m+1,j}}|x_j-\overline{x}_j|^r\right]u_j\\
&\quad+\sum_{j=1}^{n}\frac{1}{r}\left[\sum_{k=1}^{m}q_k|\alpha_{ij}|^{\frac{r\gamma_{kj}}{q_k}}|x_i-\overline{x}_i|^r+|\alpha_{ij}|^{r\gamma_{m+1,j}}|x_j-\overline{x}_j|^r\right]\sigma_j\\
&\quad+\sum_{j=1}^{n}\frac{1}{r}\left[\sum_{k=1}^{m}q_k|\beta_{ij}|^{\frac{r\xi_{kj}}{q_k}}|x_i-\overline{x}_i|^r+|\beta_{ij}|^{r\xi_{m+1,j}}|x_j-\overline{x}_j|^r\right]\sigma_j\Bigg\}\\
&=\sum_{i=1}^{n}\Bigg\{-rc_id_i|x_i-\overline{x}_i|^r+c_i\left[\sum_{j=1}^{n}\sum_{k=1}^{m}q_k|a_{ij}|^{\frac{r\delta_{kj}}{q_k}}u_j+\sum_{j=1}^{n}\sum_{k=1}^{m}q_k|\alpha_{ij}|^{\frac{r\gamma_{kj}}{q_k}}\sigma_j\right.\\
&\quad\left.+\sum_{j=1}^{n}\sum_{k=1}^{m}q_k|\beta_{ij}|^{\frac{r\xi_{kj}}{q_k}}\sigma_j\right]|x_i-\overline{x}_i|^r\\
&\quad+\sum_{j=1}^{n}c_i\left[u_j|a_{ij}|^{r\delta_{m+1,j}}+\sigma_j|\alpha_{ij}|^{r\gamma_{m+1,j}}+\sigma_j|\beta_{ij}|^{r\xi_{m+1,j}}\right]|x_j-\overline{x}_j|^r\Bigg\}\\
&=\sum_{i=1}^{n}\Bigg\{-rd_ic_i|x_i-\overline{x}_i|^r+c_i\left[\sum_{j=1}^{n}\sum_{k=1}^{m}q_k|a_{ij}|^{\frac{r\delta_{kj}}{q_k}}u_j+\sum_{j=1}^{n}\sum_{k=1}^{m}q_k|\alpha_{ij}|^{\frac{r\gamma_{kj}}{q_k}}\sigma_j\right.\\
&\quad\left.+\sum_{j=1}^{n}\sum_{k=1}^{m}q_k|\beta_{ij}|^{\frac{r\xi_{kj}}{q_k}}\sigma_j\right]|x_i-\overline{x}_i|^r+\left[\sum_{j=1}^{n}|a_{ji}|^{r\delta_{m+1,j}}c_ju_i\right.
\end{aligned}
$$

$$+\sum_{j=1}^{n}|\alpha_{ji}|^{r\gamma_{m+1,j}}c_j\sigma_i+\sum_{j=1}^{n}|\beta_{ji}|^{r\xi_{m+1,j}}c_j\sigma_i\Bigg]|x_i-\overline{x}_i|^r\Bigg\}$$

$$=-\sum_{i=1}^{n}c_i\Bigg\{rd_i-\Bigg[\sum_{j=1}^{n}\sum_{k=1}^{m}q_k|a_{ij}|^{\frac{r\delta_{kj}}{q_k}}u_j+\sum_{j=1}^{n}\sum_{k=1}^{m}q_k|\alpha_{ij}|^{\frac{r\gamma_{kj}}{q_k}}\sigma_j$$

$$+\sum_{j=1}^{n}\sum_{k=1}^{m}q_k|\beta_{ij}|^{\frac{r\xi_{kj}}{q_k}}\sigma_j\Bigg]-\frac{1}{c_i}\Bigg[\sum_{j=1}^{n}|a_{ij}|^{r\delta_{m+1,j}}c_ju_i+\sum_{j=1}^{n}|\alpha_{ij}|^{r\gamma_{m+1,j}}c_j\sigma_i$$

$$+\sum_{j=1}^{n}|\beta_{ij}|^{r\xi_{m+1,j}}c_j\sigma_i\Bigg]\Bigg\}|x_i-\overline{x}_i|^r$$

From (A2) we derive

$$\sum_{i=1}^{n}\mathrm{sgn}(x_i-\overline{x}_i)rc_i[h_i(x_i)-h_i(\overline{x}_i)]|x_i-\overline{x}_i|^{r-1}<0.$$

Which implies that there is at least one index i such that $h_i(x_i)-h_i(\overline{x}_i)\neq 0$, therefore $H(x)\neq H(\overline{x})$, namely, the map H is injective.

Now we only need to prove that $\|H(x)\|\to\infty$ as $\|x\|\to\infty$.
Let

$$H^*(x)=(h_1^*(x_1),h_2^*(x_2),\ldots,h_n^*(x_n))^T$$

where

$$h_i^*(x_i)=-d_ix_i+\sum_{j=1}^{n}a_{ij}(f_j(x_j)-f_j(0))+\bigwedge_{j=1}^{n}\alpha_{ij}g_j(x_j)-\bigwedge_{j=1}^{n}\alpha_{ij}g_j(0)$$
$$+\bigvee_{j=1}^{n}\beta_{ij}g_j(x_j)-\bigvee_{j=1}^{n}\beta_{ij}g_j(0)\tag{10}$$

To prove that $\|H(x)\|\to\infty$ as $\|x\|\to\infty$. it suffices to show that $\|H^*(x)\|\to\infty$ as $\|x\|\to\infty$, we have
$\sum_{i=1}^{n}rc_i\mathrm{sgn}(x_i)h_i^*(x_i)|x_i|^{r-1}$

$$\leq\sum_{i=1}^{n}rc_i\Bigg\{-d_i|x_i|^r+\sum_{j=1}^{n}|a_{ij}|u_j|x_j||x_i|^{r-1}+\sum_{j=1}^{n}|\alpha_{ij}|\sigma_j|x_j||x_i|^{r-1}$$

$$+\sum_{j=1}^{n}|\beta_{ij}|\sigma_j|x_j||x_i|^{r-1}\Bigg\}$$

$$=\sum_{i=1}^{n}rc_i\Bigg\{-d_i|x_i|^r+\sum_{j=1}^{n}u_j|a_{ij}^{\delta_{m+1,j}}x_j|\prod_{k=1}^{m}|a_{ij}^{\frac{\delta_{kj}}{q_k}}x_i|^{q_k}$$

$$
\begin{aligned}
&+\sum_{j=1}^{n}\sigma_j|\alpha_{ij}^{\gamma_{m+1,j}}x_j|\prod_{k=1}^{m}|\alpha_{ij}^{\frac{\gamma_{kj}}{q_k}}x_i|^{q_k}+\sum_{j=1}^{n}\sigma_j|\beta_{ij}^{\xi_{m+1,j}}x_j|\prod_{k=1}^{m}|\beta_{ij}^{\frac{\xi_{kj}}{q_k}}x_i|^{q_k}\Bigg\}\\
&\le\sum_{i=1}^{n}rc_i\Bigg\{-d_i|x_i|^r+\sum_{j=1}^{n}u_j\frac{1}{r}\left[\sum_{k=1}^{m}q_k|a_{ij}|^{\frac{r\delta_{kj}}{q_k}}|x_i|^r+|a_{ij}|^{r\delta_{m+1,j}}|x_j|^r\right]\\
&\quad+\sum_{j=1}^{n}\sigma_j\frac{1}{r}\left[\sum_{k=1}^{m}q_k|\alpha_{ij}|^{\frac{r\gamma_{kj}}{q_k}}|x_i|^r+|\alpha_{ij}|^{r\gamma_{m+1,j}}|x_j|^r\right]\\
&\quad+\sum_{j=1}^{n}\sigma_j\frac{1}{r}\left[\sum_{k=1}^{m}q_k|\beta_{ij}|^{\frac{r\xi_{kj}}{q_k}}|x_i|^r+|\beta_{ij}|^{r\xi_{m+1,j}}|x_j|^r\right]\Bigg\}\\
&=-\sum_{i=1}^{n}c_i\Bigg\{rd_i-\left[\sum_{j=1}^{n}\sum_{k=1}^{m}q_k|a_{ij}|^{\frac{r\delta_{kj}}{q_k}}u_j+\sum_{j=1}^{n}\sum_{k=1}^{m}q_k|\alpha_{ij}|^{\frac{r\gamma_{kj}}{q_k}}\sigma_j\right.\\
&\quad\left.+\sum_{j=1}^{n}\sum_{k=1}^{m}q_k|\beta_{ij}|^{\frac{r\xi_{kj}}{q_k}}\sigma_j\right]-\frac{1}{c_i}\left[\sum_{j=1}^{n}|a_{ji}|^{r\delta_{m+1,j}}c_ju_i+\sum_{j=1}^{n}|\alpha_{ji}|^{r\gamma_{m+1,j}}c_j\sigma_i\right.\\
&\quad\left.+\sum_{j=1}^{n}|\beta_{ji}|^{r\xi_{m+1,j}}c_j\sigma_i\right]|x_i|^r\Bigg\}\\
&\le-\eta\sum_{i=1}^{n}c_i|x_i|^r
\end{aligned}
$$

where

$$
\begin{aligned}
\eta=\min_{1\le i\le n}\Bigg\{&rd_i-\left[\sum_{j=1}^{n}\sum_{k=1}^{m}q_k|a_{ij}|^{\frac{r\delta_{kj}}{q_k}}u_j+\sum_{j=1}^{n}\sum_{k=1}^{m}q_k|\alpha_{ij}|^{\frac{r\gamma_{kj}}{q_k}}\sigma_j+\sum_{j=1}^{n}\sum_{k=1}^{m}\right.\\
&\left.\times q_k|\beta_{ij}|^{\frac{r\xi_{kj}}{q_k}}\sigma_j\right]-\frac{1}{c_i}\left[\sum_{j=1}^{n}|a_{ji}|^{r\delta_{m+1,j}}c_ju_i+\sum_{j=1}^{n}|\alpha_{ji}|^{r\gamma_{m+1,j}}c_j\sigma_i\right.\\
&\left.+\sum_{j=1}^{n}|\beta_{ji}|^{r\xi_{m+1,j}}c_j\sigma_i\right]\Bigg\}\\
>{}&0
\end{aligned}
$$

Thus we obtain

$$
\begin{aligned}
\eta c_i|x_i|^r&\le\eta\sum_{i=1}^{n}c_i|x_i|^r\\
&\le\left|\sum_{i=1}^{n}rc_i\ \mathrm{sgn}(x_i)h_i^*(x_i)|x_i|^{r-1}\right|
\end{aligned}
$$

$$\leq \sum_{i=1}^{n} rc_i \ \mathrm{sgn}(x_i) h_i^*(x_i) |x_i|^{r-1}$$

$$\leq \|H^*(x)\|_\infty \|x\|_\infty^{r-1} \sum_{i=1}^{n} rc_i.$$

That is $\eta \underline{c} \|x\|_\infty \leq (\sum_{i=1}^{n} rc_i) \|H^*(x)\|_\infty$, where $\underline{c} = \min_{1 \leq i \leq n}\{c_i\}$. Therefore,it follows that $\|H^*(x)\| \geq \frac{\eta \underline{c}}{\sum_{i=1}^{n} rc_i} \|x\|_\infty$, which directly implies that $\|H^*(x)\| \to \infty$ as $\|x\| \to \infty$, in view of the equivalance of the norms $\|\cdot\|_\infty$ and $\|\cdot\|$, thus H is a homeomorphism on R^n. Hence there is a unique equilibrium point $x = x^*$ such that $H(x^*) = 0$.

3 Global Exponential Stability of FCNNs

Let $x^* = (x_1^*, x_2^*, \ldots, x_n^*)^T$ be the equilibrium point of system(1), we make a transform for system (1): $z_i(t) = x_i(t) - x_i^*, (i = 1, 2, \ldots, n)$, we have

$$\dot{z}_i(t) = -d_i z_i(t) + \sum_{j=1}^{n} a_{ij}(f_j(z_j(t) + x_j^*) - f_j(x_j^*)) + \bigwedge_{j=1}^{n} \alpha_{ij} g_j(z_j(t - \tau_{ij}) + x_j^*)$$
$$- \bigwedge_{j=1}^{n} \alpha_{ij} g_j(x_j^*) + \bigvee_{j=1}^{n} \beta_{ij} g_j(z_j(t - \tau_{ij}) + x_j^*) - \bigvee_{j=1}^{n} \beta_{ij} g_j(x_j^*) \tag{11}$$

where $z_i(t) = \Phi_i(t), \Phi_i(t) = \phi_i(t) - x_i^*, i, j = 1, 2, \ldots, n. -\tau \leq t \leq 0$,

Clearly, the equilibrium point x^* of system (1) is GES if and only if the equilibrium point O of system (11) is GES. In the following, we only study global exponential stability of the equilibrium point O for system (11).

Theorem 2. *If the conditions (A1)-(A2) hold, then the system (11) has a unique equilibrium point O of system which is GES and satisfies*

$$\|z(t)\|_\infty \leq M \|\Phi\|_\tau e^{-\lambda t} \tag{12}$$

where

$$M = \left\{ \frac{1}{\underline{c}} \left[\sum_{i=1}^{n} c_i + \sum_{i=1}^{n} \sum_{j=1}^{n} c_i \sigma_j (|\alpha_{ij}|^{r\gamma_{m+1,j}} + |\beta_{ij}|^{r\xi_{m+1,j}}) \tau e^{\lambda r \tau} \right] \right\}^{\frac{1}{r}} \geq 1$$

$\lambda > 0$ *is a constant,* $z(t) = (z_1(t), z_2(t), \ldots, z_n(t))^T, \Phi = (\Phi_1, \Phi_2, \ldots, \Phi_n)^T$.

Proof. The existence and uniqueness of an equilibrium point O is guaranteed by theorem 1. So we only need to show that inequality (12) holds. From (A2), there is a small constant $0 < \lambda < d_i$, such that

$$\min_{1 \leq i \leq n} \left\{ r(d_i - \lambda) - \left[\sum_{j=1}^{n} \sum_{k=1}^{m} q_k |a_{ij}|^{\frac{r\delta_{kj}}{q_k}} u_j + \sum_{j=1}^{n} \sum_{k=1}^{m} q_k |\alpha_{ij}|^{\frac{r\gamma_{kj}}{q_k}} \sigma_j \right. \right.$$

$$+\sum_{j=1}^{n}\sum_{k=1}^{m} q_k|\beta_{ij}|^{\frac{r\xi_{kj}}{q_k}}\sigma_j\Bigg] - \frac{1}{c_i}\left[\sum_{j=1}^{n}|a_{ji}|^{r\delta_{m+1,j}}c_j u_i + \sum_{j=1}^{n}|\alpha_{ji}|^{r\gamma_{m+1,j}}c_j\sigma_i + \sum_{j=1}^{n}|\beta_{ji}|^{r\xi_{m+1,j}}c_j\sigma_i\right]\Bigg\} \geq 0$$

Now we consider the following Lyapunov functional

$$V(t) = \sum_{i=1}^{n} c_i \left[|z_i(t)|^r e^{\lambda rt} + \sum_{j=1}^{n}\sigma_j(|\alpha_{ij}|^{r\gamma_{m+1,j}} + |\beta_{ij}|^{r\xi_{m+1,j}}) \times \int_{t-\tau_{ij}}^{t}|z_j(s)|^r e^{\lambda r(s+\tau)}ds\right] \tag{13}$$

Calculating the upper right derivative of V along system (11) and using Lemma 2, we obtain

$$\begin{aligned}
D^+V(t) \leq & \sum_{i=1}^{n} c_i \left\{r|z_i(t)|^{r-1}D^+|z_i(t)|e^{\lambda rt} + |z_i(t)|^r\lambda r e^{\lambda rt} \right.\\
& \left. + \sum_{j=1}^{n}\sigma_j(|\alpha_{ij}|^{r\gamma_{m+1,j}} + |\beta_{ij}|^{r\xi_{m+1,j}})[|z_j(t)|^r e^{\lambda r(t+\tau)} - |z_j(t-\tau_{ij})|e^{\lambda r\tau}]\right\}\\
\leq & \sum_{i=1}^{n} rc_i \left\{e^{\lambda rt}\left[-d_i|z_i(t)|^r + \sum_{j=1}^{n}|a_{ij}||u_j||z_i(t)|^{r-1}|z_j(t)| + \sum_{j=1}^{n}(|\alpha_{ij}|\right.\right.\\
& \left. + |\beta_{ij}|)\sigma_j|z_i(t)|^{r-1}|z_j(t-\tau_{ij})|\right] + |z_i(t)|^r\lambda e^{\lambda rt} + \frac{1}{r}\left[\sum_{j=1}^{n}\sigma_j(|\alpha_{ij}|^{r\gamma_{m+1,j}}\right.\\
& \left. + |\beta_{ij}|^{r\xi_{m+1,j}})|z_j(t)|^r e^{\lambda r(t+\tau)} - \sum_{j=1}^{n}\sigma_j(|\alpha_{ij}|^{r\gamma_{m+1,j}} + |\beta_{ij}|^{r\xi_{m+1,j}})\right.\\
& \left.\left. \times|z_j(t-\tau_{ij})|^r e^{\lambda rt}\right]\right\}\\
= & \sum_{i=1}^{n} rc_i \left\{e^{\lambda rt}\left[-(d_i-\lambda)|z_i(t)|^r + \sum_{j=1}^{n}u_j|a_{ij}^{\delta_{m+1,j}}z_j(t)|\prod_{k=1}^{m}|a_{ij}^{\frac{\delta_{kj}}{q_k}}z_i(t)|^{q_k}\right.\right.\\
& + \sum_{j=1}^{n}\sigma_j|\alpha_{ij}^{\gamma_{m+1,j}}z_j(t-\tau_{ij})|\prod_{k=1}^{m}|\alpha_{ij}^{\frac{\gamma_{kj}}{q_k}}z_i(t)|^{q_k} + \sum_{j=1}^{n}\sigma_j|\beta_{ij}^{\xi_{m+1,j}}z_j(t-\tau_{ij})|\\
& \left.\left. \times\prod_{k=1}^{m}|\beta_{ij}^{\frac{\xi_{kj}}{q_k}}z_i(t)|^{q_k}\right] + \frac{1}{r}\left[\sum_{j=1}^{n}\sigma_j(|\alpha_{ij}|^{r\gamma_{m+1,j}} + |\beta_{ij}|^{r\xi_{m+1,j}})|z_j(t)|^r e^{\lambda r(t+\tau)}\right.\right.
\end{aligned}$$

$$
\begin{aligned}
&\left. -\sum_{j=1}^{n}\sigma_j(|\alpha_{ij}|^{r\gamma_{m+1,j}}+|\beta_{ij}|^{r\xi_{m+1,j}})|z_j(t-\tau_{ij})|^r e^{\lambda rt}\right]\Bigg\}\\
\le&\sum_{i=1}^{n} rc_i e^{\lambda rt}\Bigg\{-(d_i-\lambda)|z_i(t)|^r+\frac{1}{r}\sum_{j=1}^{n}u_j\Bigg[\sum_{k=1}^{m}q_k|a_{ij}|^{\frac{r\delta_{kj}}{q_k}}|z_i(t)|^r+|a_{ij}|^{r\delta_{m+1,j}}\\
&\times.|z_j(t)|^r]+\frac{1}{r}\sum_{j=1}^{n}\sigma_j\left[\sum_{k=1}^{m}q_k|\alpha_{ij}|^{\frac{r\gamma_{kj}}{q_k}}|z_i(t)|^r+|\alpha_{ij}|^{r\gamma_{m+1,j}}|z_j(t-\tau_{ij})|^r\right]\\
&+\frac{1}{r}\sum_{j=1}^{n}\sigma_j\left[\sum_{k=1}^{m}q_k|\beta_{ij}|^{\frac{r\xi_{kj}}{q_k}}|z_i(t)|^r+|\beta_{ij}|^{r\xi_{m+1,j}}|z_j(t-\tau_{ij})|^r\right]\\
&+\frac{1}{r}\Bigg[\sum_{j=1}^{n}\sigma_j(|\alpha_{ij}|^{r\gamma_{m+1,j}}+|\beta_{ij}|^{r\xi_{m+1,j}})||z_j(t)|^r e^{\lambda r\tau}\\
&\left.-\sum_{j=1}^{n}\sigma_j(|\alpha_{ij}|^{r\gamma_{m+1,j}}+|\beta_{ij}|^{r\xi_{m+1,j}})|z_j(t-\tau_{ij})|^r\right]\Bigg\}\\
=&\,e^{\lambda rt}\sum_{i=1}^{n}\Bigg\{-rc_i(d_i-\lambda)|z_i(t)|^r+c_i|z_i(t)|^r\left[\sum_{j=1}^{n}u_j\sum_{k=1}^{m}q_k|a_{ij}|^{\frac{r\delta_{kj}}{q_k}}\right.\\
&\left.+\sum_{j=1}^{n}\sigma_j\sum_{k=1}^{m}q_k\left(|\alpha_{ij}|^{\frac{r\gamma_{kj}}{q_k}}+|\beta_{ij}|^{\frac{r\xi_{kj}}{q_k}}\right)\right]\\
&+\sum_{j=1}^{n}c_i\left[u_j|a_{ij}|^{r\delta_{m+1,j}}+(|\alpha_{ij}|^{r\gamma_{m+1,j}}+|\beta_{ij}|^{r\xi_{m+1,j}})\sigma_j e^{\lambda r\tau}\right]|z_j(t)|^r\Bigg\}\\
=&-e^{\lambda rt}\sum_{i=1}^{n}c_i\Bigg\{(d_i-\lambda)r-\left[\sum_{j=1}^{n}\sum_{k=1}^{m}q_k|a_{ij}|^{\frac{r\delta_{kj}}{q_k}}u_j\right.\\
&\left.+\sum_{j=1}^{n}\sum_{k=1}^{m}q_k\left(|\alpha_{ij}|^{\frac{r\gamma_{kj}}{q_k}}+|\beta_{ij}|^{\frac{r\xi_{kj}}{q_k}}\right)\sigma_j\right]-\frac{1}{c_i}\left[\sum_{j=1}^{n}|a_{ji}|^{r\delta_{m+1,j}}c_j u_i\right.\\
&\left.\left.+\sum_{j=1}^{n}(|\alpha_{ji}|^{r\gamma_{m+1,j}}+|\beta_{ji}|^{r\xi_{m+1,j}})c_j\sigma_i e^{\lambda r\tau}\right]\right\}|z_i(t)|^r\le 0
\end{aligned}
$$

So $V(t)\le V(0)$. From Eq.(13) we have

$$
V(0)=\sum_{i=1}^{n}c_i\left[|\Phi_i(0)|^r+\sum_{j=1}^{n}\sigma_j(|\alpha_{ij}|^{r\gamma_{m+1,j}}+|\beta_{ij}|^{r\xi_{m+1,j}})\times\int_{-\tau_{ij}}^{0}|\Phi_j(s)|^r e^{\lambda r(s+\tau)}ds\right]
$$

$$\leq \sum_{i=1}^{n} \left[c_i + c_i \sum_{j=1}^{n} \sigma_j (|\alpha_{ij}|^{r\gamma_{m+1,j}} + |\beta_{ij}|^{r\xi_{m+1,j}}) \tau e^{\lambda r \tau} \right] \|\Phi\|_\tau^r$$

and it is obvious that

$$V(t) \geq \sum_{i=1}^{n} c_i |z_i(t)|^r e^{\lambda r t} \geq c_i |z_i(t)|^r e^{\lambda r t}$$

Therefore

$$c_i |z_i(t)|^r e^{\lambda r t} \leq \sum_{i=1}^{n} \left[c_i + c_i \sum_{j=1}^{n} \sigma_j (|\alpha_{ij}|^{r\gamma_{m+1,j}} + |\beta_{ij}|^{r\xi_{m+1,j}}) \tau e^{\lambda r \tau} \right] \|\Phi\|_\tau^r$$

That is

$$\|z(t)\|_\infty \leq M e^{-\lambda t} \|\Phi\|_\tau, \quad t \geq 0$$

where

$$M = \left\{ \frac{1}{\underline{c}} \left[\sum_{i=1}^{n} c_i + \sum_{i=1}^{n} \sum_{j=1}^{n} c_i \sigma_j (|\alpha_{ij}|^{r\gamma_{m+1,j}} + |\beta_{ij}|^{r\xi_{m+1,j}}) \tau e^{\lambda r \tau} \right] \right\}^{\frac{1}{r}} \geq 1.$$

This implies that the equilibrium point O of system (11) is GES, namely the equilibrium point x^* of system (11) is GES . The proof is completed .

Remark 1. In Theorem 2, if we don't consider fuzzy AND and fuzzy OR operations, it becomes traditional cellular neural networks. The results in References [7] are the corollary of theorem 2. Therefore the results of this paper are new and extend the previous known publication.

Remark 2. In this paper, we don't need delay with deferential and continuous. Clearly, the proposed results are different from those in reference [16].

4 Conclusion

By constructing a new Lyapunov functional, employing the inequality (2) and the Homeomorphism theory, we have derived a new conditions of the existence,uniqueness of the equilibrium point and its GES for the fuzzy cellular neural networks with bounded unknown delays. In contrast with the previous paper, these conditions are independent of delays, which need not be differentiable.

Acknowledgments. This work is partially supported by the Scientific Research Foundation of Hunan Provincial Education Department(No. 08C261) and partially supported by Hunan Institute of Technology (No.HGQ0705).

References

1. Chua, L.O., Yang, L.: Cellular Neural Networks: Theory. IEEE Transactions on Circuits and Systems I 35, 1257–1272 (1988)
2. Chua, L.O., Yang, L.: Cellular Neural Networks: Application. IEEE Transactions on Circuits and Systems I 35, 1273–1290 (1988)
3. Cao, J.D.: Global Stability Analysis In Delayed Cellular Networks. Physical Review E 59, 5940–5944 (1999)
4. Cao, J.D., Zhou, D.M.: Stability Analysis of Delayed Cellular Neural Networks. Neural Networks 11, 1601–1605 (1998)
5. Li, X.M., Huang, L.H., Zhu, H.Y.: Global Stability of Cellular Neural Networks with Constant and Variable Delays. Nonliner Analysis 53, 319–333 (2003)
6. Huang, H., Cao, J.D., Wang, J.: Global Stability and Periodic Solutions of Recurrent Neural Networks with Delays. Physics Letters A 298, 393–404 (2002)
7. Zhao, H.Y., Cao, J.D.: New Conditons for Global Exponential Stability of Cellular Neural Networks with Delays. Neural Networks 18, 1332–1340 (2005)
8. Zhang, J.Y.: Global Stability Analysis in Delayed Cellular Neural Networks. Computers and Mathematics with applications 45, 1707–1720 (2003)
9. Zhang, J.Y.: Absolute Stability Analysis in Cellular Neural Networks with Variable Delays and Unbounded Delay. Computers and Mathematics with applications 47, 183–194 (2004)
10. Jiang, H.J., Teng, Z.D.: Global Exponential Stability of Cellular Neural Networks with Time-Varing Coeffiences And Delays. Neural Networks 17, 1415–1425 (2004)
11. Huang, T.W., Cao, J.D., Li, C.D.: Necessary and sufficient Condition for the Absolute Exponential Stability of a Class of Neural Networks With Finite Delay. Physics Letters A 352, 94–98 (2006)
12. Yang, T., Yang, L.B.: The Global Stability Of Fuzzy Cellular Neural Networks. IEEE Transactions on Circuits and Systems I 43, 880–883 (1996)
13. Yang, T., Yang, L.B., Wu, C.W., Chua, L.O.: Fuzzy Cellular Neural Networks: Theory. In: Proc. IEEE Int. Workshop Cellular Neural Networks Appl., pp. 181–186 (1996)
14. Huang, T.W.: Exponential Stability of Fuzzy Cellular Neural Networks with Distributed Delay. Physics Letters A 351, 48–52 (2006)
15. Zhang, Q.H., Xiang, R.G.: Global Asymptotic Stability of Fuzzy Cellular Neural Networks with Time-Varing Delays. Physics Letters A 372, 3971–3977 (2008)
16. Liu, Y.Q., Tang, W.S.: Exponential Stability of Fuzzy Cellular Neural Networks with Costant and Time-Varying Delays. Physics Letters A 323, 224–233 (2004)
17. Huang, T.W.: Exponential Stability of Delayed Fuzzy Cellular Neural Networks with Diffusion. Chaos, Solitons and Fractals 31, 658–664 (2007)
18. Forti, M., Tesi, A.: New Conditions ffor Global Stability of Neural Networks with Application To Linear And Quadratic Programming Problems. IEEE Transactions on Circuits and Systems I 42, 354–366 (1995)

Finite-Time Boundedness Analysis of a Class of Neutral Type Neural Networks with Time Delays

Jianfeng Wang, Jigui Jian, and Peng Yan

Institute of Nonlinear and Complex Systems, China Three Gorges University, Yichang 443002, China
wjfwjffff@126.com, jiguijian65@hotmail.com

Abstract. In this paper, the finite time boundedness (FTB) for certain and uncertain neutral type neural networks are investigated. The concept of FTB for time delay system is extended first. Then, based on the Lyapunov stability theory and linear matrix inequality (LMI) technique, some sufficient conditions are derived to guarantee FTB, and our results are less conservative than exiting results. Finally, some examples are given to demonstrate the effectiveness and improvement of the proposed results.

Keywords: Finite time boundness, Neural network, Neutral type, Time-delay, Linear matrix inequality.

1 Introduction

In recent years, there exists an extensive literature on various aspects of different neural networks with or without time delays, such as the Hopfield neural networks, cellular neural networks, bidirectional associative memory neural networks and so on. Time delays are unavoidably encountered in implementation of artificial networks. As is well known, time delays may degrade system performance and induce oscillation in a network, causing instability. So, it is very important to study time delays effects on stability and convergent dynamics of neural networks. It has received considerable attention in the past decades [1-6]. In many practical applications, some systems may be unstable, in this case, the main concern is the behavior of the system over a fixed finite time interval, it could be required that the trajectories of the controlled system do not exceed given bounds. In order to deal with this problem, Peter Dorato [7] presented the concept of finite-time stability (FTS). After that, Amato [8-11] extended the definition of FTS to the definition of finite-time boundedness (FTB), which takes into external constant disturbances. Recently, there are many papers about FTB analysis, such as [12, 13].

As far as we know, there are two types of time delays: discrete and neutral type. The first one contains delays only in its states. The second one contains delays in both its states and the derivatives of its states. So far, there are only

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 395–404, 2009.

a few papers that have taken neutral type phenomenon into account in delay neural networks [14-20]. However, there is few result about FTB for certain or uncertain neutral type neural networks in exiting works. Based on the above discussion, in this paper, we further extend the results of FTB to certain and uncertain neural networks with time delays described by nonlinear delay differential equations of the neutral type. Some sufficient conditions are presented to ensure the delayed neural networks are FTB. The conditions can be reduced to a feasibility problem. Finally, some examples are given to demonstrate the effectiveness and improvement of the proposed results.

Throughout this paper, for real symmetric matrices X and Y, the notation $X \geq Y$(respectively, $X > Y$) means that the matrix $X - Y$ is positive semi-definite (respectively, positive definite). The superscript T presents the transpose. We use $\lambda_{min}(\cdot)$ and $\lambda_{max}(\cdot)$ to denote the minimum and maximum eigenvalue of a real symmetric matrix, respectively. The notation denotes $\|x\|$ a vector norm defined by $\|x\| = (\sum_{i=1}^{n} x_i^2)^{\frac{1}{2}}$, where x is a vector, while $\|A\|$ denotes a matrix norm defined by $\|A\| = (\lambda_{max}(A^T A))^{\frac{1}{2}}$, where A is a matrix. Matrices, if not explicitly stated, are assumed to have compatible dimensions.

2 Problem Formulation and Preliminaries

Consider the following delayed neural networks with norm-bounded parametric uncertainties which is described by a nonlinear neutral delay differential equation

$$\dot{u}(t) = -Au(t)+(W+\triangle W)g(u(t))+(W_1+\triangle W_1)g(u(t-\tau))+W_2\dot{u}(t-\tau)+J, \quad (1)$$

$$u(t) = \varphi(t), \dot{u}(t) = \phi(t), t \in [-\tau, 0]. \quad (2)$$

Where $u(t) = (u_1(t), u_2(t), \cdots, u_n(t))^T$ is the state vector associated with n neurons, and $g(u(t)) = (g_1(u_1(t)), g_2(u_2(t)), \cdots, g_n(u_n(t)))^T$ denotes the neuron activation function. The diagonal matrix $A = diag\{a_1, a_2, \cdots, a_n\}$. A, W, W_1 and W_2 are interconnection weight matrices. $\varphi(t), \phi(t)$ denote the initial condition function that are continuously differentiable on $[-\tau, 0]$. J is a constant external input vector. $\triangle W, \triangle W_1$ are parametric uncertainties. The scalar $\tau > 0$ represents the transmission delay. Throughout the paper we assume that the activation function satisfies the following assumption.

Assumption 1. The activation function $g(u)$ satisfies

$$0 \leq \frac{g_i(\xi_1) - g_i(\xi_2)}{\xi_1 - \xi_2} \leq k_i, i = 1, 2, \cdots, n \quad (3)$$

for any $\xi_1, \xi_1 \in R$, where real constant $k_i > 0$ for $i = 1, 2, \cdots, n$. The uncertainties $\triangle W, \triangle W_1$ are defined by

$$\triangle W = HFE, \triangle W_1 = H_1 F_1 E_1, \quad (4)$$

where H, H_1, E, E_1 are known constant matrices of appropriate dimensions, and F, F_1 are unknown matrices representing the parameter uncertainties, which satisfy

$$F^T F \leq I, F_1^T F_1 \leq I. \tag{5}$$

Assume u^* is an equilibrium point of (1). Let $x(t) = u(t) - u^*$, then it is easy to see that system (1) can be transformed to

$$\dot{x}(t) = -Ax(t) + (W + \triangle W)f(x(t)) + (W_1 + \triangle W_1)f(x(t-\tau)) + W_2\dot{x}(t-\tau), \tag{6}$$

where $x(t) = (x_1(t), x_2(t), \cdots, x_n(t))^T, f(x(t)) = g(x(t) + u^*) - g(u^*), f(x(t)) = (f_1(x_1(t)), f_2(x_2(t)), \cdots, f_n(x_n(t)))^T$. Then, it is easy to see that $f_i(0) = 0 (i = 1, 2, \cdots, n)$ and $f_i(\cdot)$ satisfies (3), that is

$$0 \leq \frac{f_i(\xi_1) - f_i(\xi_2)}{\xi_1 - \xi_2} \leq k_i, i = 1, 2, \cdots, n. \tag{7}$$

The problem to be addressed in this paper is to develop some sufficient conditions which guarantee that the state of time delay neural networks with norm-bounded parametric uncertainties is finite time boundedness (FTB).

Definition 1. System (6) is said to be finite time boundedness(FTB) with respect to (c_1, c_2, T), if

$$\sup_{t\in[-\tau,0]} \|\varphi(t)\|^2 \leq c_1^2 \Longrightarrow \|x(t)\|^2 \leq c_2^2, \forall t \in [0, T]. \tag{8}$$

Lemma 1. (Schur complement) Given constant symmetric matrices $\Sigma_1, \Sigma_2, \Sigma_3$, where $\Sigma_1 = \Sigma_1^T$ and $0 < \Sigma_2 = \Sigma_2^T$, then $\Sigma_1 + \Sigma_3^T \Sigma_2^{-1} \Sigma_3 < 0$ if and only if

$$\begin{pmatrix} \Sigma_1 & \Sigma_3^T \\ \Sigma_3 & -\Sigma_2 \end{pmatrix} < 0, or \begin{pmatrix} -\Sigma_2 & \Sigma_3 \\ \Sigma_3^T & \Sigma_1 \end{pmatrix} < 0.$$

Lemma 2. For matrices Y, D and E of appropriate dimensions, where Y is a symmetric matrix, then $Y + DFE + E^T F^T D^T < 0$ holds for all matrix F satisfying $F^T F \leq I$, if and only if there exist a constant $\epsilon > 0$, such that $Y + \epsilon DD^T + \epsilon^{-1} E^T E < 0$ holds

3 Main Results

Firstly, we consider a special case of system (6) with certain parameters, i.e., it becomes

$$\dot{x}(t) = -Ax(t) + Wf(x(t)) + W_1 f(x(t-\tau)) + W_2\dot{x}(t-\tau). \tag{9}$$

Theorem 1. System (9) is FTB with respect to (c_1, c_2, T) for any delay $0 < \tau \leq \bar{\tau}$, if there exsit matrices $P > 0, Q_1 > 0, Q_2 > 0, Q_3 > 0$ and two diagonal matrices $S > 0, Y > 0$, and a scalar α such that the following conditions hold:

$$\prod = \begin{pmatrix} -\bar{Q} & -\bar{Q}A & \bar{Q}W & \bar{Q}W_1 & \bar{Q}W_2 & 0 \\ * & \Phi_1 & \Psi_1 & \Psi_2 & PW_2 & 0 \\ * & * & \Phi_2 & 0 & 0 & 0 \\ * & * & * & \Phi_3 & 0 & -\bar{\tau}Y \\ * & * & * & * & -Q_2 & 0 \\ * & * & * & * & * & -\bar{\tau}Q_3 \end{pmatrix} < 0, \tag{10}$$

and

$$\frac{c_1^2[\lambda_{max}(P) + \bar{\tau}\lambda_{max}(Q_1)\lambda_{max}(\Sigma^T\Sigma)] + \mu^2[\bar{\tau}\lambda_{max}(Q_2) + \bar{\tau}^2\lambda_{max}(Q_3)]}{e^{-\alpha T}\lambda_{min}(P)} < c_2^2, \tag{11}$$

where $\Phi_1 = -PA - AP - \alpha P, \Phi_2 = Q_1 - S, \Phi_3 = -Q_1 - Y\Sigma^{-1} - \Sigma^{-1}Y, \Psi_1 = PW + \frac{1}{2}S\Sigma, \Psi_2 = PW_1 + Y, \bar{Q} = Q_2 + \bar{\tau}Q_3, \sup_{t\in[-\tau,0]} \|\varphi(t)\|^2 \le c_1^2, \Sigma = diag\{k_1, k_2, \cdots, k_n\}, \sup_{t\in[-\tau,0]} \|\phi(t)\|^2 = \mu^2$, and $k_i > 0 (1, 2, \cdots, n)$ are given in Assumption 1.

Proof. Let

$$\begin{aligned} V(x(t)) = {} & x^T(t)Px(t) + \int_{t-\tau}^{t} f^T(x(s))Q_1 f(x(s))ds \\ & + \int_{t-\tau}^{t} \dot{x}^T(s)Q_2\dot{x}(s)ds + \int_{-\tau}^{0}\int_{t+\beta}^{t} \dot{x}^T(s)Q_3\dot{x}(s)dsd\beta, \end{aligned} \tag{12}$$

where $x(t) = x(t+\theta), -\tau \le \theta \le 0$. Then, the time-derivative of $V(x(t))$ along the solution of (9) is

$$\begin{aligned} \dot{V}(x(t)) = {} & 2x^T(t)P[-Ax(t) + Wf(x(t)) + W_1 f(x(t-\tau)) + W_2\dot{x}(t-\tau)] \\ & + f^T(x(t))Q_1 f(x(t)) - f^T(x(t-\tau))Q_1 f(x(t-\tau)) + \dot{x}^T(t)Q_2\dot{x}(t) \\ & - \dot{x}^T(t-\tau)Q_2\dot{x}(t-\tau) + \tau\dot{x}^T(t)Q_3\dot{x}(t) - \int_{t-\tau}^{t} \dot{x}^T(s)Q_3\dot{x}(s)ds. \end{aligned} \tag{13}$$

By the Newton-Leibniz formula, it is easy to see that

$$x(t-\tau) = x(t) - \int_{t-\tau}^{t} \dot{x}(s)ds. \tag{14}$$

It follows from (13) and (14) that

$$\begin{aligned} \dot{V}(x(t)) = {} & \tfrac{1}{\tau}\int_{t-\tau}^{t}\{2x^T(t)P[-Ax(t) + Wf(x(t)) + W_1 f(x(t-\tau)) + W_2\dot{x}(t-\tau)] \\ & + f^T(x(t))Q_1 f(x(t)) - f^T(x(t-\tau))Q_1 f(x(t-\tau)) \\ & + \dot{x}^T(t)(Q_2 + \tau Q_3)\dot{x}(t) - \dot{x}^T(t-\tau)Q_2\dot{x}(t-\tau) - \tau\dot{x}^T(t)Q_3\dot{x}(t) \\ & + 2f^T(x(t-\tau))Y[x(t) - x(t-\tau)] - 2f^T(x(t-\tau))\tau Y\dot{x}(s) \\ & - f^T(x(t))Sf(x(t)) + f^T(x(t))Sf(x(t))\}ds. \end{aligned} \tag{15}$$

Noting that $Y > 0$ and $S > 0$ are diagonal matrices and using (7), we can obtain

$$-f^T(x(t-\tau))Yx(t-\tau) \le -f^T(x(t-\tau))Y\Sigma^{-1}f(x(t-\tau)), \tag{16}$$

$$f^T(x(t))Sf(x(t)) \le f^T(x(t))S\Sigma x(t). \tag{17}$$

Taking (16) and (17) into (15), we can obtain the following matrix inequality

$$\dot{V}(x(t)) \leq \frac{1}{\tau}\int_{t-\tau}^{t} \xi^T(t,s)\Omega\xi(t,s)ds, \tag{18}$$

where $\xi(t,s) = [x^T(t), f^T(x(t)), f^T(x(t-\tau)), \dot{x}^T(t-\tau), \dot{x}^T(s)]^T$,

$$\Omega = \begin{pmatrix} -PA-AP & -\bar{\Psi}_1 & \bar{\Psi}_2 & \bar{\Psi}_3 & 0 \\ * & \bar{\Phi}_2 & \Gamma_1 & \Gamma_2 & 0 \\ * & * & \bar{\Phi}_3 & \Gamma_3 & -\tau Y \\ * & * & * & \bar{\Phi}_4 & 0 \\ * & * & * & * & -\tau Q_3 \end{pmatrix}, \tag{19}$$

$\bar{\Psi}_1 = (P - A\bar{Q})W + \frac{1}{2}S\Sigma, \bar{\Psi}_2 = (P - A\bar{Q})W_1 + Y, \bar{\Psi}_3 = (P - A\bar{Q})W_2, \bar{\Phi}_2 = Q_1 - S + W^T\bar{Q}W, \bar{\Phi}_3 = -Q_1 - Y\Sigma^{-1} - \Sigma^{-1}Y + W_1^T\bar{Q}W_1, \bar{Q} = Q_2 + \tau Q_3, \bar{\Phi}_4 = W_2^T\bar{Q}W_2 - Q_2, \Gamma_1 = W^T\bar{Q}W_1, \Gamma_2 = W^T\bar{Q}W_2, \Gamma_3 = W_1^T\bar{Q}W_2$. Then, pre-multiply and post-multiply the inequality (10) by the matrix $diag\{\bar{Q}^{-1}, I, I, I, I, I\}$, for any delay $0 < \tau \leq \bar{\tau}$, we can obtain that (10) is equivalent to

$$\begin{pmatrix} -\bar{Q}^{-1} & -A & W & W_1 & W_2 & 0 \\ * & \Phi_1 & \Psi_1 & \Psi_2 & PW_2 & 0 \\ * & * & \Phi_2 & 0 & 0 & 0 \\ * & * & * & \Phi_3 & 0 & -\tau Y \\ * & * & * & * & -Q_2 & 0 \\ * & * & * & * & * & -\tau Q_3 \end{pmatrix} < 0, \tag{20}$$

by Lemma 1, (20) is equivalent to

$$\begin{pmatrix} \Phi_1 & \bar{\Psi}_1 & \bar{\Psi}_2 & \bar{\Psi}_3 & 0 \\ * & \bar{\Phi}_2 & \Gamma_1 & \Gamma_2 & 0 \\ * & * & \bar{\Phi}_3 & \Gamma_3 & -\tau Y \\ * & * & * & \bar{\Phi}_4 & 0 \\ * & * & * & * & -\tau Q_3 \end{pmatrix} < 0, \tag{21}$$

then, it follows from (21) and (19), we have

$$\begin{pmatrix} -PA-AP & \bar{\Psi}_1 & \bar{\Psi}_2 & \bar{\Psi}_3 & 0 \\ * & \bar{\Phi}_2 & \Gamma_1 & \Gamma_2 & 0 \\ * & * & \bar{\Phi}_3 & \Gamma_3 & -\tau Y \\ * & * & * & \bar{\Phi}_4 & 0 \\ * & * & * & * & -\tau Q_3 \end{pmatrix} < \begin{pmatrix} \alpha P & 0 & 0 & 0 & 0 \\ * & 0 & 0 & 0 & 0 \\ * & * & 0 & 0 & 0 \\ * & * & * & 0 & 0 \\ * & * & * & * & 0 \end{pmatrix}, \tag{22}$$

from (18) and (22), we can get

$$\begin{aligned} \dot{V}(x(t)) &< \alpha x^T(t)Px(t) \\ &\leq \alpha[x^T(t)Px(t) + \textstyle\int_{t-\tau}^{t} f^T(x(s))Q_1 f(x(s))ds \\ &\quad + \textstyle\int_{t-\tau}^{t} \dot{x}^T(s)Q_2\dot{x}(s)ds + \int_{-\tau}^{0}\int_{t+\beta}^{t} \dot{x}^T(s)Q_3\dot{x}(s)dsd\beta] \\ &= \alpha V(x(t)). \end{aligned} \tag{23}$$

Multiplying (23) by $e^{-\alpha t}$, we can obtain

$$\frac{d}{dt}(e^{-\alpha t}V) < 0, \tag{24}$$

integrating (24) from 0 to t, with $t \in [0,T]$, we have $e^{-\alpha t}V(x(t)) < V(x(0))$, then

$$\begin{aligned} V(x(t)) &< e^{\alpha t}V(x(0)) \\ &= e^{\alpha t}[x^T(0)Px(0) + \textstyle\int_{-\tau}^{0} f^T(x(s))Q_1 f(x(s))ds \\ &\quad + \textstyle\int_{-\tau}^{0} \dot{x}^T(s)Q_2\dot{x}(s)ds + \int_{-\tau}^{0}\int_{\beta}^{0} \dot{x}^T(s)Q_3\dot{x}(s)dsd\beta] \\ &\le e^{\alpha T}[\lambda_{max}(P)x^T(0)x(0) + \lambda_{max}(Q_1)\lambda_{max}(\Sigma^T\Sigma)\textstyle\int_{-\tau}^{0} f^T(x(s))f(x(s))ds \\ &\quad + \lambda_{max}(Q_2)\textstyle\int_{-\tau}^{0} \dot{x}^T(s)\dot{x}(s)ds + \lambda_{max}(Q_3)\int_{-\tau}^{0}\int_{\beta}^{0} \dot{x}^T(s)\dot{x}(s)dsd\beta]. \end{aligned} \tag{25}$$

Noting that

$$x^T(t)Px(t) \le V(x(t)) \Longrightarrow \lambda_{max}(P)x^T(t)x(t) \le V(x(t)). \tag{26}$$

Using (25), (26) and $0 < \tau \le \bar{\tau}$, we have

$$\|x(t)\|^2 \le \frac{c_1^2(\lambda_{max}(P) + \bar{\tau}\lambda_{max}(Q_1)\lambda_{max}(\Sigma^T\Sigma)) + \mu^2(\bar{\tau}\lambda_{max}(Q_2) + \bar{\tau}^2\lambda_{max}(Q_3))}{e^{-\alpha T}\lambda_{min}(P)}, \tag{27}$$

where $\sup_{t\in[-\tau,0]}\|\varphi(t)\|^2 \le c_1^2$, $\sup_{t\in[-\tau,0]}\|\phi(t)\|^2 = \mu^2$. Then condition (11) implies $\|x(t)\|^2 \le c_2^2$ for all $t \in [0,T]$. Therefore, the proof is completed.

Theorem 2. System (6) is FTB with respect to (c_1, c_2, T) for any delay $0 < \tau \le \bar{\tau}$, if there exsit matrices $P > 0, Q_1 > 0, Q_2 > 0, Q_3 > 0$, constant $\varepsilon_1 > 0, \varepsilon_2 > 0$ and two diagonal matrices $S > 0, Y > 0$, and a scalar α such that the following conditions hold:

$$\Theta = \begin{pmatrix} -\varepsilon_1 I & 0 & H^T\bar{Q} & H^TP & 0 & 0 & 0 & 0 \\ * & -\varepsilon_2 I & H_1^T\bar{Q} & H_1^TP & 0 & 0 & 0 & 0 \\ * & * & -\bar{Q} & -\bar{Q}A & \bar{Q}W & \bar{Q}W_1 & \bar{Q}W_2 & 0 \\ * & * & * & \Phi_1 & \Psi_1 & \Psi_2 & PW_2 & 0 \\ * & * & * & * & \Phi_2 + \varepsilon_1 E^TE & 0 & 0 & 0 \\ * & * & * & * & * & \Phi_3 + \varepsilon_2 E_1^TE_1 & 0 & -\bar{\tau}Y \\ * & * & * & * & * & * & -Q_2 & 0 \\ * & * & * & * & * & * & * & -\bar{\tau}Q_3 \end{pmatrix} < 0, \tag{28}$$

and

$$\frac{c_1^2[\lambda_{max}(P) + \bar{\tau}\lambda_{max}(Q_1)\lambda_{max}(\Sigma^T\Sigma)] + \mu^2[\bar{\tau}\lambda_{max}(Q_2) + \bar{\tau}^2\lambda_{max}(Q_3)]}{e^{-\alpha T}\lambda_{min}(P)} < c_2^2, \tag{29}$$

where $c_1^2, \bar{Q}, \mu^2, \Phi_1, \Phi_2, \Phi_3, \Psi_1, \Psi_2, \Sigma$ are given in Theorem 1.

Proof. To proof that the system (6) is FTB, according to Theorem 1, we only need proof

$$\Lambda = \begin{pmatrix} -\bar{Q} & -\bar{Q}A & \bar{Q}\bar{W} & \bar{Q}\bar{W}_1 & \bar{Q}\bar{W}_2 & 0 \\ * & \Phi_1 & \tilde{\Psi}_1 & \tilde{\Psi}_2 & PW_2 & 0 \\ * & * & \Phi_2 & 0 & 0 & 0 \\ * & * & * & \Phi_3 & 0 & -\bar{\tau}Y \\ * & * & * & * & -Q_2 & 0 \\ * & * & * & * & * & -\bar{\tau}Q_3 \end{pmatrix} < 0, \tag{30}$$

where $\bar{W} = W + \triangle W, \bar{W}_1 = W_1 + \triangle W_1, \tilde{\Psi}_1 = P\bar{W} + \frac{1}{2}S\Sigma, \tilde{\Psi}_2 = P\bar{W}_1 + Y$. By (4) and (10), (30) can be rewritten as

$$\Pi + M_1 F M_2 + M_2^T F^T M_1^T + M_3 F_1 M_4 + M_4^T F_1^T M_3^T < 0, \tag{31}$$

where P is given in (10), $M_1^T = \begin{pmatrix} H^T\bar{Q} & H^T P & 0 & 0 & 0 & 0 \end{pmatrix}$, $M_2 = \begin{pmatrix} 0 & 0 & E & 0 & 0 & 0 \end{pmatrix}$, $M_3^T = \begin{pmatrix} H_1^T\bar{Q} & H_1^T P & 0 & 0 & 0 & 0 \end{pmatrix}$, $M_4 = \begin{pmatrix} 0 & 0 & 0 & E_1 & 0 & 0 \end{pmatrix}$. With (5), by Lemma 2, (31) is equivalent to

$$\Pi + \varepsilon_1^{-1} M_1 M_1^T + \varepsilon_1 M_2^T M_2 + \varepsilon_2^{-1} M_3 M_3^T + \varepsilon_2 M_4^T M_4 < 0, \tag{32}$$

where $\varepsilon_1 > 0, \varepsilon_2 > 0$ are certain constants. By Lemma 1, (32) is equivalent to $\Theta < 0$ in (28). Then, we can obtain that $\Lambda < 0$. So, the system (6) is FTB. Therefore, the proof is completed.

Remark 1. By above discussion, (10) is equivalent to (21), so when $\alpha = 0$, the condition (10) in Theorem 1 is equivalent to the condition (9) of Theorem 1 in [14], which is the sufficient condition for the globally exponential stability of neutral systems. In other words, the results in this paper can also be the criteria of the globally exponential stability of this class of neutral systems.

Remark 2. Let $W_2 = 0$ in Theorem 1 and Theorem 2, then the results in this paper can also be the criteria of FTB for the time delayed systems, and our results are less conservative than exiting results, which can be illustrated in Example 2.

4 Numerical Examples

In this section, examples are given to show the validity of our results.

Example 1. Consider system (6) with parameter as $W = \begin{pmatrix} 1 & -3.1 & -2 \\ 3.1 & -1 & -2.5 \\ 2 & -1.4 & 0.6 \end{pmatrix}$, $W_1 = \begin{pmatrix} -1 & -1 & 3.8 \\ 0.1 & 0.7 & -2.4 \\ -0.7 & 1.3 & -1.8 \end{pmatrix}$, $W_2 = \begin{pmatrix} 0.2 & 0.1 & 0.4 \\ -0.1 & 0.3 & 0.1 \\ -0.3 & 0.1 & 0.4 \end{pmatrix}$, $A = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 9 \end{pmatrix}$, $H = \begin{pmatrix} 0.2 & 0.5 & 0.2 \\ 0.1 & -0.3 & 0.1 \\ 0.4 & 0.1 & 0.2 \end{pmatrix}$, $H_1 = \begin{pmatrix} -0.4 & 0.3 & 0.3 \\ 0.3 & 0.1 & 0.4 \\ 0.1 & 0.2 & 0.1 \end{pmatrix}$, $\Sigma = \begin{pmatrix} 0.1 & 0 & 0 \\ 0 & 0.4 & 0 \\ 0 & 0 & 0.07 \end{pmatrix}$, $E = H, E_1 = H_1, c_1 = \mu = 1$.

Giving $\alpha = 0.2$, by Theorem 2, we can obtain $0 < \tau \leq \bar{\tau} = 1.2212$, and the solutions as follows

$$P = \begin{pmatrix} 3.004 & 0.9835 & -2.2900 \\ 0.9835 & 3.6519 & -1.0467 \\ -2.2900 & -1.0467 & 8.4949 \end{pmatrix}, Q_1 = \begin{pmatrix} 14.4012 & 7.5704 & 18.8816 \\ 7.5704 & 23.7533 & 1.6504 \\ 18.8816 & 1.6504 & 31.5849 \end{pmatrix},$$

$$Q_2 = \begin{pmatrix} 0.6310 & 0.2356 & 0.0348 \\ 0.2356 & 1.1929 & 0.2716 \\ 0.0348 & 0.2716 & 0.4338 \end{pmatrix}, Q_3 = \begin{pmatrix} 0.2546 & 0.2714 & -0.2509 \\ 0.2714 & 0.7892 & -0.4885 \\ -0.2509 & -0.4885 & 0.5145 \end{pmatrix},$$

$$S = \begin{pmatrix} 83.0850 & 0 & 0 \\ 0 & 83.0850 & 0 \\ 0 & 0 & 083.0850 \end{pmatrix}, Y = \begin{pmatrix} 2.0771 & 0 & 0 \\ 0 & 2.0771 & 0 \\ 0 & 0 & 2.0771 \end{pmatrix},$$

$\varepsilon_1 = 10.4143, \varepsilon_2 = 15.7534$. When $T = 0.9$, it is obtained that the minimum $c_2 = 3.6553$ from (29). So system (6) is FTB with respect to (c_1, c_2, T).

Example 2. Consider the system

$$\dot{x}(t) = -Ax(t) + (W + \triangle W)f(x(t)) + (W_1 + \triangle W_1)f(x(t - \tau))$$

with parameter as:

$$A = \begin{pmatrix} 0.2 & 0 \\ 0 & 0.4 \end{pmatrix}, W = \begin{pmatrix} 1 & -0.1 \\ -0.1 & 1 \end{pmatrix}, W_1 = \begin{pmatrix} -1 & 0.7 \\ 1 & -2 \end{pmatrix}, H = H_1 = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix},$$

$$E = E_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \Sigma = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.3 \end{pmatrix}, c_1 = \mu = 1.$$

Giving $\alpha = 0.1$, by Theorem 2, we can obtain $0 < \tau \leq \bar{\tau} = 1.6379$, and the solutions as follows

$$P = \begin{pmatrix} 5.4041 & 1.3414 \\ 1.3414 & 6.4172 \end{pmatrix}, Q_1 = \begin{pmatrix} 41.0526 & 4.5805 \\ 4.5805 & 14.0863 \end{pmatrix}, Q_2 = \begin{pmatrix} 0.0412 & 0.0161 \\ 0.0161 & 0.0065 \end{pmatrix},$$

$$Q_3 = \begin{pmatrix} 2.0654 & 0.0387 \\ 0.0387 & 4.2514 \end{pmatrix}, S = \begin{pmatrix} 85.2453 & 0 \\ 0 & 85.2453 \end{pmatrix}, Y = \begin{pmatrix} 10.8433 & 0 \\ 0 & 10.8433 \end{pmatrix},$$

$\varepsilon_1 = 2.8096, \varepsilon_2 = 1.7766$. When $T = 0.6$, it is obtained that the minimum $c_2 = 2.4347$, so system (6) is FTB with respect to (c_1, c_2, T). But when we apply this example to the Theorem 1 in [13], the LMIs are not strictly feasible. So it is very obvious that the proposed results in this paper are less conservative.

5 Conclusion

The paper has mainly investigated the problem of FTB for a class of neutral type neural networks. Based on the Lyapunov stability theory and linear matrix inequality (LMI) technique, some sufficient conditions are derived to guarantee FTB, and our results are less conservative than exiting results[13]. Two examples are given to demonstrate the effectiveness and improvement of the proposed results.

Acknowledgments

This work was partially supported by the Scientific Innovation Team Project T200809 and the Scientific Research Projects D20081306 of Hubei Provincial Department of Education, and the Hubei Provincial Natural Science Foundation 2008CDB316 and the Doctoral Pre-Research Foundation of China Three Gorges University.

References

1. Liao, X.F., Chen, G., Sanchez, E.: LMI-Based Approach for Asymptotically Stability Analysis of Delayed Neural Networks. IEEE Trans. Circ. Syst (I) 49, 1033–1039 (2002)
2. Liao, X.F., Chen, G., Sanchez, E.: Delay-Dependent Exponential Stability Analysis of Delayed Neural Network. Neural Networks 15, 855–866 (2002)
3. Singh, V.: A Generalized LMI-Based Approach to the Global Asymptotic Stability of Delayed Cellular Neural Networks. IEEE Trans Neural Networks 15, 223–225 (2004)
4. Ou, O.: Global Robust Exponential Stability of Delayed Neural Networks: an LMI Approach. Chaos, Solitons and Fractals 22, 1742–1748 (2006)
5. Xu, S., James, L.: A New Approach to Exponential Stability Analysis of Neural Networks with Time-Varying Delays. Neural Networks 19, 76–83 (2006)
6. Sun, C., Zhang, K., Fei, S., Feng, C.: On Exponential Stability of Delayed Neural Networks with a General Class of Activation Functions. Physics Letters A (298), 122–132 (2002)
7. Dorat, P.: Short Time Stability in Linear Time-Varying System. In: Proc. IRE International Convention Record Part 4, pp. 83–87 (1961)
8. Amato, F., Ariola, M., Dorato, P.: Finite-Time Control of Linear Systems Subject to Parametric Uncertainties and Disturbances. Automatica 37, 1459–1463 (2001)
9. Amato, F., Ariola, M., Abdallah, C.T., Dorato, P.: Dynamic Output Feedback Finite-Time Control of LTI Systems Subject to Parametric Uncertainties and Disturbances. In: Proc. European Control Conference Karlsruhe, CA, pp. 1176–1180 (1999)
10. Amato, F., Ariola, M., Cosentino, C.: Finite-Time Control of Linear Time-Varying Systems via Output Feedback. In: American Control Conference (2005)
11. Amato, F., Ariola, M., Dorate, P.: Finite-Time Stabilization via Dynamic Output Feedback. Automatic 42, 337–342 (2006)
12. Shen, Y.J., Li, C.C.: LMI-Based Finite-Time Boundedness Analysis of Neural Networks with Parametric Uncertainties. Neurocomputing 71, 502–507 (2008)
13. Shen, Y.J., Zhu, L.: Finite-Time Boundness Analysis of Uncertain Neural Networks with Time Delay: an LMI Approach. In: Liu, D., Fei, S., Hou, Z.-G., Zhang, H., Sun, C. (eds.) ISNN 2007. LNCS, vol. 4491, pp. 904–909. Springer, Heidelberg (2007)
14. Xu, S.Y., James, L.: Delay-Dependent Exponential Stability for a Class of Neural Networks with Time Delays. Journal of Computational and Applied Mathematics 183, 16–28 (2005)
15. Cao, J., Zhong, S., Hu, Y.: Global Stability Analysis for a Class of Neural Networks with Time Varying Delays and Control Input. Appl. Math. Comput. 189, 1480–1490 (2007)

16. Bai, C.Z.: Global Stability of Almost Periodic Solutions of Hopfield Neural Networks with Neutral Time-Varying Delays. Appl. Math. Comput. (in Press, 2008), doi:10.1016/j.amc. 2008.04.002
17. Park, J.H., Kwon, O.M., Lee, S.M.: LMI Optimization Approach on Stability for Delayed Neural Networks of Neutral-Type. Appl. Math. Comput. 196, 236–244 (2008)
18. Park, J.H., Kwon, O.M., Lee, S.M.: State Estimation for Neural Networks of Neutral-Type with Interval Time-Varying Delay. Appl. Math. Comput. (2008), doi:10.1016/j.amc.2008.04.025
19. Park, J.H., Park, C.H., Kwon, O.M., Lee, S.M.: A New Stability Criterion for Bidirectional Associative Memory Neural Networks of Neutral-Type. Appl. Math. Comput. 199, 716–722 (2008)
20. Qiu, J., Cao, J.: Delay-Dependent Robust Stability of Neutral-Type Neural Networks with Time Delays. J. Math. Cont. Sci. Appl. 1, 179–188 (2007)

Global Passivity of Stochastic Neural Networks with Time-Varying Delays

Jinming Liang[1] and Qiankun Song[2]

[1] School of Computer Science, Sichuan University of Science and Engineering, Sichuan 643000, China
[2] Department of Mathematics, Chongqing Jiaotong University, Chongqing 400074, China
qiankunsong@163.com

Abstract. In this paper, the passivity problem is investigated for a class of stochastic neural networks with time-varying delays as well as generalized activation functions. By employing a combination of Lyapunov functional, the free-weighting matrix method and stochastic analysis technique, a delay-independent criterion for the passivity of the addressed neural networks is established in terms of linear matrix inequalities (LMIs), which can be checked numerically using the effective LMI toolbox in MATLAB. An example is given to show the effectiveness and less conservatism of the proposed criterion. It is noteworthy that the traditional assumptions on the differentiability of the time-varying delays and the boundedness of its derivative are removed.

Keywords: Passivity, Stochastic neural networks, Time-varying delays.

1 Introduction

It is well known that many artificial neural networks have been extensively investigated and successfully applied to various areas such as signal processing, pattern recognition, associative memory and optimization problems. In such applications, it is of prime importance to ensure that the designed neural networks are stable [1]. In hardware implementation, time delays are likely to be present due to the finite switching speed of amplifiers and communication time. It has also been shown that the processing of moving images requires the introduction of delay in the signal transmitted through the networks [2]. The time delays are usually variable with time, which will affect the stability of designed neural networks and may lead to some complex dynamic behaviors such as oscillation, bifurcation, or chaos [3]. Therefore, the study of neural dynamics with consideration of time delays becomes extremely important to manufacture high quality neural networks [4]. Many important results on the dynamical behaviors have been reported for delayed neural networks, see [1]-[16] and the references therein for some recent publications.

Just as pointed out in [17], in real nervous systems, synaptic transmission is a noisy process brought on by random fluctuations from the release of neurotransmitters and other probabilistic causes. In the implementation of artificial

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 405–412, 2009.

neural networks, noise is unavoidable and should be taken into consideration in modelling. Therefore, it is of significant importance to consider stochastic effects to the dynamical behaviors of neural networks [18]. Some recent interest results on stochastic neural networks can be found, see [18]-[20] and references therein.

On the other hand, the passivity theory is another effective tool to the stability of nonlinear system [21]. The main idea of passivity theory is that the passive properties of system can keep the system internal stability [22]. Thus, the passivity theory has received a lot of attention from the control community since 1970s [23]. Recently, the passivity theory for delayed neural networks was investigated, some criteria checking the passivity were provided for certain or uncertain neural networks with time-varying delays, see [24]-[26] and references therein. It is worth pointing out that, the given criteria in [24]-[26] have been based on the following assumptions: 1) the time-varying delays are continuously differentiable; 2) the derivative of time-varying delay is bounded and is smaller than one; and 3) the activation functions are bounded and monotonically nondecreasing. However, time delays can occur in an irregular fashion, and sometimes the time-varying delays are not differentiable. In such a case, the methods developed in [24]-[26] may be difficult to be applied, and it is therefore necessary to further investigate the passivity problem of neural networks with time-varying delays under *milder* assumptions. To the best of our knowledge, few authors have considered the passivity problem for stochastic neural networks with time-varying delays as well as generalized activation functions.

Motivated by the above discussions, the objective of this paper is to study the passivity of stochastic neural networks with time-varying delays as well as generalized activation functions by employing a combination of Lyapunov functional, the free-weighting matrix method and stochastic analysis technique. The obtained sufficient conditions require *neither* the differentiability of time-varying delays *nor* the monotony of the activation functions, and are expressed in terms of linear matrix inequalities (LMIs), which can be checked numerically using the effective LMI toolbox in MATLAB. An example is given to show the effectiveness and less conservatism of the proposed criterion.

2 Problem Formulation and Preliminaries

In this paper, we consider the following neural network model

$$\begin{aligned} dx(t) = [-Dx(t) + Af(x(t)) + Bf(x(t-\tau(t))) + u(t)]dt \\ +\sigma(t, x(t), x(t-\tau(t)))d\omega(t) \end{aligned} \tag{1}$$

for $t \geq 0$, where $x(t) = (x_1(t), x_2(t), \cdots, x_n(t))^T \in R^n$ is the state vector of the network at time t, n corresponds to the number of neurons; $D = \mathrm{diag}(d_1, d_2, \cdots, d_n)$ is a positive diagonal matrix, $A = (a_{ij})_{n\times n}$ and $B = (b_{ij})_{n\times n}$ are the interconnection weight matrices; $\sigma \in R^{n\times q}$ is the diffusion coefficient vector and $\omega(t) = (\omega_1(t), \omega_2(t), \cdots, \omega_q(t))^T$ is an q-dimensional Brownian motion defined on a complete probability space $(\Omega, F, \{F_t\}_{t\geq 0}, P)$ with a filtration $\{F_t\}_{t\geq 0}$ satisfying the usual conditions (i.e., it is right continuous and F_0 contains all P-null

sets); $f(x(t)) = (f_1(x_1(t)), f_2(x_2(t)), \cdots, f_n(x_n(t)))^T$ denotes the neuron activation at time t; $u(t) = (u_1(t), u_2(t), \cdots, u_n(t))^T \in R^n$ is a varying external input vector; $\tau(t) > 0$ is the time-varying delay, and is assumed to satisfy $0 \leq \tau(t) \leq \tau$, where τ is constant.

The initial condition associated with model (1) is given by

$$x(s) = \phi(s), \qquad s \in [-\tau, 0].$$

Let $x(t, \phi)$ denote the state trajectory of model (1) from the above initial condition and $x(t, 0)$ is the corresponding trajectory with zero initial condition.

Throughout this paper, we make the following assumptions:

(H1).([14]) For any $j \in \{1, 2, \cdots, n\}$, $f_j(0) = 0$ and there exist constants F_j^- and F_j^+ such that

$$F_j^- \leq \frac{f_j(\alpha_1) - f_j(\alpha_2)}{\alpha_1 - \alpha_2} \leq F_j^+$$

for all $\alpha_1 \neq \alpha_2$.

(H2).([19]) There exist constant matrices R_1 and R_2 of appropriate dimensions such that the following inequality trace

$$trace\Big(\sigma^T(t, x(t), x(t-\tau(t)))\sigma(t, x(t), x(t-\tau(t)))\Big) \leq \|R_1 u\|^2 + \|R_2 v\|^2$$

holds for all $(t, u, v) \in \mathbf{R} \times \mathbf{R}^n \times \mathbf{R}^n$.

Definition 1. *([23]) System (1) is called globally passive in the sense of expectation if there exists a scalar $\gamma > 0$ such that*

$$2\mathbf{E}\Big\{\int_0^{t_p} f^T(x(s))u(s)ds\Big\} \geq -\mathbf{E}\Big\{\gamma \int_0^{t_p} u^T(s)u(s)ds\Big\}$$

for all $t_p \geq 0$ and for all $x(t, 0)$.

3 Main Results

For presentation convenience, in the following, we denote

$$F_1 = \mathrm{diag}(F_1^- F_1^+, \cdots, F_n^- F_n^+), \quad F_2 = \mathrm{diag}(\frac{F_1^- + F_1^+}{2}, \cdots, \frac{F_n^- + F_n^+}{2}).$$

Theorem 1. *Under assumptions (**H1**) and (**H2**), model (1) is passive in the sense of expectation if there exist two scalar $\gamma > 0$, $\lambda > 0$, a symmetric positive definite matrices P, two positive diagonal matrices L and S, and matrices M, N and W such that the following two LMIs hold:*

$$P < \lambda I, \tag{2}$$

$$\Omega = \begin{bmatrix} \Omega_1 & PA+F_2L & PB & P & W-M^T & N-M^T \\ * & -L & 0 & -I & 0 & 0 \\ * & * & -S & 0 & 0 & F_2S \\ * & * & * & -\gamma I & 0 & 0 \\ * & * & * & * & -W-W^T & -N-W^T \\ * & * & * & * & * & \Omega_2 \end{bmatrix} < 0, \tag{3}$$

where $\Omega_1 = -PD-DP+\lambda R_1^T R_1-F_1L+M+M^T$, $\Omega_2 = \lambda R_2^T R_2-F_1S-N-N^T$.

Proof. Consider the following Lyapunov functional as

$$V(t,x(t)) = x^T(t)Px(t), \tag{4}$$

By Itô differential rule, the mathematical expectation of the stochastic derivative of $V(t)$ along the trajectory of system (1) can be obtained as

$$\begin{aligned} \mathbf{E}\{dV(t,x(t))\} = \mathbf{E}\Big\{\Big[2x^T(t)P\Big(-Dx(t)+Af(x(t))+Bf(x(t-\tau(t)))+u(t)\Big) \\ +trace\Big(\sigma^T(x(t),x(t-\tau(t)))P\sigma(x(t),x(t-\tau(t)))\Big)\Big]dt\Big\}. \end{aligned} \tag{5}$$

From assumption (**H2**) and inequality (2), we have

$$\begin{aligned} &trace\Big(\sigma^T(x(t),x(t-\tau(t)))P\sigma(x(t),x(t-\tau(t)))\Big) \\ &\le \lambda\Big[x^T(t)R_1^TR_1x(t)+x^T(t-\tau(t))R_2^TR_2x(t-\tau(t))\Big]. \end{aligned} \tag{6}$$

It follows from inequalities (5)and (6) that

$$\begin{aligned} \mathbf{E}\{dV(t,x(t))\} \le \mathbf{E}\Big\{\Big[x^T(t)\Big(-PD-DP+\lambda R_1^TR_1\Big)x(t) \\ +2x^T(t)PAf(x(t))+2x^T(t)PBf(x(t-\tau(t))) \\ +2x^T(t)Pu(t)+\lambda x^T(t-\tau(t))R_2^TR_2x(t-\tau(t))\Big]dt\Big\} \end{aligned} \tag{7}$$

From assumption (**H1**), we have

$$\Big(f_i(x_i(t))-F_i^-x_i(t)\Big)\Big(f_i(x_i(t))-F_i^+x_i(t)\Big) \le 0, \quad i=1,2,\cdots,n,$$

which are equivalent to

$$\begin{bmatrix} x_i(t) \\ f_i(x_i(t)) \end{bmatrix}^T \begin{bmatrix} F_i^-F_i^+e_ie_i^T & -\frac{F_i^-+F_i^+}{2}e_ie_i^T \\ -\frac{F_i^-+F_i^+}{2}e_ie_i^T & e_ie_i^T \end{bmatrix} \begin{bmatrix} x_i(t) \\ f_i(x_i(t)) \end{bmatrix} \le 0, \quad i=1,2,\cdots,n,$$

where e_r denotes the unit column vector having 1 element on its rth row and zeros elsewhere. Let

$$L = \mathrm{diag}\{l_1,l_2,\cdots,l_n\}, \quad S = \mathrm{diag}\{s_1,s_2,\cdots,s_n\},$$

then

$$\sum_{i=1}^{n} l_i \begin{bmatrix} x_i(t) \\ f_i(x_i(t)) \end{bmatrix}^T \begin{bmatrix} F_i^- F_i^+ e_i e_i^T & -\frac{F_i^- + F_i^+}{2} e_i e_i^T \\ -\frac{F_i^- + F_i^+}{2} e_i e_i^T & e_i e_i^T \end{bmatrix} \begin{bmatrix} x_i(t) \\ f_i(x_i(t)) \end{bmatrix} \le 0,$$

that is

$$\begin{bmatrix} x(t) \\ f(x(t)) \end{bmatrix}^T \begin{bmatrix} F_1 L & -F_2 L \\ -F_2 L & L \end{bmatrix} \begin{bmatrix} x(t) \\ f(x(t)) \end{bmatrix} \le 0. \tag{8}$$

Similarly, one has

$$\begin{bmatrix} x(t-\tau(t)) \\ f(x(t-\tau(t))) \end{bmatrix}^T \begin{bmatrix} F_1 S & -F_2 S \\ -F_2 S & S \end{bmatrix} \begin{bmatrix} x(t-\tau(t)) \\ f(x(t-\tau(t))) \end{bmatrix} \le 0. \tag{9}$$

From Newton-Leibniz formulation $x(t) - x(t-\tau(t)) - \int_{t-\tau(t)}^{t} dx(s) = 0$, we have

$$\begin{aligned} 0 = {} & 2\Big(x(t) - x(t-\tau(t)) - \int_{t-\tau(t)}^{t} dx(s)\Big)^T \\ & \times \Big(Mx(t) + Nx(t-\tau(t)) + W \int_{t-\tau(t)}^{t} dx(s)\Big) = 0. \end{aligned} \tag{10}$$

It follows from (7)-(10) that

$$\begin{aligned} & \mathbf{E}\{dV(t,x(t)) - 2f^T(x(t))u(t)dt - \gamma u^T(t)u(t)dt\} \\ \le {} & \mathbf{E}\Big\{\Big[x^T(t)\Big(-PD - DP + \lambda R_1^T R_1\Big)x(t) + 2x^T(t)PAf(x(t)) \\ & + 2x^T(t)PBf(x(t-\tau(t))) + 2x^T(t)Pu(t) \\ & + \lambda x^T(t-\tau(t))R_2^T R_2 x(t-\tau(t)) \\ & - \begin{bmatrix} x(t) \\ f(x(t)) \end{bmatrix}^T \begin{bmatrix} F_1 L & -F_2 L \\ -F_2 L & L \end{bmatrix} \begin{bmatrix} x(t) \\ f(x(t)) \end{bmatrix} \\ & - \begin{bmatrix} x(t-\tau(t)) \\ f(x(t-\tau(t))) \end{bmatrix}^T \begin{bmatrix} F_1 S & -F_2 S \\ -F_2 S & S \end{bmatrix} \begin{bmatrix} x(t-\tau(t)) \\ f(x(t-\tau(t))) \end{bmatrix} \\ & + 2\Big(x(t) - x(t-\tau(t)) - \int_{t-\tau(t)}^{t} dx(s)\Big)^T \\ & \times \Big(Mx(t) + Nx(t-\tau(t)) + W \int_{t-\tau(t)}^{t} dx(s)\Big) \\ & - 2f^T(x(s))u(s) - \gamma u^T(t)u(t)\Big]dt\Big\} \\ = {} & \mathbf{E}\Big\{\alpha^T(t)\Omega\alpha(t)dt\Big\}, \end{aligned} \tag{11}$$

where $\alpha(t) = \Big(x^T(t), f^T(x(t)), f^T(x(t-\tau(t))), u^T(t), \int_{t-\tau(t)}^{t} dx(s), x^T(t-\tau(t))\Big)^T$. One can derive from (3) and (11) that

$$\frac{\mathbf{E}\{dV(t,x(t))\}}{dt} - \mathbf{E}\{2f^T(x(t))u(t) + \gamma u^T(t)u(t)\} \le 0. \tag{12}$$

From (12) and the definition of $V(t, x(t))$, we can get

$$2\mathbf{E}\{\int_0^{t_p} f^T(x(s))u(s)ds\} \geq -\gamma\mathbf{E}\{\int_0^{t_p} u^T(s)u(s)ds\}$$

From Definition 1, we know that the stochastic neural networks (1) is globally passive in the sense of expectation, and the proof of Theorem 1 is then completed.

Remark 1. Assumption (**H1**) was first proposed in [14] and [15]. The constants F_j^- and F_j^+ $(i = 1, 2, \cdots, n)$ in assumption (**H1**) are allowed to be positive, negative or zero. Hence, Assumption (**H1**) is weaker than the assumption in [24]-[26]. In addition, the conditions in [24]-[26] that the time-varying delay is differentiable and the derivative is smaller than one have been removed in this paper.

4 An Example

Consider a two-neuron neural network (1), where

$$D = \begin{bmatrix} 1.8 & 0 \\ 0 & 2.5 \end{bmatrix}, \quad A = \begin{bmatrix} 0.3 & 0.1 \\ -0.2 & -0.2 \end{bmatrix}, \quad B = \begin{bmatrix} 0.1 & -0.5 \\ 0.3 & 0.7 \end{bmatrix},$$

$$f_1(z) = \tanh(-0.2z), \quad f_2(z) = \tanh(0.4z), \quad \tau(t) = 5|\sin t|,$$

and σ satisfies

$$\begin{aligned} &trace\Big(\sigma^T(t, x(t), x(t-\tau(t)))\sigma(t, x(t), x(t-\tau(t)))\Big) \\ &\leq 0.04x_1^2(t) + 0.01x_2^2(t) + 0.01x_1^2(t-\tau(t)) + 0.01x_2^2(t-\tau(t)). \end{aligned}$$

It can be verified that assumptions (**H1**) and (**H2**) are satisfied, and $F_1 = 0$, $F_2 = \text{diag}\{-0.1, 0.2\}$, $R_1 = \text{diag}\{0.2, 0.1\}$, $R_2 = \text{diag}\{0.1, 0.1\}$.

By the Matlab LMI Control Toolbox, we find a solution to the LMIs in (2) and (3) as follows:

$$P = 10^{-7}\begin{bmatrix} 0.6997 & 0.1565 \\ 0.1565 & 0.7805 \end{bmatrix}, \quad L = 10^{-4}\begin{bmatrix} 0.1048 & 0 \\ 0 & 0.0228 \end{bmatrix},$$

$$S = 10^{-6}\begin{bmatrix} 0.1047 & 0 \\ 0 & 0.0363 \end{bmatrix}, \quad M = 10^{6}\begin{bmatrix} -2.2397 & -0.0685 \\ 0.0498 & -6.9419 \end{bmatrix},$$

$$N = 10^{6}\begin{bmatrix} 2.2397 & 0.0685 \\ -0.0498 & 6.9419 \end{bmatrix}, \quad W = 10^{6}\begin{bmatrix} 2.2397 & 0.0685 \\ -0.0498 & 6.9419 \end{bmatrix},$$

$$\gamma = 2.2536 \times 10^7, \lambda = 1.7865 \times 10^{-7}.$$

Therefore, by Theorem 1, we know that the considered model is passive in the sense of Definition 1. It should be pointed out that the conditions in [24]-[26] cannot be applied to this example since it requires the differentiability of the time-varying delay.

5 Conclusions

In this paper, the passivity has been investigated for a class of stochastic neural networks with time-varying delays as well as generalized activation functions. By employing a combination of Lyapunov functional, the free-weighting matrix method and inequality technique, a new delay-independent criterion for the passivity of the addressed neural networks has been established in terms of linear matrix inequalities (LMIs), which can be checked numerically using the effective LMI toolbox in MATLAB. The obtained results generalize and improve the earlier publications, and remove the traditional assumptions on the differentiability of the discrete time-varying delay and the boundedness of its derivative. An example has been provided to demonstrate the effectiveness and less conservatism of the proposed criterion.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 10772152.

References

1. Liao, X.X., Wang, J.: Algebraic Criteria for Global Exponential Stability of Cellular Neural Networks with Multiple Time Delays. IEEE Transactions on Circuits and Systems I 50, 268–275 (2003)
2. Arik, S.: An Analysis of Exponential Stability of Delayed Neural Networks with Time Varying Delays. Neural Networks 17, 1027–1031 (2004)
3. Cao, J.D., Feng, G., Wang, Y.Y.: Multistability and Multiperiodicity of Delayed Cohen-Grossberg Neural Networks with A General Class of Activation Functions. Physica D 237, 1734–1749 (2008)
4. Cao, J.D., Song, Q.K.: Stability in Cohen-Grossberg Type BAM Neural Networks with Time-varying Delays. Nonlinearity 19, 1601–1617 (2006)
5. Chen, T.P.: Global Exponential Stability of Delayed Hopfield Neural Networks. Neural Networks 14, 977–980 (2001)
6. Li, C.D., Liao, X.F., Zhang, R.: A Global Exponential Robust Stability Criterion for Interval Delayed Neural Networks with Variable Delays. Neurocomputing 69, 803–809 (2006)
7. Liao, X.F., Liu, Q., Zhang, W.: Delay-dependent Asymptotic Stability for Neural Networks with Distributed Delays. Nonlinear Analysis: Real World Applications 7, 1178–1192 (2006)
8. Liao, X.X., Luo, Q., Zeng, Z.G.: Positive Invariant and Global Exponential Attractive Sets of Neural Networks with Time-varying Delays. Neurocomputing 71, 513–518 (2008)
9. Lu, H.T., Chung, F.L., He, Z.Y.: Some Sufficient Conditions for Global Exponential Stability of Hopfield Neural Networks. Neural Networks 17, 537–544 (2004)
10. Lu, W.L., Chen, T.P.: R^n_+-Global Stability of A Cohen-Grossberg Neural Network System with Nonnegative Equilibria. Neural Networks 20, 714–722 (2007)

11. Song, Q.K., Wang, Z.D.: A delay-dependent LMI approach to dynamics analysis of discrete-time recurrent neural networks with time-varying delays. Physics Letters A 368, 134–145 (2007)
12. Zeng, Z.G., Wang, J.: Improved Conditions for Global Exponential Stability of Recurrent Neural Networks with Time-varying Delays. IEEE Transactions on Neural Networks 17, 623–635 (2006)
13. Zhao, H.Y., Wang, L., Ma, C.X.: Hopf Bifurcation and Stability Analysis on Discrete-time Hopfield Neural Network with Delay. Nonlinear Analysis: Real World Applications 9, 103–113 (2008)
14. Liu, Y.R., Wang, Z.D., Liu, X.H.: Global Exponential Stability of Generalized Recurrent Neural Networks with Discrete and Distributed Delays. Neural Networks 19, 667–675 (2006)
15. Wang, Z., Shu, H., Liu, Y., Ho, D.W.C., Liu, H.: Robust stabilityanalysis of generalized neural networks with discrete and distributed time delays. Chaos, Solitons and Fractals 30, 886–896 (2006)
16. Zhang, H.G., Wang, Z.S., Liu, D.R.: Robust Exponential Stability of Recurrent Neural Networks With Multiple Time-Varying Delays. IEEE Transactions on Circuits and Systems II 54, 730–734 (2007)
17. Xu, D.Y., Yang, Z.G., Huang, Y.M.: Existence-uniqueness andcontinuation theorems for stochastic functional differential equations. Journal of Differential Equations 245, 1681–1703 (2008)
18. Shen, Y., Wang, J.: Noise-Induced Stabilization of the Recurrent Neural Networks With Mixed Time-Varying Delays and Markovian-Switching Parameters. IEEE Transactions on NeuralNetworks 18, 1857–1862 (2007)
19. Wang, Z.D., Liu, Y.R., Fraser, K., Liu, X.H.: Stochastic Stability of Uncertain Hopfield Neural Networks with Discrete and Distributed Delays. Physics Letters A 354, 288–297 (2006)
20. Liu, X.W., Chen, T.P.: Robust -stability for Uncertain Stochastic Neural Networks with Unbounded Time-varying Delays. Physica A 387, 2952–2962 (2008)
21. Yu, W.: Passivity Analysis for Dynamic Multilayer Neuro Identifier. IEEE Transactions on Circuits and Systems I 50, 173–178 (2003)
22. Li, C.G., Zhang, H.B., Liao, X.F.: Passivity and Passification of Fuzzy Systems with Time Delays. Computers and Mathematics with Applications 52, 1067–1078 (2006)
23. Gao, H., Chen, T., Chai, T.: Passivity and Passification for Networked Control Systems. SIAM Journal on Control and Optimization 46, 1299–1322 (2007)
24. Li, C.G., Liao, X.F.: Passivity Analysis of Neural Networks with Time Delay. IEEE Transactions on Circuits and Systems II 52, 471–475 (2005)
25. Park, J.H.: Further Results on Passivity Analysis of Delayed Cellular Neural Networks. Chaos, Solitons and Fractals 34, 1546–1551 (2007)
26. Lou, X.Y., Cui, B.T.: Passivity Analysis of Integro-differential Neural Networks with Time-varying Delays. Neurocomputing 70, 1071–1078 (2007)

Exponential Stability of High-Order Fuzzy Cellular Neural Networks with Time-Varying Delays

Haijun Jiang, Bianjing Guo, and Zhidong Teng

College of Mathematics and System Sciences, Xinjiang University, Urumqi 830046, China
jianghai@xju.edu.cn

Abstract. In this paper, the existence and global exponential stability of equilibrium point of high-order fuzzy cellular neural networks (HFCNNs) with time-varying delays is studied. Employing nonsingular M-matrix and Lyapunov functional method, some new sufficient conditions are derived for checking the existence and global exponential stability of equilibrium point of the HFCNNs with time-varying delays.

Keywords: Cellular neural networks, Equilibrium point, Exponential stability, Nonsingular M-matrix, Lyapunov functional.

1 Introduction

Fuzzy cellular neural networks (FCNNs) is introduced by Yang, Yang, Wu and Chua in [1], combines fuzzy logic with the traditional CNNs. Studies have shown the potential of FCNNs in image processing and pattern recognition. In these applications, it is required that the neural networks be exponentially stable. In [2], the authors have obtained some conditions for the existence of the equilibrium point and the exponential stability of FCNNs without delay. The FCNNs with the constant and time-varying delays, with distributed delay, with diffusion have been also studied in [3-6].

It is well known that the high-order neural networks (HNNs) dose better than the ordinary neural networks in the applications, that is to say, HNNs have stronger approximation property, faster convergence rate, great stronger capacity and higher fault tolerance(see [7-9]). In [10, 11] the stability of HNNs with impulsive effects was studied by using the linear matrix inequality (LMI) method, Halanay inequality and fixed point theorem. In [12], the exponential stability of high-order bidirectional associative memory neural networks with time delays was studied by employing LMI. In [13, 14] the existence and stability of periodic solution for delayed HNNs was studied by using coincidence degree theory.

However, to the best of our knowledge, there does not seem to be much (if any) study on the dynamics of high-order fuzzy cellular neural networks (HFCNNs). Therefore, In this paper, the global exponential stability of HFCNNs with time-varying delays is proposed. Employing the nonsingular M-matrix and method of

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 413–422, 2009.

Lyapunov functional, some sufficient conditions are derived for checking global exponential stability of the HFCNNs with time-varying delays.

In this paper, we will study the following delayed HFCNNs:

$$\begin{aligned}\frac{\mathrm{d}x_i(t)}{\mathrm{d}t} &= -d_i x_i(t) + \sum_{j=1}^{n} a_{ij}\widetilde{f}_j(x_j(t-\tau_{ij}(t))) \\ &+ \sum_{j=1}^{n}\sum_{l=1}^{n} b_{ijl}\widetilde{f}_j(x_j(t-\tau_{ij}(t)))\widetilde{f}_l(x_l(t-\tau_{ij}(t))) \\ &+ \bigwedge_{j=1}^{n} \alpha_{ij}\widetilde{f}_j(x_j(t-\tau_{ij}(t))) + \bigvee_{j=1}^{n} \beta_{ij}\widetilde{f}_j(x_j(t-\tau_{ij}(t))) \\ &+ \bigwedge_{j=1}^{n}(\bigwedge_{l=1}^{n} \gamma_{ijl}\widetilde{f}_j(x_j(t-\tau_{ij}(t)))\widetilde{f}_l(x_l(t-\tau_{ij}(t)))) \\ &+ \bigvee_{j=1}^{n}(\bigvee_{l=1}^{n} \sigma_{ijl}\widetilde{f}_j(x_j(t-\tau_{ij}(t)))\widetilde{f}_l(x_l(t-\tau_{ij}(t)))) \\ &+ \bigwedge_{j=1}^{n} T_{ij}u_j + \bigvee_{j=1}^{n} H_{ij}u_j + I_i, \quad i = 1, 2, \cdots, n. \end{aligned} \tag{1}$$

where, x_i denotes the potential (or voltage) of the cell i at time t; d_i is positive constant, denotes the rate with which the cell i reset its potential to the resting state when isolated from the other cells and inputs; Time delays $\tau_{ij}(t)\,(i, j = 1, 2, \cdots, n)$ are non-negative, continuously differentiable functions, it correspond to finite speed of axonal signal transmission; a_{ij} and b_{ij} are the first- and second-order connection weights of neural network, respectively; $\alpha_{ij}, \beta_{ij}, T_{ij}$ and H_{ij} are elements of the first-order fuzzy feedback MIN template, first-order fuzzy feedback MAX template, first-order fuzzy feed-forward MIN template and first-order fuzzy feed-forward MAX template, respectively; γ_{ijl} and σ_{ijl} are elements of the second-order fuzzy feedback MIN template, second-order fuzzy feedback MAX template, respectively; $\bigwedge$ and $\bigvee$ denote the fuzzy AND and fuzzy OR operations, respectively; u_i and I_i denote input and bias of the ith neurons, respectively; $\widetilde{f}_i$ is the activation function.

2 Preliminaries

Throughout this paper, the following notations will be used. Let $A = (a_{ij})$ be an $n \times n$ dimensional real matrix. A^{-1} denotes the inverse of matrix A. For $x \in \mathbf{R}^n$, its norm is defined by $\|x\| = \sqrt{x^T x}$.

For system (1), we introduce the following assumptions:

(H_1) For each $i \in 1, 2, \cdots, n$, there exist a constant $M_i > 0$, such that

$$|\widetilde{f}_i(u)| \leq M_i, \text{ for all } u \in \mathbf{R}.$$

(H_2) For each $i \in 1, 2, \cdots, n$, there exist a constant $L_i > 0$, such that

$$|\widetilde{f}_i(u) - \widetilde{f}_i(v)| \leq L_i|u - v|, \text{ for all } u \in \mathbf{R}.$$

(H_3) For $i, j = 1, 2, \cdots, n$, $\tau_{ij}(t)$ are nonnegative, bounded, differentiable and there exist a constant $0 \leq \delta < 1$, such that $\dot{\tau}_{ij}(t) \leq \delta$, where $\dot{\tau}_{ij}(t) = \frac{\mathrm{d}\tau_{ij}(t)}{\mathrm{d}t}$.

Let $\tau = \sup\{\tau_{ij}(t) : t \in [0, +\infty\}, i, j = 1, 2, \cdots, n\}$. We introduce $C([-\tau, 0], \mathbf{R}^n)$ as the initial function space of system (1), which is the Banach space of all continuous functions $\phi = (\phi_1, \phi_2, \cdots, \phi_n)^T : [-\tau, 0] \to \mathbf{R}^n$ with normal $\|\phi\| = \sup_{-\tau \leq \theta \leq 0} |\phi(\theta)|$, where $|\phi(\theta)| = (\sum_{i=1}^n |\phi_i(\theta)|^2)^{\frac{1}{2}}$.

Definition 1. *Equilibrium point x^* of system (1) is said to be globally exponentially stable, if there exist constants $k > 0$ and $\gamma > 0$ such that, for all $t \geq 0$,*

$$\|x(t, \varphi) - x^*\| \leq \gamma e^{-kt}.$$

Definition 2. *(see [15]). Let matrix $A = (a_{ij})_{n \times n}$ have nonpositive off-diagonal elements, then A is said to be a nonsingular M-matrix if A have all positive diagonal elements and there exists a positive diagonal matrix $\Lambda = diag(\lambda_1, \lambda_2, \cdots, \lambda_n,)$ such that $A\Lambda$ or $A^T\Lambda$ is strictly diagonally dominant; That is*

$$a_{ii}\lambda_i > \sum_{j \neq i}^{n} |a_{ij}|\lambda_j, \quad \text{or} \quad a_{ii}\lambda_i > \sum_{j \neq i}^{n} |a_{ji}|\lambda_j \quad i = 1, 2, \cdots, n,$$

which can be rewritten as

$$\sum_{j=1}^{n} a_{ij}\lambda_j > 0, \quad \text{or} \quad \sum_{j=1}^{n} a_{ji}\lambda_j > 0 \quad i = 1, 2, \cdots, n.$$

To obtain our main results, we need the following lemma.

Lemma 1. *(see [12]). Suppose $x = (x_1, x_2, \cdots, x_n)$ and $y = (y_1, y_2, \cdots, y_n)$ are two states of system (1), then we have*

$$|\bigwedge_{j=1}^{n} \alpha_{ij}\widetilde{f}_i(x_j) - \bigwedge_{j=1}^{n} \alpha_{ij}\widetilde{f}_i(y_j)| \leq \sum_{j=1}^{n} |\alpha_{ij}||\widetilde{f}_i(x_j) - \widetilde{f}_i(y_j)|,$$
$$|\bigvee_{j=1}^{n} \alpha_{ij}\widetilde{f}_i(x_j) - \bigvee_{j=1}^{n} \alpha_{ij}\widetilde{f}_i(y_j)| \leq \sum_{j=1}^{n} |\alpha_{ij}||\widetilde{f}_i(x_j) - \widetilde{f}_i(y_j)|.$$

3 Existence Uniqueness of Equilibrium Point

In the following discussions, we denote $L = \mathrm{diag}(L_1, L_2, \cdots, L_n)$, $D = \mathrm{diag}(d_1, d_2, \cdots, d_n)$, $A = (|a_{ij}| + |\alpha_{ij}| + |\beta_{ij}|)_{n \times n}$ and $B = (B_{ij})_{n \times n}$ with $B_{ij} = \sum_{l=1}^{n} M_l (|b_{ijl}| + |\gamma_{ijl}| + |\sigma_{ijl}| + |b_{ilj}| + |\gamma_{ilj}| + |\sigma_{ilj}|)$.

Theorem 1. *Under assumptions* (H_1) *and* (H_2)*, then system (1) has a unique equilibrium* x^* *if* $D-(A+B)L$ *is a nonsingular M-matrix.*

Proof. Let $g(x)=(g_1(x),g_2(x),\cdots,g_n(x))$ with

$$\begin{aligned} g_i(x) &= d_i x_i - \sum_{j=1}^{n} a_{ij}\widetilde{f}_j(x_j) - \sum_{j=1}^{n}\sum_{l=1}^{n} b_{ijl}\widetilde{f}_j(x_j)\widetilde{f}_l(x_l) \\ &\quad - \bigwedge_{j=1}^{n} \alpha_{ij}\widetilde{f}_j(x_j) - \bigvee_{j=1}^{n} \beta_{ij}\widetilde{f}_j(x_j) \\ &\quad - \bigwedge_{j=1}^{n} (\bigwedge_{l=1}^{n} \gamma_{ijl}\widetilde{f}_j(x_j)\widetilde{f}_l(x_l)) - \bigvee_{j=1}^{n} (\bigvee_{l=1}^{n} \sigma_{ijl}\widetilde{f}_j(x_j)\widetilde{f}_l(x_l)) \\ &\quad - \bigwedge_{j=1}^{n} T_{ij}u_j - \bigvee_{j=1}^{n} H_{ij}u_j - I_i, \quad i=1,2,\cdots,n. \end{aligned} \tag{2}$$

Obviously, the solution of equation $g(x)=0$ is the equilibrium point of system (1). We can define homotopic mapping as follow

$$G(x,\lambda)=\lambda g(x)+(1-\lambda)x,$$

where $\lambda\in[0,\,1]$, $G(x,\,\lambda)=(G_1(x,\lambda),G_2(x,\lambda),\cdots,G_n(x,\lambda))^T$, then it follows from (H_1) and (H_2) that for $1\le i\le n$

$$\begin{aligned} |G_i(x,\,\lambda)| &\ge |\lambda d_i x_i + (1-\lambda)x_i| - \lambda\sum_{j=1}^{n}|a_{ij}||\widetilde{f}_j(x_j)| \\ &\quad -\lambda|\sum_{j=1}^{n}\sum_{l=1}^{n} b_{ijl}\widetilde{f}_j(x_j)\widetilde{f}_l(x_l)| \\ &\quad -\lambda|\bigwedge_{j=1}^{n}\alpha_{ij}\widetilde{f}_j(x_j)| - \lambda|\bigvee_{j=1}^{n}\beta_{ij}\widetilde{f}_j(x_j)| \\ &\quad -\lambda|\bigwedge_{j=1}^{n}(\bigwedge_{l=1}^{n}\gamma_{ijl}\widetilde{f}_j(x_j)\widetilde{f}_l(x_l))| - \lambda|\bigvee_{j=1}^{n}(\bigvee_{l=1}^{n}\sigma_{ijl}\widetilde{f}_j(x_j)\widetilde{f}_l(x_l))| \\ &\quad -\lambda|\bigwedge_{j=1}^{n}T_{ij}u_j| - \lambda|\bigvee_{j=1}^{n}H_{ij}u_j| - \lambda|I_i| \\ &\ge (1-\lambda)|x_i| + \lambda[d_i|x_i| - \sum_{j=1}^{n}A_{ij}L_j|x_j| - \sum_{j=1}^{n}B_{ij}L_j|x_j|] \\ &\quad -\lambda[\sum_{j=1}^{n}|a_{ij}||\widetilde{f}_j(0)| + |\bigwedge_{j=1}^{n}\alpha_{ij}\widetilde{f}_j(0)| + |\bigvee_{j=1}^{n}\beta_{ij}\widetilde{f}_j(0)| \end{aligned}$$

$$+|\sum_{j=1}^{n}\sum_{l=1}^{n} b_{ijl}\widetilde{f}_j(0)\widetilde{f}_l(0)| + |\bigwedge_{j=1}^{n}(\bigwedge_{l=1}^{n}\gamma_{ijl}\widetilde{f}_j(0)\widetilde{f}_l(0))|$$
$$+|\bigvee_{j=1}^{n}(\bigvee_{l=1}^{n}\sigma_{ijl}\widetilde{f}_j(0)\widetilde{f}_l(0))| + |\bigwedge_{j=1}^{n}T_{ij}u_j| + |\bigvee_{j=1}^{n}H_{ij}u_j| + |I_i|]. \tag{3}$$

Since $D - AL - BL$ is a nonsingular M-matrix, hence, there exist constants $r_i > 0$ such that

$$r_i d_i - \sum_{j=1}^{n} r_j A_{ji} L_j - \sum_{j=1}^{n} r_j B_{ji} L_j > 0, \quad i = 1, 2, \cdots, n.$$

Further, we have

$$\begin{aligned}
&\sum_{i=1}^{n} r_i |G_i(x,\lambda)| \\
&\geq \lambda \sum_{i=1}^{n}[r_i d_i |x_i| - r_i \sum_{j=1}^{n} A_{ij} L_j |x_j| - r_i \sum_{j=1}^{n} B_{ij} L_j |x_j|] \\
&\quad -\lambda \sum_{i=1}^{n} r_i [\sum_{j=1}^{n} |a_{ij}||\widetilde{f}_j(0)| + |\bigwedge_{j=1}^{n}\alpha_{ij}\widetilde{f}_j(0)| + |\bigvee_{j=1}^{n}\beta_{ij}\widetilde{f}_j(0)| \\
&\quad +|\sum_{j=1}^{n}\sum_{l=1}^{n} b_{ijl}\widetilde{f}_j(0)\widetilde{f}_l(0)| + |\bigwedge_{j=1}^{n}(\bigwedge_{l=1}^{n}\gamma_{ijl}\widetilde{f}_j(0)\widetilde{f}_l(0))| \\
&\quad +|\bigvee_{j=1}^{n}(\bigvee_{l=1}^{n}\sigma_{ijl}\widetilde{f}_j(0)\widetilde{f}_l(0))| + |\bigwedge_{j=1}^{n}T_{ij}u_j| + |\bigvee_{j=1}^{n}H_{ij}u_j| + |I_i|]. \\
&\geq \lambda \sum_{i=1}^{n}[r_i d_i - \sum_{j=1}^{n} r_j A_{ji} L_j - \sum_{j=1}^{n} r_j L_j B_{ji}]|x_i| - \lambda n I_0 \\
&\geq \lambda r_0 \|x\|_1 - \lambda n I_0,
\end{aligned}$$

where

$$\begin{aligned}
r_0 &= \min_{1\leq i\leq n}\{r_i d_i - \sum_{j=1}^{n} r_j A_{ji} L_j - \sum_{j=1}^{n} r_j L_j B_{ji}\}, \\
I_0 &= \max_{1\leq i\leq n}\{r_i [\sum_{j=1}^{n} |a_{ij}||\widetilde{f}_j(0)| + |\bigwedge_{j=1}^{n}\alpha_{ij}\widetilde{f}_j(0)| + |\bigvee_{j=1}^{n}\beta_{ij}\widetilde{f}_j(0)| \\
&\quad +|\sum_{j=1}^{n}\sum_{l=1}^{n} b_{ijl}\widetilde{f}_j(0)\widetilde{f}_l(0)| + |\bigwedge_{j=1}^{n}(\bigwedge_{l=1}^{n}\gamma_{ijl}\widetilde{f}_j(0)\widetilde{f}_l(0))| \\
&\quad +|\bigvee_{j=1}^{n}(\bigvee_{l=1}^{n}\sigma_{ijl}\widetilde{f}_j(0)\widetilde{f}_l(0))| + |\bigwedge_{j=1}^{n}T_{ij}u_j| + \bigvee_{j=1}^{n}H_{ij}u_j| + |I_i|]\}.
\end{aligned}$$

Let

$$B(H_0) = \{x | \|x\|_1 < H_0 = \frac{n(I_0+1)}{r_0}\},$$

then, we have $\|x\|_1 = H_0 = \frac{n(I_0+1)}{r_0}$, for any $x \in \partial B(H_0)$, so, we obtain

$$\sum_{i=1}^{n} r_i |G_i(x,\lambda)| \geq \lambda r_0 \frac{n(I_0+1)}{r_0} - \lambda n I_0 \quad \lambda \in (0,1].$$

When $\lambda = 0$, $G(x,\lambda) = i_d(x) = x \neq 0$, for any $x \in \partial B(H_0)$, here, i_d is identity mapping. Consequently, we have $G(x,\lambda) \neq 0$, for any $x \in \partial B(H_0), \lambda \in [0,1]$.

From (H_2), it is easy to prove $\deg(i_d, B(H_0), 0) = 1$, hence, we have from homotopy invariance theorem that

$$deg(g, B(H_0), 0) = deg(i_d, B(H_0), 0) = 1.$$

This shows that system (1) has at least one equilibrium point $x^* = (x_1^*, x_2^*, \cdots, x_n^*)$.

Assume that $y^* = (y_1^*, y_2^*, \cdots, y_n^*)^T$ is also an equilibrium point of system (1), then, we have

$$g_i(x^*) = 0,\ g_i(y^*) = 0,\ i = 1, 2, \cdots, n.$$

This implies that

$$\begin{aligned} d_i|x_i^* - y_i^*| &\leq \sum_{j=1}^{n} A_{ij}|\widetilde{f}_j(x_j^*) - \widetilde{f}_j(y_j^*)| + \sum_{j=1}^{n} B_{ij}|\widetilde{f}_j(x_j^*) - \widetilde{f}_j(y_j^*)| \\ &\leq \sum_{j=1}^{n} A_{ij}L_j|x_j^* - y_j^*| + \sum_{j=1}^{n} B_{ij}L_j|x_j^* - y_j^*| \\ &\quad i = 1, 2, \cdots, n. \end{aligned} \tag{4}$$

(3) can be rewriten as

$$(D - AL - BL)(|x_1^* - y_1^*|, |x_2^* - y_2^*|, \cdots, |x_n^* - y_n^*|) \leq 0. \tag{5}$$

Since $D - AL - BL$ is a nonsingular M-matrix, so $(D - AL - BL)^{-1}$ is a nonnegative matrix. Thus multiplying both sides of (4) by $(D - AL - BL)^{-1}$, we obtain

$$|x_1^* - y_1^*|, |x_2^* - y_2^*|, \cdots, |x_n^* - y_n^*| \leq 0.$$

This means that $x^* = y^*$. This implies that system (1) has one unique equilibrium point.

4 Exponential Stability of Equilibrium

Theorem 2. *Under assumptions* $(H_1)-(H_3)$*, equilibrium point* x^* *of system (1) is globally exponentially stable if* $D(1-\delta)-(A+B)L$ *is a nonsingular M-matrix.*

Proof. If $D(1-\delta)-(A+B)L$ is a nonsingular M-matrix, then $D-(A+B)L$ is also a nonsingular M-matrix. Hence, from Theorem 1, equilibrium point x^* of system (1) exists and is unique. Let $x(t)=(x_1(t),x_2(t),\cdots,x_n(t))$ be any solution of system (1) with initial condition $x_i(s)=\varphi_i(s)$, for all $s\in[-\tau,0]$ $(i=1,2,\cdots,n)$ and $x^*=(x_1^*,x_2^*,\cdots,x_n^*)^T$. We define

$$y_i(t)=x_i(t)-x_i^*,\ f_i(y_i(t))=\widetilde{f}_i(x_i(t)+x_i^*)-\widetilde{f}_i(x_i^*),\ i=1,2,\cdots,n,$$

then, from system (1) we have

$$\begin{aligned}\frac{\mathrm{d}y_i(t)}{\mathrm{d}t}=&-d_iy_i(t)+\sum_{j=1}^{n}a_{ij}f_j(y_j(t-\tau_{ij}(t)))\\&+\sum_{j=1}^{n}\sum_{l=1}^{n}[b_{ijl}\widetilde{f}_l(x_l(t-\tau_{ij}(t)))+b_{ilj}\widetilde{f}_l(x_l^*)]f_j(y_j(t-\tau_{ij}(t)))\\&+\bigwedge_{j=1}^{n}\alpha_{ij}\widetilde{f}_j(y_j(t-\tau_{ij}(t))+x_j^*)-\bigwedge_{j=1}^{n}\alpha_{ij}\widetilde{f}_j(x_j^*)\\&+\bigvee_{j=1}^{n}\beta_{ij}\widetilde{f}_j(y_j(t-\tau_{ij}(t))+x_j^*)-\bigvee_{j=1}^{n}\beta_{ij}\widetilde{f}_j(x_j^*).\\&+\bigwedge_{j=1}^{n}(\bigwedge_{l=1}^{n}\gamma_{ijl}\widetilde{f}_j(y_j(t-\tau_{ij}(t))+x_j^*)\widetilde{f}_l(y_l(t-\tau_{ij}(t))+x_l^*))\\&-\bigwedge_{j=1}^{n}(\bigwedge_{l=1}^{n}\gamma_{ijl}\widetilde{f}_j(x_j^*)\widetilde{f}_l(x_l^*))\\&+\bigvee_{j=1}^{n}(\bigvee_{l=1}^{n}\sigma_{ijl}\widetilde{f}_j(y_j(t-\tau_{ij}(t))+x_j^*)\widetilde{f}_l(y_l(t-\tau_{ij}(t))+x_l^*))\\&-\bigvee_{j=1}^{n}(\bigvee_{l=1}^{n}\sigma_{ijl}\widetilde{f}_j(x_j^*)\widetilde{f}_l(x_l^*)).\end{aligned}\tag{6}$$

According to Lemma 1, we have

$$\frac{d|y_i(t)|}{dt}\le-d_i|y_i(t)|+\textstyle\sum_{j=1}^{n}(A_{ij}+B_{ij})|f_j(y_j(t-\tau_{ij}(t)))|.$$

If $D(1-\delta)-(A+B)L$ is a nonsingular M-matrix, then there exist constants $r_i>0$ and a sufficiently small positive constant λ such that

$$(d_i-\lambda)r_i-\sum_{j=1}^{n}r_jL_i\frac{A_{ji}+B_{ji}}{1-\delta}e^{\lambda\tau}>0,\quad i=1,2,\cdots,n.$$

Take Lyapunov functional as

$$V(y(t)) = \sum_{i=1}^{n} r_i |y_i(t)| e^{\lambda t} + \sum_{i=1}^{n} r_i \int_{t-\tau_{ij}(t)}^{t} \sum_{j=1}^{n} \frac{A_{ij} + B_{ij}}{1 - \dot{\tau}_{ij}(\psi_{ij}^{-1}(t))} \times |f_j(y_j(s))| e^{\lambda(s+\tau_{ij}(\psi_{ij}^{-1}(s)))} \mathrm{d}s,$$

where $\psi_{ij}^{-1}(t)$ is the inverse function of $\psi_{ij}(t) = t - \tau_{ij}(t)$.

Calculating the derivative of $V(y(t))$, we have

$$\begin{aligned} \frac{\mathrm{d}V(y(t))}{\mathrm{d}t} &\le \sum_{i=1}^{n} r_i(-d_i|y_i(t)| + \sum_{j=1}^{n}(A_{ij} + B_{ij})|f_j(y_j(t - \tau_{ij}(t)))|)e^{\lambda t} \\ &\quad + \sum_{i=1}^{n} r_i \lambda |y_i(t)| e^{\lambda t} + \sum_{i=1}^{n} r_i \sum_{j=1}^{n} \frac{A_{ij} + B_{ij}}{1 - \dot{\tau}_{ij}(\psi_{ij}^{-1}(t))} L_j |y_j(t)| \\ &\quad \times e^{\lambda(t+\tau_{ij}(\psi_{ij}^{-1}(t)))} - \sum_{i=1}^{n} r_i \sum_{j=1}^{n} (A_{ij} + B_{ij}) L_j |y_j(t - \tau_{ij}(t))| e^{\lambda t}. \\ &\le \sum_{i=1}^{n} [(\lambda - d_i) r_i + \sum_{j=1}^{n} r_j L_i \frac{A_{ij} + B_{ij}}{1 - \delta} e^{\lambda \tau}] |y_i(t)| e^{\lambda t} < 0. \end{aligned}$$

Which implies that

$$V(y(t)) \le V(y(0)), \text{ for all } t \ge 0.$$

So, we have

$$\sum_{i=1}^{n} r_i |y_i(t)| e^{\lambda t} \le \sum_{i=1}^{n} |y_i(0)| + \sum_{i=1}^{n} r_i \max_{1 \le j \le n} \frac{A_{ij} + B_{ij}}{1 - \delta} \|\phi\| L_j \int_{-\tau}^{0} e^{\lambda(s+\tau)} \mathrm{d}s.$$

Let

$$\gamma = \frac{\sum_{i=1}^{n} |y_i(0)| + \sum_{i=1}^{n} r_i \max_{1 \le j \le n} \frac{A_{ij} + B_{ij}}{1 - \delta} \|\phi\| L_j \int_{-\tau}^{0} e^{\lambda(s+\tau)} \mathrm{d}s}{\min_{1 \le i \le n} r_i},$$

consequently,

$$\sum_{i=1}^{n} |y_i(t)| = \gamma e^{-\lambda t}.$$

This implies that the equilibrium point x^* is globally exponentially stable.

Remark 1. For system (1), if $\tau_{ij}(t) = \tau$ $(i, j = 1, 2, \cdots, n)$, where τ is a constant, then we have the following result as corollary of Theorem 1.

Corollary 1. *Under assumptions $(H_1) - (H_2)$, equilibrium point x^* of system (1) is globally exponentially stable if $D - AL - BL$ is a nonsingular M-matrix.*

5 Illustrative Example

In this section, we give a numerical example to demonstrate the effectiveness of our results.

Consider 2-dimension HFCNNs with time delays. For system (1), we take activation functions $f_i(x) = \frac{1}{2}(|x+1| - |x-1|)\,(i = 1, 2)$. Obviously, $f_i(x)$ satisfies Hypotheses (H_1) and (H_2) with $L_i = 0.8\ (i = 1, 2)$. Further, we take delays $\tau_{ij}(t) = \frac{1}{10} + \frac{9}{10}|\sin t|\,(i, j = 1, 2, \cdots, n)$ and initial functions $(\varphi_1, \varphi_2) = (-\frac{1}{2}\sin t, \cos t)^T$ for all $t \in [-1, 0]$. Further, we take

$$D = \begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix}, a = \begin{pmatrix} -0.2 & 0.1 \\ 0.1 & 0.1 \end{pmatrix}, B_1 = \begin{pmatrix} -0.1 & -0.1 \\ 0.1 & 0.11 \end{pmatrix},$$
$$B_2 = \begin{pmatrix} 0.1 & -0.1 \\ -0.14 & 0.13 \end{pmatrix}, \alpha = \begin{pmatrix} 0.2 & -0.1 \\ 0.1 & 0.1 \end{pmatrix}, \beta = \begin{pmatrix} 0.2 & 0.1 \\ -0.1 & 0.1 \end{pmatrix},$$
$$R_1 = \begin{pmatrix} 0.1 & -0.1 \\ 0.1 & 0.13 \end{pmatrix}, R_2 = \begin{pmatrix} 0.1 & -0.1 \\ -0.1 & 0.11 \end{pmatrix}, Q_1 = \begin{pmatrix} 0.1 & -0.14 \\ 0.13 & 0.12 \end{pmatrix},$$
$$Q_2 = \begin{pmatrix} 0.1 & -0.1 \\ -0.1 & 0.13 \end{pmatrix}, T = H = \begin{pmatrix} 0.3 & -0.23 \\ 0.2 & 0.3 \end{pmatrix},$$

$$a = (a_{ij})_{2\times 2},\ B_i = (b_{ijl})_{2\times 2},\ \alpha = (\alpha_{ij})_{2\times 2},\ \beta = (\beta_{ij})_{2\times 2},$$
$$R_i = (\gamma_{ijl})_{2\times 2},\ Q_i = (\sigma_{ijl})_{2\times 2},\ T = (T_{ij})_{2\times 2},\ T = (T_{ij})_{2\times 2}.$$

For this example, it is easy to see that if $(I_1, I_2)^T = (-1, 1)$ and $u_1 = u_2 = 1$, the conditions of Theorem 2 are satisfied. So the equilibrium point is unique and globally exponentially stable.

6 Conclusion

In this paper, we have investigated the existence and global exponential stability of equilibrium point for Fuzzy cellular neural networks. Using the nonsingular M-matrix and Lyapunov functional method, we gave a sufficient criterion ensuring the existence and global exponential stability of equilibrium point of system (1). The obtained result improves and extend several earlier publications and is useful in applications of manufacturing high quality neural networks.

Acknowledgement. This work was supported by The National Natural Science Foundation of P.R. China (60764003), The Major Project of The Ministry of Education of P.R. China (207130) and The Scientific Research Programmes of Colleges in Xinjiang (XJEDU2007G01, XJEDU2006I05).

References

1. Yang, T., Yang, L., Wu, C.W., Chua, L.O.: Fuzzy cellular neural networks: theory. In: Proceedings of the IEEE International Workshop on Cellular Neural Networks and their Applications, pp. 181–186 (1996)

2. Yang, T., Yang, L.: The global stability of fuzzy cellular network. IEEE Trans. Circ. Syst. -I 43, 880–883 (1996)
3. Huang, T.: Exponential stability of fuzzy cellular networks with distributed delay. Physics Letters A 351, 48–52 (2006)
4. Huang, T.: Exponential stability of delayed fuzzy cellular networks with diffusion. Chaos, Solitons and Fractals 31, 658–664 (2007)
5. Liu, Y., Tang, W.: Exponential stability of fuzzy cellular networks with constant and time-varying delays. Physics Letters A 323, 224–233 (2004)
6. Yuan, K., Cao, J., Deng, J.: Exponential stability and periodic solutions of fuzzy cellular networks with time-varying delays. Neurocomputing 69, 1619–1627 (2006)
7. Boyd, S., Ghaoui, L.E., Feron, E., Balakrishnan, V.: Linear Matrix Inequalities in System and Control Theory. SIAM, Philadelphia (1994)
8. Cao, J., Daniel, W.C.H.: A deneral framework for global asymptotic stability analysis of delayed neural networks based on LMI approach. Chaos, Solitons and Fractals 24, 1317–1329 (2005)
9. Vidyasagar, M.: Nonliear Systems Analysis, Englewood Cliffs, New Fersey (1993)
10. Gu, H., Jiang, H., Teng, Z.: Stability and periodicity in high-order neural networks with impulsive effects. Nonlinear Analysis Series A: TMA 68, 3186–3200 (2008)
11. Liu, X., Teo, K.L., Xu, B.: Exponential stability of impulsive high-order Hopfield-type neural networks with time-varying delays. IEEE Trans. Neural Networks 16, 1329–1339 (2005)
12. Cao, J., Liang, J., Lam, J.: Exponential stability of high-order bidirectional associative memory neural networks with time delays. Physica D 199, 425–436 (2004)
13. Wu, C., Ruan, J., Lin, W.: On the existence and stability of the periodic solution in the Cohen-Grossberg neural networks with time delay and high-order terms. Appl. Math. Comput. 177, 194–210 (2006)
14. Xiang, H., Yan, K., Wang, B.: Existence and global exponential stability of periodic solution for delayed high-order Hopfield-type neural networks. Physics Letters A 352, 341–349 (2006)
15. Cao, J., Wang, J.: Global Asymptotic Stability of a General Class of Recurrent Neural Networks With Time-Varying Delays. IEEE Trans. Circ. Syst. Fundamental theory and applications 50, 34–44 (2003)

Further Stability Analysis for Neural Networks with Time-Varying Interval Delay

Qiufeng Cai and Jianjiang Yu

School of Information Science and Technology, Yancheng Teachers University, Yancheng 224002, China
jsyccqf@163.com, jjyuseu@gmail.com

Abstract. This paper provides improved results on stability condition for a class of neural networks (NNs) with time-varying interval delay. The activation functions of the NNs are assumed to be more general. Based on a new augmented Lyapunov-Krasovskii functional, the improved delay-dependent stability criterion for delay NNs is obtained in terms of linear matrix inequalities (LMIs). It is shown that the new criterion can provide less conservative results than some existing ones. A numerical example is given to demonstrate the effectiveness and the benefits of the proposed method.

Keywords: Neural networks (NNs), Lyapunov-Krasovskii functional, Delay-dependent, Time-varying interval delay.

1 Introduction

In the past few decades, neural networks (NNs) have received considerable attention due to their extensive applications in a variety of areas, such as signal processing, pattern recognition, and combinatorial optimization [1]. Time delay is frequently encountered in NNs and it is often a source of instability and oscillations in a system, so increasing interest has been focused on stability analysis of NNs with time delays. Generally speaking, the so-far obtained stability results for delay NNs can be classified into two types; that is, delay-independent stability [2]-[4] and delay-dependent stability [5]-[16]; the former does not include any information on the size of delay while the latter employs such information.

For delay-dependent type, much attention has been paid to reduce the conservatism of stability conditions. Recently, a free-weighting matrices method was proposed in [8] to study the delay-dependent stability problems for NNs with time-varying delay. It has been shown effective in reducing conservatism by the introduction of free-weighting matrices. By considering the additional useful terms, some less conservative delay-dependent stability criteria for NNs were presented in [12] and [13]. With aid of an augmented Lyapunov-Krasovskii functional, an improved delay-dependent stability criterion was established in [16].

In this paper, by constructing a new augmented Lyapunov-Krasovskii functional, improved delay-dependent stability criterion for NNs with time-varying

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 423–432, 2009.

interval delay is obtained. In addition, the activation functions of NNs considered in this paper are more general than those in [13], [16], etc. Furthermore, a illustrative example is given to show the effectiveness of the proposed criteria.

Notation. Throughout this paper, a real symmetric matrix $P > 0 (\geq 0)$ denotes P being a positive definite (positive semi-definite) matrix. I is used to denote an identity matrix with proper dimension. Matrices, if not explicitly stated, are assumed to have compatible dimensions. The symmetric terms in a symmetric matrix are denoted by $*$. The superscript "T" represents the transpose.

2 Problem Formulation and Preliminaries

Consider the following NNs with time-varying delays:

$$\dot{x}(t) = -Cx + Ag(x(t)) + Bg(x(t - d(t))) + J \tag{1}$$

where $x(t) = [x_1(t),\ x_2(t), \cdots,\ x_n(t)]^T \in \mathbb{R}^n$, $g(x(t)) = [g_1(x_1(t)),\ g_2(x_2(t)),\ \cdots,\ g_n(x_n(t))]^T \in \mathbb{R}^n$, $g(x(t-d(t))) = [g_1(x_1(t-d(t))),\ g_2(x_2(t-d(t))), \cdots, g_n(x_n(t-d(t)))]^T \in \mathbb{R}^n$, $J = [J_1(\cdot),\ J_2(\cdot), \cdots,\ J_n(\cdot)]^T \in \mathbb{R}^n$, $C = diag\{c_1,\ c_2,\ \cdots,\ c_n\}$, $A = (a_{ij})_{n\times n}$ and $B = (b_{ij})_{n\times n}$. In the following, we assume that each neuron activation function in (1), $g_i(\cdot), i = 1, 2, ..., n$, satisfies the following condition:

$$k_i^- \leq \frac{g_i(x) - g_i(y)}{x - y} \leq k_i^+,\ \forall\ x, y \in \mathbb{R}, x \neq y,\ i = 1, 2, ..., n \tag{2}$$

where $k_i^-,\ k_i^+,\ i = 1, 2, ..., n$ are some constants. $d(t)$ corresponds to the time-varying transmission delay and satisfies

$$h_1 \leq d(t) \leq h_2, \qquad \dot{d}(t) \leq \mu, \tag{3}$$

where $0 \leq h_1 < h_2$ and $\mu > 0$ are constants. Note that h_1 may not be equal to 0.

Remark 1. The previous results in [12], [14], and [16] only considered the case that the range of the time-varying delay from 0 to an upper bound. Indeed, the lower bound of time-varying delay for NNs is always not equal to 0 in practice.

Remark 2. As pointed out in [10], the constants $k_i^-,\ k_i^+$ are allowed to be positive, negative or zero. Hence, the resulting activation functions could be non-monotonic, and more general than the usual sigmoid functions.

Assume $x^* = [x_1^*,\ x_2^*, ..., x_n^*]^T$ is an equilibrium of system (1), one can derive from (1) that the transformation $z(\cdot) = x(\cdot) - x^*$ transforms system (1) into the following system:

$$\dot{z}(t) = -Cz(t) + Af(z(t)) + Bf(z(t - d(t))) \tag{4}$$

where $z(t) = [z_1(t),\ z_2(t), \cdots, z_n(t)]^T$ is the state vector of the transformed system, $f(z(t)) = [f_1(z_1(t)),\ f_2(z_2(t)), \cdots, f_n(z_n(t))]^T$, $f(x(t-d(t))) = [f_1(x_1(t-d(t))),\ f_2(x_2(t-d(t))), \cdots, f_n(x_n(t-d(t)))]^T \in \mathbb{R}^n$ and $f_i(z_i(t)) = g_i(z_i(t) + x_i^*) - g_i(x_i^*),\ i = 1, 2, ..., n$. Note that the functions $f_i(\cdot), i = 1, 2, ..., n$ satisfy the following condition:

$$k_i^- \leq \frac{f_i(x) - f_i(y)}{x - y} \leq k_i^+,\ \forall\ x, y \in R, x \neq y,\ i = 1, 2, ..., n \tag{5}$$

The main purpose of this paper is to establish LMI-based sufficient conditions guaranteeing the global asymptotic stability of delay NNs (4). To obtain our main results, we need the following lemmas:

Lemma 1. *([13]) Given constant matrices $\Omega_1, \Omega_2, \Omega_3$ where $\Omega_1 = \Omega_1^T$ and $\Omega_2 > 0$, then*

$$\Omega_1 + \Omega_3^T \Omega_2^{-1} \Omega_3 < 0 \tag{6}$$

if only if

$$\begin{bmatrix} \Omega_1 & \Omega_3^T \\ \Omega_3 & -\Omega_2 \end{bmatrix} < 0,\ or\ \begin{bmatrix} -\Omega_2 & \Omega_3 \\ \Omega_3^T & \Omega_1 \end{bmatrix} < 0 \tag{7}$$

Lemma 2. *([17]) For any constant matrix $M \in \mathbb{R}^{n \times n}$, $M = M^T > 0$, a scalar $\gamma > 0$, vector function $\omega :\ [0,\ \gamma] \to \mathbb{R}^n$, then*

$$\left(\int_0^\gamma \omega(s) ds \right)^T M \left(\int_0^\gamma \omega(s) ds \right) \leq \gamma \int_0^\gamma \omega^T(s) M \omega(s) ds. \tag{8}$$

3 Main Results

In the section, a new augmented Lyapunov-Krasovskii functional is constructed and the following asymptotic stability criterion is obtained.

Theorem 1. *For given scalars $0 \leq h_1 < h_2,\ \mu > 0$, and $h_{12} = h_2 - h_1$. Let matrices $K_1 = diag\{k_1^+ k_1^-,\ k_2^+ k_2^-, ..., k_n^+ k_n^-\},\ K_2 = diag\{(k_1^+ + k_1^-),\ (k_2^+ + k_2^-), ..., (k_n^+ + k_n^-)\}$, and $K_3 = diag\{k_1^-,\ k_2^-, ..., k_n^-\}$. Then, for any delay $d(t)$ satisfy (3), the origin of system (4) with (5) is globally asymptotically stable, if there exist positive diagonal matrices $\Lambda = diag\{\lambda_1,\ \lambda_2, ...\lambda_n\},\ D_1 = diag\{d_{11},\ d_{12}, ...d_{1n}\},\ D_2 = diag\{d_{21},\ d_{22}, ...d_{2n}\}$ and matrices $\begin{bmatrix} P_{11} & P_{12} \\ * & P_{22} \end{bmatrix} > 0,\ \begin{bmatrix} Q_{11} & Q_{12} \\ * & Q_{22} \end{bmatrix} > 0,\ X = \begin{bmatrix} X_{11} & X_{12} \\ * & X_{22} \end{bmatrix} > 0,\ Y = \begin{bmatrix} Y_{11} & Y_{12} \\ * & Y_{22} \end{bmatrix} > 0,\ R_i > 0\ (i = 1, 2),\ Q_i > 0\ (i = 1, 2, 3, 4)\ ,\ Z_i > 0\ (i = 1, 2)$ and any matrices $P_i\ (i = 2, \cdots, 7)$ with appropriate dimensions, such that the following LMIs(9)-(12) are feasible:*

$$\Xi = \begin{bmatrix} \Xi_{11} & \Xi_{12} & \Xi_{13} & \Xi_{14} & P_4^T & -P_6^T & \Xi_{17} & 0 & -C^TU & \mu P_{12} & 0 \\ * & \Xi_{22} & 0 & D_2K_2 & P_5^T & \Xi_{26} & \Xi_{27} & -\frac{h_2}{h_{12}}Q_{12}^T & 0 & 0 & \mu P_{22} \\ * & * & \Xi_{33} & \Lambda B & 0 & 0 & A^TP_{12} & 0 & A^TU & 0 & 0 \\ * & * & * & \Xi_{44} & 0 & 0 & B^TP_{12} & 0 & B^TU & 0 & 0 \\ * & * & * & * & -Q_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ * & * & * & * & * & \Xi_{66} & 0 & \frac{h_2}{h_{12}}Q_{12}^T & 0 & 0 & 0 \\ * & * & * & * & * & * & \Xi_{77} & 0 & 0 & 0 & 0 \\ * & * & * & * & * & * & * & -\frac{h_2}{h_{12}}Q_{11} & 0 & 0 & 0 \\ * & * & * & * & * & * & * & * & -U & 0 & 0 \\ * & * & * & * & * & * & * & * & * & -\mu Z_1 & 0 \\ * & * & * & * & * & * & * & * & * & * & -\mu Z_2 \end{bmatrix}$$
$$< 0, \tag{9}$$

$$\Pi_1 = \begin{bmatrix} X_{11} & X_{12} & P_2^T \\ * & X_{22} & P_3^T \\ * & * & R_1 \end{bmatrix} \geq 0, \tag{10}$$

$$\Pi_2 = \begin{bmatrix} Y_{11} & Y_{12} & P_4^T \\ * & Y_{22} & P_5^T \\ * & * & R_2 \end{bmatrix} \geq 0, \tag{11}$$

$$\Pi_3 = \begin{bmatrix} X_{11}+Y_{11} & X_{12}+Y_{12} & P_6^T \\ * & X_{22}+Y_{22} & P_7^T \\ * & * & R_1+R_2 \end{bmatrix} \geq 0, \tag{12}$$

where

$$\begin{aligned}
\Xi_{11} &= -P_{11}C - C^TP_{11} + P_2 + P_2^T + \Lambda K_3C + C^TK_3^T\Lambda^T + Q_1 \\
&\quad +Q_3 + Q_4 + h_2^2Q_{11} - h_2^2C^TQ_{12}^T - h_2^2Q_{12}C - Q_{22} \\
&\quad +P_{12} + P_{12}^T - 2D_1K_1 + h_2X_{11} + h_{12}Y_{11}, \\
\Xi_{12} &= P_3 - P_2^T + P_6^T - P_4^T - P_{12} + Q_{22} + h_2X_{12} + h_{12}Y_{12}, \\
\Xi_{13} &= P_{11}A - C^T\Lambda - \Lambda K_3A + h_2^2Q_{12}A + D_1K_2, \\
\Xi_{14} &= P_{11}B - \Lambda K_3B + h_2^2Q_{12}B, \\
\Xi_{17} &= -Q_{12}^T - C^TP_{12} + P_{22}^T, \\
\Xi_{22} &= -P_3 - P_3^T - P_5 - P_5^T + P_7 + P_7^T + \mu Z_1 - (1-\mu)Q_1 \\
&\quad -Q_{22} - \frac{h_2}{h_{12}}Q_{22} - 2D_2K_1 + h_2X_{22} + h_{12}Y_{22}, \\
\Xi_{26} &= -P_7^T + \frac{h_2}{h_{12}}Q_{12}^T, \\
\Xi_{27} &= -P_{22}^T + Q_{12}^T, \\
\Xi_{33} &= \Lambda A + A^T\Lambda + Q_2 - 2D_1, \\
\Xi_{44} &= -(1-\mu)Q_2 - 2D_2, \\
\Xi_{66} &= -Q_4 - \frac{h_2}{h_{12}}Q_{22}, \\
\Xi_{77} &= \mu Z_2 - Q_{11}, \\
U &= h_2R_1 + h_{12}R_2 + h_2^2Q_{22}.
\end{aligned}$$

Proof. Choose a new augmented Lyapunov-Krasovskii functional candidate as follows:

$$V(z(t)) = V_1(z(t)) + V_2(z(t)) + V_3(z(t)) + V_4(z(t)), \tag{13}$$

where

$$V_1(z(t)) = \xi_0^T(t)EP\xi_0(t) + 2\sum_{i=1}^{n}\lambda_i\int_0^{z_i(t)}(f_i(s) - k_i^- s)ds,$$

$$V_2(z(t)) = \int_{t-d(t)}^{t}[z^T(s)Q_1z(s)ds + f^T(z(s))Q_2f(z(s))]ds + \int_{t-h_1}^{t}z^T(s)Q_3z(s)ds + \int_{t-h_2}^{t}z^T(s)Q_4z(s)ds,$$

$$V_3(z(t)) = \int_{-h_2}^{0}\int_{t+\theta}^{t}\dot{z}^T(s)R_1\dot{z}(s)dsd\theta + \int_{-h_2}^{-h_1}\int_{t+\theta}^{t}\dot{z}^T(s)R_2\dot{z}(s)dsd\theta,$$

$$V_4(z(t)) = h_2\int_{-h_2}^{0}\int_{t+\theta}^{t}\begin{bmatrix}z(s)\\ \dot{z}(s)\end{bmatrix}^T\begin{bmatrix}Q_{11} & Q_{12}\\ * & Q_{22}\end{bmatrix}\begin{bmatrix}z(s)\\ \dot{z}(s)\end{bmatrix}dsd\theta,$$

where $E = \begin{bmatrix}I & 0 & 0 & 0 & 0\\ 0 & I & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0\end{bmatrix}$, $P = \begin{bmatrix}P_{11} & P_{12} & 0\\ P_{12}^T & P_{22} & 0\\ P_2 & 0 & P_3\\ P_4 & 0 & P_5\\ P_6 & 0 & P_7\end{bmatrix}$, $\xi_0(t) = \begin{bmatrix}z(t)\\ \int_{t-d(t)}^{t}z(s)ds\\ z(t-d(t))\end{bmatrix}$, $Q_i = Q_i^T > 0, i = 1, \cdots, 4$, $R_i = R_i^T > 0, i = 1, 2$, $\Lambda = diag\{\lambda_1, \lambda_2, \cdots, \lambda_n\} \geq 0$, $\begin{bmatrix}P_{11} & P_{12}\\ * & P_{22}\end{bmatrix} > 0$, and $P_j, j = 2, \cdots, 7$ are any matrices with appropriate dimensions. It is easy to see that $EP = P^TE^T > 0$, and $\xi_0^T(t)EP\xi_0(t)$ is actually $\begin{bmatrix}z(t)\\ \int_{t-d(t)}^{t}z(s)ds\end{bmatrix}^T\begin{bmatrix}P_{11} & P_{12}\\ * & P_{22}\end{bmatrix}\begin{bmatrix}z(t)\\ \int_{t-d(t)}^{t}z(s)ds\end{bmatrix}$.

On the other hand, from the Leibniz-Newton formula, the following equations are true

$$\alpha_1 := z(t) - z(t-d(t)) - \int_{t-d(t)}^{t}\dot{z}(s)ds = 0$$

$$\alpha_2 := z(t-h_1) - z(t-d(t)) - \int_{t-h_2}^{t-h_1}\dot{z}(s)ds = 0$$

$$\alpha_3 := z(t-d(t)) - z(t-h_2) - \int_{t-h_2}^{t-d(t)}\dot{z}(s)ds = 0$$

Then calculating the time derivative of $V(z(t))$ along the solution of (4) yields

$$\dot{V}_1(z(t)) = -2z^T(t)P_{11}Cz(t) + 2z^T(t)P_{11}Af(z(t)) + 2z^T(t)P_{11}Bf(z(t-d(t)))$$

$$
\begin{aligned}
&-2\left(\int_{t-d(t)}^{t} z(s)ds\right)^T P_{12}^T C z(t) + 2\left(\int_{t-d(t)}^{t} z(s)ds\right)^T P_{12}^T A f(z(t)) \\
&+2\left(\int_{t-d(t)}^{t} z(s)ds\right)^T P_{12}^T B f(z(t-d(t))) + 2z^T(t)P_{12}z(t) \\
&-2z^T(t)P_{12}z(t-d(t)) + 2\left(\int_{t-d(t)}^{t} z(s)ds\right)^T P_{22}z(t) \\
&-2\left(\int_{t-d(t)}^{t} z(s)ds\right)^T P_{22}z(t-d(t)) + 2\dot{d}(t)z^T(t)P_{12}z(t-d(t)) \\
&+2\dot{d}(t)\left(\int_{t-d(t)}^{t} z(s)ds\right)^T P_{22}z(t-d(t)) \\
&+2\xi_0^T(t)P^T\begin{bmatrix}0\\0\\ \alpha_1\\ \alpha_2\\ \alpha_3\end{bmatrix} + 2\sum_{i=1}^{n}\lambda_i(f_i(z_i(t)) - k_i^- z_i(t))\dot{z}_i(t) \qquad (14)
\end{aligned}
$$

On the other hand, for some matrices $Z_1 > 0,\ Z_2 > 0$, the following inequities always hold based on (3)

$$
\begin{aligned}
&2\dot{d}(t)z^T(t)P_{12}z(t-d(t)) \\
&\le \mu z^T(t)P_{12}Z_1^{-1}P_{12}^T z(t) + \mu z^T(t-d(t))Z_1 z(t-d(t)) \qquad (15)\\
&2\dot{d}(t)\left(\int_{t-d(t)}^{t} z(s)ds\right)^T P_{22}z(t-d(t)) \\
&\le \mu\left(\int_{t-d(t)}^{t} z(s)ds\right)^T Z_2\left(\int_{t-d(t)}^{t} z(s)ds\right) \\
&\quad +\mu z^T(t-d(t))P_{22}Z_2^{-1}P_{22}^T z(t-d(t)) \qquad (16)
\end{aligned}
$$

Then,

$$
\begin{aligned}
\dot{V}_1(z(t)) = {} & \xi^T(t)\tilde{\Xi}\xi(t) - 2\xi_0^T(t)\begin{bmatrix}P_2^T\\0\\P_3^T\end{bmatrix}\int_{t-d(t)}^{t}\dot{z}(s)ds \\
& -2\xi_0^T(t)\begin{bmatrix}P_4^T\\0\\P_5^T\end{bmatrix}\int_{t-h_2}^{t-h_1}\dot{z}(s)ds - 2\xi_0^T(t)\begin{bmatrix}P_6^T\\0\\P_7^T\end{bmatrix}\int_{t-h_2}^{t-d(t)}\dot{z}(s)ds \\
& +\mu z^T(t)P_{12}Z_1^{-1}P_{12}^T z(t) \\
& +\mu z^T(t-d(t))P_{22}Z_2^{-1}P_{22}^T z(t-d(t)) \qquad (17)
\end{aligned}
$$

where

$$\xi^T(t) = [z^T(t) \quad z^T(t-d(t)) \quad f^T(z(t)) \quad f(z(t-d(t))) \\ z^T(t-h_1) \quad z^T(t-h_2) \quad (\int_{t-d(t)}^{t} z(s)ds)^T \quad (\int_{t-h_2}^{t-d(t)} z(s)ds)^T],$$

and

$$\tilde{\Xi} = \begin{bmatrix} \tilde{\Xi}_{11} & \tilde{\Xi}_{12} & \tilde{\Xi}_{13} & \tilde{\Xi}_{14} & P_4^T & -P_6^T & \tilde{\Xi}_{17} & 0 \\ * & \tilde{\Xi}_{22} & 0 & 0 & P_5^T & -P_7^T & -P_{22} & 0 \\ * & * & \Lambda A + A^T\Lambda & \Lambda B & 0 & 0 & A^T P_{12} & 0 \\ * & * & * & 0 & 0 & 0 & B^T P_{12} & 0 \\ * & * & * & * & 0 & 0 & 0 & 0 \\ * & * & * & * & * & 0 & 0 & 0 \\ * & * & * & * & * & * & \mu Z_2 & 0 \\ * & * & * & * & * & * & * & 0 \end{bmatrix},$$

$$\tilde{\Xi}_{11} = -P_1C - C^TP_1 + \Lambda K_3C + C^TK_3\Lambda + P_2 + P_2^T + P_{12} + P_{12}^T,$$
$$\tilde{\Xi}_{12} = -P_2^T + P_3 - P_4^T + P_6^T - P_{12},$$
$$\tilde{\Xi}_{13} = P_1A - C^T\Lambda - \Lambda K_3A,$$
$$\tilde{\Xi}_{14} = P_1B - \Lambda K_3B,$$
$$\tilde{\Xi}_{17} = -C^TP_{12} + P_{22},$$
$$\tilde{\Xi}_{22} = -P_3 - P_3^T - P_5 - P_5^T + P_7 + P_7^T + \mu Z_1.$$

In addition,

$$\begin{aligned} \dot{V}_2(z(t)) \le\ & z^T(t)(Q_1 + Q_3 + Q_4)z(t) + f^T(z(t))Q_2f(z(t)) \\ & -(1-\mu)z^T(t-d(t))Q_1z(t-d(t)) \\ & -(1-\mu)f^T(z-d(t))Q_2f(z-d(t)) \\ & -z^T(t-h_1)Q_3z(t-h_1) - z^T(t-h_2)Q_4z(t-h_2), \end{aligned} \tag{18}$$

and,

$$\begin{aligned} \dot{V}_3(z(t)) =\ & \dot{z}^T(t)(h_2R_1 + h_{12}R_2)\dot{z}(t) - \int_{t-d(t)}^{t} \dot{z}^T(s)R_1\dot{z}(s)ds \\ & - \int_{t-d(t)}^{t-h_1} \dot{z}^T(s)R_2\dot{z}(s)ds - \int_{t-h_2}^{t-d(t)} \dot{z}^T(s)(R_1+R_2)\dot{z}(s)ds, \end{aligned} \tag{19}$$

and,

$$\begin{aligned} \dot{V}_4(z(t)) =\ & h_2^2 \begin{bmatrix} z(t) \\ \dot{z}(t) \end{bmatrix}^T \begin{bmatrix} Q_{11} & Q_{12} \\ * & Q_{22} \end{bmatrix} \begin{bmatrix} z(t) \\ \dot{z}(t) \end{bmatrix} \\ & -h_2 \int_{t-h_2}^{t} \begin{bmatrix} z(s) \\ \dot{z}(s) \end{bmatrix}^T \begin{bmatrix} Q_{11} & Q_{12} \\ * & Q_{22} \end{bmatrix} \begin{bmatrix} z(s) \\ \dot{z}(s) \end{bmatrix} ds, \end{aligned} \tag{20}$$

with

$$\begin{aligned} & h_2^2 \begin{bmatrix} z(t) \\ \dot{z}(t) \end{bmatrix}^T \begin{bmatrix} Q_{11} & Q_{12} \\ * & Q_{22} \end{bmatrix} \begin{bmatrix} z(t) \\ \dot{z}(t) \end{bmatrix} \\ =\ & h_2^2 z^T(t) \left[Q_{11} - 2Q_{12}C\right] z(t) + z^T(t)(2h_2^2Q_{12}A)f(z(t)) \\ & + z^T(t)2h_2^2Q_{12}Bf(z(t-d(t))) + \dot{z}^T(t)h_2^2Q_{22}\dot{z}(t). \end{aligned} \tag{21}$$

It follows from Lemma 2, that the rightmost term of (20) satisfies

$$
\begin{aligned}
&-h_2\int_{t-h_2}^{t}\begin{bmatrix}z(s)\\ \dot z(s)\end{bmatrix}^T\begin{bmatrix}Q_{11}&Q_{12}\\ *&Q_{22}\end{bmatrix}\begin{bmatrix}z(s)\\ \dot z(s)\end{bmatrix}dt\\
&\le\begin{bmatrix}z(t)\\ z(t-d(t))\\ \int_{t-d(t)}^{t}z(s)ds\end{bmatrix}^T\begin{bmatrix}-Q_{22}&Q_{22}&-Q_{12}^T\\ *&-Q_{22}&Q_{12}^T\\ *&*&-Q_{11}\end{bmatrix}\begin{bmatrix}z(t)\\ z(t-d(t))\\ \int_{t-d(t)}^{t}z(s)ds\end{bmatrix}\\
&\quad+\begin{bmatrix}z(t-d(t))\\ z(t-h_2)\\ \int_{t-h_2}^{t-d(t)}z(s)ds\end{bmatrix}^T\begin{bmatrix}-\frac{h_2}{h_{12}}Q_{22}&\frac{h_2}{h_{12}}Q_{22}&-\frac{h_2}{h_{12}}Q_{12}^T\\ *&-\frac{h_2}{h_{12}}Q_{22}&\frac{h_2}{h_{12}}Q_{12}^T\\ *&*&-\frac{h_2}{h_{12}}Q_{11}\end{bmatrix}\begin{bmatrix}z(t-d(t))\\ z(t-h_2)\\ \int_{t-h_2}^{t-d(t)}z(s)ds\end{bmatrix}
\end{aligned}\tag{22}
$$

Defining $D_1=diag\{d_{11},\ d_{12},...d_{1n}\}$, $D_2=diag\{d_{21},\ d_{22},...d_{2n}\}$, $K_1=diag\{k_1^+k_1^-,\ k_2^+k_2^-,...,k_n^+k_n^-\}$, $K_2=diag\{(k_1^++k_1^-),\ (k_2^++k_2^-),...,(k_n^++k_n^-)\}$, one can infer from (5) that

$$
\begin{aligned}
&-2\sum_{i=1}^{n}d_{1i}\begin{bmatrix}z(t)\\ f(z(t))\end{bmatrix}^T\begin{bmatrix}k_i^+k_i^-e_ie_i^T&-\frac{k_i^++k_i^-}{2}e_ie_i^T\\ -\frac{k_i^++k_i^-}{2}e_ie_i^T&e_ie_i^T\end{bmatrix}\begin{bmatrix}z(t)\\ f(z(t))\end{bmatrix}-2\sum_{i=1}^{n}d_{2i}\times\\
&\begin{bmatrix}z(t-d(t))\\ f(z(t-d(t)))\end{bmatrix}^T\begin{bmatrix}k_i^+k_i^-e_ie_i^T&-\frac{k_i^++k_i^-}{2}e_ie_i^T\\ -\frac{k_i^++k_i^-}{2}e_ie_i^T&e_ie_i^T\end{bmatrix}\begin{bmatrix}z(t-d(t))\\ f(z(t-d(t)))\end{bmatrix}\ge 0.
\end{aligned}\tag{23}
$$

Moreover, for any appropriately dimensioned matrices $X=X^T\ge 0$ and $Y=Y^T\ge 0$, the following equations hold:

$$
0=h_2\xi_0(t)X\xi_0(t)-\int_{t-d(t)}^{t}\xi_0(t)X\xi_0(t)ds-\int_{t-h_2}^{t-d(t)}\xi_0(t)X\xi_0(t)ds \tag{24}
$$

$$
0=h_{12}\xi_0(t)Y\xi_0(t)-\int_{t-d(t)}^{t-h_1}\xi_0(t)Y\xi_0(t)ds-\int_{t-h_2}^{t-d(t)}\xi_0(t)Y\xi_0(t)ds \tag{25}
$$

then, Considering (14)-(22), (23), (24), and (25), we have

$$
\begin{aligned}
\dot V(z(t))\le{}&\xi^T(t)\Xi_0\xi(t)+\dot z^T(t)U\dot z(t)+\mu z^T(t)P_{12}Z_1^{-1}P_{12}^Tz(t)\\
&+\mu z^T(t-d(t))P_{22}Z_2^{-1}P_{22}^Tz(t-d(t))-\int_{t-d(t)}^{t}\xi_1^T(t,s)\Pi_1\xi_1(t,s)ds\\
&-\int_{t-d(t)}^{t-h_1}\xi_1^T(t,s)\Pi_2\xi_1(t,s)ds-\int_{t-h_2}^{t-d(t)}\xi_1^T(t,s)\Pi_2\xi_1(t,s)ds,
\end{aligned}\tag{26}
$$

with

$$
\Xi_0=\begin{bmatrix}
\Xi_{11}&\Xi_{12}&\Xi_{13}&\Xi_{14}&P_4^T&-P_6^T&\Xi_{17}&0\\
*&\Xi_{22}&0&D_2K_2&P_5^T&\Xi_{26}&\Xi_{27}&-\frac{h_2}{h_{12}}Q_{12}^T\\
&&\Xi_{33}&\Lambda B&0&0&A^TP_{12}&0\\
&&*&\Xi_{44}&0&0&B^TP_{12}&0\\
&&*&*&-Q_3&0&0&0\\
&&*&*&*&\Xi_{66}&0&\frac{h_2}{h_{12}}Q_{12}^T\\
&&*&*&*&*&\Xi_{77}&0\\
&&*&*&*&*&*&-\frac{h_2}{h_{12}}Q_{11}
\end{bmatrix},
$$

$$\xi_1(t,s) = \left[z^T(t)\ z^T(t-d(t))\ \dot{z}^T(s) \right]^T .$$

Applying the Schur complement equivalence to (9) gives $\dot{V}(z(t)) < -\epsilon\|z(t)\|^2$ for a sufficiently small $\epsilon > 0$. Then, the system (4) is asymptotically stable.

Lemma 3. *It is worth pointing out that $d(t), h_2 - d(t)$, and $d(t) - h_1$ are not simply enlarged as h_2, $h_2 - h_1$, and $h_2 - h_1$, respectively. Instead the term $h_2 - d(t)$ is also considered, which may reduce conservatism.*

4 Numerical Example

This section presents a numerical example to demonstrate the validity of the method described above.

Example 1. [6]: Consider the delayed NN (4) with:

$$C = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix},\ A = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix},\ B = \begin{bmatrix} 0.88 & 1 \\ 1 & 1 \end{bmatrix},\ k_1^+ = 0.4,\ k_2^+ = 0.8.$$

The corresponding upper bound of h_2 for various h_1 and μ derived by Theorem 1 in this paper and those in [6], [12], and [13] are listed in Table 1. It is clear that our results are better than those in [6], [12], and [13]. For the general activation functions, the maximum allowed delay for $\mu = 0.8$ and $h_1 = 1$ are illustrated in Table 2. But [6], [12], and [13] fail to conclude whether this system is asymptotically stable or not.

Table 1. Comparison of delay-dependent stability criterion of example 2

h_1	Methods	$\mu = 0.8$	$\mu = 0.9$	unknown μ
$h_1 = 0$	[6]	1.2281	0.8636	0.8298
	[12]	1.6831	1.1493	1.0880
	[13]	2.3534	1.6050	1.5103
	Theorem 1	**2.7262**	**1.6330**	**1.5103**
$h_1 = 1$	[13]	3.2575	2.4769	2.3606
	Theorem 1	**3.6207**	**2.4904**	**2.3606**
$h_1 = 2$	[13]	4.2552	3.4769	3.3606
	Theorem 1	**4.5548**	**3.4826**	**3.3606**

Table 2. The maximum allowed delay of generalized activation functions ($\mu = 0.8$)

	$k_1^- = k_2^- = 0.1$	$k_1^- = 0.1, k_2^- = -0.1$	$k_1^- = -0.2, k_2^- = -0.1$
$h_1 = 1$	3.6609	$+\infty$	2.7036

5 Conclusions

In this paper, an augmented Lyapunov-Krasovskii functional is proposed to investigate the stability problem of NNs with time-varying interval delay. The obtained stability condition is expressed in terms of LMIs. It can be shown that

the derived criterion is less conservative than previously existing results through the numerical example.

Acknowledgments. This work is partially supported by the Natural Science Foundation of China (60874030, 60404006, 60574006), and the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (Grant No.07KJB510125, 08KJD520011, 08KJD510008).

References

1. Liu, G.: Nonlinear identification and control: a neural network approach. Springer, New York (2001)
2. Liao, T., Wang, F.: Global stability for cellular neural networks with time delay. IEEE Trans. Neural Netw. 11, 1481–1484 (2000)
3. Cao, J., Wang, L.: Exponential stability and periodic oscillatory solution in BAM networks with delays. IEEE Trans. Neural Netw. 13, 457–463 (2002)
4. Xu, S., Lam, J., Ho, D., Zou, Y.: Improved global robust asymptotic stability criteria for delayed cellular neural networks. IEEE Trans. Syst. Man Cybern. 6, 1317–1321 (2005)
5. Liao, X., Chen, G., Sanchez, E.: Delay-dependent exponential stability analysis of delayed neural networks: An LMI approach. Neural Netw. 15, 855–866 (2002)
6. Xu, S., Lam, J., Ho, D., Zou, Y.: Novel global asymptotic stability criteria for delayed cellular neural networks. IEEE Trans. Circuits Syst. II 52, 349–353 (2005)
7. Ensari, T., Arik, S.: Global stability analysis of neural networks with multiple time varying delays. IEEE Trans. Automatic Control 50, 1781–1785 (2005)
8. He, Y., Wu, M., She, J.: An improved global asymptotic stability criterion for delayed cellular neural networks. IEEE Trans. Neural Netw. 17, 250–252 (2006)
9. Yuan, K., Cao, J., Li, H.: Robust stability of switched Cohen-Grossberg neural networks with mixed time-varying delays. IEEE Trans. Syst. Man Cybern. B. Cybern. 36, 1356–1363 (2006)
10. Liu, Y., Wang, Z., Liu, X.: Global exponential stability of generalized recurrent neural networks with discrete and distributed delays. Neural Netw. 19, 667–675 (2006)
11. Song, Q., Cao, J.: Global robust stability of interval neural networks with multiple time-varying delays. Mathematics and Computers in Simulation 74, 38–46 (2007)
12. He, Y., Liu, G., Rees, D.: New delay-dependent stability criteria for neural networks with time-varying delay. IEEE Trans. Neural Netw. 18, 310–314 (2007)
13. He, Y., Liu, G., Rees, D., Wu, M.: Stability analysis for neural networks with time-varying interval delay. IEEE Trans. Neural Netw. 18, 1850–1854 (2007)
14. Li, T., Guo, L., Sun, C.: Further result on asymptotic stability criterion of neural networks with time-varying delays. Neurocomputing 71, 439–447 (2007)
15. Li, T., Fei, S.: Stability analysis of Cohen-Grossberg neural networks with time-varying and distributed delays. Neurocomputing 71, 823–833 (2008)
16. Li, T., Guo, L., Sun, C., Lin, C.: Further results on delay-dependent stability criteria of neural networks with time-varying delays. IEEE Trans. Neural Netw. 19, 726–730 (2008)
17. Gu, K.: An integral inequality in the stability problem of time-delay systems. In: Proceedings of the 39th IEEE Conference on Decision and Control, Sydney, Australia, pp. 2805–2810 (2000)

Dynamic Analysis of Delayed Fuzzy Cellular Neural Networks with Time-Varying Coefficients

Manchun Tan

Department of Mathematics, Jinan University, Guangzhou 510632, China

Abstract. Some new sufficient conditions are obtained guaranteeing the exponential convergence behavior of each solution of fuzzy cellular neural networks (FCNNs) with variable delays and time-varying coefficients. Compared with some earlier works, the new criteria do not require the Lipschitz continuous condition on activation functions and the differentiability of variable delays.

Keywords: Delayed fuzzy neural network, Exponential convergence, Time-varying coefficients.

1 Introduction

The study on various neural networks is known to be important in theory and application [1]-[7]. Researchers have found that fuzzy cellular neural networks (FCNNs) are useful in image processing, and some results have been reported on stability and periodicity of FCNNs [8]-[17]. Liu and Tang (2004)[12] and Yuan et al.(2006)[15] studied a class of delayed FCNNs using M-matrix theory. Liu et al. (2008) studied such FCNNs by LMI approach [13]. Most of the results are obtained under the Lipschitz continuous condition in the literature [9]-[16]. In addition, some stability criteria, such as those in [12], [13] and [15], require the differentiability of time-varying delays.

Most studies in the literature focused on the dynamics of autonomous neural network model [12]-[16]. However, non-autonomous phenomena often occur in many realistic systems. The parameters of the system usually will change along with time, when we consider a long-term dynamical behaviors of the system. Thus, it is of prime importance and significance to study the dynamic behavior of FCNNs with variable coefficients.

To the best of the author's knowledge, for the non-autonomous FCNNs with variable coefficients and time-varying delays, up till now, the study works are very few. In this paper, we propose some new criteria for FCNNs with variable delays and time-varying coefficients, without the assumption of the Lipschitz continuous condition on activation functions and the differentiability of variable delays.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 433–439, 2009.

2 Problem Statement and Preliminaries

We consider the following fuzzy cellular neural networks (FCNNs) with variable delays and time-varying coefficients:

$$\dot{x}_i(t) = -d_i(t)x_i(t) + \sum_{j=1}^{n} a_{ij}(t)f_j(x_j(t)) + \sum_{j=1}^{n} c_{ij}u_j(t) + I_i(t)$$

$$+ \bigwedge_{j=1}^{n} \alpha_{ij}(t)f_j(x_j(t-\tau_{ij}(t))) + \bigwedge_{j=1}^{n} \varsigma_{ij}u_j(t)$$

$$+ \bigvee_{j=1}^{n} \beta_{ij}(t)f_j(x_j(t-\tau_{ij}(t))) + \bigvee_{j=1}^{n} \delta_{ij}u_j(t), \tag{1}$$

$i = 1, 2, \cdots, n$, where α_{ij}, β_{ij}, ς_{ij} and δ_{ij} are elements of fuzzy feedback MIN template, fuzzy feedback MAX template, fuzzy feed-forward MIN template and fuzzy feed-forward MAX template, respectively. a_{ij} is element of feedback template, and c_{ij} is element of feed-forward template. $\bigwedge$ and $\bigvee$ denote the fuzzy AND and fuzzy OR operation, respectively. x_i, u_i and I_i denote state, input and bias of the ith neurons, respectively. $f_i(\cdot)$ is the activation function. $\tau_{ij}(t)$ is the bounded transmission delay with $0 \le \tau_{ij}(t) \le \tau$, $i = 1, 2, \cdots, n$.

We make the following assumptions:
(H$_1$) For each $i \in \{1, 2, \cdots, n\}$, there exist constants $k_i \ge 0$ such that $|f_i(x)| \le k_i |x|, \quad \forall x \in R$.

(H$_2$) For $i = 1, 2, \cdots, n$, there exist positive constants λ and η, such that for all $t > 0$, there holds

$$\lambda - d_i(t) + \sum_{j=1}^{n} k_j |a_{ij}(t)| + \bigwedge_{j=1}^{n} k_j |\alpha_{ij}(t)| e^{\lambda\tau} + \bigvee_{j=1}^{n} k_j |\beta_{ij}(t)| e^{\lambda\tau} < -\eta < 0.$$

(H$_3$) $|u_i(t)| = O(e^{-\lambda t}), \quad |I_i(t)| = O(e^{-\lambda t})$, $i = 1, 2, \cdots, n$.

For convenience, we use the following notations.

$$X(t) = (x_1(t), x_2(t), \cdots, x_n(t))^T, \quad \|X(t)\| = \max_{1\le i\le n} |x_i(t)|,$$

$$\omega = \max_{1\le i\le n} \{\sup_{t\ge 0} \left| \sum_{j=1}^{n} c_{ij}u_j(t) + I_i(t) + \bigwedge_{j=1}^{n} \varsigma_{ij}u_j(t) + \bigvee_{j=1}^{n} \delta_{ij}u_j(t) \right| \}.$$

The initial conditions associated with system (1) are of the forms

$$x_i(s) = \varphi_i(s), \quad s \in [-\tau, 0], \quad i = 1, 2, \cdots, n, \tag{2}$$

where $\varphi_i(\cdot)$ are real-valued continuous functions defined on $[-\tau, 0]$.

3 Exponential Convergence Analysis

Theorem 1. *Under assumptions* (H_1), (H_2) *and* (H_3), *for every solution* $X(t) = (x_1(t), x_2(t), \cdots, x_n(t))^T$ *of FCNN* (1) *with initial condition* (2), *there holds*

$$|x_i(t)| = O(e^{-\lambda t}), \quad i = 1, 2, \cdots, n.$$

Proof. From assumption (H_3), we know that there exist $\kappa > 0$ and $T > 0$, such that for all $t > T$ and $i \in \{1, 2, \cdots, n\}$

$$\left| \sum_{j=1}^{n} c_{ij} u_j(t) + I_i(t) + \bigwedge_{j=1}^{n} \varsigma_{ij} u_j(t) + \bigvee_{j=1}^{n} \delta_{ij} u_j(t) \right|$$

$$\leq \sum_{j=1}^{n} \left(|c_{ij}| + |\varsigma_{ij}| + |\delta_{ij}| \right) |u_j(t)| + |I_i(t)| \leq \eta \kappa e^{-\lambda t}. \tag{3}$$

Let $X(t) = (x_1(t), x_2(t), \cdots, x_n(t))^T$ be a solution of FCNN (1) with initial condition (2), and i_t be such an index that

$$|x_{i_t}(t)| = \|X(t)\|. \tag{4}$$

In view of (4), the upper right Dini derivative of $e^{\lambda s} |x_{i_s}(s)|$ along the trajectory of system (1) is

$$D^+ \left(e^{\lambda s} |x_{i_s}(s)| \right)\big|_{s=t} = \lambda e^{\lambda t} |x_{i_t}(t)| + e^{\lambda t} \operatorname{sgn}(x_{i_t}(t)) \left\{ -d_{i_t}(t) x_{i_t}(t) + \sum_{j=1}^{n} c_{i_t j} u_j(t) \right.$$

$$+ \sum_{j=1}^{n} a_{i_t j}(t) f_j(x_j(t)) + I_{i_t}(t) + \bigwedge_{j=1}^{n} \alpha_{i_t j}(t) f_j(x_j(t - \tau_{i_t j}(t))) + \bigwedge_{j=1}^{n} \varsigma_{i_t j} u_j(t)$$

$$\left. + \bigvee_{j=1}^{n} \beta_{i_t j}(t) f_j(x_j(t - \tau_{i_t j}(t))) + \bigvee_{j=1}^{n} \delta_{i_t j} u_j(t) \right\}$$

$$\leq e^{\lambda t} \left\{ (\lambda - d_{i_t}(t)) |x_{i_t}(t)| + \sum_{j=1}^{n} k_j |a_{i_t j}(t)| |x_j(t)| \right.$$

$$+ \bigwedge_{j=1}^{n} k_j |\alpha_{i_t j}(t)| |x_j(t - \tau_{i_t j}(t))| + \bigvee_{j=1}^{n} k_j |\beta_{i_t j}(t)| |x_j(t - \tau_{i_t j}(t))|$$

$$\left. + \left| \sum_{j=1}^{n} c_{i_t j} u_j(t) + I_{i_t}(t) + \bigwedge_{j=1}^{n} \varsigma_{i_t j} u_j(t) + \bigvee_{j=1}^{n} \delta_{i_t j} u_j(t) \right| \right\}, \tag{5}$$

where $t \geq T$.

Let

$$M(t) = \max_{s \le t}\{e^{\lambda s} \|X(s)\|\}. \tag{6}$$

It is obvious that $M(t) \ge e^{\lambda t} \|X(t)\|$, and $M(t)$ is non-decreasing.

In the following, we prove that

$$M(t) \le \max\{M(T), \kappa\}, \quad t > T. \tag{7}$$

Assume, by way of contradiction, that (7) does not hold. Then, there exists $\gamma > T$ such that $M(\gamma) > \max\{M(T), \kappa\}$. In view of the continuousness of $M(t)$, there exists a interval $[t_0, t_1] \subseteq [T, \gamma]$ such that $M(t)$ is strictly increasing in $[t_0, t_1]$, and

$$M(t) = e^{\lambda t} \|X(t)\| = e^{\lambda t} |x_{i_t}(t)| \ge \kappa, \quad \forall t \in [t_0, t_1]. \tag{8}$$

From (5), (9), (8) and assumption (H_2), we get

$$\begin{aligned}
D^+(M(s))\big|_{s=t_0} &= D^+\left(e^{\lambda s} |x_{i_s}(s)|\right)\big|_{s=t_0} \le e^{\lambda t_0} \Big\{(\lambda - d_{i_{t_0}}(t_0)) \left|x_{i_{t_0}}(t_0)\right| \\
&+ \sum_{j=1}^{n} k_j \left|a_{i_{t_0}j}(t_0)\right| |x_j(t_0)| + \bigwedge_{j=1}^{n} k_j \left|\alpha_{i_{t_0}j}(t_0)\right| \left|x_j(t_0 - \tau_{i_{t_0}j}(t_0))\right| \\
&+ \bigvee_{j=1}^{n} k_j \left|\beta_{i_{t_0}j}(t_0)\right| \left|x_j(t_0 - \tau_{i_{t_0}j}(t_0))\right| \Big\} + \eta\kappa \\
&= e^{\lambda t_0}(\lambda - d_{i_{t_0}}(t_0)) \left|x_{i_{t_0}}(t_0)\right| + e^{\lambda t_0} \sum_{j=1}^{n} k_j \left|a_{i_{t_0}j}(t_0)\right| |x_j(t_0)| \\
&+ e^{\lambda(t_0 - \tau_{i_{t_0}j}(t_0))} e^{\lambda \tau_{i_{t_0}j}(t_0)} \left\{ \bigwedge_{j=1}^{n} k_j \left|\alpha_{i_{t_0}j}(t_0)\right| \left|x_j(t_0 - \tau_{i_{t_0}j}(t_0))\right| \right. \\
&\left. + \bigvee_{j=1}^{n} k_j \left|\beta_{i_{t_0}j}(t_0)\right| \left|x_j(t_0 - \tau_{i_{t_0}j}(t_0))\right| \right\} + \eta\kappa \\
&\le M(t_0) \left\{ (\lambda - d_{i_{t_0}}(t_0)) + \sum_{j=1}^{n} k_j \left|a_{i_{t_0}j}(t_0)\right| e^{\lambda\tau} + \bigwedge_{j=1}^{n} k_j \left|\alpha_{i_{t_0}j}(t_0)\right| e^{\lambda\tau} \right. \\
&\left. + \bigvee_{j=1}^{n} k_j \left|\beta_{i_{t_0}j}(t_0)\right| e^{\lambda\tau} \right\} + \eta\kappa < -\eta M(t_0) + \eta\kappa \le 0.
\end{aligned} \tag{9}$$

This contradicts that $M(t)$ is non-decreasing. Hence, we obtain $e^{\lambda t} \|X(t)\| \le M(t) \le \max\{M(T), \kappa\}$, for any $t > T$. This completes the proof.

4 Numerical Example

Example 1. Consider the FCNNs (1) with the following parameters:

$$X(t) = [x_1(t), x_2(t)]^T, \quad d_1(t) = 3 - \frac{1}{3}\cos t, \quad d_2(t) = 3 - \frac{1}{3}\sin t,$$

$$a_{11}(t) = \frac{|t|\sin t}{1+4\,|t|^2}, \quad a_{12}(t) = \frac{|t|\cos t}{2(1+2\,|t|)}, \quad a_{21}(t) = \frac{|t|\cos t}{2(1+|t|^2)},$$

$$a_{22}(t) = \frac{|t|\sin t}{2(1+2\,|t|)}, \quad \alpha_{11}(t) = \frac{|t|\cos t^2}{8(1+4\,|t|^2)}, \quad \alpha_{12}(t) = \frac{|t|\sin t^2}{8(1+4\,|t|)},$$

$$\alpha_{21}(t) = \frac{|t|\cos t^2}{4(1+16\,|t|^2)}, \quad \alpha_{22}(t) = \frac{|t|\sin t}{8(1+4\,|t|)}, \quad \beta_{11}(t) = \frac{|t|\cos t}{4(1+16\,|t|^2)},$$

$$\beta_{12}(t) = \frac{|t|\,(\sin t)^2}{16(1+|t|^2)}, \quad \beta_{21}(t) = \frac{|t|\cos t}{8(1+4\,|t|^2)}, \quad \beta_{22}(t) = \frac{|t|\sin t}{16(1+2\,|t|)},$$

$$f_i(x) = x\cos(x^2), \quad I_i(t) = \frac{1}{3}e^{-t}\cos t, \quad u_i(t) = \frac{1}{8}e^{-t}\cos t,$$

$$\tau_{ij}(t) = \frac{1}{4}(1+|\sin t|), \quad c_{ij} = \varsigma_{ij} = \delta_{ij} = 1, \quad i,j = 1,2.$$

From the parameters above, we obtain

$$\tau = 0.5, \quad k_i = 1, \quad |a_{ij}(t)| \le \frac{1}{4}, \quad |\alpha_{ij}(t)| \le \frac{1}{32}, \quad |\beta_{ij}(t)| \le \frac{1}{32}.$$

Letting $\lambda = 1$, $\eta = 1$, we have

$$(\lambda - d_i(t)) + \sum_{j=1}^{n} k_j\,|a_{ij}(t)| + \bigwedge_{j=1}^{n} k_j\,|\alpha_{ij}(t)|\,e^{\lambda\tau} + \bigvee_{j=1}^{n} k_j\,|\beta_{ij}(t)|\,e^{\lambda\tau}$$

$$\le \lambda - d_i(t) + \sum_{j=1}^{2} k_j\frac{1}{4} + (\frac{1}{32} + \frac{1}{32})e^{\lambda\tau} < -1 = -\eta, \quad t > 0.$$

Hence, the parameters in this FCNNs satisfy assumptions (H_1), (H_2) and (H_3). From Theorem 1, we know that the solution of this FCNNs model with initial condition (2) converge exponentially to the zero point $[0,0]^T$.

Remark 1. The commonly used assumption on activation functions is the Lipschitz continuous condition (see [7-14]), i.e., there exist constants $k_i > 0$ such that for $|f_i(\xi_1) - f_i(\xi_2)| \le k_i\,|\xi_1 - \xi_2|, \forall \xi_1, \xi_2 \in R$. It is obvious that functions $f_i(x) = x\cos(x^2)$ in Example 1 don't satisfy the Lipschitz continuous condition. Therefore, the results in [7]-[14] are not applicable to this example.

Remark 2. In [10], [11] and [13], results are obtained under the assumptions that each time-varying delay $\tau_{ij}(t)$ is differentiable. Such limitation is removed in this paper.

5 Conclusion

In this paper, we study a class of delayed fuzzy cellular neural networks (FCNNs) involving variable delays and time-varying coefficients. Some new sufficient conditions that ensure the exponential convergence behavior of each solution of such FCNNs are obtained. We do not assume that the considered model has any equilibriums. The Lipschitz continuous conditions on the activation functions and the differentiability of time-varying delays, which are needed in most other papers, are not required in the new criteria. These obtained conditions are new and they complement previously known results.

Acknowledgements. The research is supported by grants from the National Natural Science Foundation of China (No. 50578064) and the Natural Science Foundation of Guangdong Province in China (No.06025219).

References

1. Zhang, H., Wang, Z., Liu, D.: Global Asymptotic Stability of Recurrent Neural Networks With Multiple Time-Varying Delays. IEEE Transactions on Neural Networks 19, 855–873 (2008)
2. Arik, S.: Global asymptotic stability of hybrid bidirectional associative memory neural networks with time delays. Physics Letters A 351, 85–91 (2006)
3. Cao, J.D., Wang, J.: Global exponential stability and periodicity of recurrent neural networks with time delays. IEEE Transactions on Circuits and Systems I-Regular Papers 52, 920–931 (2005)
4. Zhang, Y.N., Wang, J.: Global exponential stability of recurrent neural networks for synthesizing linear feedback control systems via pole assignment. IEEE Transactions on Neural Networks 13, 633–644 (2002)
5. Zhang, Y.N., Peng, H.F.: Zhang neural network for linear time-varying equation solving and its robotic application. In: 6th International Conference on Machine Learning and Cybernetics, Hong Kong, CHINA, pp. 3543–3548 (2007)
6. Tan, M.C., Zhang, Y.N.: New sufficient conditions for global asymptotic stability of Cohen-Grossberg neural networks with time-varying delays. Nonlinear Analysis: Real World Applications (2008), doi:10.1016/j.nonrwa.2008.03.022
7. Liu, B.: Exponential convergence for a class of delayed cellular neural networks with time-varying coefficients. Physics Letters A 372, 424–428 (2008)
8. Hou, Y.Y., Liao, T.L., Yan, J.J.: Stability analysis of Takagi-Sugeno fuzzy cellular neural networks with time-varying delays. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 37, 720–726 (2007)
9. Chen, L., Zhao, H.: Stability analysis of stochastic fuzzy cellular neural networks with delays. Neurocomputing 72, 436–444 (2008)
10. Huang, T.W.: Exponential stability of fuzzy cellular neural networks with distributed delay. Physics Letters A 351, 48–52 (2006)
11. Huang, T.W.: Exponential stability of delayed fuzzy cellular neural networks with diffusion. Chaos, Solitons and Fractals 31, 658–664 (2007)
12. Liu, Y.Q., Tang, W.S.: Exponential stability of fuzzy cellular neural networks with constant and time-varying delays. Physics Letters A 323, 224–233 (2004)

13. Liu, Z., Zhang, H., Wang, Z.: Novel stability criterions of a new fuzzy cellular neural networks with time-varying delays. Neurocomputing (2008), doi:10.1016/j.neucom.2008.04.001
14. Wang, J., Lu, J.G.: Global exponential stability of fuzzy cellular neural networks with delays and reaction-diffusion terms. Chaos, Solitons & Fractals 38, 878–885 (2008)
15. Yuan, K., Cao, J.D., Deng, J.M.: Exponential stability and periodic solutions of fuzzy cellular neural networks with time-varying delays. Neurocomputing 69, 1619–1627 (2006)
16. Zhang, Q., Xiang, R.: Global asymptotic stability of fuzzy cellular neural networks with time-varying delays. Physics Letters A 372, 3971–3977 (2008)
17. Zhou, Q.H., Wan, L., Sun, J.H.: Exponential stability of reaction-diffusion fuzzy recurrent neural networks with time-varying delays. International Journal of Bifurcation and Chaos 17, 3099–3108 (2007)

Delay-Dependent Exponential Stability of Discrete-Time BAM Neural Networks with Time Varying Delays

Rui Zhang , Zhanshan Wang, Jian Feng, and Yuanwei Jing

School of Information Science and Engineering, Northeastern University, Shenyang 110004, China
zhangrui0325@163.com, wangzhanshan@ise.neu.edu.cn

Abstract. In this paper, the global exponential stability is discussed for discrete-time bidirectional associative memory (BAM) neural networks with time varying delays. By the linear matrix inequality (LMI) technique and discrete Lyapunov functional combined with inequality techniques, a new global exponential stability criterion of the equilibrium points is obtained for this system. The proposed result is less restrictive than those given in the earlier literatures, and easier to check in practice. Remarks are made with other previous works to show the superiority of the obtained results, and the simulation example is used to demonstrate the effectiveness of our result.

Keywords: Discrete-time system, BAM neural network, Discrete lyapunov functional, Global exponential stability, Linear matrix inequality (LMI).

1 Introduction

The bidirectional associative memory (BAM) neural network was first introduced by Kosko in [1] and [2]. It is an extension of the unidirectional autoassociator of Hopfield [3]. This class of neural network has potential application in many fields such as pattern recognition and artificial intelligence. Many applications of BAM neural networks are dependent on the stability of the equilibrium points. So it is significantly important in theory and practice to study the stability of BAM neural networks. And time delays often occur in realizing network, neural processing and signal transmission. The system can be unstable and oscillations due to time-delays. So the asymptotic or exponential stability analysis of BAM neural networks with time delays has received great attention in the past years in [4-9].

It should be pointed out that all of the above references are concerned with continuous-time BAM neural networks. However, discretization in numerical simulations and practical implementations continuous neural networks is necessary. And the dynamics of discrete-time neural networks can be different from those of continuous-time ones. So, it is important in theory and practice to study the stability of discrete-time neural networks. In [10], it was studied exponential stability of discrete-time neural networks by M-matrix method with constant delays. Exponential stability of discrete-time neural networks with time varying delays was studied in [11], but it was

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 440–449, 2009.

unreasonable supposition on the delay functions. But many delay functions do not satisfy such constraints. By using the Lyapunov functional method and LMIs technique, the exponential stability for discrete-time BAM neural networks with variable delays were established in [12] and [13]. But the structures of Lyapunov functional were simple, and no more information on the neurons was used, which directly lead to the conservativeness of their results.

In this paper, we propose a new global exponential stability criterion for the discrete-time BAM neural network with time-varying delays based on the LMIs technique and a new Lyapunov functional with inequality techniques. The proposed result has less conservativeness and is easy to be verified. Simulation example is used to demonstrate the effectiveness of proposed result.

2 System Description and Preliminaries

The discrete-time BAM neural network with time-varying delays can be described as:

$$\begin{cases} u_i(k+1) = a_i u_i(k) + \sum_{j=1}^{m} w_{ij} \overline{f}_j(v_j(k-\tau(k))) + I_i, i = 1, \cdots, n \ , \\ v_j(k+1) = b_j v_j(k) + \sum_{i=1}^{n} v_{ji} \overline{g}_i(u_i(k-\sigma(k))) + J_j, j = 1, \cdots, m \ . \end{cases} \tag{1}$$

where $k = 1, 2, \cdots$ represents the discrete time. $u_i(k)$ and $v_j(k)$ are the states of the i-th neuron from the neural field F_u and the j-th neuron from the neural field F_v at time k, respectively. $a_i, b_j \in (0,1)$ describe the stability of internal neuron processes on the U-layer and the V-layer, respectively. The interconnection weights w_{ij} and v_{ji} are real constants, which are the strengths of connectivity between the i-th neuron from the neural field F_u and the j-th neuron from the neural field F_v. $\overline{f}_j(\cdot)$ and $\overline{g}_i(\cdot)$ are the activation functions of the j-th neuron from the neural field F_v and the i-th neuron from the neural field F_u, respectively. $\tau(k)$ and $\sigma(k)$ represent time varying delays satisfying $\tau_m \le \tau(k) \le \tau_M$, $\sigma_m \le \sigma(k) \le \sigma_M$, where $\tau_m, \tau_M, \sigma_m, \sigma_M$ are non-negative integer values. I_i and J_j denote the external constant inputs from outside the network acting on the i-th neuron from the neural field F_u and the j-th neuron from the neural field F_v, respectively.

The activation functions are assumed to satisfy the following assumptions:

Assumption 1. The activation functions $\overline{f}_j(\cdot)$ and $\overline{g}_i(\cdot)$ ($i = 1, \cdots, n$, $j = 1, \cdots, m$) are bounded. And $\forall \zeta_1, \zeta_2 \in R$, there exist positive constants l_{1j} and l_{2i} such that $0 \le \left| \overline{f}_j(\zeta_1) - \overline{f}_j(\zeta_2) \right| \le l_{1j} \left| \zeta_1 - \zeta_2 \right|$, $0 \le \left| \overline{g}_i(\zeta_1) - \overline{g}_i(\zeta_2) \right| \le l_{2i} \left| \zeta_1 - \zeta_2 \right|$.

Suppose that $u^* = (u_1^*, u_2^*, \cdots, u_n^*)^T$, $v^* = (v_1^*, v_2^*, \cdots, v_m^*)^T$ are equilibrium points of network (1). For convenience, in network (1) we make the following transformation $x_i(k) = u_i(k) - u_i^*$, $y_j(k) = v_j(k) - v_j^*$. Then it becomes:

$$\begin{cases} x_i(k+1) = a_i x_i(k) + \sum_{j=1}^{m} w_{ij} f_j(y_j(k-\tau(k))), \ i = 1, \cdots, n \ , \\ y_j(k+1) = b_j y_j(k) + \sum_{i=1}^{n} v_{ji} g_i(x_i(k-\sigma(k))), \ j = 1, \cdots, m \ . \end{cases} \tag{2}$$

Obviously, the equilibrium points are changed to the origin and the activation functions $f_j(\cdot)$ and $g_i(\cdot)$ satisfy the following condition:

For any $\zeta \in R$, there exist positive scalars l_{ij}, l_{2i} such that

$$0 \le \left| f_j(\zeta) \right| \le l_{1j} \left| \zeta \right|, \ 0 \le \left| g_i(\zeta) \right| \le l_{2i} \left| \zeta \right|, \ j = 1, \cdots, m \ , \ i = 1, \cdots, n \ . \tag{3}$$

The system (2) can be rewritten as vector form

$$\begin{cases} x(k+1) = Ax(k) + Wf(y(k-\tau(k))) \ , \\ y(k+1) = By(k) + Vg(x(k-\sigma(k))) \ , \end{cases} \tag{4}$$

where $x(k) = (x_1(k), \cdots, x_n(k))^T$, $y(k) = (y_1(k), \cdots, y_n(k))^T$, $A = diag\{a_1, \cdots, a_n\}$, $B = diag\{b_1, \cdots, b_m\}$, $W = (w_{ij})_{m \times n}$, $V = (v_{ij})_{n \times m}$, $f_j(y_j(k)) = \overline{f}_j(v_j(k)) - \overline{f}_j(v_j^*)$, $f(y(k)) = [f_1(y_1(k)), \cdots, f_m(y_m(k))]^T$, $g_i(x_i(k)) = \overline{g}_i(u_i(k)) - \overline{g}_i(u_i^*)$, $g(x(k)) = [g_1(x_1(k)), \cdots, g_n(x_n(k))]^T$, $x(s) = \phi(s)$, $\phi(s) = [\phi_1(s), \cdots, \phi_n(s)]^T$, $y(s) = \varphi(s)$, $\varphi(s) = [\varphi_1(s), \cdots, \varphi_m(s)]^T$, $s \in [-\tau, 0]$, $\tau = \max\{\tau_M, \sigma_M\}$.

Definition 1 [13]**.** *The BAM neural network given in (4) is said to be global exponential stable if there exist scalar* $r > 1$, $K > 1$ *such that*

$$\left\| x(k) \right\|^2 + \left\| y(k) \right\|^2 \le K(\sup_{-\sigma_M < s < 0} \left\| \phi(s) \right\|^2 + \sup_{-\tau_M < s < 0} \left\| \varphi(s) \right\|^2) r^{-k}, \ k = 1, 2, \cdots \ .$$

Lemma 1.[14]**:** *For any real vectors* x, y *and any matrix* $Q > 0$ *with appropriate dimensions, it follows that* $2x^T y \le x^T Q x + y^T Q^{-1} y$.

3 Main Result

Theorem 1. *Under Assumption 1, the origin of neural network (4) is global exponential stable, if there exist positive definite matrices* P_1, P_2, Q_1, Q_2, R_1, R_2, Z_1 *and* Z_2, *matrices* F_j, M_j, N_j *and* E_j ($j = 1, \cdots, 4$), *and positive diagonal matrices* D_1, D_2, *such that the following LMI holds:*

$$\begin{bmatrix} \Omega & \sqrt{\sigma_M}F & \sqrt{\sigma_M-\sigma_m}M \\ * & -Z_1 & 0 \\ * & * & -Z_1 \end{bmatrix} < 0, \tag{5}$$

$$\begin{bmatrix} \Upsilon & \sqrt{\tau_M}N & \sqrt{\tau_M-\tau_m}E \\ * & -Z_2 & 0 \\ * & * & -Z_2 \end{bmatrix} < 0, \tag{6}$$

where

$$\Omega = \begin{bmatrix} \Omega_{11} & \Omega_{12} & F_1 - F_3^T - M_1 & -F_4^T + M_1 \\ * & \Omega_{22} & F_2 - M_2 & M_2 \\ * & * & \Omega_{33} & F_4^T + M_3 - M_4^T \\ * & * & * & \Omega_{44} \end{bmatrix} < 0, \tag{7}$$

$$\Omega_{11} = rAP_1A - P_1 + (\sigma_M - \sigma_m + 1)Q_1 + R_1 + r^{\sigma_M}\sigma_M(A-I)^T Z_1(A-I) - F_1 - F_1^T,$$

$$\Omega_{12} = rAP_1W + r^{\sigma_M}\sigma_M(A-I)^T Z_1 W - F_2^T,$$

$$\Omega_{22} = rW^T P_1 W + r^{\sigma_M}\sigma_M W^T Z_1 W - L_1^{-1}D_1L_1^{-1},$$

$$\Omega_{33} = -r^{-\sigma(k)}Q_1 + F_3 + F_3^T + D_2 - M_3 - M_3^T,$$

$$\Omega_{44} = -r^{-\sigma_M}R_1 + M_4 + M_4^T,\ F = [F_1^T\ F_2^T\ F_3^T\ F_4^T]^T,\ M = [M_1^T\ M_2^T\ M_3^T\ M_4^T]^T,$$

$$\Upsilon = \begin{bmatrix} \Upsilon_{11} & \Upsilon_{12} & N_1 - N_3^T - E_1 & -N_4^T + E_1 \\ * & \Upsilon_{22} & N_2 - E_2 & E_2 \\ * & * & \Upsilon_{33} & N_4^T + E_3 - E_4^T \\ * & * & * & \Upsilon_{44} \end{bmatrix} < 0, \tag{8}$$

$$\Upsilon_{11} = rBP_2B - P_P + (\tau_M - \tau_m + 1)Q_2 + R_2 + r^{\tau_M}\tau_M(B-I)^T Z_2(B-I) - N_1 - N_1^T,$$

$$\Upsilon_{12} = rBP_2V + r^{\tau_M}\tau_M(B-I)^T Z_2 V - N_2^T,$$

$$\Upsilon_{22} = rV^T P_2 V + r^{\tau_M}\tau_M V^T Z_2 V - L_2^{-1}D_2L_2^{-1},$$

$$\Upsilon_{33} = -r^{-\tau(k)}Q_2 + N_3 + N_3^T + D_1 - E_3 - E_3^T,$$

$$\Upsilon_{44} = -r^{-\tau_M}R_2 + E_4 + E_4^T,\ N = [N_1^T\ N_2^T\ N_3^T\ N_4^T]^T, E = [E_1^T\ E_2^T\ E_3^T\ E_4^T]^T,$$

I *is an identity matrix with appropriate dimensions.*

Proof. For the real number $r>1$, consider the following discrete Lyapunov-Krasovskii functional of system (4) as follows:

$$V(k)=V_1(k)+V_2(k)+V_3(k)+V_4(k)+V_5(k)+V_6(k)+V_7(k)+V_8(k), \quad (9)$$

where

$$V_1(k)=r^k x^T(k)P_1x(k),$$

$$V_2(k)=\sum_{l=k-\sigma(k)}^{k-1} r^l x^T(l)Q_1x(l)+\sum_{s=-\sigma_M+1}^{-\sigma_m}\sum_{l=k+s}^{k-1} r^l x^T(l)Q_1x(l),$$

$$V_3(k)=\sum_{l=k-\sigma_M}^{k-1} r^l x^T(l)R_1x(l),\ V_4(k)=r^{\sigma_M}\sum_{s=-\sigma_M}^{-1}\sum_{l=k+s}^{k-1} r^l\eta_1^T(l)Z_1\eta_1(l),$$

$$V_5(k)=r^k y^T(k)P_2y(k),$$

$$V_6(k)=\sum_{l=k-\tau(k)}^{k-1} r^l y^T(l)Q_2y(l)+\sum_{s=-\tau_M+1}^{-\tau_m}\sum_{l=k+s}^{k-1} r^l y^T(l)Q_2y(l),$$

$$V_7(k)=\sum_{l=k-\tau_M}^{k-1} r^l y^T(l)R_2y(l),\ V_8(k)=r^{\tau_M}\sum_{s=-\tau_M}^{-1}\sum_{l=k+s}^{k-1} r^l\eta_2^T(l)Z_2\eta_2(l),$$

$$\eta_1(l)=x(l+1)-x(l),\ \eta_2(l)=y(l+1)-y(l).$$

Define $\Delta V(k)=V(k+1)-V(k)$, then along the solution of (4) we have

$$\Delta V_1(k)=r^k[x^T(k)(rAP_1A-P_1)x(k)+2x^T(k)(rAP_1W)f(y(k-\tau(k)))$$
$$+f^T(y(k-\tau(k)))(rW^TP_1W)f(y(k-\tau(k)))], \quad (10)$$

$$\Delta V_2(k)\le r^k(\sigma_M-\sigma_m+1)x^T(k)Q_1x(k)-r^{k-\sigma(k)}x^T(k-\sigma(k))Q_1x(k-\sigma(k)), \quad (11)$$

$$\Delta V_3(k)=r^k x^T(k)R_1x(k)-r^{k-\sigma_M}x^T(k-\sigma_M)R_1x(k-\sigma_M), \quad (12)$$

$$\Delta V_4(k)\le r^k\{r^{\sigma_M}\sigma_M[(A-I)x(k)+Wf(y(k-\tau(k)))]^TZ_1[(A-I)x(k)$$
$$+Wf(y(k-\tau(k)))]-\sum_{l=k-\sigma(k)}^{k-1}\eta_1^T(l)Z_1\eta_1(l)-\sum_{l=k-\sigma_M}^{k-\sigma(k)-1}\eta_1^T(l)Z_1\eta_1(l)\}, \quad (13)$$

By Lemma 1, we have

$$-\sum_{l=k-\sigma(k)}^{k-1}\eta_1^T(l)Z_1\eta_1(l)\le -2\sum_{l=k-\sigma(k)}^{k-1}\eta_1^T(l)F^T\xi(k)+\sum_{l=k-\sigma(k)}^{k-1}\xi^T(k)FZ_1^{-1}F^T\xi(k)$$
$$=-2[x^T(k)-x^T(k-\sigma(k))]F^T\xi(k)+\sigma(k)\xi^T(k)FZ_1^{-1}F^T\xi(k)$$
$$=-\xi^T(k)(\psi_1F^T+F\psi_1^T)\xi(k)+\sigma(k)\xi^T(k)FZ_1^{-1}F^T\xi(k), \quad (14)$$

$$-\sum_{l=k-\sigma_M}^{k-\sigma(k)-1}\eta_1^T(l)Z_1\eta_1(l)\le -2\sum_{l=k-\sigma_M}^{k-\sigma(k)-1}\eta_1^T(l)M^T\xi(k)+\sum_{l=k-\sigma_M}^{k-\sigma(k)-1}\xi^T(k)MZ_1^{-1}M^T\xi(k)$$
$$=-2[x^T(k-\sigma(k))-x^T(k-\sigma_M)]M^T\xi(k)+(\sigma_M-\sigma(k))\xi^T(k)MZ_1^{-1}M^T\xi(k)$$
$$=-\xi^T(k)(\psi_2M^T+M\psi_2^T)\xi(k)+(\sigma_M-\sigma(k))\xi^T(k)MZ_1^{-1}M^T\xi(k), \quad (15)$$

where $F=[F_1^T\ F_2^T\ F_3^T\ F_4^T]^T$, $M=[M_1^T\ M_2^T\ M_3^T\ M_4^T]^T$, $\psi_1=[I\ \ 0\ \ -I\ \ 0]^T$, $\psi_2=[0\ \ 0\ \ I\ \ -I]^T$, $\xi(k)=[x^T(k)\ f^T(y(k-\tau(k)))\ x^T(k-\sigma(k))\ x^T(k-\sigma_M)]^T$. And $\Delta V_5(k)$ - $\Delta V_8(k)$ are similar $\Delta V_1(k)$ - $\Delta V_4(k)$, respectively. So we have

$$\Delta V_5(k)=r^k[y^T(k)(rBP_2B-P_2)y(k)+2y^T(k)(rBP_2V)g(x(k-\sigma(k)))+g^T(x(k-\sigma(k)))(rVP_2V)g(x(k-\sigma(k)))], \quad (16)$$

$$\Delta V_6(k)\le r^k(\tau_M-\tau_m+1)y^T(k)Q_2y(k)-r^{k-\tau(k)}y^T(k-\tau(k))Q_2y(k-\tau(k)), \quad (17)$$

$$\Delta V_7(k)=r^k y^T(k)R_2y(k)-r^{k-\tau_M}y^T(k-\tau_M)R_2y(k-\tau_M), \quad (18)$$

$$\Delta V_8(k)\le r^k\{r^{\tau_M}\tau_M[(B-I)y(k)+Vg(x(k-\sigma(k)))]^TZ_2[(B-I)y(k)+Vg(x(k-\sigma(k)))]-\sum_{l=k-\tau(k)}^{k-1}\eta_2^T(l)Z_2\eta_2(l)-\sum_{l=k-\tau_M}^{k-\tau(k)-1}\eta_2^T(l)Z_2\eta_2(l)\}, \quad (19)$$

$$-\sum_{l=k-\tau(k)}^{k-1}\eta_2^T(l)Z_2\eta_2(l)\le -2\sum_{l=k-\tau(k)}^{k-1}\eta_2^T(l)N^T\zeta(k)+\sum_{l=k-\tau(k)}^{k-1}\zeta^T(k)NZ_2^{-1}N^T\zeta(k)$$
$$=-\zeta^T(k)(\psi_3N^T+N\psi_3^T)\zeta(k)+\tau(k)\zeta^T(k)NZ_2^{-1}N^T\zeta(k), \quad (20)$$

$$-\sum_{l=k-\tau_M}^{k-\tau(k)-1}\eta_2^T(l)Z_2\eta_2(l)\le -2\sum_{l=k-\tau_M}^{k-\tau(k)-1}\eta_2^T(l)E^T\zeta(k)+\sum_{l=k-\tau_M}^{k-\tau(k)-1}\zeta^T(k)EZ_2^{-1}E^T\zeta(k)$$
$$=-\zeta^T(k)(\psi_4E^T+E\psi_4^T)\zeta(k)+(\tau_M-\tau(k))\zeta^T(k)EZ_2^{-1}E^T\zeta(k), \quad (21)$$

where $N=[N_1^T\ N_2^T\ N_3^T\ N_4^T]^T$, $E=[E_1^T\ E_2^T\ E_3^T\ E_4^T]^T$, $\psi_3=[I\ \ 0\ \ -I\ \ 0]^T$, $\psi_4=[0\ \ 0\ \ I\ \ -I]^T$, $\zeta(k)=[y^T(k)\ \ g^T(x(k-\sigma(k)))\ \ y^T(k-\tau(k))\ \ y^T(k-\tau_M)]^T$.

By condition (3), it is well known that there exist non-negative diagonal matrices D_1 and D_2, such that the following inequalities hold

$$[y^T(k-\tau(k))D_1y(k-\tau(k))-f^T(y(k-\tau(k)))L_1^{-1}D_1L_1^{-1}f(y(k-\tau(k)))]\ge 0, \quad (22)$$

$$[x^T(k-\sigma(k))D_2x(k-\sigma(k))-g^T(x(k-\sigma(k)))L_2^{-1}D_2L_2^{-1}g(x(k-\sigma(k)))]\ge 0, \quad (23)$$

where $L_1=diag\{l_{11},l_{12},\cdots,l_{1m}\}$, $L_2=diag\{l_{21},l_{22},\cdots,l_{2n}\}$.

And by (10)-(23), we can obtain that

$$\Delta V(k)\le r^k\{\xi^T(k)[\Omega+\sigma_MFZ_1^{-1}F^T+(\sigma_M-\sigma_m)MZ_1^{-1}M^T]\xi(k)+\zeta^T(k)[\Upsilon+\tau_MNZ_2^{-1}N^T+(\tau_M-\tau_m)EZ_2^{-1}E^T]\zeta(k)\}, \quad (24)$$

where Ω is defined in (5), and Υ is defined in (6).

Applying (5)-(8) and Schur Complement Theorem [15], we have $\Delta V(k)\le 0$. Hence, $V(k)\le V(0)$.

Next, we will establish the global exponential stability for network (4).
From (9), we have

$$V_1(0) = x^T(0)P_1x(0) \le \lambda_M(P_1) \sup_{-\sigma_M<s<0} \|\phi(s)\|^2,$$

$$V_2(0) = \sum_{l=-\sigma(0)}^{-1} r^l x^T(l)Q_1x(l) + \sum_{s=-\sigma_M+1}^{-\sigma_m} \sum_{l=s}^{-1} r^l x^T(l)Q_1x(l)$$

$$\le \lambda_M(Q_1)\frac{1-r^{-\sigma_M}}{r-1} \sup_{-\sigma_M<s<0} \|\phi(s)\|^2 + (\sigma_M-\sigma_m)\lambda_M(Q_1)\frac{1-r^{-\sigma_M}}{r-1} \sup_{-\sigma_M<s<0} \|\phi(s)\|^2,$$

$$V_3(0) = \sum_{l=-\sigma_M}^{-1} r^l x^T(l)R_1x(l) \le \lambda_M(R_1)\frac{1-r^{-\sigma_M}}{r-1} \sup_{-\sigma_M<s<0} \|\phi(s)\|^2,$$

$$V_4(0) = r^{\sigma_M} \sum_{s=-\sigma_M}^{-1} \sum_{l=s}^{-1} r^l \eta_1^T(l)Z_1\eta_1(l) \le r^{\sigma_M}\sigma_M\lambda_M(Z_1) \sum_{l=-\sigma_M}^{-1} r^l \eta_1^T(l)Z_1\eta_1(l)$$

$$\le r^{\sigma_M}\sigma_M\lambda_M(Z_1)\frac{1-r^{-\sigma_M}}{r-1} \sup_{-\sigma_M<s<-1} \|\eta_1(s)\|^2,$$

$$\sup_{-\sigma_M<s<-1} \|\eta_1(s)\|^2 = \sup_{-\sigma_M<s<-1} \|x(s+1)-x(s)\|^2$$

$$\le 2 \sup_{-\sigma_M<s<-1} [\|x(s+1)\|^2 + \|x(s)\|^2] \le 4 \sup_{-\sigma_M<s<0} \|\phi(s)\|^2,$$

$$V_5(0) = y^T(0)P_2y(0) \le \lambda_M(P_2) \sup_{-\tau_M<s<0} \|\varphi(s)\|^2,$$

$$V_6(0) = \sum_{l=-\tau(0)}^{-1} r^l y^T(l)Q_2y(l) + \sum_{s=-\tau_M+1}^{-\tau_m} \sum_{l=s}^{-1} r^l y^T(l)Q_2y(l)$$

$$\le \lambda_M(Q_2)\frac{1-r^{-\tau_M}}{r-1} \sup_{-\tau_M<s<0} \|\varphi(s)\|^2 + (\tau_M-\tau_m)\lambda_M(Q_2)\frac{1-r^{-\tau_M}}{r-1} \sup_{-\tau_M<s<0} \|\varphi(s)\|^2,$$

$$V_7(0) = \sum_{l=-\tau_M}^{-1} r^l y^T(l)R_2y(l) \le \lambda_M(R_2)\frac{1-r^{-\tau_M}}{r-1} \sup_{-\tau_M<s<0} \|\varphi(s)\|^2,$$

$$V_8(0) = r^{\tau_M} \sum_{s=-\tau_M}^{-1} \sum_{l=s}^{-1} r^l \eta_2^T(l)Z_2\eta_2(l) \le r^{\tau_M}\tau_M\lambda_M(Z_2) \sum_{l=-\tau_M}^{-1} r^l \eta_2^T(l)Z_2\eta_2(l)$$

$$\le r^{\tau_M}\tau_M\lambda_M(Z_2)\frac{1-r^{-\tau_M}}{r-1} \sup_{-\tau_M<s<-1} \|\eta_2(s)\|^2,$$

$$\sup_{-\tau_M<s<-1} \|\eta_2(s)\|^2 = \sup_{-\tau_M<s<-1} \|y(s+1)-y(s)\|^2$$

$$\le 2 \sup_{-\sigma_M<s<-1} [\|y(s+1)\|^2 + \|y(s)\|^2] \le 4 \sup_{-\tau_M<s<0} \|\varphi(s)\|^2.$$

So we have

$$V(0) = V_1(0) + V_2(0) + V_3(0) + V_4(0) + V_5(0) + V_6(0) + V_7(0) + V_8(0)$$

$$\leq \rho_1 \sup_{-\sigma_M < s < 0} \left\|\phi(s)\right\|^2 + \rho_2 \sup_{-\tau_M < s < 0} \left\|\varphi(s)\right\|^2, \tag{25}$$

where

$$\rho_1 = \lambda_M(P_1) + [(\sigma_M - \sigma_m + 1)\lambda_M(Q_1) + \lambda_M(R_1) + 4r^{\sigma_M}\sigma_M\lambda_M(Z_1)]\frac{1 - r^{-\sigma_M}}{r - 1},$$

$$\rho_2 = \lambda_M(P_2) + [(\tau_M - \tau_m + 1)\lambda_M(Q_2) + \lambda_M(R_2) + 4r^{\tau_M}\tau_M\lambda_M(Z_2)]\frac{1 - r^{-\tau_M}}{r - 1}.$$

On the other hand,

$$V(k) \geq V_1(k) + V_5(k) \geq r^k x^T(k) P_1 x(k) + r^k y^T(k) P_2 y(k)$$

$$\geq r^k \lambda_m(P_1)\left\|x(k)\right\|^2 + r^k \lambda_m(P_2)\left\|y(k)\right\|^2, \tag{26}$$

From (25) and (26), we can obtain

$$\left\|x(k)\right\|^2 + \left\|y(k)\right\|^2 \leq \frac{\alpha}{\beta}\left(\sup_{-\sigma_M < s < 0}\left\|\phi(s)\right\|^2 + \sup_{-\tau_M < s < 0}\left\|\varphi(s)\right\|^2\right) r^{-k},$$

where $\alpha = \max\{\rho_1, \rho_2\}$, $\beta = \min\{\lambda_m(P_1), \lambda_m(P_2)\}$. By Definition 1, the system (2) is global exponential stable. This ends the proof.

Remark 1. The novel Lyapunov functional $V(k)$ is employed. That is to say, $V_3(k)$ and $V_7(k)$ provide the information on $x^T(k - \sigma_M)$ and $y^T(k - \tau_M)$, $V_4(k)$ and $V_8(k)$ provide the information on neural states change rate. Thus, this Lyapunov functional is more general and less conservative.

Remark 2. In [11], exponential stability of discrete-time neural networks with time varying delays was studied, but it requires the assumption on the delay functions, $1 < k(n+1) < 1 + k(n)$, $1 < l(n+1) < 1 + l(n)$, where $k(n)$, $l(n)$ were variable delays. But many delay functions do not satisfy such constraints. In this paper we do not need this condition.

Remark 3. Note that in our result, the conditions in Theorem 1 are expressed in the form of linear matrix inequality. Therefore, by using the Matlab LMI Toolbox, it is easy to check the feasibility of our result without tuning any parameters. Moreover, the conditions in Theorem 1 consider the signs of the entries in the synaptic connection weights, that is to say, the differences between the neuronal excitatory and the inhibitory effects have been considered, which lead to the less conservativeness of our result.

4 An Illustrative Example

In this section, we will use an example to demonstrate the effectiveness of the present result.

Consider the discrete-time BAM neural network (4) with the following parameters,

$$A=\begin{bmatrix}0.8 & 0\\ 0 & 0.9\end{bmatrix},\ B=\begin{bmatrix}0.4 & 0\\ 0 & 0.7\end{bmatrix},\ W=\begin{bmatrix}-0.1 & 0.01\\ -0.2 & -0.1\end{bmatrix},\ V=\begin{bmatrix}0.3 & -0.1\\ -1.2 & 0.1\end{bmatrix},$$

$1=\tau_m\le\tau(k)\le\tau_M=3$, $1=\sigma_m\le\sigma(k)\le\sigma_M=2$, $L_1=\mathrm{diag}(0.3,\ 0.3)$, $L_2=\mathrm{diag}(0.4,\ 0.4)$.

Pertaining to this example, Theorem 2 in [10], Theorem 1 in [11], Theorem 1 in [12] and Corollary 1 in [13] are not satisfied. Using Theorem 1 of the present paper, we have

$$P_1=\begin{bmatrix}2.3832 & 0.0537\\ 0.0537 & 2.0948\end{bmatrix},P_2=\begin{bmatrix}3.9551 & 0.4618\\ 0.4618 & 2.5228\end{bmatrix},Q_1=\begin{bmatrix}1.4436 & 0.0011\\ 0.0011 & 1.4337\end{bmatrix},$$

$$Q_2=\begin{bmatrix}1.1336 & 0.0006\\ 0.0006 & 1.1276\end{bmatrix},R_1=\begin{bmatrix}1.6649 & 0.0011\\ 0.0011 & 1.6581\end{bmatrix},R_2=\begin{bmatrix}1.7266 & 0.0012\\ 0.0012 & 1.7186\end{bmatrix},$$

$$Z_1=\begin{bmatrix}2.1491 & -0.0988\\ -0.0988 & 2.4393\end{bmatrix},Z_2=\begin{bmatrix}0.5221 & 0.1261\\ 0.1261 & 0.5594\end{bmatrix}.$$

The other parameters are omitted due to the limited space. Therefore, the concerned discrete-time BAM neural network is global exponential stable. The state response curves are depicted in Figure 1.

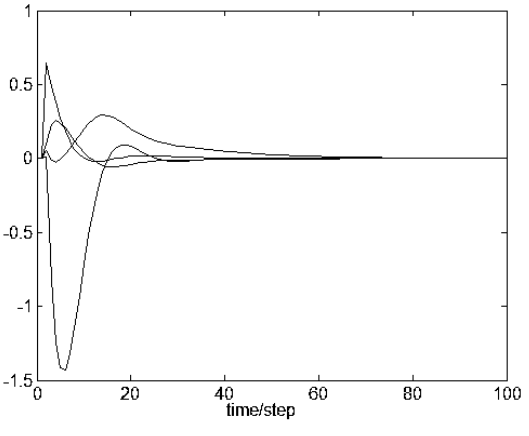


Fig. 1. The state response curves of discrete time BAM neural network

5 Conclusion

The global exponential stability of discrete-time BAM neural network with time varying delays is studied. A delay dependent exponential stability criterion has been obtained by means of a Lyapunov functional and inequality techniques. The proposed result has less conservativeness and is easy to be checked. A simulation example is used to demonstrate the effectiveness of our result.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (Grant Nos. 60534010, 60572070, 60728307, 60774048, 60774093), the Natural Science Foundation of Liaoning Province (Grant No. 20072025), the Postdoctoral Science Foundation of China (Grant No. 20080431150), and the Postdoctoral Foundation of Northeastern University (Grant No. 20080314).

References

1. Kosko, B.: Adaptive Bi-directional Associative Memories. Appl. Opt. 26, 4947–4960 (1987)
2. Kosko, B.: Bi-directional Associative Memories. IEEE Trans. Syst. Man Cybernet 18, 49–60 (1988)
3. Hopfield, J.: Neurons with Graded Response have Collective Computational Properties like Those of Two-state Neurons. Proceedings of the National Academy of Sciences of the United States of America 81, 3088–3092 (1984)
4. Zhang, H.G., Wang, Z.S.: Stability Analysis of BAM Neural Networks with Time-varying Delays. Progress in Natural Science 17, 206–211 (2007)
5. Guan, H.X., Wang, Z.S., Zhang, H.G.: Stability Analysis of Uncertain Bi-directional Associative Memory Neural Networks with Variable Delays. Control Theory & Applications 25, 421–426 (2008)
6. Cui, B., Lou, X.: Global Asymptotic Stability of BAM Neural Networks with Distributed Delays and Reaction–diffusion Terms. Chaos, Solitons & Fractals 27, 1347–1354 (2006)
7. Cao, J., Dong, M.: Exponential Stability of Delayed Bidirectional Associative Memory Networks. Appl. Math. Comput. 135, 105–112 (2003)
8. Guan, H.X., Wang, Z.S., Zhang, H.G.: Robust Stability of BAM Neural Networks with Delays. J. Northeastern University (Natural Science) 28, 1069–1072 (2007)
9. 'Sheng, L., Yang, H.: Novel Global Robust Exponential Stability Criterion for Uncertain BAM Neural Networks with Time-varying Delays. Chaos, Solitons & Fractals (in press)
10. Liang, J., Cao, J.: Exponential Stability of Continuous-time and Discrete-time Bidirectional Associative Memory Networks with Delays. Chaos, Solitons & Fractals 22, 773–785 (2004)
11. Liang, J., Cao, J., Ho, D.W.C.: Discrete-time Bidirectional Associative Memory Neural Networks with Variable Delays. Phys. Lett. A 335, 226–234 (2005)
12. Liu, X., Tang, M., Martin, R., Liu, X.: Discrete-time BAM Neural Networks with Variable Delays. Phys. Lett. A 367, 322–330 (2007)
13. Gao, M., Cui, B.T.: Global Robust Exponential Stability of Discrete-time Interval BAM Neural Networks with Time-varying Delays. Appl. Math. Modell. (in press)
14. Huang, Y.S., Wu, C.W.: A Unifying Proof of Global Asymptotical Stability of Neural Networks with Delay. IEEE Trans. Circ. Syst. II: Express Briefs 52, 181–184 (2005)
15. Boyd, S., Ghaoui, L.E., Feron, E., Balakrishnan, V.: Linear Matrix Inequalities in System and Control Theory. SIAM, Philadelphia (1994)

Memory State Feedback Stabilization for Time-Varying Delayed Neural Networks Systems

Aijun Zhou [1,2], Guang Ren [1], Shubo Liu [2], and Yuan Zhang [1,2]

[1] Dalian Maritime University, Dalian 116026, China
[2] Dalian Naval Academy, Dalian 116018, China

Abstract. In order to improve speed of dynamic response, this paper studied the memory state feedback stabilization for time-varying delayed neural networks systems. By using the second method of Lyapunov, the state feedback controller is given to ensure that the system is asymptotically stable. The related theories are expressed in terms of linear matrix inequalities (LMIs). An example is given to illustrate the effectiveness of the proposed criterion. The simulation results show that this method has excellent control effect.

1 Introduction

Neural networks have received considerable attention due to their extensive applications in various signal processing problems such as optimization, fixed-point computations, and other areas in the past decades[1]. Because the applications of neural networks rely heavily on the dynamical behaviors of the networks, the stability of delayed neural networks have been investigated by many researchers and presented a number of useful and interesting results[2,3]. LMI-based technique is a powerful tool to derive various stability problems for neural networks with time delays[4]. It should be noted that the LMI condition can be checked very easily by using the LMI toolbox of Matlab[5].

The neural networks systems with time-varying delay are special time-delay systems. The problem related to the stability analysis of delayed neural networks systems is the stabilization. The stabilization is to find a feedback controller to ensure the closed system stable. The feedback adopted contains the state feedback and the output feedback. By comparison, the state feedback, that is the complete feedback of the system characteristic is superior to the output feedback in characteristic.

The design of the feedback control without memory based the second Lyapunov method is simple to analyse the robust stability of the closed system. But presently there are less researches into memory feedback controller. In this paper, a memory feedback state controller for time-varying delayed neural networks systems is designed under this background.

2 System Description and Main Results

We consider the following delayed neural networks:

$$\dot{x}(t) = -Ax(t) + W_1 f(x(t)) + W_2 f(x(t-\tau(t))) + Du(t) \tag{1a}$$

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 450–454, 2009.

The initial state is:

$$x(t_0+\theta)=\phi(\theta)\ \ \forall\theta\in[-\tau_0,0] \tag{1b}$$

Where $x(t)=[x_1(t),x_2(t),\ldots,x_n(t)]^{\mathrm{T}}$ is the neuron state vector, $f(x(t))=[f_1(x_1(t)),f_2(x_2(t)),\ldots,f_n(x_n(t))]^{\mathrm{T}}=[f(x_1(t)),f(x_2(t)),\ldots,f(x_n(t))]^{\mathrm{T}}$, $f_i(0)=0$, Note that functions $f_i(\cdot)$ here satisfies the following:

$$0\le\frac{f_i(\xi_1)-f_i(\xi_2)}{\xi_1-\xi_2}\le\sigma_i \qquad i=1,2,\ldots,n$$

$u(t)\in R^n$ represents the control input matrix, $\phi(\cdot)$ is the continuous differentiable function in given interval $[-\tau_0,0]$. The time delay $\tau(t)$, is time-varying differentiable function and satisfies the following condition:

$$0\le\tau(t)\le\tau_0;\dot{\tau}(t)\le d$$

In the following, we will design a memory state feedback controller $u(t)=Kx(t)+Hx(t-\tau(t))$ such that the closed system is asymptotically stable.

Theorem 1: Assume the time-varying delay, the origin of the delayed neural networks in (1) is asymptotically stable under the controller $u(t)=MX^{-1}x(t)+NX^{-1}x(t-\tau(t))$, if there exist matrices $\overline{P}>0$, $\overline{Q}>0$, $\overline{R}>0$, $\overline{Z}>0$, diagonal matrix, $\overline{T}>0$, $\overline{S}>0$, X, $\overline{N}_1$, $\overline{N}_2$, M ,such that the following LMI below holds:

$$M=\begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} & M_{15} & M_{16} \\ * & M_{22} & M_{23} & 0 & M_{25} & M_{26} \\ * & * & M_{33} & M_{34} & 0 & M_{36} \\ * & * & * & M_{44} & 0 & 0 \\ * & * & * & * & M_{55} & 0 \\ * & * & * & * & * & M_{66} \end{bmatrix}<0 \tag{2}$$

Where: $M_{11}=-AX-X^{\mathrm{T}}A^{\mathrm{T}}+\overline{N}_1+\overline{N}_1^{\mathrm{T}}+\overline{Q}+DM+M^{\mathrm{T}}D^{\mathrm{T}}$; $M_{12}=\overline{P}-X+M^{\mathrm{T}}D^{\mathrm{T}}$; $M_{13}=-\overline{N}_1+\overline{N}_2^{\mathrm{T}}+DN$; $M_{14}=-\tau_0\overline{N}_1$; $M_{15}=W_1X+\sum\overline{T}$; $M_{16}=W_2X$; $M_{22}=-X^{\mathrm{T}}-X+\tau_0\overline{Z}$; $M_{23}=DN$; $M_{25}=W_1X$; $M_{26}=W_2X$; $M_{33}=-(1-d)\overline{Q}-(\overline{N}_2+\overline{N}_2^{\mathrm{T}})$; $M_{34}=-\tau_0\overline{N}_2$; $M_{36}=\sum\overline{S}$; $M_{44}=-\tau_0\overline{Z}$; $M_{55}=\overline{R}-2\overline{T}$; $M_{66}=-(1-d)\overline{R}-2\overline{S}$; $\Sigma=diag(\sigma_1,\sigma_2,\ldots,\sigma_n)$; $T=diag(t_1,t_2,\ldots,t_n)$; $S=diag(s_1,s_2,\ldots,s_n)$; $\overline{P}=X^{\mathrm{T}}PX$, $\overline{R}=X^{\mathrm{T}}RX$, $\overline{Q}=X^{\mathrm{T}}QX$, $\overline{Z}=X^{\mathrm{T}}ZX$, $\overline{T}=X^{\mathrm{T}}TX$, $\overline{S}=X^{\mathrm{T}}SX$, $\overline{N}_1=X^{\mathrm{T}}N_1X$, $\overline{N}_2=X^{\mathrm{T}}N_2X$

Proof: Firstly, rewrite system (1) in the following equivalent descriptor system

$$\dot{x}(t)=y(t) \tag{3a}$$

$$y(t)=-Ax(t)+W_1f(x(t))+W_2f(x(t-\tau(t)))+Du(t) \tag{3b}$$

Choose a Lyapunov-Krasovskii functional as follows, thinking of the closed system (3) under the controller $u(t) = Kx(t) + Hx(t-\tau(t))$:

$$V(x(t)) = V_1 + V_2 + V_3 + V_4 \tag{4}$$

$$V_1 = x^{\mathbf{T}}(t)Px(t),\ V_2 = \int_{t-\tau(t)}^{t} x^{\mathbf{T}}(s)Qx(s)ds,\ V_3 = \int_{-\tau}^{0}\int_{t+\theta}^{t} \dot{\mathrm{x}}^{\mathbf{T}}(\mathrm{s})Z\dot{\mathrm{x}}(\mathrm{s})dsd\theta,$$

$$V_4 = \int_{t-\tau(t)}^{t} f^{\mathbf{T}}(x(\mu))Rf((x(\mu))d\mu$$

Now we consider the derivation of V along the trajectories of (3).For $\forall P_1, N_1, N_2$,

$$\begin{aligned}
\dot{V}_1 &= 2x^{\mathbf{T}}(t)P\dot{x}(t) \\
&= 2x^{\mathbf{T}}(t)Py(t) + 2[x^{\mathbf{T}}(t)N_1 + x^{\mathbf{T}}(t-\tau(t))N_2][x(t) - x(t-\tau(t)) - \int_{t-\tau(t)}^{t} y(s)ds] \\
&\quad +2[x^{\mathbf{T}}(t)P_1^{\mathbf{T}} + y^{\mathbf{T}}(t)P_1^{\mathbf{T}}][-y(t) - Ax(t) + W_1 f(x(t)) + W_2 f(x(t-\tau(t))) + Du(t)] \\
\dot{V}_2 &= x^{\mathbf{T}}(t)Qx(t) - (1-\dot{\tau}(t))x^{\mathbf{T}}(t-\tau(t))Qx(t-\tau(t)) \\
&\le x^{\mathbf{T}}(t)Qx(t) - (1-d)x^{\mathbf{T}}(t-\tau(t))Qx(t-\tau(t)) \\
\dot{V}_3 &= \tau(t)\, y^{\mathbf{T}}(t)Zy(t) - \int_{t-\tau(t)}^{t} y^{\mathbf{T}}(\theta)Zy(\theta)d\theta \le \tau_0 y^{\mathbf{T}}(t)Zy(t) - \int_{t-\tau(t)}^{t} y^{\mathbf{T}}(\theta)Zy(\theta)d\theta \\
&\le \tau_0 y^{\mathbf{T}}(t)Zy(t) - (\frac{1}{\tau_0}\int_{t-\tau(t)}^{t} y^{\mathbf{T}}(\theta)Zy(\theta)d\theta)^{\mathbf{T}}\tau_0 Z(\frac{1}{\tau_0}\int_{t-\tau(t)}^{t} y^{\mathbf{T}}(\theta)Zy(\theta)d\theta) \\
\dot{V}_4 &= f^{\mathbf{T}}(x(t))Rf(x(t)) - (1-\dot{\tau}(t))f^{\mathbf{T}}(x(t-\tau(t)))Rf(x(t-\tau(t))) \\
&\le f^{\mathbf{T}}(x(t))Rf(x(t)) - (1-d)f^{\mathbf{T}}(x(t-\tau(t)))Rf(x(t-\tau(t)))
\end{aligned}$$

Then, we have that

$$\begin{aligned}
\dot{V}(x(t)) &= \dot{V}_1 + \dot{V}_2 + \dot{V}_3 + \dot{V}_4 \\
&\le \dot{V}_1 + \dot{V}_2 + \dot{V}_3 + \dot{V}_4 - 2f^{\mathbf{T}}(x(t))Tf(x(t)) + 2x^{\mathbf{T}}(t)\sum Tf(x(t)) \\
&\quad - 2f^{\mathbf{T}}(x(t-\tau(t)))Sf(x(t-\tau(t))) + 2x^{\mathbf{T}}(t-\tau(t))\sum Sf(x(t-\tau(t))) \\
&\le Z^{\mathbf{T}}M_0 Z
\end{aligned} \tag{5}$$

$$Z = [x^{\mathbf{T}}(t)\ ,\ y^{\mathbf{T}}(t)\ ,\ x^{\mathbf{T}}(t-\tau(t)), (\frac{1}{\tau_0}\int_{t-\tau(t)}^{t} \dot{x}(\theta)d\theta)^{\mathbf{T}}, f^{\mathbf{T}}(x(t))\ \ f^{\mathbf{T}}(x(t-\tau(t)))]^{\mathbf{T}}$$

Remark 1. The shortcoming of the original design method for controller is to know the delay beforehand so that it limits the application in many practical engineering systems. The time delay, which is researched in this paper, can be unknown, time-varying, dependent on the state before.

3 Simulation and Results

In this section, one example is given to show the effectiveness of the theorem presented in this paper. The LMI is solved by the LMI-Toolbox in Matlab.

Example 1: Consider a delayed neural networks with time-varying delay in (1) with parameters as:

$$\dot{x}(t) = -Ax(t) + W_1 f(x(t)) + W_2 f(x(t-\tau(t))) + Du(t) \tag{6}$$

Where $A=\begin{bmatrix}2 & 0\\ 0 & 2\end{bmatrix}, W_1=\begin{bmatrix}1 & 1\\ -1 & -1\end{bmatrix}, W_2=\begin{bmatrix}0.8 & 1\\ 1 & 1\end{bmatrix}, D=\begin{bmatrix}1\\ 1\end{bmatrix}\ \sigma_1=0.9, \sigma_2=0.3$

In this example, we have the activation function as $f_1(x)=f_2(x)=[|x+1|-|x-1|]/4$. And when $d=1$, $\tau_0\le 3.422$, the LMI in (2) holds. When $\tau_0=3.422$, we solve the LMI in (2) and obtain:

$$\bar{P}=\begin{bmatrix}186.3922 & 173.1872\\ 173.1872 & 224.4770\end{bmatrix}, \bar{Q}=\begin{bmatrix}80.6020 & 70.2552\\ 70.2552 & 62.1601\end{bmatrix}, \bar{S}=\begin{bmatrix}0.1049 & 0\\ 0 & 46.3884\end{bmatrix}$$

$$X=\begin{bmatrix}16.3562 & -2.5500\\ -14.7262 & 27.4044\end{bmatrix},\quad M=\begin{bmatrix}-178.8313 & -173.7971\end{bmatrix},$$

$$\bar{Z}=\begin{bmatrix}0.6967 & -0.4023\\ -0.4023 & 7.6299\end{bmatrix}, \bar{N}_1=\begin{bmatrix}0.0389 & 6.3057\\ 0.3040 & 5.3385\end{bmatrix}, \bar{N}_2=\begin{bmatrix}0.1998 & -0.0979\\ -0.1138 & 2.1757\end{bmatrix}$$

$$\bar{T}=\begin{bmatrix}10.3388 & 0\\ 0 & 235.3151\end{bmatrix},\quad \bar{R}=\begin{bmatrix}2.1825 & -0.0954\\ -0.0954 & 38.8218\end{bmatrix},$$

$$K=\begin{bmatrix}-18.1654 & -8.0323\end{bmatrix}, H=\begin{bmatrix}-0.1366 & -0.1501\end{bmatrix},\quad N=\begin{bmatrix}-0.0231 & -3.7652\end{bmatrix}$$

We can see that the roots satisfy the conditions provided in theorem 1.

For a given initial state $x=[-1,1]^{\mathrm{T}}$, when emulating this system, we can obtain the simulation results as figure 1 and figure 2, where figure 1 shows that the state response before control, and figure 2 shows that the state response after control. And from the simulation curve we can conclude that the dynamical response speed is obviously rapid after control and easy to compute. The memory feedback controller is:

$$u(t)=\begin{bmatrix}-18.1654 & -8.0323\end{bmatrix}x(t)+\begin{bmatrix}-0.1366 & -0.1501\end{bmatrix}x(t-\tau_0).$$

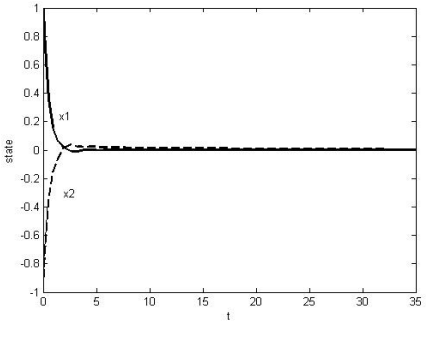


Fig. 1. the state curve before control

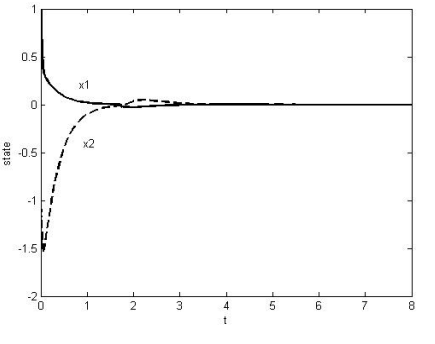


Fig. 2. the state curve after control

4 Conclusion

We can see that the system is also asymptotically stable and the control effect is improved after control from the simulation result.

The author feel that there is a lot of work to be done in this field and hope that we can do further research in the following aspects:

(1) The conservation of the conclusion. The Lyapunov stable theory which we use as the main tool to study the stability and the stabilization of the delayed neural networks is a necessary and sufficient condition, but the conclusion we obtain is a sufficient condition after realized using the linear matrix inequality. In addition, we bring a lot of conservation when quoting some lemmas during the course of proof.

(2) The constitution of the Lyapunov function. From the proof, we can see that the constitution of the Lyapunov function, especially in the delayed neural networks, has much influence on the conclusion. Unfortunately up to now, the constitution of the Lyapunov function is usually established by experience. How to establish a Lyapunov function with less conservation is another direction we should turn toward.

(3) The application of the stabilization. Presently, there is plenty of study on the stability analysis, but less on the stabilization and application in the practice.

References

1. Borkar, V.S., Soumyanatha, K.: An Analog Scheme for Fixed Point Computation-Part I: Theory. IEEE Trans. Circuits and Systems I 144, 351–355 (1997)
2. Arik, S.: Stability Analysis of Delayed Neural Networks. IEEE Trans. Circuits and Systems I 47(7), 1089–1092 (2000)
3. Liao, T.L., Wang, F.C.: Global Stability Condition for Cellular Neural Networks with Delay. Electron. Lett. 35(16), 1347–1349 (1999)
4. Liao, X., Chen, G., Sanchez, E.N.: LMI-based Approach for Asymptotically Stability Analysis of Delayed Neural Networks. IEEE Trans.Circuits ans systems I 49(7), 1033–1039 (2002)
5. Gahinet, P., Nemirovskii, A., Laub, A.J., Chilali, M.: LMI Control Toolbox, Math Works, Naticvk, Massachusetts (1995)
6. Zhang, H.B., Li, H.G., Liao, X.F.: A Note on The Robust Stability of Neural Networks With Time Delays. Chaos, Solitons and Fractals 25, 357–360 (2005)
7. Xu, S.Y., James, L., Daniel, W.C., et al.: Delay-dependent Exponential Stability for a Class of Neural Networks with Time Delays. Journal of computational and applied mathematics 183(1), 16–28 (2005)

Global Stability of Neural Networks with Delays and Impulses*

Fengjian Yang[1], Chaolong Zhang[1], Dongqing Wu[1,**], Jianfu Yang[1], Yanshan Zeng[1], Lishi Liang[2], and Qun Hong[3]

[1] Computational Science Department of Zhongkai University of Agriculture and Engineering, Guangzhou, 510225
wwddqq@126.com
[2] The 5th Middle School of Guangzhou City, Guangzhou, 510220
[3] The Haizhu District Chigang Primary School of Guangzhou City, Guangzhou, 510310

Abstract. This paper is concerned with a class of neural networks with delays and impulses. Some sufficient conditions are obtained for the existence and globally exponential stability of a unique equilibrium solution without assuming the activation function to be bounded, monotonic or differentiable. An example is given to demonstrate the effectiveness of the obtained results.

Keywords: impulse; neural network; delay; stability.

1 Introduction

The dynamics of Hopfield neural networks aroused extensive researches by numerous scholars duo to its applicability in solving image processing, signal processing and pattern recognition problems. In 1988, Chua and Yang [1-2] put forward the following model of cellular neural networks with time delay:

$$\frac{dx_i(t)}{dt} = -x_i(t) + \sum_{j=1}^{n} a_{ij} g_j(x_j(t-\tau)) + I_i, \quad i = 1, 2, \cdots n,$$

and presented the electric circuits to utilize the model. Some authors discussed dynamics property of this kind of neural networks in [3-7]. Cao etc [8-10] discussed asymptotic stability of the more general neural network of DCNN type:

$$\frac{dx_i(t)}{dt} = -x_i(t) + \sum_{j=1}^{n} a_{ij} g_j(x_j(t)) + \sum_{j=1}^{n} b_{ij} g_j(x_j(t-\tau_j)) + I_i, \quad i = 1, 2, \cdots, n.$$

Most widely used neural networks is neither purely continuous-time nor purely discrete-time ones, these are called impulsive neural networks. In this paper, we consider the dynamics of the following neural networks with delays and impulses:

* This work is supported by Science and technology plan foundation of Guangzhou under Grant 2006j1-C0341.

** Corresponding author.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 455–462, 2009.

$$\begin{cases} \dfrac{dx_j(t)}{dt} = -a_j x_j(t) + \sum\limits_{i=1}^{n} b_{ij} g_i(x_i(t)) + \sum\limits_{i=1}^{n} c_{ij} g_i(x_i(t-\tau_i)) + I_j, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad j = 1,2,\cdots,n, \\ \Delta x_j(t_k) = I_k(x_j(t_k)), \qquad j = 1,2,\cdots,n; k \in N = \{1,2,\cdots\}. \end{cases} \tag{1}$$

In which, n corresponds to the number of units in the neural networks; $x_i(t)$ corresponds to the state variable of the i-th unit at time t; $a_j > 0$, b_{ij}, c_{ij}, $I_j \in R$, b_{ij} denotes the strength of the i-th unit on the j-th unit at time t, c_{ij} denotes the strength of the i-th unit on the j-th unit at time $t-\tau_i$; I_j denotes the external bias on the j-th unit; the delays τ_i satisfy $0 \le \tau_i \le \tau (i = 1,2,\cdots,n, \tau$ is a constant); $g_i(x_i(t))$ denotes the output of the i-th unit at time t. $\Delta x_j(t_k) = x_j(t_k + 0) - x_j(t_k - 0)$ are the impulses at moments t_k and $t_1 < t_2 < \cdots$ is a strictly increasing sequence such that $\lim\limits_{k\to\infty} t_k = +\infty$.

Equation (1) indicates that neural cell is also subjected to outputs of other neural cell at past time. The system satisfies initial condition as follow:

$$x_i(s) = \varphi_i(s), s \in [t_0 - \tau, t_0], \quad i = 1,2,\cdots n, \tag{2}$$

where $\varphi_i(\cdot) \in C([t_0 - \tau, t_0], R), i = 1,2,\cdots,n$.

Throughout this paper, we assume that there exist constant L_j (j = 1, 2, ⋯, n) such that

$$\left| g_j(s_1) - g_j(s_2) \right| \le L_j \left| s_1 - s_2 \right|, \quad s_1, s_2 \in R. \tag{3}$$

As usual, vector function

$$x(t) = (x_1(t), x_2(t), \cdots, x_n(t))^T$$

is said to be a solution of system (1), if $x(t)$ satisfies (1) and (2). At the points of discontinuity t_k of the solution, we denote

$$x_i(t_k) \equiv x_i(t_k - 0), i = 1,2,\cdots,n; k \in N, \tag{4}$$

$$x_i'(t_k) \equiv x_i'(t_k - 0), i = 1,2,\cdots,n; k \in N. \tag{5}$$

For $w = (w_1, w_2, \cdots, w_n)^T \in R^n$. we define norm by

$$\|w\| = \sum_{l=1}^{n} |w_l|. \tag{6}$$

2 Existence of Equilibria

Definition 1. An equilibrium solution of (1) is a constant vector $(x_1^*, x_2^*, \cdots, x_n^*)^T$ which satisfies the system

$$a_j x_j^* = \sum_{i=1}^{n} (b_{ij} + c_{ij}) g_i(x_i^*) + I_j, \quad j = 1, 2, \cdots, n \tag{7}$$

when the impulsive jumps $I_k(\bullet)$ as assumed to satisfy $I_k(x_i^*) = 0$, $i = 1, 2, \cdots, n$; $k \in Z^+$, where Z^+ denotes the set of all positive integers.

Theorem 1. Suppose that

$$a_i > L_i \sum_{j=1}^{n} \left(\left| b_{ij} + c_{ij} \right| \right), \quad i = 1, 2, \cdots, n. \tag{8}$$

Then there exists a unique solution of the system (7).

Proof. It follows from (8) that

$$\alpha := \max_{1 \le i \le n} \left[\frac{L_i}{a_i} \sum_{j=1}^{n} \left(\left| b_{ij} + c_{ij} \right| \right) \right] < 1.$$

Let $a_i x_i^* = u_i^*, i = 1, 2, \cdots, n$ in (7), we have

$$u_j^* = \sum_{i=1}^{n} (b_{ij} + c_{ij}) g_i\left(\frac{u_i^*}{a_i}\right) + I_j, \quad j = 1, 2, \cdots, n. \tag{9}$$

Then the result of Theorem 1 is equivalent to that system (9) has a unique solution. Consider a mapping $\Phi : R^n \to R^n$ defined by

$$\Phi(u_1, u_2, \cdots, u_n) = \begin{pmatrix} \sum_{i=1}^{n} (b_{i1} + c_{i1}) g_i\left(\frac{u_i}{a_i}\right) + I_1 \\ \sum_{i=1}^{n} (b_{i2} + c_{i2}) g_i\left(\frac{u_i}{a_i}\right) + I_2 \\ \vdots \\ \sum_{i=1}^{n} (b_{in} + c_{in}) g_i\left(\frac{u_i}{a_i}\right) + I_n \end{pmatrix}.$$

For $\bar{u} = (\bar{u}_1, \bar{u}_2, \cdots, \bar{u}_n)^T, u = (u_1, u_2, \cdots, u_n)^T \in R^n$, we have

$$\begin{aligned} &\left\| \Phi(\bar{u}_1, \cdots, \bar{u}_n) - \Phi(u_1, \cdots, u_n) \right\| \\ &= \sum_{j=1}^{n} \left| \sum_{i=1}^{n} (b_{ij} + c_{ij}) \left[g_i\left(\frac{\bar{u}_i}{a_i}\right) - g_i\left(\frac{u_i}{a_i}\right) \right] \right| \end{aligned}$$

$$\leq \sum_{j=1}^{n}\sum_{i=1}^{n}\left(\left|b_{ij}+c_{ij}\right|\right)L_i\left|\frac{\overline{u}_i-u_i}{a_i}\right|$$

$$\leq \max_{1\leq i\leq n}\left(\frac{L_i}{a_i}\sum_{j=1}^{n}\left|b_{ij}+c_{ij}\right|\right)\cdot\sum_{i=1}^{n}\left|\overline{u}_i-u_i\right|$$

$$=\alpha\left\|u-\overline{u}\right\|.$$

Since $0<\alpha<1$, then $\Phi: R^n \to R^n$ is a contraction mapping on R^n. Hence by the contraction mapping principle, there exist a unique fixed point of the map $\Phi: R^n \to R^n$ which is a solution of the system (9). This completes the proof.

3 Exponential Stability

Definition 2. The equilibrium point x^* of system (1) is said to be globally exponentially stable, if there exist constants $\beta > 0$ and $\delta > 0$ such that

$$\left\|x(t)-x^*\right\| \leq \beta e^{-\delta(t-t_0)}, \quad t \geq t_0, \tag{10}$$

where $x(t)$ is any solution of system (1).

Lemma 1 [11]**.** Suppose that $x(t)$ is a derivable function of single variable, then

$$D^+\left|x(t)\right| = \begin{cases} x'(t)\operatorname{sgn}x(t), & if \quad x(t)\neq 0, or \ x(t)=x'(t)=0, \\ x'(t), & if \quad x(t)=0 \ \text{ and } \ x'(t)>0, \\ -x'(t), & if \quad x(t)=0 \ \text{ and } \ x'(t)<0, \end{cases}$$

where $D^+x(t)$ is the right upper Dini derivative of $x(t)$.

Theorem 2. Assume that conditions (3) and

$$a_i > L_i\sum_{j=1}^{n}\left(\left|b_{ij}\right|+\left|c_{ij}\right|\right), \quad i=1,2,\cdots,n. \tag{11}$$

$$I_k(x_j(t_k)) = -\gamma_{jk}(x_j(t_k)-x_j^*),\ 0<\gamma_{jk}<2,\ j=1,2,\cdots,n,k\in Z^+ \tag{12}$$

Hold, then system (1) has a unique equilibrium solution $x^* = (x_1^*, x_2^*, \cdots, x_n^*)^T$, moreover, this solution is globally exponentially stable.

Proof. In view of condition (12) we have $I_k(x_j^*) = 0$, noting condition (11), from Theorem 1 we know that system (1) has a unique equilibrium point. Let $x(t) = (x_1(t), x_2(t), \cdots, x_n(t))^T$ be an arbitrary solution of system (1). Lemma 1 yields

$$D^+\left|x_j-x_j^*\right| \le -a_j\left|x_j(t)-x_j^*\right| + \sum_{i=1}^{n}\left|b_{ij}\right| L_i\left|x_i(t)-x_i^*\right| + \sum_{i=1}^{n}\left|c_{ij}\right| L_i\left|x_i(t-\tau_i)-x_i^*\right|, \tag{13}$$

for $t>0, t\neq t_k$, $j=1,2,\cdots,n, k\in Z^+$, and

$$\begin{aligned} & x_j(t_k+0)-x_j^* \\ &= x_j(t_k)+I_k(x_j(t_k))-x_j^* \\ &= (1-\gamma_{jk})(x_j(t_k)-x_j^*),\ j=1,2,\cdots,n, k\in Z^+. \end{aligned}$$

Hence

$$\begin{aligned} & \left|x_j(t_k+0)-x_j^*\right| \\ &= \left|1-\gamma_{jk}\right|\left|x_j(t_k)-x_j^*\right| \\ &\le \left|x_j(t_k)-x_j^*\right|, \qquad j=1,2,\cdots,n, k\in Z^+. \end{aligned}$$

Let F_i be defined by

$$F_i(\xi_i)=a_i-\xi_i-L_i\sum_{j=1}^{n}\left(\left|b_{ij}\right|+\left|c_{ij}\right|e^{\tau_i\xi_i}\right), i=1,2,\cdots,n, \tag{14}$$

where $\xi_i\in[0,\infty)$. Condition (11) yields $F_i(0)>0$, $i=1,2,\cdots,n$. Since $F_i(\cdot)$ are continuous on $[0,\infty)$ and $F_i(\xi_i)\to-\infty$ as $\xi_i\to+\infty$, there must exist $\eta>0$ and a number $k\in Z^+$ such that $F_k(\eta)=0$ but $F_i(t)>0$, $0\le t<\eta$, $i=1,2,\cdots,n$. Let

$$u_j(t)=e^{\eta t}\left|x_j(t)-x_j^*\right|,\ t\in[-\tau,\infty), j=1,2,\cdots,n. \tag{15}$$

It follows from (13) and (15) that

$$D^+u_j(t)\le(\eta-a_j)u_j(t)+\sum_{i=1}^{n}\left|b_{ij}\right|L_iu_i(t) + \sum_{i=1}^{n}\left|c_{ij}\right|L_ie^{\eta\tau_i}u_i(t-\tau_i),\ j=1,2,\cdots,n, \tag{16}$$

for $t>0, t\neq t_k, k\in Z^+$. Also

$$u_j(t_k+0)=\left|1-\gamma_{jk}\right|u_j(t_k)\le u_j(t_k),\ j=1,2,\cdots,n, k\in Z^+. \tag{17}$$

Consider a Lyapunov function defined by

$$V(t)=\sum_{i=1}^{n}\left(u_i(t)+\sum_{j=1}^{n}\left|c_{ij}\right|L_ie^{\eta\tau_i}\int_{t-\tau_i}^{t}u_i(s)ds\right). \tag{18}$$

It is clear that $V(t)>0$ for $t\geq 0$. Calculating the right upper Dini derivative of $V(t)$ along the solution of derivatives of (16), noting

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_{ij}\rho_i=\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_{ji}\rho_j,$$

We have

$$\begin{aligned}D^{+}V(t)&\leq\sum_{i=1}^{n}\left[(\eta-a_i)u_i(t)+\sum_{j=1}^{n}\left(\left|b_{ij}\right|+\left|c_{ij}\right|e^{\eta\tau_i}\right)L_i u_i(t)\right]\\&=-\sum_{i=1}^{n}F_i(\eta)u_i(t)\leq 0,\qquad t>0,t\neq t_k,k\in Z^{+}.\end{aligned}$$

Also, (17) yields

$$\begin{aligned}&V(t_k+0)\\&=\sum_{i=1}^{n}\left(u_i(t_k+0)+\sum_{j=1}^{n}\left|c_{ij}\right|L_i e^{\eta\tau_i}\int_{t_k+0-\tau_i}^{t_k+0}u_i(s)ds\right)\\&\leq\sum_{i=1}^{n}\left(u_i(t_k)+\sum_{j=1}^{n}\left|c_{ij}\right|L_i e^{\eta\tau_i}\int_{t_k-\tau_i}^{t_k}u_i(s)ds\right)\\&=V(t_k),\quad k\in Z^{+}.\end{aligned}$$

It follows that $V(t)\leq V(0)$ for $t>0$. Thus, (18) yields

$$\begin{aligned}\sum_{i=1}^{n}u_i(t)&\leq V(t)\leq V(0)\\&=\sum_{i=1}^{n}\left(u_i(0)+\sum_{j=1}^{n}\left|c_{ij}\right|L_i e^{\eta\tau_i}\int_{-\tau_i}^{0}u_i(s)ds\right)\end{aligned}\tag{19}$$

for $t>0$. It follows from (15) and (19) that

$$\begin{aligned}\sum_{i=1}^{n}\left|x_i(t)-x_i^{*}\right|&\leq e^{-\eta t}\left[\sum_{i=1}^{n}\left(1+\sum_{j=1}^{n}\left|c_{ij}\right|L_i e^{\eta\tau_i}\tau\right)\sup_{s\in[-\tau,0]}\left|x_i(s)-x_i^{*}\right|\right]\\&\leq\beta e^{-\eta t},\quad t>0,\end{aligned}$$

where

$$\beta=\sum_{i=1}^{n}\left(1+\sum_{j=1}^{n}\left|c_{ij}\right|L_i e^{\eta\tau_i}\tau\right)\sup_{s\in[-\tau,0]}\left|x_i(s)-x_i^{*}\right|.$$

This completes the proof.

4 Example

Example 1. Consider the following neural networks system with delays and impulses:

$$\begin{cases} \dfrac{dx(t)}{dt} = -5x(t) - 2g(x(t)) + g(y(t-\tau)) + 4, \\ \dfrac{dy}{dt} = -3y(t) - 3g(x(t)) + 2g(y(t-\sigma)) + 2, \\ \Delta x(t_k) = -\gamma_k (x(t_k) - 1), \quad k \in Z^+, \\ \Delta y(t_k) = -\overline{\gamma}_k (y(t_k) - 2), \quad k \in Z^+, \end{cases} \tag{20}$$

where $g(u) = -|u|, \tau, \sigma \in (0, \infty)$, and $t_1 < t_2 < \cdots$ is a strictly increasing sequence such that $\lim_{k \to \infty} t_k = +\infty$,

$$\gamma_k = 1 + \frac{2}{3}\sin(1+k), \ \overline{\gamma}_k = 1 + \frac{1}{2}\cos(3k^2). \tag{21}$$

It is easy to verify that (20) satisfies all the conditions of Theorem 2. The equilibrium of (20) is $(x^*, y^*) = (1,1)$. According to Theorem 2, this equilibrium is globally exponentially stable.

Example 2. Consider the following neural networks system with delays and impulses:

$$\begin{cases} x_1'(t) = -2x_1(t) + \dfrac{1}{3}g(x_1(t)) + \dfrac{1}{2}g(x_2(t)) - \dfrac{1}{3}g(x_1(t-\tau_1)) + \dfrac{1}{2}g(x_2(t-\tau_2)) + 1, t \neq t_k, \\ x_2'(t) = -2x_2(t) + \dfrac{1}{2}g(x_1(t)) + \dfrac{1}{3}g(x_2(t)) + \dfrac{1}{2}g(x_1(t-\tau_1)) - \dfrac{1}{3}g(x_2(t-\tau_2)) - 2, t \neq t_k, \\ I_{1k}(x_1(t_k)) = -\dfrac{2}{3}(x_1(t_k) - x_1^*), \quad k \in N, \\ I_{2k}(x_2(t_k)) = -\dfrac{3}{2}(x_2(t_k) - x_2^*), \quad k \in N. \end{cases} \tag{22}$$

where

$$g_i(x_i(\cdot)) = \frac{1}{2}\left(|x_i(\cdot)+1| - |x_i(\cdot)-1|\right), i = 1,2,\cdots,n, \tag{23}$$

$\tau_1, \tau_2 \in (0, \infty)$, and $t_1 < t_2 < \cdots$ is a strictly increasing sequence such that $\lim_{k \to \infty} t_k = +\infty$.

Equation (23) yields

$$|g_i(u) - g_i(v)| \le |u - v|, \forall u, v \in R, i = 1,2,\cdots,n.$$

This means that $L_i = 1 (i = 1,2)$. We can prove that $(0,-1)$ is a unique equilibrium solution of (22). It is easy to verify that (22) satisfies all the conditions of Theorem 2. According to Theorem 2, this equilibrium is globally exponentially stable.

5 Conclusion

In this paper, we consider the exponential stability of a class of neural networks with delays and impulses. Under assumption that the activation function is not bounded, monotonic or differentiable, we obtain some sufficient conditions for the existence and globally exponential stability of a unique equilibrium. An example is given to demonstrate the effectiveness of the obtained results.

References

1. Chua, L.O., Yang, T.: Cellular Neural Network: Theory. IEEE Trans CAS-I 35, 1257–1272 (1988)
2. Chua, L.O., Yang, T.: Cellular Neural Network: Application. IEEE Trans CAS-I 35, 1273–1290 (1988)
3. Goplasam, K., He, X.: Stability in Asymmetric Hopfield Nets with Transmission Delays. Phys D 76, 344–358 (1994)
4. Morita, M.: Associative Memory with Non-monotone Dynamics. IEEE Trans on Neural Networks 6, 115–126 (1993)
5. Liao, X.: Stability of Hopfield-type Neural Networks(I). Science in China 38, 407–418 (1995)
6. Liao, X., Liao, Y.: Stability of Hopfield-type Neural Networks(II). Science in China 40, 813–816 (1997)
7. Cao, J., Lin, Y.: Stability of a Class of Neural Network Models with Delay. Applied Mathematics and Mechanics 20, 851–855 (1999)
8. Cao, J.: Global Asymptotic Stability Analysis of Delayed Cellular Neural Networks. Journal of Electronics 22, 253–258 (2000)
9. Zhong, S.: Stability of Cellular Neural Networks with Delay. Acta Electronica Sinica 25, 125–127 (1997)
10. Zhou, D., Cao, J.: Stability Analysis on Delayed Cellular Neural Networks. Information and Control 27, 31–36 (1998)
11. Chen, A., Cao, J.: Existence and Attractivity of Almost Periodic Solutions for Cellular Neural Networks with Distributed Delays and Variable Coefficients. Applied Math. and Comp. 134, 125–140 (2003)

LMI Based Global Asymptotic Stability Criterion for Recurrent Neural Networks with Infinite Distributed Delays*

Zhanshan Wang[1], Huaguang Zhang[1], Derong Liu[2], and Jian Feng[1]

[1] School of Information Science and Engineering, Northeastern University, Shenyang, Liaoning, 110004, People's Republic of China
wangzhanshan@ise.neu.edu.cn, hgzhang@ieee.org
[2] Department of Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, IL60607, USA
dliu@ece.uic.edu

Abstract. Global asymptotic stability problem for a class of recurrent neural networks with infinite distributed delay is investigated based on the linear matrix inequality (LMI) technique. Using a matrix decomposition method, a vector-matrix form of recurrent neural networks with infinite distributed delay is obtained. Then by constructing a suitable Lyapunov functional and using an inequality, new LMI-based criteria are established to ensure the global asymptotic stability of the class of neural networks, which considers the effects of neuron's excitatory and inhibitory action in the term of infinite delay on the networks. The obtained results are independent of the size of delay and are easily verified. Numerical example shows the effectiveness of the obtained results.

1 Introduction

The dynamical characteristic such as global asymptotic stability of recurrent neural networks plays an important role in the image processing, pattern recognition and associative memory and optimization problems. As a consequence, many researchers have forced their attention on the study of global stability of recurrent neural networks without delays or with discrete delays[1-29]. As is well known, the use of discrete delays in models of delayed feedback provides of a good approximation in simple circuits consisting of a small number of cells. However, neural networks usually have a spatial extent due to the presence of a multitude of parallel pathways with a variety of axon sizes and lengths, and hence there is a distribution of delays over a period of time. Thus, there should be a distribution of conduction velocities along the sepathways, and the

* This work was supported by the National Natural Science Foundation of China (Grant Nos. 60534010, 60572070, 60728307, 60774048, 60774093), the Program for Cheung Kong Scholars and Innovative Research Groups of China (Grant No. 60521003) and the National High Technology Research and Development Program of China (Grant No. 2006AA04Z183–B08015), the Natural Science Foundation of Liaoning Province (Grant No. 20072025), the Postdoctoral Science Foundation of China (Grant No. 20080431150) and the Postdoctoral Foundation of Northeastern University (Grant No. 20080314).

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 463–471, 2009.

distribution of propagation is not instantaneous and cannot be modelled with discrete delays. A more appropriate description is to incorporate continuously distributed delays. At present, there are many results on global stability of recurrent neural networks with infinite distributed delays using different analysis method, for example, M-matrix method, matrix measure method, and differential inequality method [2,4]. However, all above results didn't consider the signs of the interconnection weight coefficients, and this leads to the ignorance of the effects of the neuron's excitatory and inhibitory action, which increases the conservativeness of the stability results. Because the stability results based on linear matrix inequality (LMI) method can efficiently reduce the conservativeness of the stability results and also can be easily checked using interior algorithms, the LMI-based stability results have been extensively studied in recent years [32,33,34,35].

Nowadays, there exists many LMI-based stability results for recurrent neural networks with different delays, for example, discrete delay, neutral delay and infinite distributed delay etc, [2,28,30,31,32,33]. To the best of our knowledge, no LMI-based stability result for recurrent neural networks with infinite distributed delays has been reported in the literatures. It is important to study the LMI-based stability problem for recurrent neural networks with infinite distributed delays as it is for recurrent neural networks with other kind of delays. Therefore, the purpose of this paper is to propose a novel LMI-based stability result for recurrent neural networks with infinite distributed delays. By using the matrix decomposition method and suitably constructing the lyapunov functional, the corresponding stability criterion is derived, which is less conservative than the existing results by a numerical simulation.

2 Problem Formulation

Consider the following recurrent neural networks with continuous distributed delays,

$$\begin{aligned}\dot{u}_i(t) = &- a_i u_i(t) + \sum_{j=1}^{n} w_{ij} g_j(u_j(t)) + \sum_{j=1}^{n} w_{ij}^1 g_j(u_j(t-\tau(t))) \\ &+ \sum_{j=1}^{n} c_{ij} \int_{-\infty}^{t} k_{ij}(t-s) g_j(u_j(s)) ds + U_i, \end{aligned} \tag{1}$$

where $u_i(t)$ is the neural state, $a_i > 0$, w_{ij} and w_{ij}^1 are connection weight coefficients and delayed connection weight coefficients, respectively, U_i is the constant external input, time-varying delay satisfies $0 \leq \tau(t) \leq \tau_M$, $\dot{\tau}(t) \leq \mu < 1$, $g_j(u_j(t)))$ is the activation function, $i, j = 1, \ldots, n$.

Assumption 1. The activation function $g_j(u_j(t)))$ satisfies $0 \leq (g_j(\zeta) - g_j(\xi))/(\zeta - \xi) \leq \delta_j$ for $\forall \zeta \neq \xi, \zeta, \xi \in \Re$, $\delta_j > 0$, $j = 1, \ldots, n$

Assumption 2. The activation function $g_j(u_j(t)))$ satisfies $|g_j(\zeta) - g_j(\xi)| \leq \delta_j |\zeta - \xi|$ for $\forall \zeta, \xi \in \Re$, $\delta_j > 0$, $j = 1, \ldots, n$

Obviously, Assumption 2 is rather general as it merely requires the activation function to be Lipschitzian. In contrast, Assumption 1 requires a non-decreasing feature of the activation function. Clearly, Assumption 1 is a special case of Assumption 2.

Assumption 3. The delay kernel $k_{ij}(s)$, $i, j = 1, \cdots, n$, is a real value non-negative continuous function defined on $[0, \infty)$ and, for each i, j, it satisfies $\int_0^\infty k_{ij}(s)ds = 1$.

Lemma 1. [29](Cauchy's Inequality) For continuous functions $g(s)$ and $h(s)$, which are well defined in the integral interval, the following inequality holds,

$$\Big(\int_{-\infty}^{b} g(s)h(s)ds\Big)^2 \leq \Big(\int_{-\infty}^{b} g^2(s)ds\Big)\Big(\int_{-\infty}^{b} h^2(s)ds\Big),$$

where $b \geq 0$.

It is well known that the bounded activation function can always ensures that system (1) has an equilibrium point. Let u_i^* be an equilibrium point, then by state transformation $x_i(t) = u_i(t) - u_i^*$, system (1) is changed into the following form

$$\begin{aligned}\dot{x}_i(t) = &- a_i x_i(t) + \sum_{j=1}^{n} w_{ij} f_j(x_j(t)) + \sum_{j=1}^{n} w_{ij}^1 f_j(x_j(t-\tau(t))) \\ &+ \sum_{j=1}^{n} c_{ij} \int_{-\infty}^{t} k_{ij}(t-s) f_j(x_j(s))ds, \end{aligned} \tag{2}$$

or in a vector-matrix form

$$\dot{x}(t) = -Ax(t) + Wg(x(t)) + W_1 f(x(t-\tau(t))) + \sum_{i=1}^{n} E_i \int_{-\infty}^{t} K_i(t-s) f(x(s))ds, \tag{3}$$

where $f_j(x_j(t)) = g_j(x_j(t) + u_j^*) - g_j(u_j^*)$, $A = \text{diag}(a_1, \ldots, a_n)$, $W = (w_{ij})_{n\times n}$, $W_1 = (w_{ij}^1)_{n\times n}$, $x(t) = (x_1(t), \ldots, x_n(t))^T$, $f(x(t)) = (f_1(x_1(t)), \ldots, f_n(x_n(t)))^T$, $f(x(t-\tau(t))) = (f_1(x_1(t-\tau(t))), \ldots, f_n(x_n(t-\tau(t))))^T$. $C = (c_{ij})_{n\times n}$, E_i is an $n \times n$ matrix, whose i-th row is composed by the i-th row of matrix C, and the other rows are all zeros. $K_i(t-s) = \text{diag}(k_{i1}(t-s), k_{i2}(t-s), \cdots, k_{in}(t-s))$ is a diagonal matrix, $i = 1, \ldots, n$. Obviously, by Assumption 1 and Assumption 2, $0 \leq f_j(x_j(t))/x_j(t) \leq \delta_j$ for $\forall x_j(t) \neq 0$, or $|f_j(x_j)| \leq \delta_j |x_j|$, respectively, $j = 1, \ldots, n$.

3 Global Asymptotic Stability Results

We now state and prove our main result of the paper.

Theorem 1. Suppose that Assumption 1 holds. If there exist positive definite symmetric matrices P and Q, positive diagonal matrices G, D, H^i, $i = 1, 2, \cdots, n$, such that the following LMI holds,

$$\Xi = \begin{bmatrix} \Psi_{11} & \Psi_{12} & PW_1 & PE_1 & PE_2 & \cdots & PE_n \\ * & \Psi_{22} & DW_1 & DE_1 & DE_2 & \cdots & DE_n \\ * & * & -(1-\mu)Q & 0 & 0 & \cdots & 0 \\ * & * & * & -H^1 & 0 & \cdots & 0 \\ * & * & * & * & -H^2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ * & * & * & * & * & \cdots & -H^n \end{bmatrix} < 0, \tag{4}$$

then the equilibrium point of system (3) is globally asymptotically stable, where $*$ denotes the symmetric part in a matrix, $\Delta = \text{diag}(\delta_1, \ldots, \delta_n)$,
$\Psi_{11} = -PA - (PA)^T$,
$\Psi_{12} = PW - (DA)^T + \Delta G$,
$\Psi_{22} = DW + (DW)^T + Q + \sum_{i=1}^{n} H^i - 2G$.

Proof. Consider the following Lyapunov functional $V = V_1 + V_2$, where,

$$V_1 = x^T Px + 2\sum_{i=1}^{n} \int_0^{x_i} d_i f_i(s)ds + \int_{t-\tau(t)}^{t} f^T(x(s))Qf(x(s))ds,$$

$$V_2(t) = \sum_{i=1}^{n}\sum_{j=1}^{n} H_j^i \int_0^{\infty} k_{ij}(s) \int_{t-s}^{t} f_j^2(x_j(\theta))d\theta ds,$$

$d_i > 0, H_j^i > 0$, P and Q are positive definite symmetric matrices, $i, j = 1, \cdots, n$.

The derivative of $V_1(t)$ along the trajectories of system (3) is as follows,

$$\begin{aligned} \dot{V}_1(t) =& 2x^T P[-Ax(t) + Wf(x(t)) + W_1 f(x(t-\tau(t))) \\ &+ \sum_{i=1}^{n} E_i \int_{-\infty}^{t} K_i(t-s)f(x(s))ds] + 2f^T(x)D[-Ax(t) + Wf(x(t)) \\ &+ W_1 f(x(t-\tau(t))) + \sum_{i=1}^{n} E_i \int_{-\infty}^{t} K_i(t-s)f(x(s))ds] \\ &+ f^T(x(t))Qf(x(t)) - (1-\dot{\tau}(t))f^T(x(t-\tau(t)))Qf(x(t-\tau(t))). \end{aligned} \tag{5}$$

The derivative of $V_2(t)$ along the trajectories of system (3) is as follows,

$$\dot{V}_2(t) = \sum_{i=1}^{n}\sum_{j=1}^{n} H_j^i \int_0^{\infty} k_{ij}(s) f_j^2(x_j(t))ds - \sum_{i=1}^{n}\sum_{j=1}^{n} H_j^i \int_0^{\infty} k_{ij}(s) f_j^2(x_j(t-s))ds. \tag{6}$$

By Assumption 3 and Lemma 1, we have

$$\begin{aligned} \dot{V}_2(t) &= \sum_{i=1}^{n}\sum_{j=1}^{n} H_j^i f_j^2(x_j(t)) - \sum_{i=1}^{n}\sum_{j=1}^{n} H_j^i \int_0^{\infty} k_{ij}(s) f_j^2(x_j(t-s))ds \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} H_j^i f_j^2(x_j(t)) - \sum_{i=1}^{n}\sum_{j=1}^{n} H_j^i \int_0^{\infty} k_{ij}(s)ds \int_0^{\infty} k_{ij}(s) f_j^2(x_j(t-s))ds \end{aligned}$$

$$
\begin{aligned}
&\leq \sum_{i=1}^{n} f^T(x) H^i f(x) - \sum_{i=1}^{n}\sum_{j=1}^{n} H_j^i \Big(\int_0^{\infty} k_{ij}(s) f_j(x_j(t-s)) ds \Big)^2 \\
&= \sum_{i=1}^{n} f^T(x) H^i f(x) - \sum_{i=1}^{n}\sum_{j=1}^{n} H_j^i \Big(\int_{-\infty}^{t} k_{ij}(t-s) f_j(x_j(s)) ds \Big)^2 \\
&= \sum_{i=1}^{n} f^T(x) H^i f(x) \\
&\quad - \sum_{i=1}^{n} \Big(\int_{-\infty}^{t} K_i(t-s) f(x(s)) ds \Big)^T H^i \Big(\int_{-\infty}^{t} K_i(t-s) f(x(s)) ds \Big), \qquad (7)
\end{aligned}
$$

where $H^i = \mathrm{diag}(H_1^i, H_2^i, \cdots, H_n^i), i = 1, 2, \cdots, n$.

Note that, by Assumption 1, the following inequality holds

$$
2(f^T(x) \Delta G x - f^T(x) G f(x)) \geq 0, \qquad (8)
$$

for positive diagonal matrix G.

Combining (5), (7) and (8), we have

$$
\dot{V}(t) = \zeta^T \Xi \zeta < 0, \qquad (9)
$$

for $\zeta \neq 0$, where Ξ is defined in (4),

$$
\begin{aligned}
\zeta = \Big(& x(t) \; f(x(t)) \;\; f(x(t-\tau(t))) \\
& \int_{-\infty}^{t} K_1(t-s) f(x(s)) ds \; \cdots, \int_{-\infty}^{t} K_n(t-s) f(x(s)) ds \Big)^T.
\end{aligned}
$$

$\dot{V}(t) = 0$ if and only if $\zeta = 0$. According to Lyapunov stability theory [1], the equilibrium point of system (3) is globally asymptotically stable.

Now we consider the stability problem of neural network (3) under Assumption 2. result.

Theorem 2. Suppose that Assumption 2 holds. If there exist positive definite symmetric matrices P and Q, positive diagonal matrices G, D, H^i, $i = 1, 2, \cdots, n$, such that the following LMI holds,

$$
\Xi_1 = \begin{bmatrix}
\bar{\Psi}_{11} & \bar{\Psi}_{12} & PW_1 & PE_1 & PE_2 & \cdots & PE_n \\
* & \Psi_{22} & DW_1 & DE_1 & DE_2 & \cdots & DE_n \\
* & * & -(1-\mu)Q & 0 & 0 & \cdots & 0 \\
* & * & * & -H^1 & 0 & \cdots & 0 \\
* & * & * & * & -H^2 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
* & * & * & * & * & \cdots & -H^n
\end{bmatrix} < 0, \qquad (10)
$$

then the equilibrium point of system (3) is globally asymptotically stable, where $*$ denotes the symmetric part in a matrix,

$\Delta = \text{diag}(\delta_1, \ldots, \delta_n)$,
$\bar{\Psi}_{11} = -PA - (PA)^T + \Delta G \Delta$,
$\bar{\Psi}_{12} = PW - (DA)^T$,
$\Psi_{22} = DW + (DW)^T + Q + \sum_{i=1}^{n} H^i - 2G$.

Proof. By Assumption 2, $|f_j(x_j)| \leq \delta_j |x_j|$ or $f_j^2(x_j) \leq \delta_j^2 x_j^2$, $j = 1, 2, \cdots, n$. It means that the following inequality holds

$$f^T(x) G f(x) \leq x^T \Delta G \Delta x, \tag{11}$$

for positive diagonal matrix G.

Substituting (8) by (11) in the proof of Theorem 1, in a similar manner we can obtained the Theorem 2. The details are omitted.

Remark 1. Many stability results have been proposed for system (2) or its special case using different mathematical methods in the literature. For example, the results in [2,4,5,7,8,9,11,15,16,17,18,19] take absolute value operation on the interconnection weight coefficients, which ignore the effects of the neuron's excitatory and inhibitory action on the networks. The results in [12,13,14] also ignore the effects of the neuron's excitatory and inhibitory action on the networks, and are difficult to verify due to more unknown parameters to be tuned. LMI-based stability results are presented in [3,6], however, the sign difference in the infinite distributed delay weight coefficients is not considered, and the corresponding results are not easy to check. In contrast, our proposed results completely consider the effects of the neuron's excitatory and inhibitory action on the networks, which are easy to check and are less conservative than most of the existing results.

Remark 2. Note that the inequalities (8) and (11) represent the differences between Assumption 1 and Assumption 2. Obviously, inequality (8) can lead to inequality (11), i.e., $f^T(x) G f(x) \leq f^T(x) G \Delta x \leq x^T \Delta G \Delta x$. Conversely, it is not true. This means that the general condition in Assumption 2 will lead to a relatively conservative result (Theorem 2) than that (Theorem 1) under Assumption 1. Different Assumptions often gives different trade-off between the conservativeness and the generality.

4 Illustrative Examples

In this section, we will use an example to show the effectiveness of the obtained results.

Example 1. Let us consider a second-order neural network (3), where $\tau(t) > 0$ is any bounded constant delay, $f_i(x_i) = \tanh(x_i), i = 1, 2$, i.e., Δ is an identity matrix, $k_{11}(s) = k_{12}(s) = \frac{2}{\pi(1+s^2)}$, $k_{21}(s) = k_{22}(s) = 3e^{-3s}$. One can verify that Assumption 3 holds. The other parameters are as follows,

$$A = \begin{bmatrix} 8.9665 & 0 \\ 0 & 0.3227 \end{bmatrix}, \quad W = \begin{bmatrix} 0.7340 & 0.3998 \\ 0.4109 & 0.5055 \end{bmatrix},$$

$$W_1 = \begin{bmatrix} 0.1693 & 0.6412 \\ 0.5247 & 0.0162 \end{bmatrix} \quad C = \begin{bmatrix} -0.5000 & 0.3000 \\ 0.3000 & 0.2500 \end{bmatrix}.$$

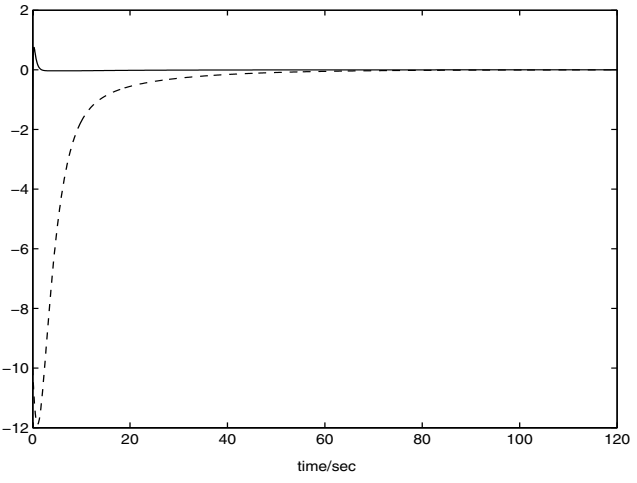


Fig. 1. State response curves of Example 1. x_1: solid line; x_2: dash-dotted line.

Pertaining to this example, because

$$M = A - (|W| + |W_1| + |C|)\Delta = \begin{bmatrix} 7.5632 & -1.3410 \\ -1.2356 & -0.4490 \end{bmatrix}$$

is not an M-matrix, Corollary 4.4 in [17] is not satisfied. Meanwhile,

$$\|W\| + \|W_1\| + \|C\| - \min(a_i) = 2.0209$$

is greater than 0, then Theorem 3 in [18] is not satisfied, here $\|\cdot\|$ denotes the 2-norm. Therefore, the results in [17,18] can not judge the stability of the example.

Applying Theorem 1 of the present paper, we have

$$Q = \begin{bmatrix} 37.3598 & -1.5058 \\ -1.5058 & 1.4514 \end{bmatrix}, \quad P = \begin{bmatrix} 9.6281 & -0.1194 \\ -0.1194 & 0.0680 \end{bmatrix}.$$

$$G = \begin{bmatrix} 76.5784 & 0 \\ 0 & 9.7606 \end{bmatrix}, \quad D = \begin{bmatrix} 0.0692 & 0 \\ 0 & 9.8066 \end{bmatrix}.$$

$$H^1 = \begin{bmatrix} 4.7524 & 0 \\ 0 & 0.4948 \end{bmatrix}, \quad H^2 = \begin{bmatrix} 15.8058 & 0 \\ 0 & 2.5104 \end{bmatrix}.$$

Therefore, the concerned neural network is globally asymptotically stable. When initial condition is $(1, \ -10)^T$, $\tau(t) = 1$, the state response curves are depicted in Figure 1.

5 Conclusions

Two LMI-based stability criteria are derived to ensure the global asymptotic stability of a class of recurrent neural networks with infinite distributed delay under different assumption on the activation function. The obtained results consider the effects of the neuron's inhibitory and excitatory action caused by the infinite distributed delay, which significantly improve the stability results existed in the literature. A numerical example is employed to demonstrate the effectiveness of the obtained result.

References

1. Wang, Z.: Stability of Continuous time Recurrent Neural Networks with Delays. Northeastern University Press, Shenyang (2007) (in Chinese)
2. Mao, Z., Zhao, H.: Dynamical Analysis of Cohen–Grossberg Neural Networks with Distributed Delays. Physics Letters A 364, 38–47 (2007)
3. Ji, Y., Lou, X., Cui, B.: Global Output Convergence of Cohen–Grossberg Neural Networks with Both Time-varying and Distributed Delays. Chaos, Solitons and Fractals (to appear)
4. Wu, W., Cui, B., Lou, X.: Global Exponential Stability of Cohen–Grossberg Neural Networks with Distributed Delays. Mathematical and Computer Modelling (to appear)
5. Huang, T., Li, C., Chen, G.: Stability of Cohen–Grossberg Neural Networks with Unbounded Distributed Delays. Chaos, Solitons and Fractals 34, 992–996 (2007)
6. Wan, L., Sun, J.: Global Asymptotic Stabilityof Cohen–Grossberg Neural Network with Continuously Distributed Delays. Physics Letters A 342, 331–340 (2005)
7. Song, Q., Cao, J.: Stability Analysis of Cohen–Grossberg Neural Network with Both Time-varying and Continuously Distributed Delays. Journal of Computational and Applied Mathematics 197, 188–203 (2006)
8. Zhang, Q., Wei, X., Xu, J.: Global Exponential Stability of Hopfield Neural Networks with Continuously Distributed Delays. Physics Letters A 315, 431–436 (2003)
9. Zhang, J., Suda, Y., Iwasa, T.: Absolutely Exponential Stability of a Class of Neural Networks with Unbounded Delay. Neural Networks 17, 391–397 (2004)
10. Liang, J., Cao, J.: Global Output Convergence of Recurrent Neural Networks with Distributed Delays. Nonlinear Analysis 23, 187–197 (2007)
11. Liu, Y., You, Z., Cao, L.: On the almost Periodic Solution of Cellular Neural Networks with Distributed Delays. IEEE Transactions on Neural Networks 18, 295–300 (2007)
12. Bao, S.: Exponential Stability of Reaction-diffusion Cohen-grossberg Neural Networks with Variable Coefficients and Distributed Delays. In: Proceedings of the 7th World Congress on Intelligent Control and Automation, Chongqing, China, pp. 8261–8264 (2008)
13. Lu, J.: Robust Global Exponential Stability for Interval Reaction-diffusion Hopfield Neural networks with Distributed Delays. IEEE Transactions on Circuits and Systems-II 54, 1115–1119 (2007)
14. Lou, X., Cui, B., Wu, W.: On Global Exponential Stability and Existence of Periodic Solutions for BAM Neural Networks with Distributed Delays and Reaction-diffusion Terms. Chaos, Solitons and Fractals 36, 1044–1054 (2008)
15. Cui, B., Wu, W.: Global Exponential Stability of Cohen–Grossberg Neural Networks with Distributed Delays. Neurocomputing 72, 386–391 (2008)
16. Zhou, L., Hu, G.: Global Exponential Periodicity and Stability of Cellular Neural Networks with Variable and Distributed Delays. Applied Mathematics and Computation 195, 402–411 (2008)
17. Lv, Y., Lv, W., Sun, J.: Convergence Dynamics of Stochastic Reaction-diffusion Recurrent Neural Networks with Continuously Distributed Delays. Nonlinear Analysis: Real World Applications 9, 1590–1606 (2008)
18. Zhou, J., Li, S., Yang, Z.: Global Exponential Stability of Hopfield Neural Networks with Distributed Delays. Applied Mathematical Modelling 33, 1513–1520 (2009)
19. Chen, W., Zheng, W.: Global Asymptotic Stability of a Class of Neural Networks with Distributed Delays. IEEE Transactions on Circuits and Systems-I 53, 644–652 (2006)
20. Yu, J., Zhang, K., Fei, S., Li, T.: Simplified Exponential Stability Analysis for Recurrent Neural Networks with Discrete and Distributed Time-varying Delays. Applied Mathematics and Computation 205, 465–474 (2008)

21. Rakkiyappan, R., Balasubramaniam, P., Lakshmanan, S.: Robust Stability Results for Uncertain Stochastic Neural Networks with Discrete Interval and Distributed Time-varying Delays. Physics Letters A 372, 5290–5298 (2008)
22. Hu, S., Liu, D.: On the Global Output Convergence of a Class of Recurrent Neural Networks with Time-varying Inputs. Neural Networks 18, 171–178 (2005)
23. Zhao, H.: Global Asymptotic Stability of Hopfield Neural Networks Involving Distributed Delays. Neural Networks 17, 47–53 (2004)
24. Sun, J., Wan, L.: Global Exponential Stability and Periodic Solutions of Cohen-Grossberg Neural Networks with Continuously Distributed Delays. Physica D 208, 1–20 (2005)
25. Zhang, J.: Absolute Stability of a Class of Neural Networks with Unbounded Delay. International Journal of Circuits Theory and Applications 32, 11–21 (2004)
26. Chen, Y.: Global Asymptotic Stability of Delayed Cohen-Grossberg Neural Networks. Transactions on Circuits and Systems-I 53, 351–357 (2006)
27. Liao, X., Wong, K., Yang, S.: Convergence Dynamics of Hybrid Bidirectional Associative Memory Neural Networks with Distributed Delays. Physics Letters A 316, 55–64 (2003)
28. Song, Q., Zhao, Z., Li, Y.: Global Exponential Stability of BAM Neural Networks with Distributed Delays and Reaction Diffusion Terms. Physics Letters A 335, 213–225 (2005)
29. Hardy, G., Littlewood, J., Polya, G.: Inequality, 2nd edn. Cambridge University Press, Cambridge (1954)
30. Gopalsamy, K., He, X.: Delay-independent Stability in Bidirectional Asociative Neural Networks. IEEE Transactions Neural Networks 15, 998–1002 (1994)
31. Liao, X., Wang, J.: Algebraic Criteria for Global Exponential Stability of Cellular Neural Networks with Multiple Time Delays. IEEE Transactions Circuits and Systems-I 50, 268–275 (2003)
32. Wang, Z., Zhang, H., Yu, W.: Robust Exponential Stability Analysis of Neural Networks with Multiple Time Delays. Neurocomputing 70, 2534–2543 (2007)
33. Wang, Z., Zhang, H.: LMI-based Criteria for Globally Asymptotic Stability of Cellular Neural Networks with Multiple Delays. Chinese Journal of Electronics 16, 111–114 (2007)
34. Zhang, H., Wang, Z.: Global Asymptotic Stability of Delayed Cellular Neural Networks. IEEE Transactions on Neural Networks 18, 947–950 (2007)
35. Zhang, H., Wang, Z., Liu, D.: Robust Exponential Stability of Cellular Neural Networks with Multiple Time Varying Delays. IEEE Transactions on Circuits and Systems-II 54, 730–734 (2007)

Existence and Stability of Periodic Solutions for BAM Neural Networks with Time-Varying Delays and Impulses

Chunxue Wu[1,2,⋆] and Bao Shi[1]

[1] Institute of Applied Mathematics, Naval Aeronautical and Astronautical University, Yantai, Shandong 264001, P R China
wuchunxue1020@163.com,
baoshi781@sohu.com
[2] School of Mathematics and Information, Yantai University
Yantai, Shandong 264005, P R China

Abstract. By using the continuation theorem of Mawhin's coincidence degree theory, some new sufficient conditions are obtained for the existence and stability of periodic solution of BAM neural networks with variable delays and impulses, and without requirement of the boundedness of the activation functions.

Keywords: BAM neural networks; periodic solutions; impulse; coincidence degree; delays.

1 Introduction

Bi-directional associative memory (BAM) neural networks were first introduced by Kosto [8]. It has been used in many fields such as pattern recognition and automatic control. Recently, in Ho, Liang and James [7], Li [10], Mohamad [14], the authors discussed the problem of stability for BAM networks with and without axonal signal transmission delays, and some sufficient conditions are obtained for the BAM networks. It is well known that studies on neural dynamical systems not only involve discussion of stability property, but also involve other dynamics behaviors such as periodic oscillatory, bifurcation and chaos. In many applications, the property of periodic oscillatory solutions are of great interest, see Gui and Ge [5], Li [9], Li, Liu and Huang [12]. Liu, Chen, Cao and Huang [13] studies the periodic oscillatory solution of BAM networks with periodic coefficients and time-varying delays by using the continuation theorem of Mawhin's coincidence degree theory and Grönwall's inequality.

As a kind of dynamic systems, BAM neural systems are generally characterized by either continuous or discrete time. Recently, there has been a somewhat new category of neural networks, which is neither purely continuous-time nor

⋆ This work is supported by the Distinguished Expert Science Foundation of the Naval Aeronautical and Astronautical University.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 472–481, 2009.

purely discrete-time and exhibits a combination of continuous and discrete characteristics. These are called impulsive neural networks, see Guan and Chen [4], Li [10,11], Xie, Wen and Li [15], Yang and Chua [16], Yang and Xu [17], Zhu and Xu [18]. Gui, Yang and Ge [6] studies the periodic solution for BAM neural networks with impulses and without delays, and with requirement of the boundedness of the activation functions.

In Gopalsamy and He [3], a modification of (1) called bi-directional associative memory (BAM) networks with delays

$$\begin{cases} x_i'(t) = -a_i x_i(t) + \sum\limits_{j=1}^{m} p_{ji} f_j\left(y_j\left(t-\tau_{ji}\right)\right) + c_i, \quad i = 1, \cdots, n, \\ y_j'(t) = -b_j y_j(t) + \sum\limits_{i=1}^{n} q_{ij} g_i\left(x_i\left(t-\sigma_{ij}\right)\right) + d_j, \quad j = 1, \cdots, m \end{cases} \tag{1}$$

has been discussed, where $x_i(t)$ and $y_j(t)$ are the state of the i-th neurons and the j-th neurons at the time t; a_i, b_j are positive constants denoting the rate with which the cell i and j reset their potential to the resting state when isolated from the other cells and inputs; time delays τ_{ji} and σ_{ij} are nonnegative constants, which correspond to the finite speed of the axonal signal transmission; f_j and g_i are activation functions; p_{ji}, q_{ij} are the connection weights denoting the strengths of connectivity between the cells j and i at time $t-\tau_{ji}$ and $t-\sigma_{ij}$, respectively; c_i and d_j denote the i-th and j-th component of an external input source introduced from outside the network to the cell i and j, respectively.

In this paper, we will study the existence and global exponential stability of periodic solution of the following BAM neural networks with time-varying delays and impulses

$$\begin{cases} x_i'(t) = -a_i(t) x_i(t) + \sum\limits_{j=1}^{m} p_{ji}(t) f_j\left(y_j\left(t-\tau_{ji}(t)\right)\right) + c_i(t), t > 0, \ t \neq t_k, \\ \triangle x_i\left(t_k\right) = I_k\left(x_i\left(t_k\right)\right), \ i = 1, \cdots, n, \ k \in \mathbb{Z}^+, \\ y_j'(t) = -b_j(t) y_j(t) + \sum\limits_{i=1}^{n} q_{ij}(t) g_i\left(x_i\left(t-\sigma_{ij}(t)\right)\right) + d_j(t), t > 0, \ t \neq t_k, \\ \triangle y_j\left(t_k\right) = J_k\left(y_j\left(t_k\right)\right), \ j = 1, \cdots, m, \ k \in \mathbb{Z}^+. \end{cases} \tag{2}$$

where $\triangle x_i\left(t_k\right)$, $\triangle y_j\left(t_k\right)$ are the impulses at moments t_k and $t_1 < t_2 < \cdots$ is a strictly increasing sequence such that $\lim\limits_{k\to\infty} t_k = +\infty$. The system (2) is supplemented with initial values given by

$$\begin{cases} x_i(s) = \varphi_i(s), s \in [-\tau, 0]_{\mathbb{Z}}, \tau = \max\limits_{1\le i\le n, 1\le j\le m} \sup\limits_{t\in[0,\omega]} \{\tau_{ji}(t)\}, \quad i = 1, \cdots, n, \\ y_j(s) = \psi_j(s), s \in [-\sigma, 0]_{\mathbb{Z}}, \sigma = \max\limits_{1\le i\le n, 1\le j\le m} \sup\limits_{t\in[0,\omega]} \{\sigma_{ij}(t)\}, \quad j = 1, \cdots, m, \end{cases}$$

where $\varphi_i(\cdot)$ and $\psi_j(\cdot)$ are continuous ω-functions defined on $[-\tau, 0]$ and $[-\sigma, 0]$.

As usual in the theory of impulsive differential equations, at the points of discontinuity t_k of the solution $t \mapsto u(t) = (x_1(t), \cdots, x_n(t), y_1(t), \cdots, y_m(t))^T$, we assume that $u\left(t_k\right) \equiv u\left(t_k^-\right)$. It is clear that, in general, the derivatives $x_i'\left(t_k\right)$

and $y_j'(t_k)$ do not exist. On the other hand, according to (2) there exist the limits $x_i'(t_k^{\mp})$ and $y_j'(t_k^{\mp})$. According to the above convention, we assume $x_i'(t_k) \equiv x_i'(t_k^-)$ and $y_j'(t_k) \equiv y_j'(t_k^-)$.

Throughout this paper, we assume that

(H_1) There exists positive number μ_j^f, μ_i^g such that

$$0 \leq \frac{f_j(u_1) - f_j(u_2)}{u_1 - u_2} \leq \mu_j^f, \quad f_j(0) = 0,$$

$$0 \leq \frac{g_i(u_1) - g_i(u_2)}{u_1 - u_2} \leq \mu_i^g, \quad g_i(0) = 0,$$

for each $u_1, u_2 \in \mathbb{R}$, $u_1 \neq u_2$, $i = 1, \cdots, n$, $j = 1, \cdots, m$.

(H_2) There exists a positive integer q such that $t_{k+q} = t_k + \omega, I_{k+q}(x) = I_k(x), J_{k+q}(x) = J_k(x)$, $k = 1, 2, \cdots$.

(H_3) $a_i(t) > 0$, $b_j(t) > 0$, $c_i(t)$, $d_j(t)$, $p_{ji}(t)$, $q_{ij}(t)$ and $\tau_{ji}(t)$, $\sigma_{ij}(t)$ are all continuous periodic functions with the period ω, and the delays $0 \leq \tau_{ji}(t) \leq \tau, 0 \leq \sigma_{ij}(t) \leq \sigma$ $(i = 1, \cdots, n,\ j = 1, \cdots, m)$ are bounded, and $0 \leq \tau_{ji}'(t) < 1$, $0 \leq \sigma_{ij}'(t) < 1$.

The organization of this paper is as follows. In Section 2, we introduce some lemma needed in later sections. In Section 3, we prove the existence of the periodic solutions, and we do not require that the activation functions are bounded. In Section 4, we establish conditions under which the periodic solution is the globally exponentially stable.

2 Preliminaries

In this section, based on the Mawhin's continuation theorem, we shall study the existence of at least one periodic solution of (2). To do so, we shall make some preparations.

Let X and Z be two Banach space. Suppose that linear mapping $L : \mathrm{Dom}L \subset X \to Z$ is a Fredholm operator of index zero, there exist continuous projectors $P : X \to \ker L$ and $Q : Z \to Z/\mathrm{Im}L$ such that $\mathrm{Im}P = \ker L$, $\mathrm{Im}L = \ker Q = \mathrm{Im}(I - Q)$. It follows that mapping $L_P = L|_{\mathrm{Dom}L \cap \ker P} : (I - P)X \to \mathrm{Im}L$ is invertible. We denote the inverse of the mapping by K_P. Let Ω be a bounded open set in X, the mapping N will be called $L-$compact on $\overline{\Omega}$ if $QN(\overline{\Omega})$ is bounded and $K_P(I - Q)(\overline{\Omega})$ is compact.

Lemma 1. *(Gaines-Mawhin continuation theorem, Gaines and Mawhin [2]) Let X and Z be two Banach space, $L : \mathrm{Dom}L \subset X \to Z$ be a Fredholm operator with index zero, Assume that Ω is a bounded open set in X, and $N : \overline{\Omega} \to Z$ is $L-$compact on $\overline{\Omega}$. Suppose that*

(1) $Lx \neq \lambda Nx$ for all $x \in \partial\Omega \cap \mathrm{Dom}L$ and $\lambda \in (0, 1)$;

(2) $QNx \neq 0$ for all $x \in \partial\Omega \cap \ker L$;

(3) $\deg\{QN, \Omega \cap \ker L, 0\} \neq 0$.

Then equation $Lx = Nx$ has at lest one solution in $\mathrm{Dom}L \cap \overline{\Omega}$.

For any nonnegative integer p, let

$$C^{(p)}[0,\omega;t_1,\cdots,t_q] = \left\{ u:[0,\omega]\to\mathbb{R}^n \middle| \begin{array}{l} u^{(p)}(t) \quad \text{is continuous with respect to} \quad t\neq t_1,\cdots,t_q; \\ u^{(p)}(t_k^+) \quad \text{and} \quad u^{(p)}(t_k^-) \quad \text{exist at} \quad t_1,\cdots,t_q; \\ u^{(j)}(t_k) = u^{(j)}\left(t_k^-\right), \quad k=1,\cdots,q, \quad j=0,1,\ldots,p. \end{array} \right\}.$$

Let $X = \{u\in C[0,\omega;t_1,\cdots,t_q] \mid u(t)=u(t+\omega)\}$, $Z = X\times\mathbb{R}^{(n+m)\times(q+1)}$ and $\|u\| = \sum_{i=1}^{n}\max_{t\in[0,\omega]}|x_i(t)| + \sum_{j=1}^{m}\max_{t\in[0,\omega]}|y_j(t)|$. Then it is standard to show that both X and Z are Banach spaces.

Let $r(t)$ be a ω-periodic continuous function defined on $\mathbb{R}$. We define $r^- = \min_{0\le t\le\omega}|r(t)|$, $r^+ = \max_{0\le t\le\omega}|r(t)|$, $\overline{r} = \frac{1}{\omega}\int_0^\omega r(t)\mathrm{d}t$, $\|r\|_2 = \left(\int_0^\omega |r(t)|^2\mathrm{d}t\right)^{\frac{1}{2}}$.

3 Existence of Periodic Solution

Theorem 1. *Assume that (H_1)—(H_3) holds. Then system (2) has at least one ω-periodic solution.*

Proof. In order to use continuation theorem of coincidence degree theory to establish the existence of an $\omega-$periodic solution of (2), we take

$$L:\mathrm{Dom}L\cap X\to Z, Lu = (u', \Delta u(t_1),\cdots,\Delta u(t_q),0)$$

$$\mathrm{Dom}L = \left\{u(t)\in C^1[0,\omega;t_1,\cdots,t_q] \mid u(0)=u(\omega)\right\}, \quad N:X\to Z,$$

$$N(u(t)) = \left(\begin{pmatrix} A_1(t)\\ \vdots \\ \vdots \\ \vdots \\ A_{n+m}(t)\end{pmatrix}, \begin{pmatrix} \triangle x_1(t_1)\\ \vdots \\ \triangle x_n(t_1)\\ \triangle y_1(t_1)\\ \vdots \\ \triangle y_m(t_1)\end{pmatrix}, \cdots, \begin{pmatrix} \triangle x_1(t_q)\\ \vdots \\ \triangle x_n(t_q)\\ \triangle y_1(t_q)\\ \vdots \\ \triangle y_m(t_q)\end{pmatrix}, \begin{pmatrix} 0\\ \vdots \\ 0\\ 0\\ \vdots \\ 0\end{pmatrix}\right),$$

where

$$A_i(t) = -a_i(t)x_i(t) + \sum_{j=1}^{m} p_{ji}(t)f_j(y_j(t-\tau_{ji}(t))) + c_i(t), \quad i=1,\cdots,n,$$

$$A_{n+j}(t) = -b_j(t)y_j(t) + \sum_{i=1}^{n} q_{ij}(t)g_i(x_i(t-\sigma_{ij}(t))) + d_j(t), \quad j=1,\cdots,m.$$

It is not difficult to show that

$$\ker L = \left\{x\in X \mid x = h\in\mathbb{R}^{n+m}\right\},$$

$$\mathrm{Im}L = \left\{ z = (f, C_1, \cdots, C_q, d) \in \mathbb{Z} \mid \frac{1}{\omega} \int_0^\omega f(s)\mathrm{d}s + \sum_{k=1}^{q} C_k + d = 0 \right\}.$$

and $\mathrm{Im}L$ is closed in Z. Therefore, L is a Fredholm mapping of index zero. Take

$$P : X \to \ker L, \quad Px = \frac{1}{\omega} \int_0^\omega x(t)\mathrm{d}t,$$

$$Q : X \to Z, \quad Qz = \left\{ \frac{1}{\omega} \left[\int_0^\omega f(s)\mathrm{d}s + \sum_{k=1}^{q} C_k + d \right], 0, \cdots, 0, 0 \right\}.$$

It is trivial to show that P and Q are continuous projectors.

Since $\dim \ker L = n + m = \mathrm{codim}\mathrm{Im}L$. Therefore, L is a Fredholm operator with the index zero. Hence, $\mathrm{Im}P = \ker L$, $\mathrm{Im}L = \ker Q = \mathrm{Im}(I - Q)$, the generalized inverse K_P exists. Furthermore, we have that N is L-compact on $\overline{\Omega}$ (see Gaines and Mawhin [2]).

Now it needs to show that there exists an domain Ω, which satisfies all the requirements given in corresponding to operator equation $Lx = \lambda Nx$, $\lambda \in (0, 1)$.

We have

$$\begin{cases} x_i'(t) = \lambda \left[-a_i(t)x_i(t) + \sum\limits_{j=1}^{m} p_{ji}(t) f_j\left(y_j\left(t - \tau_{ji}(t)\right)\right) + c_i(t) \right], \\ \qquad t > 0, t \neq t_k, \\ \triangle x_i\left(t_k\right) = \lambda I_k\left(x_i\left(t_k\right)\right), \ i = 1, \cdots, n, \ k \in \mathbb{Z}^+, \\ y_j'(t) = \lambda \left[-b_j(t)y_j(t) + \sum\limits_{i=1}^{n} q_{ij}(t) g_i\left(x_i\left(t - \sigma_{ij}(t)\right)\right) + d_j(t) \right], \\ \qquad t > 0, t \neq t_k, \\ \triangle y_j\left(t_k\right) = \lambda J_k\left(y_j\left(t_k\right)\right), j = 1, \cdots, m, k \in \mathbb{Z}^+. \end{cases} \tag{3}$$

Suppose that $u(t) = (x_1(t), \cdots, x_n(t), y_1(t), \cdots, y_m(t))^T \in X$ is a solution of system (3) for a certain $\lambda \in (0, 1)$. Integrating (3) over the interval $[0, \omega]$, we obtain

$$\int_0^\omega A_i(t)\mathrm{d}t + \sum_{k=1}^{q} I_k\left(x_i\left(t_k\right)\right) = 0,$$

$$\int_0^\omega A_{n+j}(t)\mathrm{d}t + \sum_{k=1}^{q} J_k\left(y_j\left(t_k\right)\right) = 0.$$

Hence

$$\int_0^\omega a_i(t)x_i(t)\mathrm{d}t = \int_0^\omega \left[\sum_{j=1}^{m} p_{ji}(t) f_j\left(y_j\left(t - \tau_{ji}(t)\right)\right) + c_i(t) \right] \mathrm{d}t + \sum_{k=1}^{q} I_k\left(x_i\left(t_k\right)\right). \tag{4}$$

Let $\xi_i \in [0,\omega](\neq t_k)$, $k=1,\cdots,q$, such that $x_i(\xi_i) = \inf\limits_{t\in[0,\omega]} x_i(t)$, $i=1,\cdots,n$. Then, by (4) and Hölder inequality, we have

$$\omega\overline{a}_i x_i(\xi_i) \le \sum_{j=1}^{m} p_{ji}^{+}\mu_j^f \int_0^\omega |y_j(t-\tau_{ji}(t))|\,\mathrm{d}t + \omega c_i^{+} + \sum_{k=1}^{q} |I_k(x_i(t_k))|.$$

On the other hand, let $s_{ji} = t-\tau_{ji}(t)$. Then $\frac{\mathrm{d}s_{ji}}{\mathrm{d}t} = 1-\tau'_{ji}(t) > 0$, s_{ji} is function of t. Thus there exist functions $t = \tau_{ji}^{*}(s_{ji})$, $s_{ji} \in [-\tau_{ji}(0), \omega-\tau_{ji}(\omega)]$. So

$$\int_0^\omega |y_j(t-\tau_{ji}(t))|^2\,\mathrm{d}t = \int_{-\tau_{ji}(0)}^{\omega-\tau_{ji}(\omega)} \frac{|y_j(s_{ji})|^2}{1-\tau'_{ji}\left(\tau_{ji}^{*}(s_{ji})\right)}\mathrm{d}s_{ji} = k_{ji}^2 \int_0^\omega |y_j(s)|^2\,\mathrm{d}s,$$

where $k_{ji} = \left(\max\limits_{0\le t\le\omega} \frac{1}{1-\tau'_{ji}(t)}\right)^{\frac{1}{2}}$. Therefore

$$\omega\overline{a}_i x_i(\xi_i) \le \sqrt{\omega}\sum_{j=1}^{m} p_{ji}^{+}\mu_j^f k_{ji}^2 \|y_j\|_2^2 + \omega c_i^{+} + \sum_{k=1}^{q} |I_k(x_i(t_k))|.$$

$$x_i(\xi_i) \le \frac{1}{\overline{a}_i\sqrt{\omega}}\sum_{j=1}^{m} p_{ji}^{+}\mu_j^f k_{ji}^2 \|y_j\|_2^2 + c_i^{+} + \frac{1}{\omega}\sum_{k=1}^{q} |I_k(x_i(t_k))|, \quad i=1,\cdots,n. \tag{5}$$

Similarly, let $\eta_j \in [0,\omega](\neq t_k)$, $k=1,\cdots,q$, such that $y_j(\eta_j) = \inf\limits_{t\in[0,\omega]} y_j(t)$, $j=1,\cdots,m$. Then we have

$$y_j(\eta_j) \le \frac{1}{\overline{b}_j\sqrt{\omega}}\sum_{i=1}^{n} q_{ij}^{+}\mu_j^g \ell_{ij}^2 \|x_i\|_2^2 + d_j^{+} + \frac{1}{\omega}\sum_{k=1}^{q} |J_k(y_j(t_k))|, \quad j=1,\cdots,m,$$

where $\ell_{ij} = \left(\max\limits_{0\le t\le\omega} \frac{1}{1-\sigma'_{ij}(t)}\right)^{\frac{1}{2}}$.

Set $t_0 = t_0^{+} = 0$, $t_{q+1} = \omega$. From (3), we have

$$\begin{aligned}
\int_0^\omega |x_i'(t)|\,\mathrm{d}t &= \sum_{k=1}^{q+1}\int_{t_{k-1}+0}^{t_k} |x_i'(t)|\,\mathrm{d}t + \sum_{k=1}^{q}\left|x_i\left(t_k^{+}\right) - x_i(t_k)\right| \\
&\le \int_0^\omega |a_i(t)|\,|x_i(t)|\,\mathrm{d}t + \int_0^\omega \sum_{j=1}^{m} |p_{ji}(t)|\,|f_j(y_j(t-\tau_{ji}(t)))|\,\mathrm{d}t \\
&\quad + \int_0^\omega |c_i(t)|\,\mathrm{d}t + \sum_{k=1}^{q} |I_k(x_i(t_k))| \\
&\le \sqrt{\omega}a_i^{+}\|x_i\|_2 + \sum_{j=1}^{m}\sqrt{\omega}p_{ji}^{+}\mu_j^f k_{ji}\|y_j\|_2 + c_i^{+}\omega + \sum_{k=1}^{q}|I_k(x_i(t_k))|.
\end{aligned} \tag{6}$$

Multiplying both sides of system (3) by $x_i(t)$ and integrating over $[0,\omega]$, since

$$\begin{aligned}
\int_0^\omega x_i(t)x_i'(t)\mathrm{d}t &= \tfrac{1}{2}\sum_{\ell=1}^{q}\left[x_i^2(t_\ell) - x_i^2\left(t_\ell^{+}\right)\right] \\
&= -\lambda\sum_{k=1}^{q}\left[x_i(t_k) + \tfrac{1}{2}I_k(x_i(t_k))\right] I_k(x_i(t_k)),
\end{aligned}$$

we obtain

$$0 = -\lambda \int_0^{\omega} a_i(t)x_i^2(t)\mathrm{d}t + \lambda \int_0^{\omega} \sum_{j=1}^{m} p_{ji}(t)f_j\left(y_j\left(t-\tau_{ji}(t)\right)\right)x_i(t)\mathrm{d}t + \lambda \int_0^{\omega} c_i(t)x_i(t)\mathrm{d}t$$
$$+\lambda \sum_{k=1}^{q} \left[x_i\left(t_k\right) + \tfrac{1}{2}I_k\left(x_i\left(t_k\right)\right)\right] I_k\left(x_i\left(t_k\right)\right).$$

Write $I_i = \sum_{k=1}^{q} \left[\left|x_i\left(t_k\right)\right| + \frac{1}{2}\left|I_k\left(x_i\left(t_k\right)\right)\right|\right] \left|I_k\left(x_i\left(t_k\right)\right)\right|$. Then

$$\begin{aligned} a_i^- \int_0^{\omega} \left|x_i(t)\right|^2 \mathrm{d}t &\le \int_0^{\omega} \sum_{j=1}^{m} \left|p_{ji}(t)\right| \left|f_j\left(y_j\left(t-\tau_{ji}(t)\right)\right)\right| \left|x_i(t)\right| \mathrm{d}t \\ &\quad + \int_0^{\omega} \left|c_i(t)\right| \left|x_i(t)\right| \mathrm{d}t + I_i \\ &\le \sqrt{\omega}\left(\sum_{j=1}^{m} p_{ji}^+ \mu_j^f k_{ji} \left\|y_j\right\|_2 + c_i^+\right) \left\|x_i\right\|_2 + I_i. \end{aligned} \tag{7}$$

From (7) it follows that

$$\left\|x_i\right\|_2^2 - \frac{\sqrt{\omega}}{a_i^-} \sum_{j=1}^{m} p_{ji}^+ \mu_j^f k_{ji} \left\|y_j\right\|_2 \left\|x_i\right\|_2 + \frac{\sqrt{\omega}}{a_i^-} c_i^+ \left\|x_i\right\|_2 - \frac{I_i}{a_i^-} \le 0. \tag{8}$$

Similarly, writing $J_j = \sum_{k=1}^{q} \left[\left|y_j\left(t_k\right)\right| + \frac{1}{2}\left|J_k\left(y_j\left(t_k\right)\right)\right|\right] \left|J_k\left(y_j\left(t_k\right)\right)\right|$, we have

$$\left\|y_j\right\|_2^2 - \frac{\sqrt{\omega}}{b_j^-} \sum_{i=1}^{n} q_{ij}^+ \mu_i^g \ell_{ij} \left\|x_i\right\|_2 \left\|y_j\right\|_2 + \frac{\sqrt{\omega}}{b_j^-} d_j^+ \left\|y_j\right\|_2 - \frac{J_j}{b_j^-} \le 0. \tag{9}$$

Let $A = \max\limits_{1\le i\le n, 1\le j\le m} \left\{ \frac{\sqrt{\omega}}{a_i^-} \sum\limits_{j=1}^{m} p_{ji}^+ \mu_j^f k_{ji} + \frac{\sqrt{\omega}}{b_j^-} \sum\limits_{i=1}^{n} q_{ij}^+ \mu_i^g \ell_{ij} \right\}$,
$B = \max\limits_{1\le i\le n, 1\le j\le m} \left\{ \frac{-\sqrt{\omega}}{a_i^-} c_i^+, \frac{-\sqrt{\omega}}{b_j^-} d_j^+ \right\}$, $C = \max\limits_{1\le i\le n, 1\le j\le m} \left\{ \frac{I_i}{a_i^-}, \frac{J_j}{b_j^-} \right\}$.
Adding both (8) and (9), we obtain

$$\left(1 - \frac{A}{2}\right) \left(\left\|x_i\right\|_2 + \left\|y_j\right\|_2\right)^2 \le 2B\left(\left\|x_i\right\|_2 + \left\|y_j\right\|_2\right) + 2C.$$

Case 1. If $\left\|x_i\right\|_2 + \left\|y_j\right\|_2 < 1$, then $\left\|x_i\right\|_2$ and $\left\|y_j\right\|_2$ are bounded;
Case 2. If $\left\|x_i\right\|_2 + \left\|y_j\right\|_2 \ge 1$, then we have

$$\left(1 - \frac{A}{2}\right) \left(\left\|x_i\right\|_2 + \left\|y_j\right\|_2\right) \le 2B + 2C \quad \text{or} \quad \left\|x_i\right\|_2 + \left\|y_j\right\|_2 \le \frac{2(B+C)}{1 - \frac{A}{2}}.$$

Then

$$\left\|x_i\right\|_2 + \left\|y_j\right\|_2 \le \max\left\{1, \frac{2(B+C)}{1 - \frac{A}{2}}\right\}. \tag{10}$$

From (10), there exist R_i^* and R_{n+j}^* such that

$$\|x_i\|_2 \le R_i^*, \quad \|y_j\|_2 \le R_{n+j}^* \quad i = 1, \cdots, n, \quad j = 1, \cdots, m.$$

Since, for $t \in [0, \omega]$,

$$|x_i(t)| \le |x_i(\xi_i)| + \int_0^\omega |x_i'(s)| \mathrm{d}s, \tag{11}$$

from (5), (6) and (11), there exist $n+m$ positive constants R_ℓ such that $|x_i(t)| \le R_i$ $(i = 1, \cdots, n)$. Similarly, we have $|y_j(t)| \le R_{n+j}$ $(j = 1, \cdots, m)$. Clearly, R_ℓ $(\ell = 1, \cdots, n+m)$ are independent of λ. Denote $M^* = \sum_{\ell=1}^{n+m} R_\ell + M$, where $M > 0$ is taken sufficiently large so that

$$\begin{aligned} &\min\left\{\min_{1\le i\le n}\left(\overline{a}_i - \mu_i^g \sum_{j=1}^m \left|\overline{q}_{ij}\right|\right), \min_{1\le j\le m}\left(\overline{b}_j - \mu_j^f \sum_{i=1}^n \left|\overline{p}_{ji}\right|\right)\right\} M^* \\ &> \sum_{i=1}^n \left(\overline{c}_i - \tfrac{1}{\omega}\sum_{k=1}^q \left|I_k\left(x_i\left(t_k\right)\right)\right|\right) + \sum_{j=1}^m \left(\overline{d}_j - \tfrac{1}{\omega}\sum_{k=1}^q \left|J_k\left(y_j\left(t_k\right)\right)\right|\right). \end{aligned}$$

Now we take $\Omega = \{u(t) \in X \mid \|u\| < M^*\}$.

It is clear that Ω verifies the requirement (1) in Lemma 1. When $u \in \partial\Omega \cap \mathbb{R}^{n+m}$, u is a constant vector in $\mathbb{R}^{n+m}$ with $\|u\| = \sum_{i=1}^n |x_i| + \sum_{j=1}^m |y_j| = M^*$. Then $QNu = (E_1, \cdots, E_n, E_{n+1}, \cdots, E_{n+m})^T$, where

$$E_i = -\overline{a}_i x_i + \sum_{j=1}^m \overline{p}_{ji} f_j\left(y_j\right) + \overline{c}_i - \frac{1}{\omega}\sum_{k=1}^q I_k\left(x_i\left(t_k\right)\right), \quad i = 1, \cdots, n,$$

$$E_{n+j} = -\overline{b}_j y_j + \sum_{i=1}^n \overline{q}_{ij} g_i\left(x_i\right) + \overline{d}_j - \frac{1}{\omega}\sum_{k=1}^q J_k\left(y_j\left(t_k\right)\right), \quad j = 1, \cdots, m.$$

Therefore

$$\begin{aligned} \|QNu\| &= \sum_{i=1}^n \left|\overline{a}_i x_i - \sum_{j=1}^m \overline{p}_{ji} f_j\left(y_j\right) - \overline{c}_i + \tfrac{1}{\omega}\sum_{k=1}^q I_k\left(x_i\left(t_k\right)\right)\right| \\ &\quad + \sum_{j=1}^m \left|\overline{b}_j y_j - \sum_{i=1}^n \overline{q}_{ij} g_i\left(x_i\right) - \overline{d}_j + \tfrac{1}{\omega}\sum_{k=1}^q I_k\left(x_i\left(t_k\right)\right)\right| \\ &\ge \min_{1\le i\le n}\left(\overline{a}_i - \mu_i^g \sum_{j=1}^m \left|\overline{q}_{ij}\right|\right)|x_i| - \sum_{i=1}^n \left(\left|\overline{c}_i\right| - \tfrac{1}{\omega}\sum_{k=1}^q \left|I_k\left(x_i\left(t_k\right)\right)\right|\right) \\ &\quad + \min_{1\le j\le m}\left(\overline{b}_j - \mu_j^f \sum_{i=1}^n \left|\overline{p}_{ji}\right|\right)|y_j| \\ &\quad - \sum_{j=1}^m \left(\left|\overline{d}_j\right| - \tfrac{1}{\omega}\sum_{k=1}^q \left|J_k\left(y_j\left(t_k\right)\right)\right|\right) \\ &> 0. \end{aligned}$$

Consequently, for $u(t) \in \partial\Omega \cap \ker L$. This satisfies condition (2) of Lemma 1.

Define a continuous functions $H : \mathrm{Dom}L \times [0,1] \to X$ by $Hu = -\mu u + (1-\mu)QNu$, where $u \in \partial\Omega \cap \ker L$ is a constant vector in $\mathbb{R}^{n+m}$ and $\mu \in [0,1]$. Thus, $\|H(x_1, \cdots, x_n, y_1, \cdots, y_m, \mu)\| > 0$. As a result, we have $\deg\{QN, \Omega \cap \ker L, 0\} = \deg\{-u, \Omega \cap \ker L, 0\} \neq 0$. Condition (3) of Lemma 1 is also satisfied.

We now know that Ω satisfies all the requirements in Lemma 1. Therefore, equation (3) has at least a continuous ω periodic solutions.

This completes the proof of the theorem. □

4 Globally Exponential Stability of Periodic Solutions

According to Theorem 1, suppose $u^*(t) = (x_1^*(t), \cdots, x_n^*(t), y_1^*(t), \cdots, y_m^*(t))^T$ is a periodic solution of (2).

Theorem 2. *Assume that (H_1)—(H_3) hold. Furthermore, suppose further that*

(H_4) The following inequalities hold:

$$a_i^- - \sum_{j=1}^{m} q_{ij}^+ \mu_i^g > 0, \quad b_j^- - \sum_{i=1}^{n} p_{ji}^+ \mu_j^f > 0;$$

(H_5) The impulses operators $I_k(x_i(t_k))$ and $J_k(y_j(t_k))$ satisfy

$$I_k(x_i(t_k)) = -\gamma_{ik}(x_i(t_k)), \quad 0 < \gamma_{ik} < 2, \quad i = 1, \cdots, n, k \in \mathbb{Z}^+,$$

$$J_k(y_j(t_k)) = -\sigma_{jk}(y_j(t_k)), \quad 0 < \sigma_{jk} < 2, \quad j = 1, \cdots, m, k \in \mathbb{Z}^+.$$

Then there exists constant $\alpha > 0$ and $\beta \geq 1$ such that all solutions of (2) satisfy the inequality

$$\begin{aligned} &\sum_{i=1}^{n} |x_i(t) - x_i^*(t)| + \sum_{j=1}^{m} |y_j(t) - y_j^*(t)| \\ &\leq \beta \mathrm{e}^{-\alpha t} \left[\sum_{i=1}^{n} \sup_{s \in [-\tau, 0]} |x_i(s) - x_i^*(s)| + \sum_{j=1}^{m} \sup_{s \in [-\sigma, 0]} |y_j(s) - y_j^*(s)| \right]. \end{aligned}$$

Proof. Limited to the length of the paper, we omit the proof. □

References

1. Bainov, D.D., Simeonov, P.S.: Stability Theory of Differential Equations with Impulse Effects: Theory and Applications. Ellis Horwood, Chichester (1989)
2. Gaines, R.E., Mawhin, J.L.: Coincidence Degree and Nonlinear Differential Equations. Springer, New York (1977)
3. Gopalsamy, K., He, X.Z.: Delay-independent Stability in Bi-directional Associative Memory Networks. IEEE Trans Neural Networks 5, 998–1002 (1994)
4. Guan, Z.H., Chen, G.: On Delayed Impulsive Hopfield Neural Networks. Neural Networks 12, 273–280 (1999)

5. Gui, Z.J., Ge, W.G.: Existence and Uniqueness of Periodic Solutions of Nonautonomous Cellular Neural Networks with Impuises. Phys. Lett. A 354, 84–94 (2006)
6. Gui, Z.J., Yang, X.S., Ge, W.G.: Periodic Solution for Nonautonomous Bidirectional Associative Memory Neural Networks with Impulses. Neurocomputing 70, 2517–2527 (2007)
7. Ho, W.C., Liang, J.L., James, L.: Global Exponential Stability of Impulsive High-order BAM Neural Networks with Time-varying Delays. Neural Networks 19, 1581–1590 (2006)
8. Kosko, B.: Adaptive Bi-directional Associative Memoried. Appl. Optim. 26, 4947–4960 (1987)
9. Li, Y.K.: Existence and Stability of Periodic Solutions for Cohen-Grossberg Neural Networks with Multiple Delays. Chaos, Solitons and Fractals 20, 459–466 (2004)
10. Li, Y.: Global Exponential Stability of BAM Neural Networks with Delays and Impulses. Chaos, Solitons and Fractals 24, 279–285 (2005)
11. Li, Y.K., Lu, L.: Global Exponential Stability and Existence of Periodic Solution of Hopfield-type Neural Networks with Impulses. Phys. Lett. A 333, 62–71 (2004)
12. Liu, B.W., Huang, L.H.: Existence and Exponential Stability of Periodic Solutions for a Class of Cohen-Grossberg Neural Networks with Time-varying Delays. Chaos Solitons and Fractals 32, 617–627 (2007)
13. Liu, Z.G., Chen, A., Cao, J.D., Huang, L.H.: Existence and Global Exponential Stability of Periodic Solution for BAM Neural Networks with Periodic Coefficients and Time-varying Delays. IEEE Trans. Circuits Syst. I 50, 1162–1173 (2003)
14. Mohamad, S.: Global Exponential Stability in Continuous-time and Discrete-time Delayed Bidirectional Neural Networks. Physica D 159, 233–251 (2001)
15. Xie, W., Wen, C., Li, Z.: Impulsive Control for the stabilization and synchronization of Lorenz systems. Phys. Lett. A 275, 67–72 (2000)
16. Yang, T., Chua, L.O.: Impulsive Stabilization for Control and Synchronizatin of Chaotic Systems: Theory and Application to Secure Communication. IEEE Trans Circuits and Systems—I 44, 976–988 (1997)
17. Yang, Z.C., Xu, D.Y.: Existence and Exponential Stability of Periodic Solution for Impulsive Delay Differential Equations and Applications. Nonlinear Analysis 64, 130–145 (2006)
18. Zhu, W., Xu, D.Y.: Global Exponential Stability of Fuzzy Cellular Neural Networks with Impulses and Infinite Delays. Journal of Math R E 28, 1–10 (2008)

Impulsive Exponential Synchronization of Coupled Fuzzy Neural Networks with Time-Varying Delays

Jianting Zhou[1], Qiankun Song[2], and Jianxi Yang[3]

[1] College of Civil Engineering and Architecture, Chongqing Jiaotong University, Chongqing 400074, China
[2] Department of Mathematics, Chongqing Jiaotong University, Chongqing 400074, China
qiankunsong@163.com
[3] College of Information Science and Engineering, Chongqing Jiaotong University, Chongqing 400074, China

Abstract. In this paper, the problem on impulsive exponential synchronization is investigated for coupled fuzzy neural networks with time-varying delays. Based on M-matrix theory and analytic methods, a sufficient condition of impulsive exponential synchronization of two coupled fuzzy neural networks is established. To illustrate the effectiveness of the new scheme, a numerical example with simulation is given.

Keywords: Impulsive, Exponential synchronization, Fuzzy neural networks, Time-varying delays.

1 Introduction

During the last two decades, synchronization of chaotic dynamic systems has received a great deal of interest among scientists from various research fields [1]-[5]. Since the neural networks can exhibit chaotic behavior [6], the synchronization has received much attention for neural networks, for example, see [7]-[10] and references therein.

On the other hand, the fuzzy neural network is also a kind of important neural network [11], and its dynamical behaviors have been investigated, see [11]-[17] and references therein. In [11]-[16], authors considered the stability, periodic solutions and attracting and invariant sets for delayed fuzzy neural network. In [17], authors studied the synchronization of fuzzy cellular neural networks with constant delays. By the Lyapunov-Lasall principle of functional differential equations, authors obtained two criteria on global synchronization via adaptive control. To the best of our knowledge, there has few work studying the impulsive effects on synchronization for fuzzy neural networks with time-varying delays.

This work, inspired by the above works, addresses the exponential synchronization problem of fuzzy neural networks with time-varying delays via

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 482–491, 2009.

impulsive control approach. Based on the P-cone property, M-matrix theory and analytic methods, a sufficient condition of impulsive exponential synchronization of two coupled fuzzy neural networks is established. a numerical example is given to demonstrate the effectiveness of the presented synchronization scheme.

2 Problem Formulation and Preliminaries

In this paper, we consider the following neural network model

$$\begin{cases} \frac{du_i(t)}{dt} = -c_i u_i(t) + \sum\limits_{j=1}^{n} a_{ij} f_j(u_j(t)) + \sum\limits_{j=1}^{n} b_{ij} v_j + J_i \\ \qquad + \bigwedge\limits_{j=1}^{n} \alpha_{ij} f_j(u_j(t-\tau_{ij}(t))) + \bigvee\limits_{j=1}^{n} \beta_{ij} f_j(u_j(t-\tau_{ij}(t))) \\ \qquad + \bigwedge\limits_{j=1}^{n} T_{ij} v_j + \bigvee\limits_{j=1}^{n} H_{ij} v_j, t \neq t_k, \\ u_i(t) = p_{ik}(u_1(t^-), \cdots, u_n(t^-)) \\ \qquad +q_{ik}(u_1((t-\tau_{i1}(t))^-), \cdots, u_n((t-\tau_{in}(t))^-)) + J_{ik}, \quad t = t_k, \end{cases} \tag{1}$$

for $i = 1, 2, \cdots, n$, $k \in N$, where n corresponds to the number of units in a neural network; $u(t) = (u_1(t), u_2(t), \cdots, u_n(t))^T$, $u_i(t)$ corresponds to the state of the ith unit at time t; f_j denotes the activation function; $\tau_{ij}(t)$ corresponds to the transmission delay along the axon of the jth unit from the ith unit and satisfies $0 \leq \tau_{ij}(t) \leq \tau$ (τ is a constant); $C = diag(c_1, c_2, \cdots, c_n)$, c_i represents the rate with which the ith unit will reset its potential to the resting state in isolation when disconnected from the network and external inputs; $A = (a_{ij})_{n\times n}$, $B = (b_{ij})_{n\times n}$, a_{ij} and b_{ij} are elements of feedback template and feed forward template, respectively; $\alpha = (\alpha_{ij})_{n\times n}$, $\beta = (\beta_{ij})_{n\times n}$, α_{ij} and β_{ij} are elements of the fuzzy feedback MIN template and the fuzzy feedback MAX template, respectively; $T = (T_{ij})_{n\times n}$, $H = (H_{ij})_{n\times n}$, T_{ij} and H_{ij} are elements of fuzzy feed forward MIN template and fuzzy feed forward MAX template, respectively; $V = (v_1, v_2, \cdots, v_n)^T$, $J = (J_1, J_2, \cdots, J_n)^T$, v_i and J_i denote input and bias of the ith neuron, respectively. The second part is discrete part of model (1), which describes that the evolution processes experience abrupt change of state at the moments of time t_k (called impulsive moments), where $p_{ik}(u_1(t^-), \cdots, u_n(t^-))$ represents impulsive perturbations of the ith unit at time t_k and $u_j(t^-)$ denotes the left limit of $u_j(t)$; $q_{ik}(u_1((t-\tau_{i1}(t))^-), \cdots, u_n((t-\tau_{in}(t))^-))$ represents impulsive perturbations of the ith unit at time t_k which caused by transmission delays; J_{ik} represents external impulsive input at time t_k, the fixed moments of time t_k satisfy $t_1 < t_2 < \cdots$, $\lim_{k\to+\infty} t_k = +\infty$.

From the unidirectional linear coupling approach, a response system for (1) is constructed as follows:

$$\begin{cases} \frac{dz_i(t)}{dt} = -c_i z_i(t) + \sum\limits_{j=1}^{n} a_{ij} f_j(z_j(t)) + \sum\limits_{j=1}^{n} b_{ij} v_j + J_i \\ \qquad + \bigwedge\limits_{j=1}^{n} \alpha_{ij} f_j(z_j(t-\tau_{ij}(t))) + \bigvee\limits_{j=1}^{n} \beta_{ij} f_j(z_j(t-\tau_{ij}(t))) \\ \qquad + \bigwedge\limits_{j=1}^{n} T_{ij} v_j + \bigvee\limits_{j=1}^{n} H_{ij} v_j + d_i(z_i(t) - u_i(t)), t \neq t_k, \\ z_i(t) = p_{ik}(z_1(t^-), \cdots, z_n(t^-)) \\ \qquad +q_{ik}(z_1((t-\tau_{i1}(t))^-), \cdots, z_n((t-\tau_{in}(t))^-)) + J_{ik}, \quad t = t_k, \end{cases} \tag{2}$$

where $D = diag(d_1, d_2, \cdots, d_n)$ is a controller gain matrix to be designed later. Let $y(t) = (u_1(t) - z_1(t), u_2(t) - z_2(t), \cdots, u_n(t) - z_n(t))^T$, then $y(t)$ is the synchronization error. Therefore, the synchronization error between (1) and (2) can be expressed by

$$\begin{cases} \frac{dy_i(t)}{dt} = -(c_i + d_i) y_i(t) + \sum\limits_{j=1}^{n} a_{ij}(f_j(u_j(t)) - f_j(z_j(t))) \\ \qquad + \bigwedge\limits_{j=1}^{n} \alpha_{ij} f_j(u_j(t-\tau_{ij}(t))) - \bigwedge\limits_{j=1}^{n} \alpha_{ij} f_j(z_j(t-\tau_{ij}(t))) \\ \qquad + \bigvee\limits_{j=1}^{n} \beta_{ij} f_j(u_j(t-\tau_{ij}(t))) - \bigvee\limits_{j=1}^{n} \beta_{ij} f_j(z_j(t-\tau_{ij}(t))), t \neq t_k, \\ y_i(t) = p_{ik}(y_1(t^-), \cdots, y_n(t^-)) \\ \qquad +q_{ik}(y_1((t-\tau_{i1}(t))^-), \cdots, y_n((t-\tau_{in}(t))^-)), \quad t = t_k, \end{cases} \tag{3}$$

where $p_{ik}(y_1(t^-), \cdots, y_n(t^-)) = p_{ik}(u_1(t^-), \cdots, u_n(t^-)) - p_{ik}(z_1(t^-), \cdots, z_n(t^-))$, $q_{ik}(y_1((t-\tau_{i1}(t))^-), \cdots, y_n((t-\tau_{in}(t))^-)) = q_{ik}(u_1((t-\tau_{i1}(t))^-), \cdots, u_n((t-\tau_{in}(t))^-)) - q_{ik}(z_1((t-\tau_{i1}(t))^-), \cdots, z_n((t-\tau_{in}(t))^-))$.

To prove our results, the following lemmas that are necessary can be found in [5] and [11].

Lemma 1. *([5]) Let Q be $n \times n$ matrix with non-positive off-diagonal elements, then Q is a nonsingular M-matrix if and only if there exists a vector $\xi > 0$ such that $\xi^T Q > 0$.*

When A is a nonsingular M-matrix, denote

$$\Omega(A) = \{\xi \in R^n | A\xi > 0, \xi > 0\},$$

from Lemma 1, we know that $\Omega(A)$ is nonempty.

Lemma 2. *([5]) Let A be a nonnegative matrix, then $\rho(A)$ is a eigenvalue of A, and A has at least one positive eigenvector which is provided by $\rho(A)$.*

When A is an nonnegative matrix, denote

$$\Gamma(A) = \{\xi \in R^n | A\xi = \rho(A)\xi\},$$

from Lemma 2, we know that $\Gamma(A)$ is nonempty.

Lemma 3. *([11]) Suppose u and u' are two state of model (1), then we have*

$$\Big|\bigwedge_{j=1}^{n}\alpha_{ij}f_j(u_j)-\bigwedge_{j=1}^{n}\alpha_{ij}f_j(u'_j)\Big|\le\sum_{j=1}^{n}\Big|\alpha_{ij}\Big|\cdot\Big|f_j(u_j)-f_j(u'_j)\Big|,$$

$$\Big|\bigvee_{j=1}^{n}\beta_{ij}f_j(u_j)-\bigvee_{j=1}^{n}\beta_{ij}f_j(u'_j)\Big|\le\sum_{j=1}^{n}\Big|\beta_{ij}\Big|\cdot\Big|f_j(u_j)-f_j(u'_j)\Big|.$$

3 Main Results

Theorem 1. *Assume that*
*(***H1***) For function f_j, there exists a positive diagonal matrix $F=diag(F_1, F_2, \cdots, F_n)$ such that*

$$F_j=\sup_{x_1\neq x_2}\Big|\frac{f_j(x_1)-f_j(x_2)}{x_1-x_2}\Big|$$

for all $x_1\neq x_2$, $j=1,2,\cdots,n$.
*(***H2***) $W=C+D-(|A|+|\alpha|+|\beta|)F$ is a nonsingular $M-$matrix.*
*(***H3***) There exist nonnegative matrices $P_k=(p_{ij}^{(k)})_{n\times n}$ and $Q_k=(q_{ij}^{(k)})_{n\times n}$ such that*

$$|p_{ik}(u_1,\cdots,u_n)-p_{ik}(v_1,\cdots,v_n)|\le\sum_{j=1}^{n}p_{ij}^{(k)}|u_j-v_j|,$$

$$|q_{ik}(u_1,\cdots,u_n)-q_{ik}(v_1,\cdots,v_n)|\le\sum_{j=1}^{n}q_{ij}^{(k)}|u_j-v_j|$$

for all $(u_1,\cdots u_n)^T\in R^n,(v_1,\cdots v_n)^T\in R^n,i=1,2,\cdots,n;k=1,2,\cdots$.
*(***H4***) $\Delta=\bigcap_{k=1}^{\infty}\Big[\Gamma(P_k)\bigcap\Gamma(Q_k)\Big]\bigcap\Omega(W)$ is nonempty.*
*(***H5***) There exists a constant λ such that*

$$\frac{\ln\gamma_k}{t_k-t_{k-1}}\le\lambda<\varepsilon,\quad k=1,2,\cdots,\tag{4}$$

where the scalar $\varepsilon>0$ is determined by the inequality

$$\xi_i(\varepsilon-c_i-d_i)+\sum_{j=1}^{n}\xi_jF_j\Big(|a_{ij}|+e^{\varepsilon\tau}(|\alpha_{ij}|+|\beta_{ij}|)\Big)<0\tag{5}$$

for a given $\xi=(\xi_1,\xi_2,\cdots,\xi_n)^T\in\Delta$, and

$$\gamma_k\ge\max\{1,\rho(P_k)+e^{\varepsilon\tau}\rho(Q_k)\}.\tag{6}$$

Then the origin of (3) is globally exponentially stable, which imply that the two systems (1) and (2) are globally impulsively exponentially synchronized.

Proof. From (**H2**) and (**H4**), we know that there exists a positive vector $\xi = (\xi_1, \xi_2, \cdots, \xi_n)^T > 0$ such that

$$-\xi_i(c_i + d_i) + \sum_{j=1}^{n} \xi_j F_j \Big(|a_{ij}| + (|\alpha_{ij}| + |\beta_{ij}| \Big) < 0. \tag{7}$$

Further, we can choose a $\varepsilon > 0$ such that

$$\xi_i(\varepsilon - c_i - d_i) + \sum_{j=1}^{n} \xi_j F_j \Big(|a_{ij}| + e^{\varepsilon\tau}(|\alpha_{ij}| + |\beta_{ij}|) \Big) < 0. \tag{8}$$

Let

$$x_i(t) = e^{\varepsilon(t-t_0)}|y_i(t)|.$$

Calculating the upper right derivative $D^+x_i(t)$ of $x_i(t)$ along the solutions of (3), from Lemma 3 and the assumption (**H**1), we can get

$$\begin{aligned} D^+x_i(t) &= \varepsilon e^{\varepsilon(t-t_0)}|y_i(t)| + e^{\varepsilon(t-t_0)} sgn(y_i(t)) \Big\{ -(c_i + d_i)y_i(t) \\ &\quad + \sum_{j=1}^{n} a_{ij}\big(f_j(u_j(t)) - f_j(z_j(t))\big) \\ &\quad + \bigwedge_{j=1}^{n} \alpha_{ij} f_j(u_j(t - \tau_{ij}(t))) - \bigwedge_{j=1}^{n} \alpha_{ij} f_j(z_j(t - \tau_{ij}(t))) \\ &\quad + \bigvee_{j=1}^{n} \beta_{ij} f_j(u_j(t - \tau_{ij}(t))) - \bigvee_{j=1}^{n} \beta_{ij} f_j(z_j(t - \tau_{ij}(t))) \Big\} \\ &\leq (\varepsilon - c_i - d_i)x_i(t) + \sum_{j=1}^{n} |a_{ij}| F_j x_j(t) \\ &\quad + \sum_{j=1}^{n} (|\alpha_{ij}| + |\beta_{ij}|) F_j e^{\varepsilon\tau} x_j(t - \tau_{ij}(t)) \end{aligned} \tag{9}$$

for $i = 1, 2, \cdots, n; t_{k-1} < t < t_k, k \in N$.

For any bounded initial condition $y(s) = \phi(s) \in PC([-\tau, t_0], R^n)$ of model (3), let $l_0 = \frac{\|\phi\|}{\min\limits_{1 \leq i \leq n} \{\xi_i\}}$, then

$$x_i(s) = e^{\varepsilon(s-t_0)}|y_i(s)| \leq |y_i(s)| = |\phi_i(s)| \leq \|\phi\| \leq \xi_i l_0, \quad s \in [-\tau, t_0] \tag{10}$$

for $i = 1, 2, \cdots, n$. In following, we prove that

$$x_i(t) \leq \xi_i l_0, \quad t_0 \leq t < t_1, \quad i = 1, 2, \cdots, n, \tag{11}$$

hold. In fact, if inequality (11) is not true, then there must exist some i and $t^* \in [t_0, t_1)$ such that

$$x_i(t^*) = \xi_i l_0, \quad D^+x_i(t^*) \geq 0, \quad x_j(t) \leq \xi_j l_0, t \in [-\tau, t^*], j = 1, 2, \cdots, n.$$

However, from (8) and (9), we get

$$D^+x_i(t^*) \le \Big((\varepsilon - c_i - d_i)\xi_i + \sum_{j=1}^{n}|a_{ij}|F_j\xi_j + \sum_{j=1}^{n}(|\alpha_{ij}| + |\beta_{ij}|)F_j e^{\varepsilon\tau}\xi_j\Big)l_0 < 0,$$

this is a contradiction. So inequality (11) is true. Thus, we have

$$|y_i(t)| \le \xi_i l_0 e^{-\varepsilon(t-t_0)}, \quad t_0 \le t < t_1, \quad i = 1, 2, \cdots, n. \tag{12}$$

In the following, we will use the mathematical induction to prove that

$$|y_i(t)| \le \gamma_0\gamma_1\cdots\gamma_{k-1}\xi_i l_0 e^{-\varepsilon(t-t_0)}, \quad t_{k-1} \le t < t_k, \quad k \in N, \tag{13}$$

hold for $i = 1, 2, \cdots, n$, where $\gamma_0 = 1$.

When $k = 1$, from inequality (12) we know that inequality (13) hold.

Suppose that the inequalities

$$|y_i(t)| \le \gamma_0\gamma_1\cdots\gamma_{m-1}\xi_i l_0 e^{-\varepsilon(t-t_0)}, \quad t_{k-1} \le t < t_k, \quad i = 1, 2, \cdots, n, \tag{14}$$

hold for $k = 1, 2, \cdots, m$. From assumption (**H**3) and (14), the discrete part of model (3) satisfies that

$$\begin{aligned}
|y_i(t_m)| &\le \sum_{j=1}^{n} p_{ij}^{(m)}|y_j(t_m^-)| + \sum_{j=1}^{n} q_{ij}^{(m)}|y_j((t_m - \tau_{ij}(t_m))^-)| \\
&\le \sum_{j=1}^{n} p_{ij}^{(m)}\gamma_0\gamma_1\cdots\gamma_{m-1}\xi_j l_0 e^{-\varepsilon(t_m - t_0)} \\
&\quad + \sum_{j=1}^{n} q_{ij}^{(m)}\gamma_0\gamma_1\cdots\gamma_{m-1}\xi_j l_0 e^{-\varepsilon(t_m - \tau_{ij}(t_m) - t_0)} \\
&\le \Big(\sum_{j=1}^{n} p_{ij}^{(m)}\xi_j + e^{\varepsilon\tau}\sum_{j=1}^{n} q_{ij}^{(m)}\xi_j\Big)\gamma_0\gamma_1\cdots\gamma_{m-1}l_0 e^{-\varepsilon(t_m - t_0)}
\end{aligned}$$

for $i = 1, 2, \cdots, n$. From $\xi = (\xi_1, \xi_2, \cdots, \xi_n)^T \in \Delta$ and Lemma 2, we know that $\xi \in \Gamma(P_m)$ and $\xi \in \Gamma(Q_m)$, thus

$$P_m\xi = \rho(P_m)\xi, \qquad Q_m\xi = \rho(Q_m)\xi,$$

i.e.,

$$\sum_{j=1}^{n} p_{ij}^{(m)}\xi_j = \rho(P_m)\xi_i, \qquad \sum_{j=1}^{n} q_{ij}^{(m)}\xi_j = \rho(Q_m)\xi_i, \qquad i = 1, 2, \cdots, n. \tag{15}$$

From (14), (15) and (6), we get

$$\begin{aligned}
|y_i(t_m)| &\le \Big(\rho(P_m) + e^{\varepsilon\tau}\rho(Q_m)\Big)\gamma_0\gamma_1\cdots\gamma_{m-1}\xi_i l_0 e^{-\varepsilon(t_m - t_0)} \\
&\le \gamma_0\gamma_1\cdots\gamma_{m-1}\gamma_m\xi_i l_0 e^{-\varepsilon(t_m - t_0)}
\end{aligned} \tag{16}$$

for $i = 1, 2, \cdots, n$. This, together with (13), lead to

$$|y_i(t)| \leq \gamma_0\gamma_1 \cdots \gamma_{m-1}\gamma_m \xi_i l_0 e^{-\varepsilon(t-t_0)}, \quad i = 1, 2, \cdots, n; t \in [t_m - \tau, t_m], \quad (17)$$

i.e.,

$$x_i(t) \leq \gamma_0\gamma_1 \cdots \gamma_{m-1}\gamma_m \xi_i l_0, \quad i = 1, 2, \cdots, n; t \in [t_m - \tau, t_m]. \quad (18)$$

In the following, we will prove that

$$x_i(t) \leq \gamma_0\gamma_1 \cdots \gamma_{m-1}\gamma_m \xi_i l_0, \quad i = 1, 2, \cdots, n; t \in [t_m, t_{m+1}) \quad (19)$$

hold. In fact, if (19) is not true, then there exist some i and $t^{**} \in [t_m, t_{m+1})$ such that

$$x_i(t^{**}) = \gamma_0\gamma_1 \cdots \gamma_{m-1}\gamma_m \xi_i l_0, D^+ x_i(t^{**}) \geq 0 \text{and} x_j(t) \leq \gamma_0\gamma_1 \cdots \gamma_{m-1}\gamma_m \xi_i l_0$$

for $t_m - \tau < t \leq t^{**}$, $j = 1, 2, \cdots, n$. However, from (8), (9) and (18) we get

$$\begin{aligned} D^+ x_i(t^{**}) \leq & \Big[(\varepsilon - c_i - d_i)\xi_i + \sum_{j=1}^{n} |a_{ij}| F_j \xi_j \\ & + e^{\varepsilon\tau} \sum_{j=1}^{n} (|\alpha_{ij}| + |\beta_{ij}|) F_j \xi_j \Big] \gamma_0\gamma_1 \cdots \gamma_{m-1}\gamma_m l_0 \\ < & 0, \end{aligned}$$

this is a contradiction. So (19) holds. By the mathematical induction, we can conclude that (13) holds. From (4), we have

$$\gamma_k \leq e^{\lambda(t_k - t_{k-1})}, \qquad k \in N.$$

From (13), we get

$$\begin{aligned} |y_i(t)| &\leq \frac{\xi_i}{\min\limits_{1\leq i\leq n}\{\xi_i\}} \|\phi\| e^{\lambda(t_{k-1}-t_0)} e^{-\varepsilon(t-t_0)} \\ &\leq \frac{\xi_i}{\min\limits_{1\leq i\leq n}\{\xi_i\}} \|\phi\| e^{-(\varepsilon-\lambda)(t-t_0)} \end{aligned}$$

for any $t \in [t_{k-1}, t_k), k \in N$, which imply

$$|y_i(t)| \leq \frac{\xi_i}{\min\limits_{1\leq i\leq n}\{\xi_i\}} \|\phi\| e^{-(\varepsilon-\lambda)(t-t_0)}, \quad t \geq t_0.$$

Thus

$$\|y(t)\| \leq M\|\phi\| e^{-(\varepsilon-\lambda)(t-t_0)}, \quad t \geq t_0,$$

where $M = \max\limits_{1\leq i\leq n}\{\xi_i\} / \min\limits_{1\leq i\leq n}\{\xi_i\} \geq 1$. So the origin of model (3) is globally exponentially stable, which imply that the two systems (1) and (2) are globally impulsively exponentially synchronized, and the exponential convergence rate equals $\varepsilon - \lambda$. The proof is completed.

Remark 1. In [17], authors studied the synchronization of non-impulsive fuzzy cellular neural networks with constant delays by the Lyapunov method. In this paper, the synchronization of impulsive fuzzy cellular neural networks with time-varying delays is discussed by M-matrix theory and analytic methods.

Remark 2. It should be pointed out that when the time-varying delay is not differentiable, it may be difficult to analyze the exponential synchronization of fuzzy cellular neural networks by the provided methods in [17].

4 An Example

Consider the following fuzzy delayed neural networks with two neurons

$$\frac{du_i(t)}{dt} = -c_i u_i(t) + \sum_{j=1}^{n} a_{ij} f_j(u_j(t)) + \bigwedge_{j=1}^{n} \alpha_{ij} f_j(u_j(t-\tau_{ij}(t))) + \bigvee_{j=1}^{n} \beta_{ij} f_j(u_j(t-\tau_{ij}(t))), \quad i = 1, 2, \tag{20}$$

where

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad A = \begin{bmatrix} 2.1 & -0.1 \\ -5.2 & 4.6 \end{bmatrix}, \quad \alpha = \begin{bmatrix} -1.5 & -0.1 \\ -0.2 & -4.4 \end{bmatrix}, \quad \beta = \begin{bmatrix} -1.7 & -0.1 \\ 0.6 & -3.9 \end{bmatrix},$$

$$f_i(u) = \tanh(z), \quad \tau_{ij}(t) = 1, \quad i, j = 1, 2.$$

The fuzzy delayed neural networks (20) exhibits a chaotic behavior with the initial value $x_1(s) = -0.4$, $x_2(s) = 2$, $s \in [-1, 0]$, see Figure 1.

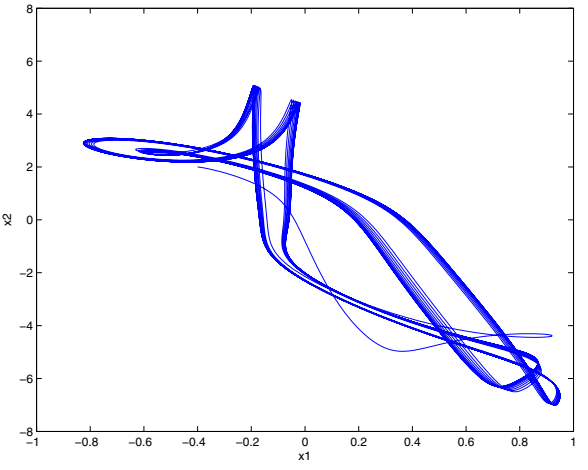


Fig. 1. The chaotic behavior of system (20)

Now the impulsive response system is designed as follows:

$$\begin{cases} \frac{dz_i(t)}{dt} = -c_i z_i(t) + \sum\limits_{j=1}^{n} a_{ij} f_j(z_j(t)) \\ \qquad + \bigwedge\limits_{j=1}^{n} \alpha_{ij} f_j(z_j(t-\tau_{ij}(t))) + \bigvee\limits_{j=1}^{n} \beta_{ij} f_j(z_j(t-\tau_{ij}(t))) \\ \qquad + d_i(z_i(t) - u_i(t)), t \neq t_k, \\ z_i(t) = p_{ik}(z_1(t^-), \cdots, z_n(t^-)) \\ \qquad + q_{ik}(z_1((t-\tau_{i1}(t))^-), \cdots, z_n((t-\tau_{in}(t))^-)), \quad t = t_k, \end{cases} \tag{21}$$

where

$$p_{1k}(z_1, z_2) = 0.08e^{0.05k} z_1 - 0.028e^{0.05k} z_2, \quad q_{1k}(z_1, z_2) = 0.2e^{0.05k} z_1,$$

$$p_{2k}(z_1, z_2) = 0.05e^{0.05k} z_1 + 0.03e^{0.05k} z_2, \quad q_{2k}(z_1, z_2) = 0.2e^{0.05k} z_2,$$

$$d_1 = 4.9, \quad d_2 = 21.5, \quad t_k = t_{k-1} + k, \quad k \in N.$$

It can be verified that assumptions (**H1**) and (**H3**) are satisfied, and $F = \text{diag}\{1, 1\}$, $P_k = e^{0.05k} \begin{pmatrix} 0.08 & 0.028 \\ 0.05 & 0.03 \end{pmatrix}$, $Q_k = 0.2e^{0.05k} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.

It is easily computing that $W = \begin{pmatrix} 0.6 & -0.3 \\ -6 & 9.6 \end{pmatrix}$ is a nonsingular M-matrix, and $\rho(P_k) = 0.1e^{0.05k}$, $\rho(Q_k) = 0.2e^{0.05k}$, $\Gamma(P_k) = \{(\xi_1, \xi_2)^T | \xi_1 = 1.4\xi_2, \xi_1 > 0, \xi_2 > 0\}$, $\Gamma(Q_k) = \{(\xi_1, \xi_2)^T | \xi_1 > 0, \xi_2 > 0\}$, $\Omega(W) = \{(\xi_1, \xi_2)^T | 0.5\xi_2 < \xi_1 < 1.6\xi_2, \xi_1 > 0, \xi_2 > 0\}$. So $\Delta = \{(\xi_1, \xi_2)^T | \xi_1 = 1.4\xi_2, \xi_1 > 0, \xi_2 > 0\}$ is non-empty.

Taking $\xi = (1.4, 1)^T \in \Delta$, from inequality (5) we can get that a maximum value of ε is 0.05413.

Take $\gamma_k = e^{0.05k}$, $\lambda = 0.05$, then

$$\gamma_k \geq \max\{1, 0.1e^{0.05k} + 0.2e^{0.05k} e^{0.05413}\}, \quad k = 1, 2, \cdots,$$

and

$$\frac{\ln \gamma_k}{t_k - t_{k-1}} = \frac{\ln e^{0.05k}}{k} = \lambda < \varepsilon, \quad k = 1, 2, \cdots,$$

Clearly, all conditions of Theorem 1 are satisfied. From Theorem 1, we know that two systems (20) and (21) are globally impulsively exponentially synchronized, and the exponential convergence rate equals 0.00413.

5 Conclusions

In this paper, the impulsive exponential synchronization has been investigated for coupled fuzzy neural networks with time-varying delays. Based on M-matrix theory and analytic methods, a new sufficient condition of impulsive exponential synchronization of two coupled fuzzy neural networks has been established. An example with simulation was also given to show the effectiveness of the obtained result.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants 50608072 and 50878219, and in part by Natural Science Foundation Project of CQ CSTC2008BA6038 and 2008BB2351.

References

1. Pecora, L.M., Carroll, T.L.: Synchronization in chaotic systems. Physical Review Letters 64, 821–824 (1990)
2. Lu, W.L., Chen, T.P.: Synchronization analysis of linearly coupled networks of discrete time systems. Physica D 198, 148–168 (2004)
3. Liu, Y.R., Wang, Z.D., Liu, X.H.: Exponential synchronization of complex networks with Markovian jump and mixed delays. Physics Letters A 372, 3986–3998 (2008)
4. Liu, X.W., Chen, T.P.: Synchronization analysis for nonlinearly-coupled complex networks with an asymmetrical coupling matrix. Physica A 387, 4429–4439 (2008)
5. Zhu, W., Xu, D.Y., Huang, Y.M.: Global impulsive exponential synchronization of time-delayed coupled chaotic systems. Chaos, Solitons and Fractals 35, 904–912 (2008)
6. Sompolinsky, H., Crisanti, A.: Chaos in random neural networks. Physical Review Letters 61, 259–262 (1988)
7. Lu, W.L., Chen, T.P.: Synchronization of coupled connected neural networks with delays. IEEE Transactions on Circuits and Systems I 51, 2491–2503 (2004)
8. Cao, J.D., Li, P., Wang, W.W.: Global synchronization in arrays of delayed neural networks with constant and delayed coupling. Physics Letters A 353, 318–325 (2006)
9. Wang, Z.S., Zhang, H.G., Wang, Z.L.: Global synchronization of a class of chaotic neural networks. Acta Physica Sinica 55, 2687–2693 (2006)
10. Li, P., Cao, J.D., Wang, Z.D.: Robust impulsive synchronization of coupled delayed neural networks with uncertainties. Physica A 373, 261–272 (2007)
11. Yang, T., Yang, L.: The global stability of fuzzy cellular neural networks. IEEE Transactions on Circuits and Systems I 43, 880–883 (1996)
12. Liu, Y.Q., Tang, W.S.: Exponential stability of fuzzy cellular neural networks with constant and time-varying delays. Physics Letters A 323, 224–233 (2004)
13. Chen, L., Zhao, H.Y.: Stability analysis of stochastic fuzzy cellular neural networks with delays. Neurocomputing 72, 436–444 (2008)
14. He, D.H., Xu, D.Y.: Attracting and invariant sets of fuzzy Cohen-Grossberg neural networks with time-varying delays. Physics Letters A 372, 7057–7062 (2008)
15. Huang, T.W., Huang, Y., Li, C.D.: Stability of periodic solution in fuzzy BAM neural networks with finite distributed delays. Neurocomputing 71, 3064–3069 (2008)
16. Liu, Z.W., Zhang, H.G., Wang, Z.S.: Novel stability criterions of a new fuzzy cellular neural networks with time-varying delays. Neurocomputing 72, 1056–1064 (2009)
17. Ding, W., Han, M.A.: Synchronization of delayed fuzzy cellular neural networks based on adaptive control. Physics Letters A 372, 4674–4681 (2008)

A Delay Fractioning Approach to Global Synchronization of Complex Networks with Distributed Delays and Stochastic Disturbances

Quanxin Cheng, Haibo Bao, and Jinde Cao

Department of Mathematics, Southeast University, Nanjing 210096, China
chengqx168@seu.edu.cn

Abstract. In this paper, the global synchronization problem is studied for a class of complex networks. This is the first time that both the distributed delays and the stochastic disturbances are considered at the same time. Based on the idea of 'delay fractioning', a sufficient condition which ensures the complex system to be globally synchronized is derived by referring to the Lyapunov functional method and the properties of Kronecker product. The condition, which is expressed in terms of linear matrix inequalities (LMIs), can be solved efficiently by the LMI toolbox in Matlab. The result obtained in this paper is proved to be much less conservative due to the fact that the delay upper bound is greatly enlarged.

Keywords: Global asymptotic synchronization, Lyapunov functional, Delay fractioning, Stochastic disturbance, Distributed delay.

1 Introduction

In the past decades, complex dynamical networks have become a popular research subject and are attracting more and more attention from many fields of scientific research [1-4]. Examples of complex networks include the wold wide web, the Internet, the electrical powder grids, food webs, biological neural networks, telephone cell graphs, coauthorship and citation networks of scientists, etc. The universality of complex networks naturally stimulates the current intensive study of the subject. In this study, one of the basic and significant characteristic is the synchronization of all dynamical nodes in a complex network [5-11]. In fact, synchronization is a ubiquitous phenomenon in nature. Loosely speaking, if two systems have something in common, a synchronization may occur between them when they are interacted.

Since a neural network usually has a spatial nature due to the parallel pathways of a variety of axon sizes and lengths, it is desirable to model them by introducing the distributed delays, see Refs. [12,13] and the references cited therein for example. On the other hand, it is worth pointing out that, uncertain components such as time delays, parameter uncertainties and noises are ubiquitous in both nature and man-made systems, among them, the stochastic disturbance effects on complex networks have drawn particular attention. In Refs.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 492–502, 2009.

[8,9,14,15,16], the synchronization problems have been intensively investigated for delayed complex networks with stochastic perturbation, where the criteria ensuring the synchronization among networks have been achieved mainly on the basis of Lyapunov approach that is capable of coping the time-delays. Therefore, one of the main aims is how to reduce the possible conservatism induced by the introduction of the Lyapunov functional. Recently, the so-called 'delay fractioning' approach has been developed in [17] which is shown to lead to much less conservative results than the most existing ones. To the best of our knowledge, so far, the synchronization problem for complex networks with *distributed delays* is still remaining as a challenging open problem.

Motivated by the above discussion, based on the Lyapunov functional method, the properties of Kronecker product and the stochastic analysis techniques combined with the 'delay fractioning' approach [17], our main aim in this paper is to shorten this gap by investigating the distributed delayed complex systems with stochastic disturbances, the derived criteria in this paper are expressed in terms of LMIs [18]. Our results are shown to be less conservative since the conservatism could be reduced by adding the number of delay fractions.

The rest of this paper is organized as follows. In section 2, a stochastic complex network model with distributed delays is introduced and some preliminaries are briefly outlined. In section 3, by utilizing the approach of 'delay fractioning' and the Lyapunov functional method, our main result which made the global synchronization realized is derived. Finally in section 4, this paper is concluded.

Notations: Throughout this paper, $P > 0$ means that matrix P is real, symmetric and positive definite. I and O denote the identity matrix and the zero matrix with compatible dimensions, respectively. The Kronecker product of matrices $Q \in \mathbb{R}^{m\times n}$ and $R \in \mathbb{R}^{p\times q}$ is a matrix in $\mathbb{R}^{mp\times nq}$ and denoted as $Q \otimes R$. We let $\tau > 0$ and $C([-\tau, 0]; \mathbb{R}^n)$ denote the family of continuous functions φ from $[-\tau, 0]$ to $\mathbb{R}^n$ with the norm $|\varphi| = \sup_{-\tau\leqslant\theta\leqslant 0} \|\varphi(\theta)\|$, where $\|\cdot\|$ is the Euclidean norm in $\mathbb{R}^n$. Moreover, let $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t\geqslant 0}, \mathcal{P})$ be a complete probability space with a filtration $\{\mathcal{F}_t\}_{t\geqslant 0}$ satisfying the usual conditions (i.e. the filtration contains all $\mathcal{P}$-null sets and is right continuous). Denote by $L^p_{\mathcal{F}_0}([-\tau, 0]; \mathbb{R}^n)$ the family of all $\mathcal{F}_0$-measurable $C([-\tau, 0]; \mathbb{R}^n)$-valued random variables $\xi = \{\xi(\theta) : -\tau \leqslant \theta \leqslant 0\}$ such that $\sup_{-\tau\leqslant\theta\leqslant 0} \mathbb{E}|\xi(\theta)|^p < \infty$,where $\mathbb{E}\{\cdot\}$ stands for the mathematical expectation operator with respect to the given probability measure $\mathcal{P}$.

2 Problem Formulation and Preliminaries

Consider the following array of identical complex networks with distributed delays and stochastic disturbances:

$$dx_i(t) = [Ax_i(t) + B\int_{-\tau}^{0} f(x_i(t+s))ds + \sum_{j=1}^{N} G_{ij}^{(1)}\Gamma_1 x_j(t) + \sum_{j=1}^{N} G_{ij}^{(2)}\Gamma_2 x_j(t-\tau)]dt + \sigma_i(t, x_i(t), x_i(t-\tau))dw(t) \quad (1)$$

where $i = 1, 2, \ldots, N$ and $x_i(t) = (x_{i1}(t), x_{i2}(t), \cdots, x_{in}(t))^T \in \mathbb{R}^n$ is the state vector of the ith network at time t; A denotes a known connection matrix; B denote the connection weight matrix; Γ_1, $\Gamma_2 \in \mathbb{R}^{n\times n}$ are matrices describing the inner-coupling between the subsystems at time t and $t-\tau$, respectively; $G^{(1)} = (G^{(1)}_{(ij)})_{(N\times N)}$ and $G^{(2)} = (G^{(2)}_{(ij)})_{(N\times N)}$ are the outer-coupling configuration matrices representing the coupling strength and the topological structure of the complex networks. The constant τ stands for the constant time delay, which satisfies $0 \leqslant \tau \leqslant h$. Furthermore, $\sigma_i(\cdot,\cdot,\cdot) : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ is the noise intensity function vector, and $\omega(t)$ is a scalar Brownian motions defined on $(\Omega, \mathcal{F}, \mathcal{P})$ satisfying

$$\mathbb{E}\{d\omega(t)\} = 0 \quad \text{and} \quad \mathbb{E}\{[d\omega(t)]^2\} = dt. \tag{2}$$

Finally, $f(x_i(t)) = (f_1(x_{i1}(t)), f_2(x_{i2}(t)), \cdots, f_n(x_{in}(t)))^T$ is an unknown but sector-bounded nonlinear function.

Throughout this letter,the following assumptions are needed.

Assumption 1. *(See [19]) The outer-coupling configuration matrices of the complex networks (1) satisfy*

$$G^{(q)}_{ij} = G^{(q)}_{ji} \geqslant 0 \ (i \neq j), \ G^{(q)}_{ii} = -\sum_{j=1, j\neq i}^{N} G^{(q)}_{ij} \ (q = 1, 2; i, j = 1, 2, \cdots, N). \tag{3}$$

Assumption 2. *(See [8]) For $\forall u, v \in \mathbb{R}^n$,the nonlinear function $f(\cdot)$ is assumed to satisfy the following sector bounded condition*

$$(f(u) - f(v) - L_f(u-v))^T (f(u) - f(v) - L^f(u-v)) \leqslant 0, \tag{4}$$

where L_f and L^f are real constant matrices with $L^f - L_f$ being symmetric and positive definite.

Assumption 3. *The noise intensity function vector $\sigma_i : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ satisfies the Lipschitz condition, i.e., there exist constant matrices W_1 and W_2 of appropriate dimensions such that the following inequality*

$$\begin{aligned}(\sigma_i(t, u_1, v_1) - \sigma_j(t, u_2, v_2))^T &(\sigma_i(t, u_1, v_1) - \sigma_j(t, u_2, v_2)) \\ &\leqslant \|W_1(u_1 - u_2)\|^2 + \|W_2(v_1 - v_2)\|^2\end{aligned} \tag{5}$$

holds for all $i, j = 1, 2, \cdots, N$ and $u_1, u_2, v_1, v_2 \in \mathbb{R}^n$.

Let $x(t) = (x_1^T(t), x_2^T(t), \cdots, x_N^T(t))^T$, $F(x(t)) = (\int_{-\tau}^0 f^T(x_1(t+s))ds$, $\int_{-\tau}^0 f^T(x_2(t+s))ds, \cdots, \int_{-\tau}^0 f^T(x_N(t+s))ds)^T$, $\sigma(t) = (\sigma_1^T(t, x_1(t), x_1(t-\tau))$, $\sigma_2^T(t, x_2(t), x_2(t-\tau)), \cdots, \sigma_N^T(t, x_N(t), x_N(t-\tau)))^T$; with the Kronecker product "$\otimes$" for matrices, system (1) can be recast into

$$\begin{aligned}dx(t) = [&(I_N \otimes A + G^{(1)} \otimes \Gamma_1)x(t) \\ &+ (G^{(2)} \otimes \Gamma_2)x(t-\tau) + (I_N \otimes B)F(x(t))]dt + \sigma(t)dw(t).\end{aligned} \tag{6}$$

The initial conditions associated with system (1) are given by

$$x_i(s) = \varphi_i(s), \qquad -h \leqslant s \leqslant 0, \quad i = 1, 2, \cdots, N \tag{7}$$

where $\varphi_i(\cdot) \in L^2_{\mathcal{F}_0}([-h, 0], \mathbb{R}^n)$, and the corresponding state trajectory is denoted as $x_i(t, \varphi_1, \varphi_2, \cdots \varphi_N)$.

Definition 1. *The set* $S = \{x = (x_1(s), x_2(s), \ldots, x_N(s)) : x_i(s) \in L^2_{\mathcal{F}_0}([-h, 0], \mathbb{R}^n), x_i(s) = x_j(s), 1 \leqslant i, j \leqslant N\}$ *is called the synchronization manifold of network (1) or (6).*

Definition 2. *The synchronization manifold* S *is said to be globally asymptotically stable in the mean square (in order words,the delayed complex network (1) is globally asymptotically synchronized in the mean square) if, for all* $\varphi_i(\cdot), \varphi_j(\cdot) \in L^2_{\mathcal{F}_0}([-h, 0], \mathbb{R}^n)$*, the following holds:*

$$\lim_{t \to +\infty} \mathbb{E}\|x_i(t, \varphi_i) - x_j(t, \varphi_j)\|^2 = 0, \qquad 1 \leqslant i < j \leqslant N. \tag{8}$$

Lemma 1. *(See [14])The Kronecker product has the following properties:*

(1) $(\alpha A) \otimes B = A \otimes (\alpha B)$;
(2) $(A + B) \otimes C = A \otimes C + B \otimes C$;
(3) $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$;
(4) $(A \otimes B)^T = A^T \otimes B^T$.

Lemma 2. *(See [20]) For scalar* $r > 0$*, let* $M \in \mathbb{R}^{m \times m}$ *be a positive semi-definite matrix and* $\rho : [0, r] \to \mathbb{R}^m$ *be a vector function. If the integration concerned are well defined, then the following inequality holds:*

$$r \int_0^r \rho^T(s) M \rho(s) ds \geqslant (\int_0^r \rho(s) ds)^T M (\int_0^r \rho(s) ds).$$

Lemma 3. *Let* $U = (\alpha_{ij})_{N \times N}$, $P \in \mathbb{R}^{n \times n}$, $x = (x_1^T, x_2^T, \cdots, x_N^T)^T$ *where* $x_i = (x_{i1}, x_{i2}, \cdots, x_{in})^T \in \mathbb{R}^n$ *and* $y = (y_1^T, y_2^T, \cdots, y_N^T)^T$ *where* $y_i = (y_{i1}, y_{i2}, \cdots, y_{in})^T \in \mathbb{R}^n$ $(i = 1, 2, \cdots, N)$*. If* $U = U^T$ *and each row sum of* U *is zero, then*

$$x^T (U \otimes P) y = - \sum_{1 \leqslant i < j \leqslant N} \alpha_{ij} (x_i - x_j)^T P (y_i - y_j).$$

3 Main Results and Proofs

In this section, we are in the position to present our main results for synchronization of the delayed complex networks with stochastic disturbances.

Theorem 1. *Consider the complex network (1) with time-delay* $\tau \in (0, h]$*. For a given integer* $r \geqslant 1$*, if there exist* $n \times n$ *matrices* $P_k > 0$, $R_k > 0$, $Q_k > 0$, M_k,

S and positive scalars λ, ϵ_k $(k = 1, 2)$ such that the following LMIs hold for all $1 \leqslant i < j \leqslant N$:

$$P_1 < \lambda I, \tag{9}$$

$$\Theta_{ij} = W_1 - W_2 - W_2^T + W_3 + W_4 + W_5 < 0, \tag{10}$$

where $W_3 = \mathrm{diag}\{Q_1, Q_2 - Q_1, -Q_2, R_1, R_2 - R_1, -R_2, -2\epsilon_1 I_n, -2\epsilon_2 I_n, -\frac{2}{h}P_2, -\frac{2}{h}P_2, hP_2 - S - S^T\}$,

$$W_1 = \begin{bmatrix} \mathbb{M} + \mathbb{M}^T & O_{2n\times 9n} \\ O_{9n\times 2n} & O_{9n\times 9n} \end{bmatrix}, W_2 = \begin{bmatrix} & \mathbb{M} & & \mathbb{M} & \\ O_{11n\times n} & & O_{11n\times 5n} & & O_{11n\times n} \\ & O_{9n\times 2n} & & O_{9n\times 2n} & \end{bmatrix},$$

$$W_4 = \begin{bmatrix} -\epsilon \otimes \hat{L} & & \epsilon \otimes \check{L} & \\ O_{4n\times 2n} & O_{11n\times 4n} & & O_{11n\times 3n} \\ * & & O_{9n\times 2n} & \\ O_{3n\times 2n} & & & \end{bmatrix}, W_5 = \begin{bmatrix} H_1 & O_{4n\times 6n} & H_2 \\ O_{6n\times 4n} & & \\ H_2^T & O_{7n\times 6n} & O_{7n\times n} \end{bmatrix}$$

and $\mathbb{M} = diag\{M_1, M_2\}$, $\epsilon = diag\{\epsilon_1, \epsilon_2\}$, $\hat{L} = L_f^T L^f + {L^f}^T L_f$, $\check{L} = L_f^T + {L^f}^T$,

$$H_1 = \begin{bmatrix} \Lambda_{ij} & O_{n\times n} & -NG_{ij}^{(2)} P_1 \Gamma_1 & P_1 B \\ O_{n\times n} & O_{n\times n} & O_{n\times n} & O_{n\times n} \\ * & O_{n\times n} & \lambda W_2^T W_2 & O_{n\times n} \\ * & O_{n\times n} & O_{n\times n} & O_{n\times n} \end{bmatrix}, \quad H_2 = \begin{bmatrix} A^T S^T - NG_{ij}^{(1)} \Gamma_1 S^T \\ O_{n\times n} \\ -NG_{ij}^{(2)} \Gamma_2^T S^T \\ B^T S^T \end{bmatrix}_{4n\times n}$$

$\Lambda_{ij} = P_1 A + A P_1^T - NG_{ij}^{(1)}(P_1\Gamma_1 + \Gamma_1^T P_1) + \lambda W_1^T W_1$;

then the asymptotic synchronization in the mean square for (8) is achieved.

Proof. By setting

$$y(t) = (I_N \otimes A + G^{(1)} \otimes \Gamma_1)x(t) + (G^{(2)} \otimes \Gamma_2)x(t-\tau) + (I_N \otimes B)F(x(t)) \tag{11}$$

system (6) becomes

$$dx(t) = y(t)dt + \sigma(t)d\omega(t).$$

Based on the delay-fractioning idea, we introduce the following new matrix functional candidate for the complex network (1) or (6):

$$V(t) = V_1(t) + V_2(t) + V_3(t) + V_4(t), \tag{12}$$

where

$$V_1(t) = x^T(t)(U \otimes P_1)x(t), \qquad V_2(t) = \int_{-\tau}^{0}\int_{t+\theta}^{t} y^T(s)(U \otimes P_2)y(s)dsd\theta,$$

$$V_3(t) = \int_{t-\frac{\tau}{2}}^{t} x^T(s)(U \otimes Q_1)x(s)ds + \int_{t-\tau}^{t-\frac{\tau}{2}} x^T(s)(U \otimes Q_2)x(s)ds,$$

$$V_4(t) = \int_{t-\frac{\tau}{2}}^{t} F^T(x(s))(U \otimes R_1)F(x(s))ds + \int_{t-\tau}^{t-\frac{\tau}{2}} F^T(x(s))(U \otimes R_2)F(x(s))ds$$

and

$$U = \begin{bmatrix} N-1 & -1 & \cdots & -1 \\ -1 & N-1 & \cdots & -1 \\ \cdots & \cdots & \cdots & \cdots \\ -1 & -1 & \cdots & N-1 \end{bmatrix}.$$

Letting $\mathcal{L}$ be the weak infinitesimal operator of the stochastic process $\{x_t = x(t+s) \mid t \geqslant 0, -h \leqslant s \leqslant 0\}$ along the trajectories of the network (1)(or(6)), then one has

$$\begin{aligned} \mathcal{L}V_1(t) &= 2x^T(t)(U \otimes P_1)[(I_N \otimes A + G^{(1)} \otimes \Gamma_1)x(t) + (G^{(2)} \otimes \Gamma_2)x(t-\tau) \\ &\quad + (I_N \otimes B)F(x(t))] + \sigma^T(t)(U \otimes P_1)\sigma(t), \end{aligned} \tag{13}$$

$$\begin{aligned} \mathcal{L}V_2(t) &= \tau y^T(t)(U \otimes P_2)y(t) - \int_{t-\tau}^{t} y^T(s)(U \otimes P_2)y(s)ds \\ &\leqslant h y^T(t)(U \otimes P_2)y(t) - \int_{t-\tau}^{t} y^T(s)(U \otimes P_2)y(s)ds, \end{aligned} \tag{14}$$

$$\begin{aligned} \mathcal{L}V_3(t) &= \begin{bmatrix} x(t) \\ x(t-\frac{\tau}{2}) \end{bmatrix}^T \begin{bmatrix} U \otimes Q_1 & \\ & U \otimes Q_2 \end{bmatrix} \begin{bmatrix} x(t) \\ x(t-\frac{\tau}{2}) \end{bmatrix} \\ &\quad - \begin{bmatrix} x(t-\frac{\tau}{2}) \\ x(t-\tau) \end{bmatrix}^T \begin{bmatrix} U \otimes Q_1 & \\ & U \otimes Q_2 \end{bmatrix} \begin{bmatrix} x(t-\frac{\tau}{2}) \\ x(t-\tau) \end{bmatrix}, \end{aligned} \tag{15}$$

$$\begin{aligned} \mathcal{L}V_4(t) &= \begin{bmatrix} F(x(t)) \\ F(x(t-\frac{\tau}{2})) \end{bmatrix}^T \begin{bmatrix} U \otimes R_1 & \\ & U \otimes R_2 \end{bmatrix} \begin{bmatrix} F(x(t)) \\ F(x(t-\frac{\tau}{2})) \end{bmatrix} \\ &\quad - \begin{bmatrix} F(x(t-\frac{\tau}{2})) \\ F(x(t-\tau)) \end{bmatrix}^T \begin{bmatrix} U \otimes R_1 & \\ & U \otimes R_2 \end{bmatrix} \begin{bmatrix} F(x(t-\frac{\tau}{2})) \\ F(x(t-\tau)) \end{bmatrix}. \end{aligned} \tag{16}$$

From Lemma 2, it follows that

$$\begin{aligned} & - \int_{t-\tau}^{t} y^T(s)(U \otimes P_2)y(s)ds \\ \leqslant & -\frac{2}{h}[(\int_{t-\frac{\tau}{2}}^{t} y(s)ds)^T(U \otimes P_2)(\int_{t-\frac{\tau}{2}}^{t} y(s)ds) \\ & + (\int_{t-\tau}^{t-\frac{\tau}{2}} y(s)ds)^T(U \otimes P_2)(\int_{t-\tau}^{t-\frac{\tau}{2}} y(s)ds)]. \end{aligned} \tag{17}$$

From the Newton-Leibniz formula, we have that for any matrices M_1, M_2,

$$2x^T(t)(U \otimes M_1)\left[x(t) - x(t-\frac{\tau}{2}) - \int_{t-\frac{\tau}{2}}^{t} y(s)ds - \int_{t-\frac{\tau}{2}}^{t} \sigma(s)d\omega(s)\right] = 0,$$

$$\begin{aligned} & 2x^T(t-\frac{\tau}{2})(U \otimes M_2)[x(t-\frac{\tau}{2}) - x(t-\tau) \\ & - \int_{t-\tau}^{t-\frac{\tau}{2}} y(s)ds - \int_{t-\tau}^{t-\frac{\tau}{2}} \sigma(s)d\omega(s)] = 0. \end{aligned} \tag{18}$$

In addition, for any matrix S, the following is true:

$$\begin{aligned}\pi &=: 2y^T(t)(U\otimes S)[(I_N\otimes A+G^{(1)}\otimes \Gamma_1)x(t)\\ &\quad +(G^{(2)}\otimes\Gamma_2)x(t-\tau)+(I_N\otimes B)F(x(t))-y(t)]=0. \end{aligned}\tag{19}$$

Noting that $UG^{(i)}=G^{(i)}U=NG^{(i)}$ $(i=1,2)$, for any matrix H with appropriate dimension, we obtain

$$(U\otimes H)(G^{(i)}\otimes\Gamma_i)=(UG^{(i)})\otimes(H\Gamma_i)=(NG^{(i)})\otimes(H\Gamma_i),\quad i=1,2. \tag{20}$$

From Lemma 3 and (13)-(20), we have

$$\begin{aligned}\mathcal{L}V_1(t)=&-\sum_{1\leqslant i<j\leqslant N}\Big\{(x_i(t)-x_j(t))^T[-(P_1A+A^TP_1)-NG_{ij}^{(1)}(P_1\Gamma_1+\Gamma_1^TP_1)]\\ &\times(x_i(t)-x_j(t))+2(x_i(t)-x_j(t))^T[-P_1B(\int_{-\tau}^{0}f(x_i(t+s))ds\\ &-\int_{-\tau}^{0}f(x_j(t+s))ds)+NG_{ij}^{(2)}P_1\Gamma_2(x_i(t-\tau)-x_j(t-\tau))\\ &-(\sigma_i(t,x_i(t),x_i(t-\tau))-\sigma_j(t,x_j(t),x_j(t-\tau)))^TP_1\\ &\times(\sigma_i(t,x_i(t),x_i(t-\tau))-\sigma_j(t,x_j(t),x_j(t-\tau)))\Big\}, \end{aligned}\tag{21}$$

where

$$\begin{aligned}&(\sigma_i(t,x_i(t),x_i(t-\tau))-\sigma_j(t,x_j(t),x_j(t-\tau)))^TP_1\\ &\qquad\times(\sigma_i(t,x_i(t),x_i(t-\tau))-\sigma_j(t,x_j(t),x_j(t-\tau)))\\ &\leqslant\lambda[(x_i(t)-x_j(t))^TW_1^TW_1(x_i(t)-x_j(t))\\ &\qquad+(x_i(t-\tau)-x_j(t-\tau))^TW_2^TW_2(x_i(t-\tau)-x_j(t-\tau))]. \end{aligned}\tag{22}$$

$$\begin{aligned}\mathcal{L}V_2(t)\leqslant& hy^T(t)(U\otimes P_2)y(t)-\frac{2}{h}[(\int_{t-\frac{\tau}{2}}^{t}y(s)ds)^T(U\otimes P_2)\int_{t-\frac{\tau}{2}}^{t}y(s)ds\\ &+(\int_{t-\tau}^{t-\frac{\tau}{2}}y(s)ds)^T(U\otimes P_2)\int_{t-\tau}^{t-\frac{\tau}{2}}y(s)ds]\\ &=h\sum_{1\leqslant i<j\leqslant N}(y_i(t)-y_j(t))^TP_2(y_i(t)-y_j(t))\\ &-\frac{2}{h}\sum_{1\leqslant i<j\leqslant N}[(\int_{t-\frac{\tau}{2}}^{t}y_i(s)ds-\int_{t-\frac{\tau}{2}}^{t}y_j(s)ds)^T\\ &P_2(\int_{t-\frac{\tau}{2}}^{t}y_i(s)ds-\int_{t-\frac{\tau}{2}}^{t}y_j(s)ds)\\ &+(\int_{t-\tau}^{t-\frac{\tau}{2}}y_i(s)ds-\int_{t-\tau}^{t-\frac{\tau}{2}}y_j(s)ds)^TP_2(\int_{t-\tau}^{t-\frac{\tau}{2}}y_i(s)ds-\int_{t-\tau}^{t-\frac{\tau}{2}}y_j(s)ds)]\\ &=\sum_{1\leqslant i<j\leqslant N}[h(y_i(t)-y_j(t))^TP_2(y_i(t)-y_j(t))\\ &-\frac{2}{h}(\mathcal{P}_i(t)-\mathcal{P}_j(t))^T\mathbb{P}_2(\mathcal{P}_i(t)-\mathcal{P}_j(t))], \end{aligned}\tag{23}$$

where

$$\mathcal{P}_i(t) = \begin{bmatrix} \int_{t-\frac{\tau}{2}}^{t} y_i(s)ds \\ \int_{t-\tau}^{t-\frac{\tau}{2}} y_i(s)ds \end{bmatrix}, \quad \mathbb{P}_2 = I_2 \otimes P_2 = \begin{bmatrix} P_2 & 0 \\ 0 & P_2 \end{bmatrix}.$$

$$\begin{aligned} \mathcal{L}V_3(t) = \sum_{1\leqslant i<j\leqslant N} & [(r_i(t) - r_j(t))^T \mathbb{Q}(r_i(t) - r_j(t)) \\ & -(r_i(t-\frac{\tau}{2}) - r_j(t-\frac{\tau}{2}))^T \mathbb{Q}(r_i(t-\frac{\tau}{2}) - r_j(t-\frac{\tau}{2}))], \end{aligned} \tag{24}$$

where

$$r_i(t) = \begin{bmatrix} x_i(t) \\ x_i(t-\frac{\tau}{2}) \end{bmatrix}, \qquad \mathbb{Q} = \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix}.$$

$$\begin{aligned} \mathcal{L}V_4(t) = \sum_{1\leqslant i<j\leqslant N} & [(\mathcal{F}_i(t) - \mathcal{F}_j(t))^T \mathbb{R}(\mathcal{F}_i(t) - \mathcal{F}_j(t)) \\ & -(\mathcal{F}_i(t-\frac{\tau}{2}) - \mathcal{F}_j(t-\frac{\tau}{2}))^T \mathbb{R}(\mathcal{F}_i(t-\frac{\tau}{2}) - \mathcal{F}_j(t-\frac{\tau}{2}))], \end{aligned} \tag{25}$$

where

$$\mathcal{F}_i(t) = \begin{bmatrix} \int_{-\tau}^{0} f(x_i(t+s))ds \\ \int_{-\tau}^{0} f(x_i(t-\frac{\tau}{2}+s))ds \end{bmatrix}, \quad \mathbb{R} = \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix}.$$

From (18)

$$\begin{aligned} & \sum_{1\leqslant i<j\leqslant N} (x_i(t)-x_j(t))^T (M_1+M_1^T) \Big[(x_i(t) - x_j(t)) - (x_i(t-\frac{\tau}{2}) - x_j(t-\frac{\tau}{2})) \\ & - (\int_{t-\frac{\tau}{2}}^{t} y_i(s)ds - \int_{t-\frac{\tau}{2}}^{t} y_j(s)ds) - (\int_{t-\frac{\tau}{2}}^{t} \sigma_i(s)d\omega(s) - \int_{t-\frac{\tau}{2}}^{t} \sigma_j(s)d\omega(s))\Big] \\ & + \sum_{1\leqslant i<j\leqslant N} (x_i(t-\frac{\tau}{2}) - x_j(t-\frac{\tau}{2}))^T (M_2+M_2^T) \Big[(x_i(t-\frac{\tau}{2}) - x_j(t-\frac{\tau}{2})) \\ & - (x_i(t-\tau) - x_j(t-\tau)) - (\int_{t-\tau}^{t-\frac{\tau}{2}} y_i(s)ds - \int_{t-\tau}^{t-\frac{\tau}{2}} y_j(s)ds) \\ & - (\int_{t-\tau}^{t-\frac{\tau}{2}} \sigma_i(s)d\omega(s) - \int_{t-\tau}^{t-\frac{\tau}{2}} \sigma_j(s)d\omega(s))\Big] \\ = & \sum_{1\leqslant i<j\leqslant N} (r_i(t) - r_j(t))^T (\mathbb{M}+\mathbb{M}^T)[(r_i(t) - r_j(t)) \\ & -(r_i(t-\frac{\tau}{2}) - r_j(t-\frac{\tau}{2})) - (\mathcal{P}_i(t) - \mathcal{P}_j(t)) - (\Omega_i(t) - \Omega_j(t))] = 0, \end{aligned} \tag{26}$$

where

$$\mathbb{M} = \begin{bmatrix} M_1 & 0 \\ 0 & M_2 \end{bmatrix}, \quad \Omega_i = \begin{bmatrix} \int_{t-\frac{\tau}{2}}^{t} \sigma_i(s)d\omega(s) \\ \int_{t-\tau}^{t-\frac{\tau}{2}} \sigma_i(s)d\omega(s) \end{bmatrix}.$$

From (19)

$$2\sum_{1\leqslant i<j\leqslant N}\Big[(x_i(t)-x_j(t))^T(A^TS^T-NG_{ij}^{(1)}\Gamma_1^TS^T)-(x_i(t-\tau)-x_j(t-\tau))^T$$
$$\times(NG_{ij}^{(2)}\Gamma_2^TS^T)+(\int_{-\tau}^{0}f(x_i(t+s))ds-\int_{-\tau}^{0}f(x_j(t+s))ds)^T(B^TS^T)\Big]$$
$$\times(y_i(t)-y_j(t))-\sum_{1\leqslant i<j\leqslant N}(y_i(t)-y_j(t))^T(S+S^T)(y_i(t)-y_j(t))=0. \quad (27)$$

From Assumption 2, for ϵ_1, $\epsilon_2>0$, it can be derived that

$$\epsilon_1\begin{bmatrix}x_i(t)-x_j(t)\\ f(x_i(t))-f(x_j(t))\end{bmatrix}^T\begin{bmatrix}\hat{L} & -\check{L}\\ * & 2I\end{bmatrix}\begin{bmatrix}x_i(t)-x_j(t)\\ f(x_i(t))-f(x_j(t))\end{bmatrix}\leqslant 0,$$
$$\epsilon_2\begin{bmatrix}x_i(t-\frac{\tau}{2})-x_j(t-\frac{\tau}{2})\\ f(x_i(t-\frac{\tau}{2}))-f(x_j(t-\frac{\tau}{2}))\end{bmatrix}^T\begin{bmatrix}\hat{L} & -\check{L}\\ * & 2I\end{bmatrix}\begin{bmatrix}x_i(t-\frac{\tau}{2})-x_j(t-\frac{\tau}{2})\\ f(x_i(t-\frac{\tau}{2}))-f(x_j(t-\frac{\tau}{2}))\end{bmatrix}\leqslant 0;$$

then we have

$$\sum_{1\leqslant i<j\leqslant N}\Big\{(r_i(t)-r_j(t))^T\Big[2(\epsilon\otimes\check{L})(\hat{\mathcal{F}}_i(t)-\hat{\mathcal{F}}_j(t))-(\epsilon\otimes\hat{L})(r_i(t)-r_j(t))\Big]$$
$$-(\hat{\mathcal{F}}_i(t)-\hat{\mathcal{F}}_j(t))^T(2\epsilon\otimes I_n)(\hat{\mathcal{F}}_i(t)-\hat{\mathcal{F}}_j(t))\Big\}\geqslant 0, \quad (28)$$

where

$$\hat{\mathcal{F}}_i(t)=\begin{bmatrix}f(x_i(t))\\ f(x_i(t-\frac{\tau}{2}))\end{bmatrix}.$$

From (21)-(28), we have

$$\mathcal{L}V(t)\leqslant\sum_{1\leqslant i<j\leqslant N}\{(x_i(t)-x_j(t))^T[(P_1A+A^TP_1)-NG_{ij}^{(1)}(P_1\Gamma_1+\Gamma_1^TP_1)]$$
$$\times(x_i(t)-x_j(t))+2(x_i(t)-x_j(t))^T[P_1B(\int_{-\tau}^{0}f(x_i(t+s))ds$$
$$-\int_{-\tau}^{0}f(x_j(t+s))ds)-NG_{ij}^{(2)}P_1\Gamma_2(x_i(t-\tau)-x_j(t-\tau))$$
$$+\lambda[(x_i(t)-x_j(t))^TW_1^TW_1(x_i(t)-x_j(t))+(x_i(t-\tau)-x_j(t-\tau))^T$$
$$\times W_2^TW_2(x_i(t-\tau)-x_j(t-\tau))]+(y_i(t)-y_j(t))^T(hP_2-S-S^T)$$
$$\times(y_i(t)-y_j(t))-\frac{2}{h}(\mathcal{P}_i(t)-\mathcal{P}_j(t))^T\mathbb{P}_2(\mathcal{P}_i(t)-\mathcal{P}_j(t))+(r_i(t)-r_j(t))^T$$
$$\times(\mathbb{Q}+\mathbb{M}+\mathbb{M}^T)(r_i(t)-r_j(t))$$

$$
\begin{aligned}
&-(r_i(t-\frac{\tau}{2})-r_j(t-\frac{\tau}{2}))^T\mathbb{Q}(r_i(t-\frac{\tau}{2})-r_j(t-\frac{\tau}{2}))\\
&-2(r_i(t)-r_j(t))^T\mathbb{M}[(r_i(t-\frac{\tau}{2})-r_j(t-\frac{\tau}{2}))+(\mathcal{P}_i(t)-\mathcal{P}_j(t))\\
&+(\Omega_i(t)-\Omega_j(t))]+(\mathcal{F}_i(t)-\mathcal{F}_j(t))^T\mathbb{R}(\mathcal{F}_i(t)-\mathcal{F}_j(t))\\
&-(\mathcal{F}_i(t-\frac{\tau}{2})-\mathcal{F}_j(t-\frac{\tau}{2}))^T\mathbb{R}(\mathcal{F}_i(t-\frac{\tau}{2})-\mathcal{F}_j(t-\frac{\tau}{2}))\\
&+2(x_i(t)-x_j(t))^T[A^TS^T-NG_{ij}^{(1)}\Gamma_1^TS^T)(y_i(t)-y_j(t))\\
&-2(x_i(t-\tau)-x_j(t-\tau))^T(NG_{ij}^{(2)}\Gamma_2^TS^T)(y_i(t)-y_j(t))\\
&+2(\int_{-\tau}^{0}f(x_i(t+s))ds-\int_{-\tau}^{0}f(x_j(t+s))ds)^TB^TS^T(y_i(t)-y_j(t))\\
&+(r_i(t)-r_j(t))^T[2(\epsilon\otimes\check{L})(\hat{\mathcal{F}}_i(t)-\hat{\mathcal{F}}_j(t))-(\epsilon\otimes\hat{L})(r_i(t)-r_j(t))]\\
&-(\hat{\mathcal{F}}_i(t)-\hat{\mathcal{F}}_j(t))^T(2\epsilon\otimes I_n)(\hat{\mathcal{F}}_i(t)-\hat{\mathcal{F}}_j(t)).
\end{aligned}
$$

Let

$$
\xi_{ij}(t)=\begin{bmatrix}
r_i(t)-r_j(t)\\
x_i(t-\tau)-x_j(t-\tau)\\
\mathcal{F}_i(t)-\mathcal{F}_j(t)\\
\int_{-\tau}^{0}f(x_i(t-\tau+s))ds-\int_{-\tau}^{0}f(x_j(t-\tau+s))ds\\
\hat{\mathcal{F}}_i(t)-\hat{\mathcal{F}}_j(t)\\
\mathcal{P}_i(t)-\mathcal{P}_j(t)\\
y_i(t)-y_j(t)
\end{bmatrix},
$$

then

$$
\mathbb{E}\{\mathcal{L}V(t)\}\leqslant\sum_{1\leqslant i<j\leqslant N}\xi_{ij}^T(t)\Theta_{ij}\xi_{ij}<0,
$$

where Θ_{ij} is defined in (10). From (10), it is guaranteed that all the subsystems in (1) are asymptotically synchronized for any fixed time delay $\tau\in(0,h]$. The proof is then completed.

4 Conclusions

In this paper, we have investigated the global synchronization of complex networks with distributed delays and stochastic disturbances. By utilizing a 'delay-fractioning' approach, firstly we built a novel Lyapunov functional, and then the properties of Kronecker product and stochastic analysis techniques are used to get the synchronization criteria in the form of LMIs which can be verified by the standard numerical software-Matlab toolbox.

References

1. Dangalchev, C.: Generation Models for Scale-free Networks. Physica A 338, 659–671 (2004)

2. Duan, Z., Chen, G., Huang, L.: Synchronization of Weighted Networks and Complex Synchronized Regions. Physics Letters A 372, 3741–3751 (2008)
3. Strogatz, S.H.: Exploring Complex Networks. Nature 410, 268–276 (2001)
4. Wang, X., Chen, G.: Complex Networks: Small-world, Scale-free and Beyond. IEEE Circuits & Systems Magazine 3, 6–20 (2003)
5. Cao, J., Li, P., Wang, W.W.: Global Synchronization in Arrays of Delayed Neural Networks with Constant and Delayed Coupling. Physics Letters A 353, 318–325 (2006)
6. Sun, Y.H., Cao, J., Wang, Z.D.: Exponential Synchronization of Stochastic Perturbed Chaotic Delayed Neural Networks. Neurocomputing 70, 2477–2485 (2007)
7. Wang, W.W., Cao, J.: Synchronization in an Array of Linearly Coupled Networks with Time-varying Delay. Physica A 366, 197–211 (2006)
8. Wang, Y., Wang, Z.D., Liang, J.L.: A Delay Fractioning Approach to Global Synchronization of Delayed Complex Networks with Stochastic Disturbances. Physics Letters A 372, 6066–6073 (2008)
9. Liang, J.L., Wang, Z.D., Liu, Y., Liu, X.: Global Synchronization Control of General Delayed Discrete-time Networks with Stochastic Coupling and Disturbances. IEEE Trans. Syst. Man Cybern. B 38, 1073–1083 (2008)
10. Li, C.H., Yang, S.Y.: Synchronization in Linearly Coupled Dynamical Networks with Distributed Time Delays. Int. J. Bifurc. Chaos 18, 2039–2047 (2008)
11. Sáchez, E., Matias, M., Muñuzuri, V.: Analysis of Synchronization of Chaotic Systems by Noise: An Experimental Study. Physical Review E 56, 4068–4071 (1997)
12. Gopalsamy, K., He, X.Z.: Stability in Asymmetric Hopfield Nets with Transmission Delays. Physica D 76, 344–358 (1994)
13. Zhao, H.: Global Stability of Neural Networks with Distributed Delays. Physical Review E 68, 051909 (2003)
14. Langville, A.N., William, W.J., Stewart, J.: The Kronecker Product and Stochastic Automata Networks. J. Comput. Appl. Math. 167, 429–447 (2004)
15. Friedman, A.: Stochastic Differential Equations and Applications. Academic Press, New York (1976)
16. Mao, X.: Exponential Stability of Stochastic Differential Equations. Marcel Deker, New York (1994)
17. Mou, S., Gao, H., Qiang, W., Chen, K.: New Delay-dependent Exponential Stability for Neural Networks with Time Delay. IEEE Trans. Syst. Man Cybern. B 38, 571–576 (2008)
18. Boyd, S., Ghaoui, L.E., Feron, E., Balakrishnan, V.: Linear Matrix Inequalities in System and Control Theory. SIAM, Philadelphia (1994)
19. Gao, H.J., Lam, J., Chen, G.R.: New Criteria for Synchronization Stability of General Complex Dynamical Networks with Coupling Delays. Physics Letters A 360, 263–273 (2006)
20. Gu, K.Q., Kharitonov, V.L., Chen, J.: Stability of Time-Delay Systems. Birkhauser, Boston (2003)

Exponential Stability of Impulsive Hopfield Neural Networks with Time Delays

Tingyan Xing, Muyao Shi, Wenjie Jiang, Nan Zhang, and Tuo Wang

School of Information Engineering,
China University of Geosciences, Beijing, 100083, China
xtygis@163.com

Abstract. This paper considers the problems of global exponential stability and exponential convergence rate for impulsive Hopfield neural networks with time delays. By using the method of Lyapunov functions, M-matrix theory and inequality technique, some sufficient conditions for ensuring global exponential stability of these networks are derived, and the estimation for exponential convergence rate index is also obtained. As an illustration, an numerical example is worked out to show the effectiveness of the obtained results.

1 Introduction

Hopfield neural networks have been extensively studied and developed in recent years, and many important applications have been found in various areas (See, e. g., [1-19]). Since the existence of delays is frequently a source of instability for neural networks, the stability of neural networks with time delays has long been a focused topic of theoretical as well as practical importance, and there has been considerable attention in the literature on Hopfield type neural networks with time delays (See, e. g., [2-4,6,7,9-11,13-19]). Besides delay effect, impulsive effects are also likely to exist in neural networks (See, e. g., [20-27]). For instance, in implementation of electronic networks, the state of the networks is subject to instantaneous perturbations and experiences abrupt change at certain instants, which may be caused by switching phenomenon, frequency change or other sudden noise, that is, it exhibits impulsive effects [23]. Therefore, it is necessary to consider both impulsive effect and delay effect on dynamical behaviors of neural networks. Some results on impulsive effect have been gained for delayed neural networks, see [20-27] and the references therein. In this paperwe consider the impulsive Hopfield neural networks with time delays. Lyapunov method, M-matrix theory and LMI technique are employed to investigate the sufficient conditions for the global exponential stability. This paper is organized as follows. In Section 2, impulsive Hopfield neural networks with time delays model is described and a lemma is given. Based on the Lyapunov stability theory, in combination with M-matrix theory, some global exponential stability criteria for neural networks are derived in Section 3. Example and conclusions are given in Section 4 and 5, respectively.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 503–511, 2009.

2 Model Description and Preliminaries

We consider the impulsive Hopfield neural networks with time delays described by

$$\begin{cases} C_i\dot{u}_i(t) = -u_i(t)/R_i + \sum\limits_{j=1}^{n} T_{ij}g_j(u_j(t-\tau_j)) + I_i, t \neq t_k \\ \Delta u_i(t) = d_i u_i(t^-) + \sum\limits_{j=1}^{n} W_{ij}h_j(u_j(t^- - \tau_j)), t = t_k, \end{cases} \quad i = 1, 2, \cdots, n, \quad (1)$$

where $\Delta u_i(t_k) = u_i(t_k) - u_i(t_k^-), u_i(t_k^-) = \lim\limits_{t \to t_k^-} u_i(t), k \in \mathcal{Z} = \{1, 2, \cdots\}$; the time sequence $\{t_k\}$ satisfies $0 < t_0 < t_1 < t_2 < \cdots < t_k < t_{k+1} < \cdots$, and $\lim\limits_{k\to\infty} t_k = \infty; C_i > 0, R_i > 0$, and I_i are, respectively, the capacitance, resistance, and external input of the ith neuron; T_{ij} and W_{ij} are the synaptic weights of the neural networks; and $\tau_i \geq 0$ is the transmission delay of the ith neuron.

The initial condition for system (1) is given by $u_i(s) = \psi_i(s),\ s \in [t_0 - \tau, t_0]$, $i = 1, 2, \cdots, n$ where $\psi_i : [t_0 - \tau, t_0] \to \Re,\ (i = 1, 2, \cdots, n)$, is a continuous function, and $\tau = \max\limits_{1\leq i\leq n}\{\tau_i\}$.

Throughout this paper, we assume that the neuron activation functions $g_i(u)$, $h_i(u),\ i = 1, 2, \cdots, n$, are continuous and satisfy the following conditions:

$$|g_i(u_i)| \leq M_i, 0 \leq \frac{g_i(u_i) - g_i(v_i)}{u_i - v_i} \leq K_i, \forall\ u_i \neq v_i, u_i, v_i \in \Re, i = 1, 2, \cdots, n, \quad (2)$$

$$|h_i(u_i)| \leq N_i, 0 \leq \frac{h_i(u_i) - h_i(v_i)}{u_i - v_i} \leq L_i, \forall\ u_i \neq v_i, u_i, v_i \in \Re, i = 1, 2, \cdots, n. \quad (3)$$

By Lemma 2.1[28] ,system (1) admits at least one equilibrium point.

Let $u^* = (u_1^*, u_2^*, \cdots, u_n^*,)^T$ be an equilibrium point of system (1), and set $x_i(t) = u_i(t) - u_i^*$, $d_i u_i^* + \sum\limits_{j=1}^{n} W_{ij}h_j(u_j^*) = 0$, $f_i(x_i(t-\tau_i)) = g_i(u_i(t-\tau_i)) - g_i(u_i^*)$, and $\varphi_i(x_i(t-\tau_i)) = h_i(u_i(t-\tau_i)) - h_i(u_i^*),\ i = 1, 2, \cdots, n$.

Then, for each $i = 1, 2, \cdots, n$,

$$|f_i(z)| \leq K_i|z|,\ zf_i(z) \geq 0,\ |\varphi_i(z)| \leq L_i|z|,\ z\varphi_i(z) \geq 0,\ \forall z \in \Re. \quad (4)$$

System (1) may be rewritten as follows.

$$\begin{cases} C_i\dot{x}_i(t) = -x_i(t)/R_i + \sum\limits_{j=1}^{n} T_{ij}f_j(x_j(t-\tau_j)), t \neq t_k \\ \Delta x_i(t) = d_i x_i(t^-) + \sum\limits_{j=1}^{n} W_{ij}\varphi_j(x_j(t^- - \tau_j)), t = t_k \end{cases}, \ i = 1, 2, \cdots, n, \quad (5)$$

Define $C = \mathrm{diag}(C_1, C_2, \cdots, C_n),\ R = \mathrm{diag}(R_1, R_2, \cdots, R_n)$, and $K = \mathrm{diag}(K_1, K_2, \cdots, K_n)$.

The following lemma will be useded in the proof of our main results.

Lemma 1.[9] Assume that the following conditions are satisfied.
(i) $D^+x_i(t) \le \sum_{j=1}^{n} a_{ij}x_j(t) + \sum_{j=1}^{n} b_{ij}\overline{x}_j(t),\ i = 1, 2, \cdots, n,$

where D^+ denotes the Dini derivative, $\overline{x}_j(t) = \sup_{t-\tau \le s \le t} x_j(s)$,

$a_{ij} \ge 0$ for $i \ne j,\ b_{ij} \ge 0,\ i, j = 1, 2, \cdots, n$, and $\sum_{j=1}^{n} \overline{x}_j(t_0) > 0$;
(ii) $G = -(a_{ij} + b_{ij})_{n\times n}$ is an M matrix.

Then, there exist constants $\alpha > 0, \gamma_i > 0, i = 1, 2, \cdots, n$ such that, for $i = 1, 2, \cdots, n$, $x_i(t) \le \gamma_i[\sum_{j=1}^{n} \overline{x}_j(t_0)]e^{-\alpha(t-t_0)}$, for $t \ge t_0$.

3 Main Results

In this section, we shall obtain some sufficient conditions for global exponential stability of impulsive Hopfield neural networks with time delays.

If $u^* = (u_1^*, u_2^*, \cdots, u_n^*)^T$ is an equilibrium point of system (1), then $x = (0, 0, \cdots, 0)^T$ is an equilibrium point of system (5). To prove the global exponential stability of the equilibrium point u^* of system (1), it is sufficient to prove the global exponential stability of the trivial solution of system (5).

Theorem 1. Assume that the following conditions are satisfied.
(i) $C^{-1}R^{-1} - C^{-1}AK$ is an M-matrix, where $A = (|T_{ij}|)_{n\times n}, i, j = 1, 2, \cdots, n$;
(ii) There exist a constant δ satisfying

$$\delta > \frac{\ln(\gamma\rho e^{\alpha\tau})}{\alpha\tau}$$

such that $\inf_{k\in Z}(t_k - t_{k-1}) > \tau\delta$, where $\alpha > 0, \gamma = \sum_{i=1}^{n} \gamma_i \ge 1,\ \gamma_i > 0,$
$i = 1, 2, \cdots, n, \rho = \max\{1, a + be^{\alpha\tau}\}, a = \max_{1\le j\le n}\{|1 + d_i|\}$, and
$b = \max_{1\le j\le n}\{|W_{ij}|L_j\}$.

Then, the equilibrium point u^* of system (1) is globally exponentially stable with convergence rate

$$\alpha - \frac{\ln(\gamma\rho e^{\alpha\tau})}{\delta\tau}.$$

Proof. For $t \ne t_k$, we computing the Dini derivative of $|x_i(t)|$ along the trajectories of system (5). By condition (2),(3) and (4), we get

$$\begin{aligned} D^+|x_i(t)||_{(5)} &= -\frac{1}{R_iC_i}|x_i(t)| + \sum_{j=1}^{n} T_{ij}\frac{f_j(x_j(t-\tau_j))}{C_i}\mathrm{sgn}(x_i(t)) \\ &\le -\frac{1}{R_iC_i}|x_i(t)| + \sum_{j=1}^{n} |T_{ij}|\frac{K_j}{C_i}|x_j(t-\tau_j)| \end{aligned}$$

$$\leq -\frac{1}{R_i C_i}|x_i(t)| + \sum_{j=1}^{n} \frac{K_j}{C_i}|T_{ij}||\overline{x}_j(t)|,$$

where $|\overline{x}_i(t)| = \sup_{t-\tau \leq s \leq t} |x_i(s)|$.

Hence, by assumption (i) and Lemma 1, there exist constants $\alpha > 0$, $\gamma_i > 0 (i = 1, 2, \cdots, n)$ such that, for each $i = 1, 2, \cdots, n$,

$$|x_i(t)| \leq \gamma_i \left(\sum_{j=1}^{n} |\overline{x}_j(t_{k-1})| \right) e^{-\alpha(t-t_{k-1})}, \text{ for all } t \in [t_{k-1}, t_k), k \in Z.$$

Let $\widehat{V}(t) = \sum_{j=1}^{n} |\overline{x}_j(t)|$.

Then, $|x_i(t)| \leq \gamma_i \widehat{V}(t_{k-1}) e^{-\alpha(t-t_{k-1})}$, for all $t \in [t_{k-1}, t_k), i = 1, 2, \cdots, n$.

From (5) we can show that

$$\begin{aligned} |x_i(t_k)| &\leq |1 + d_i||x_i(t_k^-)| + \max_{1 \leq j \leq n} \left\{ |W_{ij}|L_j \right\} \sum_{j=1}^{n} |x_j(t_k^- - \tau_j)| \\ &\leq a_i |x_i(t_k^-)| + b_i \widehat{V}(t_k^-), i = 1, 2, \cdots, n, \end{aligned}$$

where $\widehat{V}(t_k^-) = \sum_{j=1}^{n} |\overline{x}_j(t_k^-)|$, and $|\overline{x}_j(t_k^-)| = \sup_{t_k - \tau \leq t < t_k} |x_j(t)|$.

Hence,

$$\begin{cases} |x_i(t)| \leq \gamma_i \widehat{V}(t_{k-1}) e^{-\alpha(t-t_{k-1})}, t \in [t_{k-1}, t_k), k \in Z \\ |x_i(t_k)| \leq a_i |x_i(t_k^-)| + b_i \widehat{V}(t_k^-) \end{cases} \quad i = 1, 2, \cdots, n \qquad (6)$$

We claim that, for each $i = 1, 2, \cdots, n$,

$$|x_i(t)| \leq \gamma_i (\gamma \rho e^{\alpha\tau})^{k-1} \widehat{V}(t_0) e^{-\alpha(t-t_0)}, \text{for all } t \in [t_{k-1}, t_k), k \in Z. \qquad (7)$$

From (6), we get

$|x_i(t)| \leq \gamma_i \widehat{V}(t_0) e^{-\alpha(t-t_0)}$, for all $t \in [t_0, t_1), i = 1, 2, \cdots, n$,

i.e., inequality (7) is true for $k = 1$.

Assume that inequality (7) is true for $k = m$, i.e.,

$$|x_i(t)| \leq \gamma_i (\gamma \rho e^{\alpha\tau})^{m-1} \widehat{V}(t_0) e^{-\alpha(t-t_0)}, \text{for all } t \in [t_{m-1}, t_m), i = 1, 2, \cdots, n.$$

Then from (7) and assumption (ii), we have

$$\begin{aligned} |x_i(t_m)| &\leq a_i |x_i(t_m^-)| + b_i \widehat{V}(t_m^-) \\ &\leq a_i \gamma_i (\gamma \rho e^{\alpha\tau})^{m-1} \widehat{V}(t_0) e^{-\alpha(t_m - t_0)} + b_i \gamma (\gamma \rho e^{\alpha\tau})^{m-1} \widehat{V}(t_0) e^{\alpha\tau} e^{-\alpha(t_m - t_0)} \\ &= (\gamma \rho e^{\alpha\tau})^{m-1} (a_i \gamma_i + b_i \gamma e^{\alpha\tau}) \widehat{V}(t_0) e^{-\alpha(t_m - t_0)} \\ &\leq \gamma \rho_i (\gamma \rho e^{\alpha\tau})^{m-1} \widehat{V}(t_0) e^{-\alpha(t_m - t_0)}, \ i = 1, 2, \cdots, n. \end{aligned} \qquad (8)$$

Hence, by (6), we obtain

$$\begin{aligned}|x_i(t)| &\le \gamma_i \widehat{V}(t_m) e^{-\alpha(t-t_m)} \\ &= \gamma_i \sum_{j=1}^{n} \max\Big\{|\overline{x}_j(t_m^-)|, |x_j(t_m)|\Big\} e^{-\alpha(t-t_m)} \\ &\le \gamma_i \sum_{j=1}^{n} \max\Big\{\gamma_j(\gamma\rho e^{\alpha\tau})^{m-1}\widehat{V}(t_0)e^{\alpha\tau}e^{-\alpha(t_m-t_0)}, \\ &\qquad\qquad \gamma\rho_j(\gamma\rho e^{\alpha\tau})^{m-1}\widehat{V}(t_0)e^{-\alpha(t_m-t_0)}\Big\} e^{-\alpha(t-t_m)} \\ &\le \gamma_i(\gamma\rho e^{\alpha\tau})^m \widehat{V}(t_0)e^{-\alpha(t-t_0)}, \text{for all } t \in [t_m, t_{m+1}), i = 1, 2, \cdots, n.\end{aligned}$$

This shows that inequality (7) is true for $k = m + 1$.
Hence, by induction, it follows that for all $k \in Z$, inequality (7) is true.
By (7) and (8), we get

$$|x_i(t)| \le (\gamma\rho e^{\alpha\tau})^k \widehat{V}(t_0)e^{-\alpha(t-t_0)}, \text{for all } t \in [t_{k-1}, t_k), k \in Z, i = 1, 2, \cdots, n. \quad (9)$$

For any $t \in \Re^+$ there exist some $k \in Z$ such that $t \in [t_{k-1}, t_k)$.
By assumption (ii) we can show that

$$(\gamma\rho e^{\alpha\tau})^{k-1} \le e^{\frac{ln(\gamma\rho e^{\alpha\tau})}{\delta\tau}(t_{k-1}-t_0)}.$$

Thus from (9), for each $i = 1, 2, \cdots, n$, we have

$$\begin{aligned}|x_i(t)| &\le \gamma\rho e^{\alpha\tau}(\gamma\rho e^{\alpha\tau})^{k-1}\widehat{V}(t_0)e^{-\alpha(t-t_0)} \\ &\le \gamma\rho e^{\alpha\tau}\widehat{V}(t_0)e^{-\left(\alpha-\frac{ln(\gamma\rho e^{\alpha\tau})}{\delta\tau}\right)(t-t_0)}, \text{for all } t \in [t_{k-1}, t_k), k \in Z.\end{aligned}$$

Since $t \in \Re^+$ is arbitrary, it follows that for all $t \ge t_0$,

$$|x_i(t)| \le \gamma\rho e^{\alpha\tau}\widehat{V}(t_0)e^{-\left(\alpha-\frac{ln(\gamma\rho e^{\alpha\tau})}{\delta\tau}\right)(t-t_0)}, i = 1, 2, \cdots, n.$$

Thus

$$\|x(t)\| \le \sqrt{n}\gamma\rho e^{\alpha\tau}\|\overline{x}(t_0)\|e^{-\left(\alpha-\frac{ln(\gamma\rho e^{\alpha\tau})}{\delta\tau}\right)(t-t_0)}, \text{for all } t \ge t_0.$$

This completes the proof.

Theorem 2. Assume that
(i) There exist a constant $\lambda > 0$ such that $C^{-1}R^{-1} - \lambda I - e^{\lambda\tau}C^{-1}AK$ is an M-matrix, where $A = (|T_{ij}|)_{n\times n}, i, j = 1, 2, \cdots, n$;
(ii) There exist a constant δ satisfying

$$\delta > \frac{\ln(\gamma\rho e^{\lambda\tau})}{\lambda\tau}$$

such that $\inf_{k\in Z}(t_k - t_{k-1}) > \tau\delta$, where $\gamma = \sum_{i=1}^{n} \gamma_i \geq 1$, $\gamma_i > 0,\ i = 1, 2, \cdots, n$, $\rho = \max\{1, a + be^{\lambda\tau}\}, a = \max_{1\leq i\leq n}\{|1 + d_i|\}$, and $b = \max_{1\leq j\leq n}\left\{\sum_{i=1}^{n} |W_{ij}|L_j\right\}$.

Then, the equilibrium point u^* of system (1) is globally exponentially stable with convergence rate

$$\lambda - \frac{\ln(\gamma\rho e^{\lambda\tau})}{\delta\tau}.$$

Proof. Let $y_i(t) = e^{\lambda t}x_i(t), (i = 1, 2, \cdots, n)$.

Then for $t \neq t_k$, by computing the Dini derivative of $|y_i(t)|$ along the trajectories of system (5), it follows from the proof similar to that given for theorem 1 that

$$\begin{aligned} D^+|y_i(t)||_{(5)} &= e^{\lambda t}D^+|x_i(t)| + \lambda e^{\lambda t}|x_i(t)| \\ &= e^{\lambda t}\big(D^+|x_i(t)| + \lambda|x_i(t)|\big) \\ &\leq e^{\lambda t}\left[\left(\lambda - \frac{1}{R_iC_i}\right)|x_i(t)| + \sum_{j=1}^{n}\frac{K_j}{C_i}|T_{ij}||\overline{x}_j(t)|\right] \\ &\leq \left(\lambda - \frac{1}{R_iC_i}\right)|y_i(t)| + e^{\lambda\tau}\sum_{j=1}^{n}\frac{K_j}{C_i}|T_{ij}||\overline{y}_j(t)|, i = 1, 2, \cdots, n. \end{aligned}$$

where $|\overline{y}_j(t)| = \sup_{t-\tau\leq s\leq t}|y_j(s)|$.

Hence, by assumption (i) and Lemma 1, there exist constants $\alpha > 0$, $\gamma_i > 0 (i = 1, 2, \cdots, n)$, such that, for each $i = 1, 2, \cdots, n$, $|y_i(t)| \leq \gamma_i\left(\sum_{i=1}^{n}|\overline{y}_i(t_{k-1})|\right)e^{-\alpha(t-t_{k-1})}$, for all $t \in [t_{k-1}, t_k), k \in Z$.

Thus, for each $i = 1, 2, \cdots, n$,

$$\begin{aligned} |x_i(t)| &\leq \gamma_i\left(\sum_{i=1}^{n}|\overline{x}_i(t_{k-1})|\right)e^{-(\alpha+\lambda)(t-t_{k-1})} \\ &\leq \gamma_i\left(\sum_{i=1}^{n}|\overline{x}_i(t_{k-1})|\right)e^{-\lambda(t-t_{k-1})}, \text{for all } t \in [t_{k-1}, t_k), k \in Z. \end{aligned}$$

By the arguments similar to that used in the proof of Theorem 1, we can show that

$$\|x(t)\| \leq \sqrt{n}\gamma\rho e^{\lambda\tau}\|\overline{x}(t_0)\|e^{-\left(\lambda - \frac{\ln(\gamma\rho e^{\lambda\tau})}{\delta\tau}\right)(t-t_0)}, \text{for all } t \geq t_0.$$

The proof is complete.

4 Numerical Example

Consider the following impulsive Hopfield neural network with time delays

$$\begin{cases} C_i\dot{u}_i(t) = -u_i(t)/R_i + \sum\limits_{j=1}^{5} T_{ij}g_j(u_j(t-\tau_j)) + I_i,\ t \neq t_k \\ \Delta u_i(t) = d_i u_i(t^-) + \sum\limits_{j=1}^{5} W_{ij}h_j(u_j(t^- - \tau_j)),\ t = t_k \end{cases} \tag{10}$$

where $i = 1, 2, 3, 4, 5$, while $g_1(u_1) = \tanh(0.63u_1), g_2(u_2) = \tanh(0.78u_2)$, $g_3(u_3) = \tanh(0.46u_3), g_4(u_4) = \tanh(0.90u_4), g_5(u_5) = \tanh(0.98u_5)$, $h_1(u_1) = \tanh(0.09u_1), h_2(u_2) = \tanh(0.02u_2), h_3(u_3) = \tanh(0.17u_3)$, $h_4(u_4) = \tanh(0.47u_4), h_5(u_5) = \tanh(0.53u_5)$, $0 \leq \tau_i \leq 0.5, i = 1, 2, 3, 4, 5$, $C = \mathrm{diag}(C_1, C_2, C_3, C_4, C_5) = \mathrm{diag}(0.05, 0.69, 0.35, 0.66, 0.44)$, $R = \mathrm{diag}(R_1, R_2, R_3, R_4, R_5) = \mathrm{diag}(0.32, 0.28, 0.93, 0.58, 0.42)$, $D = \mathrm{diag}(D_1, D_2, D_3, D_4, D_5) = -\mathrm{diag}(0.84, 0.11, 0.59, 0.61, 0.20)$,

$$T = (T_{ij})_{5\times 5} = \begin{bmatrix} -0.02 & 0.11 & -0.03 & -0.04 & -0.09 \\ -0.04 & -0.05 & 0.02 & -0.02 & -0.12 \\ -0.03 & 0.03 & 0.06 & 0.06 & -0.14 \\ 0.01 & 0.12 & -0.11 & 0.16 & -0.02 \\ 0.02 & -0.01 & 0.12 & -0.14 & 0.27 \end{bmatrix},$$

$$W = (W_{ij})_{5\times 5} = \begin{bmatrix} 0.03 & -0.05 & -0.04 & -0.02 & 0.04 \\ -0.01 & 0.05 & 0.01 & 0.05 & 0.00 \\ -0.02 & 0.06 & 0.06 & 0.09 & -0.01 \\ 0.04 & -0.07 & 0.01 & 0.02 & 0.01 \\ 0.02 & -0.04 & 0.02 & 0.02 & 0.07 \end{bmatrix},$$

In this case, $M = N = (1, 1, 1, 1, 1)^T, K = \mathrm{diag}(0.63, 0.78, 0.46, 0.90, 0.98)$, $L = \mathrm{diag}(0.09, 0.02, 0.17, 0.47, 0.53), \tau = 0.5$, and $u^* = (0, 0, 0, 0, 0)^T$ is an equilibrium point of neural network (10).

We note that the matrix

$$C^{-1}R^{-1} - C^{-1}AK = \begin{bmatrix} 62.248 & -1.7160 & -0.2760 & -0.7200 & -1.7640 \\ -0.0365 & 5.1195 & -0.0133 & -0.0261 & -0.1704 \\ -0.0540 & -0.0669 & 2.9933 & -0.1543 & -0.3920 \\ -0.0095 & -0.1418 & -0.0764 & 2.3941 & -0.0297 \\ -0.0286 & -0.0177 & -0.1255 & -0.2864 & 4.8099 \end{bmatrix}$$

is an M-matrix. Choose constants $\alpha = 0.3, \gamma = 1.01$ such that $\rho = 1$. If we set $\delta = 1.15 > \frac{\ln(\gamma\rho e^{\alpha\tau})}{\alpha\tau} = 1.0663$, then, by Theorem 1, we see that the equilibrium point u^* of system (10) is globally exponentially stable with convergence rate 0.0218 for $\inf\limits_{k\in Z}\{t_k - t_{k-1}\} > 0.575$.

There exist a constant $\lambda = 1.2$ such that

$$C^{-1}R^{-1} - \lambda I - e^{\lambda\tau}C^{-1}AK = \begin{bmatrix} 60.8408 & -3.1268 & -0.5029 & -1.3119 & -3.2142 \\ -0.0665 & 3.8730 & -0.0243 & -0.0475 & -0.3106 \\ -0.0984 & -0.1218 & 1.7285 & -0.2811 & -0.7143 \\ -0.0174 & -0.2584 & -0.1397 & 1.0148 & -0.0541 \\ -0.0522 & -0.0323 & -0.2286 & -0.5218 & 3.1155 \end{bmatrix}$$

is an M-matrix with constant $\gamma = 1.01$ such that $\rho = 1.0613 > 1$ in Theorem 2. If we set $\delta = 1.15 > \frac{\ln(\gamma\rho e^{\alpha\tau})}{\alpha\tau} = 1.1157$, then, by Theorem 2, the equilibrium point u^* of system (10) is globally exponentially stable with convergence rate 0.0358 for $\inf_{k\in Z}\{t_k - t_{k-1}\} > 0.575$.

Hence, by Theorem1 and Theorem 2, we see that the equilibrium point u^* of system (10) is globally exponentially stable for $\inf_{k\in Z}\{t_k - t_{k-1}\} > 0.575$,, and the convergence rate computed by Theorem 1 and Theorem 2 are, respectively, 0.0218 and 0.0358.

5 Conclusion

Based on M-matrix theory and Lyapunov functions, several simple sufficient conditions ensuring the global exponential stability have been obtained for the Hopfield neural networks with time delays and impulses. Moreover, the exponential convergence rate was also estimated. The obtained results show the fact that the stability still remain under certain impulsive perturbations for the neural network with stable equilibrium point. An example has been given to show the effectiveness of the obtained results.

References

1. Hopfield, J.J.: Neurons with Graded Response Have Collective Computational Properties Like Those of Two State Neurons. Proceedings of the National Academy of Sciences, USA 81, 3088–3092 (1984)
2. Gopalsamy, K., He, X.Z.: Stability in Asymmetric Hopfield Nets with Transmission Delays. Physica D 76, 344–358 (1994)
3. Liao, X.X.: Stability of Hopfield-Type Neural Networks (I). Science in China 38, 407–418 (1995)
4. Zhang, Y., Zhong, S.M., Li, Z.L.: Periodic Solutions and Stability of Hopfield Neural Networks with Variable Delays. International Journal of Systems Sciences 27, 895–901 (1996)
5. Ma, J.W.: The Stability of the Generalized Hopfield Networks in Randomly Asynchronous Mode. Neural Networks 10, 1109–1116 (1997)
6. Liao, X.X., Liao, Y.: Stability of Hopfield-Type Neural Networks (II). Science in China(Series A) 40, 813–816 (1997)
7. Van, D.D., Zou, X.: Global Aattractivity in Delayed Hopfield Neural Networks Model. SIAM Journal of Applied Mathematics 58, 1878–1890 (1998)
8. Zhang, J.Y., Jin, X.S.: Global Stability Analysis in Delayed Hopfield Neural Network Models. Neural Networks 13, 745–753 (2000)
9. Liao, X.X., Xiao, D.M.: Global Exponential Stability of Hopfield Neural Networks with Time-Varying Delays. Acta Electronica Sinica 28, 87–90 (2000) (in Chinese)
10. Chu, T.G.: An Exponential Convergence Estimate for Analog Neural Networks with Delay. Physics Letters A 283, 113–118 (2001)
11. Peng, J.G., Qiao, H., Xu, Z.B.: A New Approach to Stability of Neural Networks with Time-Varying Delays. Neural Networks 15, 95–103 (2002)

12. Zhang, Q., Wei, X.P., Xu, J.: Global Exponential Stability of Hopfield Neural Networks with Continuously Distributed Delays. Physics Letters A 315, 431–436 (2003)
13. Zhao, H.Y.: Global Asymptotic Stability of Hopfield Neural Network Involving Distributed Delays. Neural Networks 17, 47–53 (2004)
14. Huang, H., Qu, Y.Z., Li, H.X.: Robust Stability Analysis of Switched Hopfield Neural Networks with Time-Varying Delay Under Uncertainty. Physics Letters A 345, 345–354 (2005)
15. Liu, Y.G., You, Z.S., Cao, L.P.: On Stability of Disturbed Hopfield Neural Networks with Time Delays. Neurocomputing 69, 941–948 (2006)
16. Liu, B.W.: Almost Periodic Solutions for Hopfield Neural Networks with Continuously Distributed Delays. Mathematics and Computers in Simulation 73, 327–335 (2007)
17. Zhang, Q., Wei, X.P., Xu, J.: Delay-Dependent Global Stability Condition for Delayed Hopfield Neural Networks. Nonlinear Analysis: Real World Applications 8, 997–1002 (2007)
18. Zhao, H.Y., Wang, L., Ma, C.G.: Hopf Bifurcation and Stability Analysis on Discrete-Time Hopfield Neural Network with Delay. Nonlinear Analysis: Real World Applications 9, 103–113 (2008)
19. Xu, B.J., Liu, X.Z., Liao, X.X.: Global Asymptotic Stability of High-Order Hopfield Type Neural Networks with Time Delays. Computers and Mathematics with Applications 45, 1729–1737 (2003)
20. Akca, H., Alassar, R., Covachev, V., et al.: Continuous-Time Additive Hopfield-Type Neural Networks with Impulses. Journal of Mathematical Analysis and Applications 290, 436–451 (2004)
21. Gopalsamy, K.: Stability of Artificial Neural Networks with Impulses. Applied Mathematics and Computation 154, 783–813 (2004)
22. Li, Y.K., Lu, L.H.: Global Exponential Stability and Existence of Periodic Solution of Hopfield-Type Neural Networks with Impulses. Physics Letters A 333, 62–71 (2004)
23. Xu, D.Y., Yang, Z.C.: Impulsive Delay Differential Inequality and Stability of Neural Networks. Journal of Mathematical Analysis and Applications 305, 107–120 (2005)
24. Zhang, Y., Sun, J.T.: Stability of Impulsive Neural Networks with Time Delays. Physics Letters A 348, 44–50 (2005)
25. Long, S.J., Xu, D.Y.: Delay-Dependent Stability Analysis for Impulsive Neural Networks with Time Varying Delays. Neurocomputing 71, 1705–1713 (2008)
26. Qiu, J.L.: Exponential Stability of Impulsive Neural Networks with Time-Varying Delays and Reaction-Diffusion Terms. Neurocomputing 70, 1102–1108 (2007)
27. Ho, D.W.C., Liang, J.L., Lam, J.: Global Exponential Stability of Impulsive High-Order BAM Neural Networks with Time-Varying Delays. Neural Networks 19, 1581–1590 (2006)
28. Xu, B.J., Wang, Q., Shen, Y., Liao, X.X.: Global Exponential Stability of Delayed Impulsive Hopfield Type Neural Networks. In: Wang, J., Liao, X.-F., Yi, Z. (eds.) ISNN 2005. LNCS, vol. 3496, pp. 181–186. Springer, Heidelberg (2005)

Neutral Differential Systems with Impulse and Delay

Junhao Hu[1,2] and Huafeng Chen[3,⋆]

[1] Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China
hujunhao@yahoo.com.cn
[2] College of Computer Science, South Central University for Nationalities, Wuhan 430074, China
[3] College of Science, Huazhong Agriculture University, Wuhan 430070, China
chenhf123@mail.hzau.edu.cn

Abstract. In this paper, we discuss a class of impulsive partial neutral functional differential inclusions with infinite delay. The existence of mild solutions of these neutral differential systems with impulse and delay is obtained by using the new nonlinear alternatives of Leray-Schauder type.

Keywords: Neutral differential systems, Impulses, Delays, existence.

1 Introduction

We consider the following neutral differential systems with impulse and infinite delay

$$\begin{cases} \dfrac{d}{dt}[x - g(t, x_t)] \in Ax(t) + F(t, x_t),\ t \in J, t \neq t_k \\ \Delta x(t_k) = I_k(x(t_k^-)), \quad k = 1, 2, \ldots, m, \\ x_0 = \phi(t) \in \mathcal{B}, \end{cases} \tag{1}$$

where $J := [0, b]$, A is the infinitesimal generator of an analytic semigroup of uniformly bounded linear operators, $(T(t))_{t\geq 0}$ on a Banach Space $(X, \|\cdot\|)$; the function $x_t : (-\infty, 0] \to X$, $x_t(\theta) = x(t+\theta)$, belongs to some abstract phase space $\mathcal{B}$ defined axiomatically; $g : J \times \mathcal{B} \to X$ and $F : J \times \mathcal{B} \to \mathcal{P}(X)$; $\mathcal{P}(X)$ denotes the class of all nonempty subsets of X. $0 = t_0 < t_1 < \cdots < t_m < t_{m+1} = b$, $I_k : X \to X (k = 1, 2, \ldots, m)$, $\Delta x(t_k) = x(t_k^+) - x(t_k^-)$, $x(t_k^+)$ and $x(t_k^-)$ are respectively the right and the left limit of x at $t = t_k$, and $x(t_k^+) = x(t_k)$.

The problem of the existence and controllability for differential inclusions has been extensively studied [1]–[6]. Benchohra et al.[2]-[1] considered the existence of solutions for functional and neutral functional inclusions. Benchohra et al.[4] studied the existence of solutions for integrodifferential inclusions on noncompact intervals. Benchohra et al.[3] discussed the existence of solutions for impulsive multivalued semilinear neutral functional differential inclusions. And Benchohra

⋆ Corresponding author.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 512–521, 2009.

et al.[5]-[6] studied the controllability of semilinear evolution and neutral functional differential inclusions. In this paper, we will give the existence of solutions for impulsive partial neutral functional differential inclusions with infinite delay. Our approach will be based on another nonlinear alternatives of Leray-Schauder type for multivalued maps due to D.O'Regan.

The paper is organized as follows. In Section 2, we recall some necessary preliminaries, In Section 3, we prove the existence.

2 Preliminaries

In this paper, X will be a separable Banach space with norm $\|\cdot\|$. $A : D(A) \subset X \to X$ will be the infinitesimal generator of an analytic semigroup, $(T(t))_{t\geq 0}$ of uniformly bounded linear operators on X, that is to say, there exists some constant $M \geq 1$ such that $\|T(t)\| \leq M$, for every $t \in J$. For literature relating to semigroup theory, we suggest Pazy [10]. We suppose that $0 \in \rho(A)$, for $0 < \alpha \leq 1$ then it is possible to define the fraction power $(-A)^\alpha$, as a closed linear operator on its domain $D((-A)^\alpha)$. Furthermore, the subspace $D((-A)^\alpha)$ is dense in X, and the expression $\|x\|_\alpha = \|(-A)^\alpha\|$, $x \in D((-A)^\alpha)$, defines a norm on $D((-A)^\alpha)$. Hereafter, let X_α denote the Banach space $D((-A)^\alpha)$ endowed with the norm $\|\cdot\|_\alpha$. For $0 < \beta \leq \alpha \leq 1$ $X_\alpha \hookrightarrow X_\beta$ and the imbedding is compact whenever the resolvent operator of A is compact. Also for every $0 < \alpha \leq 1$, there exists a positive constant C_α such that

$$\|(-A)^\alpha T(t)\| \leq \frac{C_\alpha}{t^\alpha}, \quad 0 < t \leq b. \tag{2}$$

We define $\mathcal{PC}$ by the set

$$\left\{ \varphi : [0,b] \to X : \begin{array}{c} \varphi(\cdot) \text{ is continuous at } t \neq t_k,\ \varphi(t_k^+) = \varphi(t_k), \text{ and} \\ \varphi(t_k^-) \text{ exist for } k = 1,2,\ldots,m. \end{array} \right\}$$

The norm $\|\cdot\|_1$ of the space $\mathcal{PC}$ is defined by $\|\varphi\|_1 = \sup_{0\leq s\leq T} \|\varphi(s)\|$. It is clear that $(\mathcal{PC}, \|\cdot\|_1)$ is a Banach space.

In this work, we will employ an axiomatic definition of the phase space $\mathcal{B}$ which is similar to that used in [11]. $\mathcal{B}$ will be a linear space of functions mapping $(-\infty, 0]$ to X endowed with a seminorm $\|\cdot\|_\mathcal{B}$. We will assume that $\mathcal{B}$ satisfies the following axioms:

(A) If $x : (-\infty, b] \to X$, is such that $x_0 \in \mathcal{B}$ and $x|_{[0,b]} \in \mathcal{PC}$,then for every $t \in [0,T]$ the following conditions hold:
  (1) x_t is in $\mathcal{B}$,
  (2) $\|x(t)\| \leq L\|x_t\|_\mathcal{B}$,
  (3) $\|x_t\|_\mathcal{B} \leq K(t)\sup\{\|x(s)\| : 0 \leq s \leq t\} + M(t)\|x_0\|_\mathcal{B}$,
  where $L > 0$ is a constant, $K, M : [0,\infty) \to [0,\infty)$,$K$ is continuous, M is locally bounded, and L, K, M are independent of $x(\cdot)$.
(B) For the function $x(\cdot)$ in (A), x_t is a $\mathcal{B}$-valued function on $[0,T]$.
(C) The space $\mathcal{B}$ is complete.

Let $\mathcal{P}(X)$ denote the class of all nonempty of subsets of X. Let $\mathcal{P}_{bd,cl}(X)$, $\mathcal{P}_{cp,cv}(X)$, $\mathcal{P}_{bd,cl,cv}(X)$ and $\mathcal{P}_{cd}(X)$ denote respectively the family of all nonempty bounded-closed, compact-convex, bounded-closed-convex and compact-acyclic [7] subsets of X.

F is called upper semicontinuous(shortly u.s.c.) on X, if for each $x_* \in X$, the set $F(x_*)$ is nonempty, closed subset of X, and if for each open set of V of X containing $F(x_*)$, there exists an open neighborhood N of x_* such that $F(N) \subseteq V$. F is said to be completely continuous if $F(V)$ is relatively compact, for every bounded subset $V \subseteq X$.

If the multivalued map F is completely continuous with nonempty compact values, then F is u.s.c. if and only if F has a closed graph,(i.e. $x_n \to x_*, y_n \to y_*, y_n \in F(x_n)$ imply $y_* \in F(x_*)$).

A point $x_0 \in X$ is called a fixed point of the multivalued map F if $x_0 \in F(x_0)$.

A multivalued map $F : J \to \mathcal{P}_{bd,cl,cv}(X)$ is said to be measurable if for each $x \in X$, the function $t \mapsto D(x, F(t))$ is a measurable function on J. For more details on the multivalued maps, see the books of Deimling [8].

Definition 1. Let $F : X \to \mathcal{P}_{bd,cl}(X)$ be a multivalued map.Then F is called a multivalued contraction if there exists a constant $k \in (0, 1)$ such that for each $x, y \in X$ we have

$$H(F(x), F(y)) \leq k\|x - y\|.$$

The constant k is called a contraction constant of F.

The consideration of this paper is based the another nonlinear alternatives of Leray-Schauder type for multivalued maps due to D.O'Regan [9].

Theorem 1.[9] Let X be a Banach space with U an open,convex subset of X and $u_0 \in U$. Suppose

(a) $F : \overline{U} \to \mathcal{P}_{cd}(X)$ has closed graph, and
(b) $F : \overline{U} \to \mathcal{P}_{cd}(X)$ is condensing map with $F(\overline{U})$ a subset of a bounded set in X

hold. Then either

(i) F has a fixed point in $\overline{U}$; or
(ii) There exist $u \in \partial U$ and $\lambda \in (0, 1)$ with $u \in \lambda F(u) + (1 - \lambda)\{u_0\}$.

Definition 2. A multivalued map $F : J \times \mathcal{B} \to \mathcal{P}_{bd,cl,cv}(X)$ is called L^1-Carathéodory if

(i) $F(t, x)$ is measurable with respect to t for each $x \in \mathcal{B}$,
(ii) $F(t, x)$ is u.s.c. with respect to x for each $t \in J$, and
(iii) for each $q > 0$, there exists a function $h_q \in L^1(J, [0, \infty)$ such that

$$\|F(t, v)\| := \sup\{\|g\| : g \in F(t, v)\} \leq h_q(t), \text{a.e. } t \in J$$

for all $v \in \mathcal{B}$ with $\|v\|_{\mathcal{B}} \leq q$.

We need the theorem due to Lasota and Opial [13].

Theorem 2. Let X be a Banach space, F be an L^1-Carathéodory multivalued map with $S_{F,\phi} \neq \emptyset$ where $S_{F,\phi} := \{g \in L^1(J,X) : g(t) \in F(t,\phi) \text{ a.e. } t \in J\}$, for each fixed $\phi \in \mathcal{B}$, and $\mathcal{K}$ be a linear continuous map from $L^1(J,X)$ to $C(J,X)$. Then the operator $\mathcal{K} \circ S_{F,\phi} : C(J,X) \to \mathcal{P}_{cp,cv}(C(J,X))$ is a closed graph operator in $C(J,X) \times C(J,X)$.

3 Main Result

Before stating and proving our main result, we give first the definition of mild solution and controllable.

Definition 3. A function $x : (-\infty, b] \to X$ is called a mild solution of the system(1.1) if $x(t) - f(t, x_t)$ is absolutely continuous on $[0,b) \setminus \{t_1, t_2, \ldots, t_m\}$, (1.1) is satisfied and for each $s \in [0,t)$, the function $AT(t-s)g(s,x_s)$ is integrable such that

$$x(t) = T(t)[\phi(0) - g(0,\phi(0))] + g(t,x_t) + \int_0^t AT(t-s)g(s,x_s)ds$$

$$+ \int_0^t T(t-s)f(s)ds + \sum_{0<t_k<t} T(t-t_k)I_k(x(t_k^-)), \tag{3}$$

where $f \in S_{F,x} = \{f \in L^1(J,X) : f(t) \in F(t,x_t), t \in J\}$.

We consider the following assumptions in the sequel

(H1) The semigroup $T(t)$ is compact for $t > 0$, and there exists $M \geq 1$ such that

$$\|T(t)\| \leq M, \quad \text{for all } t \geq 0.$$

(H2) The multivalued map $F(t,x)$ is an L^1-Carathéodory multivalued map and has compact and convex values for each $(t,x) \in J \times \mathcal{B}$.

(H3) There exist constants $0 < \beta < 1, L_1, L_2 > 0$ such that g is X_β-valued, $(-A)^\beta g$ is continuous, and

(i) $\|(-A)^\beta g(t,x_t)\| \leq L_1 \|x_t\|_{\mathcal{B}}$, $(t,x) \in J \times \mathcal{B}$

(ii) $\|(-A)^\beta g(t_1,x_{1t}) - (-A)^\beta g(t_2,x_{2t})\| \leq L_2(|t_1 - t_2| + \|x_{1t} - x_{2t}\|_{\mathcal{B}})$, $(t_i, x_{it}) \in J \times \mathcal{B}$, $i = 1,2$, with

$$L_0 := L_2 \left\{ \|(-A)^{-\beta}\| + \frac{C_{1-\beta} b^\beta}{\beta} \right\} K_b < 1.$$

where $K_b = \sup\{K(t) : 0 \leq t \leq b\}$, $M_b = \sup\{M(t) : 0 \leq t \leq b\}$.

(H4) The impulsive functions I_k are continuous and there exist positive constants β_k such that $\|I_k(x)\| \leq \beta_k$, $k = 1,2,\ldots,m$ for each $x \in X$.

(H5) There exists a positive function $p \in L^1(J,[0,\infty))$ such that

$$\|F(t,\phi)\| := \sup\{\|v\| : v \in F(t,\phi)\} \leq p(t)\Theta(\|\phi\|_{\mathcal{B}})$$

a.e. $t \in J, \phi \in \mathcal{B}$

where $\Theta : [0,\infty) \to (0,\infty)$ a continuous nondecreasing function.

Theorem 3. Let $\phi \in \mathcal{B}$, If the assumptions (H1)–(H5) are satisfied, then the system (1) has at least one solution on $(-\infty, b]$ provided that there exist a constant N_* with

$$\frac{(1 - K_b L_1 \|(-A)^{-\beta}\| - K_b L_1 C_{1-\beta} b^{\beta}/\beta) N_*}{N_1 + K_b M \Theta(N_*) \int_0^b p(s)ds} > 1, \tag{4}$$

Where $N_1 = K_b M \|g(0, \phi(0))\| + M K_b \|\phi(0)\| + M_b \|\phi\|_{\mathcal{B}} + M K_b \sum_{k=1}^{m} \beta_k$.

Proof. Let $\mathcal{B}_b$ be the space of all function $x : (-\infty, b] \to X$ such that $x_0 \in \mathcal{B}$ and the restriction $x|_J \in \mathcal{PC}$. For each $x(t) \in \mathcal{B}_b$, let $\|\cdot\|_b$ be the seminorm in $\mathcal{B}_b$ defined by

$$\|x\|_b = \|x_0\|_{\mathcal{B}} + \|x\|_1 = \|x_0\|_{\mathcal{B}} + \sup\{\|x(s)\| : \ 0 \le s \le b\}$$

The multivalued map $\Phi : \mathcal{B}_b \to \mathcal{P}(\mathcal{B}_b)$ is defined by Φx the set of $h \in \mathcal{B}_b$ such that

$$h(t) = \begin{cases} \phi(t), \quad \text{if } t \in (-\infty, 0], \\ T(t)[\phi(0) - g(0, \phi(0))] + \int_0^t AT(t-s)g(s, x_s)ds \\ +g(t, x_t) + \int_0^t T(t-s)f(s)ds + \sum\limits_{0<t_k<t} T(t-t_k)I_k(x(t_k^-)), \ t \in J, \end{cases}$$

where $f \in S_{F,x} = \{f \in L^1(J, X) : f(t) \in F(t, x_t), t \in J\}$, and Φ has a fixed point which is then a solution of the system (1.1).

We define a function

$$y(t) = \begin{cases} \phi(t), \quad \text{if } \ t \in (-\infty, 0], \\ T(t)\phi(0), \quad \text{if } t \in J, \end{cases}$$

Set $x(t) = z(t) + y(t), -\infty < t \le b$, It is clear that x satisfies (3) if and only if z satisfies $z_0 = 0$ and

$$z(t) = -T(t)g(0, \phi(0)) + g(t, z_t + y_t) + \int_0^t AT(t-s)g(s, z_s + y_s)ds$$

$$+ \int_0^t T(t-s)f(s)ds + \sum_{0<t_k<t} T(t-t_k)I_k(z(t_k^-) + y(t_k^-)),$$

where $f \in S_{F,z} = \{f \in L^1(J, X) : f(t) \in F(t, z_t + y_t), t \in J\}$

Let $\mathcal{B}_b^0$ be the space of all function $z : (-\infty, b] \to X$ such that $z_0 \equiv 0$ and the restriction $z|_J \in \mathcal{PC}$. For each $z(t) \in \mathcal{B}_b$, let $\|\cdot\|_b$ be the norm in $\mathcal{B}_b^0$ defined by

$$\|z\|_b = \sup\{\|z(s)\| : \ 0 \le s \le b\}$$

Thus $(\mathcal{B}_b^0, \|\cdot\|_b)$ is Banach space.

The multivalued map $\Phi_1 : \mathcal{B}_b^0 \to \mathcal{P}(\mathcal{B}_b^0)$ is defined by $\Phi_1 x$ the set of $h_0 \in \mathcal{B}_b$ such that

$$h_0(t) = \begin{cases} 0, \quad \text{if } t \in (-\infty, 0], \\ -T(t)g(0, \phi(0)) + \int_0^t AT(t-s)g(s, z_s + y_s)ds + g(t, z_t + y_t) \\ + \int_0^t T(t-s)f(s)ds + \sum\limits_{0<t_k<t} T(t-t_k)I_k(z(t_k^-) + y(t_k^-)), \ t \in J, \end{cases}$$

Step 1 Choose $u_0 = 0$ and a set U is convex and open in $\mathcal{B}_b^0$.
Let $\lambda \in (0,1)$ and let $x \in \lambda\Phi_1 x$, Then exists $f \in S_{F,z}$, such that

$$z(t) = \lambda \int_0^t T(t-s)f(s)ds + \lambda g(t, z_t + y_t) + \lambda \int_0^t AT(t-s)g(s, z_s + y_s)ds$$

$$-\lambda T(t)g(0, \phi(0)) + \lambda \sum_{0<t_k<t} T(t-t_k)I_k(z(t_k^-) + y(t_k^-)),$$

Then

$$\|z(t)\| \le M(\|g(0,\phi(0))\| + \int_0^t p(s)\Theta(\|z_s + y_s\|_{\mathcal{B}})ds + \sum_{k=1}^m \beta_k)$$

$$+L_1\|(-A)^{-\beta}\|\|z_t + y_t\|_{\mathcal{B}} + L_1 \int_0^t \frac{C_{1-\beta}}{(t-s)^{1-\beta}}\|z_s + y_s\|_{\mathcal{B}}\, ds$$

Since

$$\|z_t + y_t\|_{\mathcal{B}} \le \|z_t\|_{\mathcal{B}} + \|y_t\|_{\mathcal{B}}$$

$$\le K_b \sup\{\|z(s)\| : \ 0 \le s \le t\} + MK_b\|\phi(0)\| + M_b\|\phi\|_{\mathcal{B}}$$

Set $\chi(t) = K_b \sup\{\|z(s)\| : \ 0 \le s \le t\} + MK_b\|\phi(0)\| + M_b\|\phi\|_{\mathcal{B}}$, then the functions $\chi(t)$ is nondecreasing in J,
So

$$\|\chi(t)\| \le K_b M(\|g(0,\phi(0))\| + \int_0^t p(s)\Theta(\chi(s))ds + \sum_{k=1}^m \beta_k) + MK_b\|\phi(0)\| + M_b\|\phi\|_{\mathcal{B}}$$

$$+K_bL_1\|(-A)^{-\beta}\|\chi(t) + K_bL_1 \int_0^t \frac{C_{1-\beta}}{(t-s)^{1-\beta}}\chi(s)\, ds$$

Consider the norm of the function $\chi(t)$, $\|\chi\| = \sup\{\chi(t) : 0 \le t \le b\}$, Therefore, we obtain the following inequality

$$\frac{(1 - K_bL_1\|(-A)^{-\beta}\| - K_bL_1C_{1-\beta}b^{\beta}/\beta)\|\chi\|}{N_1 + K_bM\Theta(\|\chi\|)\int_0^b p(s)ds} \le 1$$

Then by (4), there exists N_* such that $\|\chi\| \ne N_*$. Set $U = \{z \in \mathcal{B}_b^0 : \|z\|_b < N_*\}$, From the choice of U, there is no $z \in \partial U$ such that $z \in \lambda\Phi_1 z$ for $\lambda \in (0,1)$.

Step 2 Φ_1 has a close graph.
Let $z^n \to z^*, h_0^n \in \Phi_1 z^n$ and $h_0^n \to h_0^*$, we shall prove that $h_0^* \in \Phi_1 z^*$. Indeed, If $h_0^n \in \Phi_1 z^n$ means that there exists $f_n \in S_{F,z^{(n)}}$, such that

$$h_0^n(t) = -T(t)g(0,\phi(0)) + \int_0^t AT(t-s)g(s, z_s^n + y_s)ds$$

$$+g(t, z_t^n + y_t) + \int_0^t T(t-s)f_n(s)ds + \sum_{0<t_k<t} T(t-t_k)I_k(z^n(t_k^-) + y(t_k^-)),$$

We must prove that there exists $f_* \in S_{F,z^*}$, such that

$$h_0^*(t) = -T(t)g(0,\phi(0)) + \int_0^t AT(t-s)g(s,z_s^* + y_s)ds + g(t,z_t^* + y_t)$$

$$+ \int_0^t T(t-s)f_*(s)ds + \sum_{0<t_k<t} T(t-t_k)I_k(z^*(t_k^-) + y(t_k^-)),$$

Clearly, we have

$$\Big\|\Big\{h_0^n(t) + T(t)g(0,\phi(0)) - \int_0^t AT(t-s)g(s,z_s^n + y_s)ds - g(t,z_t^n + y_t)$$

$$- \sum_{0<t_k<t} T(t-t_k)I_k(z^n(t_k^-) + y(t_k^-))\Big\} - \Big\{h_0^*(t) + T(t)g(0,\phi(0))$$

$$- \int_0^t AT(t-s)g(s,z_s^* + y_s)ds - g(t,z_t^* + y_t) - \sum_{0<t_k<t} T(t-t_k)I_k(z^*(t_k^-) + y(t_k^-))\Big\}\Big\|$$

$$\to 0, \text{ as } n \to \infty,$$

Consider the linear and continuous operator $\mathcal{K} : L^1(J,X) \to C(J,X)$ defined by

$$\mathcal{K}f(t) = \int_0^t T(t-s)f(s)ds$$

From Theorem 2, it follows that $\mathcal{K} \circ S_G$ is a closed graph operator, and

$$h_0^n(t) + T(t)g(0,\phi(0)) - \int_0^t AT(t-s)g(s,z_s^n + y_s)ds$$

$$-g(t,z_t^n + y_t) - \sum_{0<t_k<t} T(t-t_k)I_k(z^n(t_k^-) + y(t_k^-)) \in \mathcal{K} \circ S_{F,z^{(n)}}$$

Since $z^{(n)} \to z^*$ and $h_0^n \to h_0^*$, it follows from Theorem 2 that, there exist $f_* \in S_{F,z^*}$, such that

$$h_0^*(t) = -T(t)g(0,\phi(0)) + \int_0^t AT(t-s)g(s,z_s^* + y_s)ds + g(t,z_t^* + y_t)$$

$$+ \int_0^t T(t-s)f_*(s)ds + \sum_{0<t_k<t} T(t-t_k)I_k(z^*(t_k^-) + y(t_k^-)),$$

So we can conclude that Φ_1 has closed graph.

We define two the maps. The map $\mathcal{A} : U \to \mathcal{B}_b^0$ is defined by

$$h_1(t) = \begin{cases} 0, & \text{if } t \in (-\infty, 0], \\ -T(t)g(0,\phi(0)) + g(t,z_t + y_t) + \int_0^t AT(t-s)g(s,z_s + y_s)ds, \end{cases} \tag{5}$$

and $\mathcal{C} : U \to \mathcal{P}(\mathcal{B}_b^0)$ is defined by $\mathcal{C}z$ the set $h_2(t) \in \mathcal{B}_b^0$ such that

$$h_2(t) = \begin{cases} 0 \quad \text{if } t \in (-\infty, 0], \\ \int_0^t T(t-s)f(s)\,ds + \sum_{0<t_k<t} T(t-t_k)I_k(z(t_k^-) + y(t_k^-)), \end{cases} \tag{6}$$

Then $\Phi_1 = \mathcal{A} + \mathcal{C}$.

Step 3 , we will prove that Φ_1 is a condensing map.

It is easy to prove $\mathcal{A}$ and $\mathcal{C}$ maps bounded sets into bounded sets in $\mathcal{B}_b^0$.

$\mathcal{A}$ is a contraction on $\mathcal{B}_b^0$.

Let $z^1(t), z^2(t) \in U$, for $h_1(t) = \mathcal{A}z^1(t)$, $h_2(t) = \mathcal{A}z^2(t)$, by hypothesis (H3), we have

$$\|h_1(t) - h_2(t)\| \le L_2\{\|(-A)^{-\beta}\| + \frac{C_{1-\beta}b^\beta}{\beta}\}K_b\|z^1 - z^2\|_b$$

so

$$\|h_1(t) - h_2(t)\|_b \le L_0\|z^1 - z^2\|_b,$$

Since $L_0 < 1$, $\mathcal{A}$ is a contraction map. Also it is obviously that the map $\mathcal{A}$ has closed values.

$\mathcal{C}$ is convex for $z \in U$.

In fact, if $h_2^1(t), h_2^2(t)$ belong to $\mathcal{C}z$, then exist $f_1, f_2 \in S_{F,x}$ such that

$$h_2^i(t) = \int_0^t T(t-s)f_i(s)\,ds + \sum_{0<t_k<t} T(t-t_k)I_k(z(t_k^-) + y(t_k^-))$$

Since $F(t,z)$ has convex valued, for $0 \le \tau \le 1$, $[\tau f_1 + (1-\tau)f_2](t)$ belongs to $S_{F,x}$,

$$(\tau h_2^1(t) + (1-\tau)h_2^2(t)) = \sum_{0<t_k<t} T(t-t_k)I_k(z(t_k^-) + y(t_k^-))$$

$$+ \int_0^t T(t-s)[\tau f_1(s) + (1-\tau)f_2(s))]\,ds$$

Therefore, $(\tau h_2^1(t) + (1-\tau)h_2^2(t)) \in \mathcal{C}z$ and consequently $\mathcal{C}z$ has convex values.

$\mathcal{C}$ maps bounded sets into equicontinuous sets in U.

Let $z(t) \in U$ If $h_2(t) \in \mathcal{C}z(t)$, then exists $f \in S_{F,z}$, such that, for each $t \in J$

$$h_2(t) = \int_0^t T(t-s)f(s)\,ds + \sum_{0<t_k<t} T(t-t_k)I_k(z(t_k^-) + y(t_k^-))$$

Let $\tau_1, \tau_2 \in J, \tau_1 < \tau_2$. Then we have

$$\|h_2(\tau_2) - h_2(\tau_1)\| \le \int_{\tau_1}^{\tau_2} \|T(\tau_2 - s)f(s)\|ds + \sum_{\tau_1<t_k<\tau_2} M\beta_k$$

$$+ \left\{\int_0^{\tau_1-\epsilon} + \int_{\tau_1-\epsilon}^{\tau_1}\right\} \|T(\tau_2 - s) - T(\tau_1 - s)\|\|f(s)\|ds$$

$$+\sum_{0<t_k<\tau_1}\|T(\tau_2-t_k)-T(\tau_1-t_k)\|\beta_k.$$

We see that $\|h_2(\tau_2)-h_2(\tau_1)\|$ tends to zero as $(\tau_2-\tau_1)\to 0$ with ϵ sufficiently small, Since $T(t)$ is a strongly continuous operator and the compactness of $T(t)$ for $t>0$ implies the continuity in the uniform operator topology. Hence, $\mathcal{C}$ maps bounded sets into equicontinuous sets.

The equicontinuity for the cases $\tau_1<\tau_2\leq 0$, or $\tau_1\leq 0\leq\tau_2\leq b$ are similar.

Next we shall $\mathcal{C}$ maps U into a precompact set. Let $0<t\leq b$ be fixed and let ϵ be a real number satisfying $0<\epsilon<t$. For $z\in U$, we define

$$h_2^{\epsilon}(t)=T(\epsilon)\int_0^{t-\epsilon}T(t-\epsilon-s)f(s)\,ds$$
$$+T(\epsilon)\sum_{0<t_k<t-\epsilon}T(t-t_k-\epsilon)I_k(z(t_k^-)+y(t_k^-))$$

where $f\in S_{F,x}$, Since $T(t)$ is a compact operator, the set $Y_\epsilon(t):=\{h_2^{\epsilon}(t):z\in U\}$ is precompact for every $\epsilon, 0<\epsilon<t$. Also, for every $z\in U$, we have

$$\|h_2(t)-h_2^{\epsilon}(t)\|\leq\int_{t-\epsilon}^{t}\|T(t-s)\|\|f(s)\|ds+M\sum_{t-\epsilon<t_k<t}\beta_k$$

The right hand side of the above inequality tends to zero as $\epsilon\to 0$. Since there are precompact sets arbitrarily close to the set $Y(t):=\{h_2(t):\ z\in U\}$. Hence the set $Y(t)$ is precompact in X. By Arzelá-Ascoli theorem, we conclude that $\mathcal{C}$ $U\to\mathcal{P}(\mathcal{B}_b^0)$ is completely continuous.

These arguments enable us to conclude that $\Phi_1=\mathcal{A}+\mathcal{C}$ is a condensing map.

So, All of the conditions of Theorem 1 are satisfied. Therefore, the system (1) has at least one solution.

References

1. Benchohra, M., Gatsori, E.P., Ntouyas, S.K.: Exitence Results for Functional and Neutral Functional Integrodifferential Inclusions with Lower Semicontinuous Right-hand Side. J. Math. Anal. Appl. 281, 525–538 (2003)
2. Benchohra, M., Ntouyas, S.K.: Nonlocal Cauchy problems for Neutral Functional Differential and Integrodifferential Inclusions in Banach Spaces. J. Math. Anal. Appl. 258, 573–590 (2001)
3. Benchohra, M., Henderson, J., Ntouyas, S.K.: Existence results for impulsive multivalued semilinear neural functional differential inclusions in Banach spaces. J. Math. Anal. Appl. 263, 763–780 (2001)
4. Benchohra, M., Ntouyas, S.K.: An Existence Result on Noncompact Intervals to First Order Integrodifferential Inclusions in Banach Spaces. Libertas Math. 20, 71–78 (2000)
5. Benchohra, M., Gatsori, E.P., Ntouyas, S.K.: Controllbility of Semilinear Evolution Inclusions with Nonlocal Conditions. J. Optim. Theory Appl. 118, 493–513 (2003)
6. Benchohra, M., Ntouyas, S.K.: Controllbility Results for Multivalued Semilinear Neutral Functional Differential Equations. Math. Sci. Res. J. 6, 65–77 (2002)

7. Fitzpatrick, P.M., Petryshyn, W.V.: Fixed Point Theorems for Multivalued Noncompact Acyclic Mappings. Pacific J. Math. 54, 17–23 (1974)
8. Deimling, K.: Multivalued Differential Equations. de Gruyter, Berlin/New York (1992)
9. Regan, D.: Nonlinear Alternatives for Multivalued Maps with Applications to Operator Inclusions in Abstract Spaces. Proc. of the Amer. Math. Society 127, 3557–3564 (1999)
10. Pazy, A.: Semigroups of Linear Operators and Applications to Partial Differential Equations. In: Applied Methematical Sciences. Springer, New York (1983)
11. Hino, Y., Murakami, S., Naito, T.: Functional Differential Equations with Infinite Delay. Lecture Notes in Mathematics. Springer, New York (1991)
12. Carmichael, N., Quinn, M.D.: An Approach to Nonlinear Control Problems Using Fixed-point Methods, Degree Theory, and Pseodo-Inverses. Numer. Funct. Anal. Optim. 7, 197–219 (1984-1985)
13. Lasota, A., Opial, Z.: An Application of the Kakutani-Ky Fan Theorem in the Theory of Ordinary Differential Equations. Bull. Acad. Pol. Sci. Ser. Sci. Math. Astronom. Phys. 13, 781–786 (1965)

Finite Time Stability of Cohen-Grossberg Neural Network with Time-Varying Delays

Dingguo Jiang

College of Water Conservancy and Hydropower Engineering
Hohai University, Nanjing, China, 210098

Abstract. This paper considers the problem of finite time stability (FTS) of the Cohen-Grossberg neural networks with or without delay. Based on the Lyapunov function and linear matrix inequality (LMI) technique, some delay-dependent and delay-independent criterions are derived to guarantee finite-time stability. Finally, one example is given to demonstrate the validity of the proposed methodology and to show the differences between globally exponential stability and finite-time stability.

Keywords: Finite time stability, Lyapunov function, LMI, Time-varying delay.

1 Introduction

In the past few decades, neural networks such as Hopfield neural network [1], cellular neural network [2,3], Cohen-Grossberg neural networks [8-19] and bi-directional associative memory neural network [20-22] have attracted the attention of many mathematicians, physicists, and computer scientists due to their wide range of applications in, for example, signal processing, pattern recognition, associative memory, and combinatorial optimization. Among them, the Cohen-Grossberg neural network is an important one. In 1983, Cohen-Grossberg proposed and studied a kind of neural network, which is now called Cohen-Grossberg neural network(CGNN). It can be described by the ordinary differential equation of the form:

$$\dot{x}_i(t) = \alpha_i(x_i(t))[-a_i(x_i(t)) + \sum_{j=1}^{m} a_{ij} f_j(x_j(t)) + I_i(t)] \qquad (i = 1, \ldots, m) \quad (1)$$

So many scholars have studied the stability of Cohen-Grossberg neural networks via Lyapunov stability theory and various inequalities and get a large number of sufficient conditions. However, most of the results in this field relate to stability defined over an infinite time interval. In many practical applications, however, the main concern is the behavior of the system over a fixed finite time interval. In this sense it is reasonable to define as stable a system whose state, given some initial conditions, remains within prescribed bounds in the fixed time

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 522–531, 2009.

interval, and as unstable a system which does not. In order to deal with this problem, Peter Dorato [4] presented the concept of finite time stability (FTS). Many researchers[5,25] have studied the finite time stability problem of linear systems. But fewer scholars[23,24] studied the finite time stability problem of nonlinear systems.

Recently, LMI-based techniques have been successfully used to tackle various stability problems for neural networks. The main advantage of the LMI-based approaches is that the LMI stability conditions can be solved numerically using the effective interior-point algorithm [3]. But fewer scholars[23,24] studied the finite time stability problem of neural network systems. Thus, in this paper, we shall consider the finite time stability system of (1) via LMI approach and Lyapunov approach. Also we consider the Cohen-Grossberg neural networks with delay described as the following form:

$$\dot{x}_i(t) = \alpha_i(x_i(t))[-a_i(x_i(t)) + \sum_{j=1}^{m} a_{ij} f_j(x_j(t)) + \sum_{j=1}^{n} b_{ij} f_j(x_j(t-\tau_j)) + I_i(t)] \quad (2)$$

where $i = 1, \ldots, m$, m is the number of neurons in the network, x_i denotes the state variable associated with the neuron and a_i is an appropriately behaved function. The connection matrix $A = (a_{ij})_{m\times m}$ tells us how the neurons are connected in the network and the activation functions f_j shows how neurons respond to each other. Where $B = (b_{ij})_{m\times m}$ indicate the strength of the neuron interconnections within the network with time delay parameters τ_j.

The rest of this paper is organized as follows: in Section 2 we give some Preliminaries and Lemmas; in Section 3 we will derive a sufficient condition to ensure the equilibrium of system (1) is finite time stable; next, in Section 4 we obtain a sufficient condition to ensure the equilibrium of system (2) is finite time stable; we shall consider the Cohen-Grossberg network with time-varying delays in Section 5; one example is given to illustrate the effectiveness of our results in Section 6; finally, in Section 7 the concluding remarks are given.

2 Preliminaries and Lemmas

Throughout this paper,we need the following assumptions:

(H_1) Functions $\alpha_i(x_i(t))$ are bounded, positive and satisfy: $0 \le a_i^- \le \alpha_i(x_i(t) \le a_i^+ < \infty$.

(H_2) Behaved functions $a_i(x_i(t))$ are bounded and continuous; furthermore there exist exist $\gamma_i > 0$ $i = 1, \ldots, m$, such that $x_i(t)a_i(x_i(t)) \ge \gamma_i x_i^2(t)$.

(H_3) Activation functions f_i are bounded or globally Lipschitz continuous and there exist Lipschitz constants σ_i such that $(f_i(\mu) - f_i(\nu))(\mu - \nu) \le \sigma_i(\mu - \nu)^2$.

Throughout this letter, Let $A(a_{ij})_{m\times m}, B = (b_{ij})_{m\times m}$ and I_i are continuous functions on $\mathcal{R}$. And $a^+ = \max_{1\le i\le m}\{\alpha_i^+\}$, $a^- = \max_{1\le i\le m}\{\alpha_i^-\}$, $\Sigma = diag(\sigma_1, \ldots, \sigma_m)$, $\gamma_m = \min_{1\le i\le m}\{\gamma_i\}$, $\sigma_M = \max_{1\le i\le m}\{\sigma_i\}$, $\tau = \max_{1\le j\le m}\{\tau_j\}$, $\Gamma = diag(\gamma_1, \ldots, \gamma_m)$. Define $\|A\|_1 = \max_{1\le j\le m}\sum_{i=1}^{m}|a_{ij}|$ to be the matrix's 1-norm. The initial condition associated with (1) is $x(t) = x(0)$, for $t \in R$. The initial condition associated with (2) is $x(t) = \phi(t)$, for $t \in [-\tau, 0]$.

Definition 1. (FTS) System (1) is said to be finite time stable with respect to (c_1, c_2, T) $(c_1 < c_2)$, if $x^T(0)x(0) \leq c_1 \Rightarrow x^T(t)x(t) < c_2, \forall t \in [0, T]$.

Definition 2. (FTS) System (2) is said to be finite time stable with respect to (c_1, c_2, T) $(c_1 < c_2)$, if $\sup_{t\in[-\tau,0]} \phi^T(t)\phi(t) \leq c_1 \Rightarrow x^T(t)x(t) < c_2, \forall t \in [0, T]$.

Lemma 1. For any vectors $a, b \in R^n$ and any positive definite matrix $Y \in R^{n\times n}$, the following inequality holds $2a^Tb \leq a^TYa + b^TY^{-1}b$.

Lemma 2. For a symmetric matrix $S = (S_{ij})_{2\times 2}$, the following conditions are equivalent:

(*i*) $S < 0$,
(*ii*) $S_{11} < 0$ and $S_{22} - S_{12}^T S_{11}^{-1} S_{12} < 0$,
(*iii*) $S_{22} < 0$ and $S_{11} - S_{12} S_{22}^{-1} S_{12}^T < 0$.

3 Finite Time Stable of CGNN Network without Delay

In this section, we shall consider the finite time stable of system (1) without delay. Let $x^* = (x_1^*), \ldots, x_m^*)^T$ is the equilibrium of system (1) and $x = (x_1, \ldots, x_m)^T$ is an arbitrary solution of system (1). Then, by setting $u = x - x^* = (x_1 - x_1^*, \ldots, x_m - x_m^*)$, system (1) can be transformed into:

$$\dot{u}_i(t) = \alpha_i(u_i(t))[-a_i(u_i(t)) + \sum_{j=1}^{n} a_{ij} g_j(u_j(t))] \qquad (i = 1, \ldots, m), \tag{3}$$

where $\alpha_i(u_i(t)) = \alpha_i(x_i(t) + x_i^*)$, $a_i(u_i(t)) = a_i(x_i(t)) - a_i(x_i^*)$, $g_j(u_j(t)) = f_j(x_j(t)) - f_j(x_j^*)$. Let $u(t) = (u_1(t), \ldots, u_m(t))^T$, $A = (a_{ij})_{m\times n}$, $a(u(t)) = diag(a_1(u_1(t)), \ldots, a_m(u_m(t)))$, $\alpha(u(t)) = diag(\alpha_1(u_1(t)), \ldots, \alpha_m(u_m(t)))$, $g(u(t)) = (g_1(u_1(t)), \ldots, g_m(u_m(t)))^T$, then (3) can be rewritten into the following vector form:

$$\dot{u}(t) = \alpha(u(t))[-a(u(t)) + Ag(u(t))]. \tag{4}$$

*Theorem 1.*System (4) is FTS with respect to $(c1, c2, T)$ if there exists a diagonal positive definite matrix matrices $P > 0$, a symmetric positive definite matrix $Y > 0$ and a positive scalar α such that the following conditions hold:

$$\Omega = \begin{bmatrix} -2P\Gamma + \Sigma^T Y \Sigma - \alpha P & A^T P \\ PA & -Y \end{bmatrix} < 0 \tag{5}$$

and

$$\frac{e^{\alpha a^+ T} c_1 \lambda_{max}(P) a^+}{\lambda_{min}(P) a^-} < c_2 \tag{6}$$

where $P = diag(p_1, \ldots, p_m)$.

Proof. Let us consider a function described as $V(u(t))=2\sum_{i=1}^m p_i \int_0^{u_i(t)} \frac{s}{\alpha_i(s)} ds$. Then, the time derivative of $V(u(t))$ along the solution of (3) gives

$$\dot{V} = 2\sum_{i=1}^{m} p_i \frac{\dot{u}_i(t)}{\alpha_i(u_i(t))} = 2\sum_{i=1}^{m} p_i u_i(t)[-a_i(u_i(t)) + \sum_{j=1}^{n} a_{ij} g_j(u_j(t))]$$

$$\begin{aligned}
&\leq 2\sum_{i=1}^{m}[-p_i\gamma_i u_i^2(t) + p_i u_i(t)\sum_{j=1}^{n} a_{ij}g_j(u_j(t))] \\
&\leq -2u^T(t)P\Gamma u(t) + 2u^T(t)PAg(u(t)) - g^T(u(t))Yg(u(t)) + u^T(t)\Sigma^T Y\Sigma u(t) \\
&= (u^T(t), g^T(u(t)))\Omega_1 \begin{pmatrix} u(t) \\ g(u(t)) \end{pmatrix} \qquad (7)
\end{aligned}$$

where

$$\Omega_1 = \begin{bmatrix} -2P\Gamma + \Sigma^T Y\Sigma & A^T P \\ PA & -Y \end{bmatrix} \qquad (8)$$

According to

$$\Omega = \Omega_1 + \begin{bmatrix} -\alpha P & 0 \\ 0 & 0 \end{bmatrix} < 0 \qquad (9)$$

We have $\dot{V} < \alpha u^T(t)Pu(t)$. According to the definition of Lyapunov function $V(u(t))$, we can have:

$$\frac{1}{a^+}u^T(t)Pu(t) \leq 2\sum_{i=1}^{m} p_i \int_0^{u_i(t)} \frac{s}{\alpha_i(s)}ds = V(u(t)) \qquad (10)$$

Thus $u^T(t)Pu(t) \leq a^+V(u(t))$ holds. Thereby we have

$$\dot{V} < \alpha u^T(t)Pu(t) \leq \alpha a^+V(u(t)) \qquad (11)$$

Multiplying (11) by $e^{-\alpha a^+ t}$, it obtains that

$$e^{-\alpha a^+ t}\dot{V} - \alpha a^+ e^{-\alpha a^+ t}V(u(t)) < 0 \qquad (12)$$

Furthermore

$$\frac{d}{dt}(e^{-\alpha a^+ t}V) < 0 \qquad (13)$$

Integrating (13) from 0 to t , with $t \in [0, T]$, we have $e^{-\alpha a^+ t}V(u(t)) < V(u(0))$. Therefore

$$\begin{aligned}
\frac{1}{a^+}\lambda_{min}(P)u^T(t)u(t) &\leq \frac{1}{a^+}u^T(t)Pu(t) \leq V(u(t)) \\
&\leq \frac{1}{a^-}u^T(t)Pu(t) \leq \frac{1}{a^-}\lambda_{max}(P)u^T(t)u(t) \qquad (14)
\end{aligned}$$

So

$$\begin{aligned}
\frac{1}{a^+}\lambda_{min}(P)u^T(t)u(t) \leq e^{\alpha a^+ t}V(u(0)) &\leq \frac{e^{\alpha a^+ t}}{a^-}\lambda_{max}(P)u^T(0)u(0) \\
&\leq \frac{e^{\alpha a^+ T}}{a^-}\lambda_{max}(P)u^T(0)u(0) \qquad (15)
\end{aligned}$$

Namely

$$u^T(t)u(t) \leq \frac{e^{\alpha a^+ T} c_1 \lambda_{max}(P) a^+}{\lambda_{min}(P) a^-} \tag{16}$$

Condition (6) implies, for all $t \in [0, T]$, $u^T(t)u(t) < c2$. Therefore, the proof is completed.

4 FTS of Neural Network System with Delay

In this section, we consider system (2) with delay on the basis of the proof of Theorem 1. Let $x^* = (x_1^*, \ldots, x_m^*)^T$ is the equilibrium of system (2) and $x = (x_1, \ldots, x_m)^T$ is an arbitrary solution of system (2). Then, by setting $u = x - x^* = (x_1 - x_1^*, \ldots, x_m - x_m^*)$, and then system (1) can be transformed into:

$$\dot{u}_i(t) = \alpha_i(u_i(t))[-a_i(u_i(t)) + \sum_{j=1}^{n} a_{ij} g_j(u_j(t)) + \sum_{j=1}^{n} b_{ij} g_j(u_j(t - \tau_j))] \tag{17}$$

where the definition of every parameter is similar to the definition in Section 3, then (17) can be rewritten into the following vector form:

$$\dot{u}(t) = \alpha(u(t))[-a(u(t)) + Ag(u(t)) + Bg(u(t - \tau))]. \tag{18}$$

Theorem 2. System (17) is FTS with respect to $(c1, c2, T)$ if there exists a diagonal positive definite matrix matrices $P > 0$ and two symmetric positive definite matrices $Q > 0, Y > 0$ and a positive scalar α such that the following conditions hold:

$$\Omega = \begin{bmatrix} -2P\Gamma + \Sigma^T Y \Sigma - \alpha P & A^T P & B^T P \\ PA & Q - Y & 0 \\ PB & 0 & -Q \end{bmatrix} < 0 \tag{19}$$

and

$$\frac{e^{\alpha a^+ T} c_1 a^+ (\lambda_{max}(P) + \lambda_{max}(Q)\lambda_{max}(\Sigma^T \Sigma) a^- \tau)}{a^- \lambda_{min}(P)} < c_2 \tag{20}$$

Proof. Let us consider another function described as

$$V(u(t)) = 2 \sum_{i=1}^{m} p_i \int_0^{u_i(t)} \frac{s}{\alpha_i(s)} ds + \int_{t-\tau}^{t} g^T(u(t)) Q g(u(t)) dt \tag{21}$$

Then, the time derivative of $V(u(t))$ along the solution of (17) gives

$$\begin{aligned} \dot{V} &= 2 \sum_{i=1}^{m} p_i \frac{\dot{u}_i(t)}{\alpha_i(u_i(t))} + g^T(u(t)) Q g(u(t)) - g^T(u(t - \tau)) Q g(u(t - \tau)) \\ &= 2 \sum_{i=1}^{m} p_i u_i(t)[-a_i(u_i(t)) + \sum_{j=1}^{n} a_{ij} g_j(u_j(t)) + \sum_{j=1}^{n} b_{ij} g_j(u_j(t - \tau_j))] \end{aligned}$$

$$
\begin{aligned}
&+g^T(u(t))Qg(u(t)) - g^T(u(t-\tau))Qg(u(t-\tau)) \\
&\leq -2\sum_{i=1}^{m} p_i\gamma_i u_i^2(t) + 2u^T PAg(u(t)) + 2u^T PBg(u(t-\tau)) \\
&\quad +g^T(u(t))Qg(u(t)) - g^T(u(t-\tau))Qg(u(t-\tau)) \\
&\leq -2u^T(t)P\Gamma u(t) + 2u^T(t)PAg(u(t)) + 2u^T PBg(u(t-\tau)) + g^T(u(t))Qg(u(t)) \\
&\quad -g^T(u(t-\tau))Qg(u(t-\tau)) - g^T(u(t))Yg(u(t)) + u^T(t)\Sigma^T Y\Sigma u(t) \\
&= (u^T(t), g^T(u(t)))\Omega_1 \begin{pmatrix} u(t) \\ g(u(t)) \end{pmatrix}
\end{aligned} \tag{22}
$$

Noting that the last inequality holds because $g^T(u(t))Yg(u(t)) \leq u^T(t)\Sigma^T Y\Sigma u(t)$ holds due to Y is positive definite and using assumption (H_3). Thus

$$
\Omega_1 = \begin{bmatrix} -2P\Gamma + \Sigma^T Y\Sigma & A^T P & B^T P \\ PA & Q - Y & 0 \\ PB & 0 & -Q \end{bmatrix} \tag{23}
$$

Since

$$
\Omega = \Omega_1 + \begin{bmatrix} -\alpha P & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} < 0, \tag{24}
$$

we have $\dot{V} < \alpha u^T(t)Pu(t)$. According to the definition of Lyapunov function $V(u(t))$, we can have:

$$
\frac{1}{a^+}u^T(t)Pu(t) \leq 2\sum_{i=1}^{m} p_i \int_0^{u_i(t)} \frac{s}{\alpha_i(s)}ds \leq V(u(t)) \tag{25}
$$

Thus $u^T(t)Pu(t) \leq a^+V(u(t))$ holds. From (25),

$$
\dot{V} < \alpha u^T(t)Pu(t) \leq \alpha a^+ V(u(t)) \tag{26}
$$

Using the same method in Theorem 1, we can derive

$$
\frac{d}{dt}(e^{-\alpha a^+ t}V) < 0 \tag{27}
$$

Integrating (27) from 0 to t, with $t \in [0, T]$, we have

$$
e^{-\alpha a^+ t}V(u(t)) < V(u(0)) \tag{28}
$$

Since

$$
V(u(0)) = 2\sum_{i=1}^{m} p_i \int_0^{u_i(0)} \frac{s}{\alpha_i(s)}ds + \int_{-\tau}^{0} g^T(u(t))Qg(u(t))dt
$$

$$\leq 2\sum_{i=1}^{m} p_i \int_0^{u_i(0)} \frac{s}{\alpha_i^-} ds + \lambda_{max}(Q) \int_{-\tau}^{0} g^T(u(t))g(u(t))dt$$

$$\leq \frac{1}{\alpha^-} u^T(0)Pu(0) + \lambda_{max}(Q)\lambda_{max}(\Sigma^T\Sigma) \int_{-\tau}^{0} u^T(t)u(t)dt$$

$$\leq \frac{\lambda_{max}(P)}{\alpha^-} u^T(0)u(0) + \lambda_{max}(Q)\lambda_{max}(\Sigma^T\Sigma)c_1\tau$$

$$= c_1[\frac{\lambda_{max}(P)}{\alpha^-} + \lambda_{max}(Q)\lambda_{max}(\Sigma^T\Sigma)\tau] \tag{29}$$

Therefore

$$\frac{1}{a^+}\lambda_{min}(P)u^T(t)u(t) \leq \frac{1}{a^+}u^T(t)Pu(t) \leq V(u(t)) \tag{30}$$

Using the same method in Theorem 1, we obtain

$$\frac{1}{a^+}\lambda_{min}(P)u^T u \leq e^{\alpha a^+ t}V(u(0)) \leq c_1[\frac{\lambda_{max}(P)}{\alpha^-} + \lambda_{max}(Q)\lambda_{max}(\Sigma^T\Sigma)\tau] \tag{31}$$

Namely

$$u^T(t)u(t) \leq e^{\alpha a^+ t}V(u(0)) \leq c_1 a^+[\frac{\lambda_{max}(P)}{\alpha^-\lambda_{min}(P)} + \frac{\lambda_{max}(Q)\lambda_{max}(\Sigma^T\Sigma)\tau]}{\lambda_{min}(P)} \tag{32}$$

Condition (20) implies, for all $t \in [0, T]$, $u^T(t)u(t) < c_2$. Therefore, this complete the proof.

5 FTS of Neural Networks with Time-Varying Delay

In this section, we consider the following system with time-varying delay on the basis of the proof of Theorem 2. Now we only study the following vector form:

$$\dot{u}(t) = \alpha(u(t))[-a(u(t)) + Ag(u(t)) + Bg(u(t - \tau(t)))] \tag{33}$$

where the time-varying delay satisfy $\tau = \sup_{t\in R}\tau_j(t)$ and $\dot{\tau}_j(t) \leq \mu \leq 1$.
*Theorem 3.*System (33) is FTS with respect to $(c1, c2, T)$ if there exists a diagonal positive definite matrix matrices $P > 0$ and two symmetric positive definite matrices $Q > 0, Y > 0$ and a positive scalar α such that the following conditions hold:

$$\Omega = \begin{bmatrix} -2P\Gamma + \Sigma^T Y\Sigma - \alpha P & A^T P & B^T P \\ PA & Q - Y & 0 \\ PB & 0 & -(1-\mu)Q \end{bmatrix} < 0 \tag{34}$$

and

$$\frac{e^{\alpha a^+ T}c_1 a^+(\lambda_{max}(P) + \lambda_{max}(Q)\lambda_{max}(\Sigma^T\Sigma)a^-\tau)}{a^-\lambda_{min}(P)} < c_2 \tag{35}$$

Proof. Let us consider another function described as

$$V(u(t)) = 2\sum_{i=1}^{m} p_i \int_0^{u_i(t)} \frac{s}{\alpha_i(s)} ds + \int_{t-\tau(t)}^{t} g^T(u(t))Qg(u(t))dt \tag{36}$$

Then, the time derivative of $V(u(t))$ along the solution of (33) gives

$$\begin{aligned}
\dot{V} &= 2\sum_{i=1}^{m} p_i \frac{\dot{u}_i}{\alpha_i(u_i)} + g^T(u)Qg(u) - (1-\dot{\tau}(t))g^T(u(t-\tau(t)))Qg(u(t-\tau(t))) \\
&\leq 2\sum_{i=1}^{m} p_i u_i[-a_i(u_i) + \sum_{j=1}^{n} a_{ij}g_j(u_j) + \sum_{j=1}^{n} b_{ij}g_j(u_j(t-\tau_j(t)))] \\
&\quad + g^T(u)Qg(u) - (1-\mu)g^T(u(t-\tau(t)))Qg(u(t-\tau(t))) \\
&\leq -2\sum_{i=1}^{m} p_i\gamma_i u_i^2(t) + 2u^T PAg(u) + 2u^T PBg(u(t-\tau(t))) \\
&\quad + g^T(u)Qg(u) - (1-\mu)g^T(u(t-\tau(t)))Qg(u(t-\tau(t))) \\
&\leq -2u^T P\Gamma u + 2u^T PAg(u) + 2u^T PBg(u(t-\tau(t))) + g^T(u)Qg(u) \\
&\quad -(1-\mu)g^T(u(t-\tau(t)))Qg(u(t-\tau(t))) - g^T(u)Yg(u) + u^T\Sigma^T Y\Sigma u \\
&= (u^T(t), g^T(u(t)))\Omega_1 \begin{pmatrix} u(t) \\ g(u(t)) \end{pmatrix}
\end{aligned} \tag{37}$$

where

$$\Omega_1 = \begin{bmatrix} -2P\Gamma + \Sigma^T Y\Sigma & A^T P & B^T P \\ PA & Q-Y & 0 \\ PB & 0 & -(1-\mu)Q \end{bmatrix}$$

In this section we used u instead of $u(t)$ in order to the convenience of description. The remaining part of the proof is similar to that of Theorem 2 and its omitted here. This completes the proof.

It is easy to check that condition in Theorem 2 related to the matrix P and Q is guaranteed by imposing the conditions $\lambda_1 I \leq P \leq \lambda_2,\ \lambda_3 I < Q < \lambda_4 I$. So inequality (6) can be converted to an LMI $c_1 a^+\lambda_2 - e^{-\alpha a^+ T}c_2 a^-\lambda_1 < 0$. Similarly inequality (20)and (35) can be converted to an LMI

$$c_1 a^+\lambda_2 + c_1 a^+\lambda_4\lambda_{max}(\Sigma^T\Sigma)a^-\tau - e^{-\alpha a^+ T}c_2 a^-\lambda_1 < 0$$

Form a computation point of view, it is important to notice that, once we have fixed the value of α the feasibility of the conditions stated in Theorem 1-3 can be turned into LMIs based feasibility problem.

6 A Numerical Example

In this section, some numerical examples are presented to illustrate the feasibility and effective of our results. Consider system (33) with parameters as:

$$\alpha(u(t)) \equiv 1,\ a(u(t)) = -\begin{pmatrix} 4 & 0 \\ 0 & 6 \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix},\ A = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix},\ B = \begin{pmatrix} 1 & -2 \\ -0.5 & 1 \end{pmatrix}$$

and $c_1 = 1, c_2 = 9, f_1(u) = f_2(u) = \frac{1}{2}(|u+1| - |u-1|), \tau_1(t) = \tau_2(t) = \frac{1}{2}sin(t)$. It is clear that

$$\Gamma = \begin{pmatrix} 4 & 0 \\ 0 & 6 \end{pmatrix}, \Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, A = \begin{pmatrix} -1 & 0 \\ 0 & -2 \end{pmatrix}, B = \begin{pmatrix} -1 & 2 \\ 0.5 & -1 \end{pmatrix}$$

We can get $\gamma_m = 4, a^- = 1, a^+ = 1, \sigma_M = 1,\ \|A\|_1 = 2, \|B\|_1 = 3$. Since $\|A\|_1 + \|B\|_1 > \frac{\gamma_m a^-}{\sigma_M a^+}$, so the globally exponentially stable condition of Theorem 3 in [1] does not hold. However, using the conditions of Theorem 3 in our paper, we found that the system is FTS with respect to (c_1, c_2, T) for a maximum $T_{max} = 0.8s$, obtain for $\alpha = 1.2$ with

$$P = \begin{pmatrix} 0.6777 & 0 \\ 0 & 0.6777 \end{pmatrix}, Q = \begin{pmatrix} 1.1584 & -0.1442 \\ -0.1442 & 0.8537 \end{pmatrix}, Y = \begin{pmatrix} 2.3338 & -48.7926 \\ 48.7926 & 2.4563 \end{pmatrix}$$

7 Conclusions

This paper has studied the problems of finite time stability of the Cohen-Grossberg neural networks with or without delay. Based on LMI technique, some sufficient conditions are derived. Two example have been provided to illustrates the proposed methodology and also to show that the differences between globally exponential stability and finite-time stability.

References

1. Liao, X.X.: Stability of Hopfield-type neural networks(I). Science in China(Series A), 407–418 (1995)
2. Zhao, H.Y., Cao, J.: New Conditions for Global Exponential Stability of Cellular Neural Networks with Delays. Neural Networks 18, 1332–1340 (2005)
3. Liao, X.X.: Mathematical Theory of Cellular Neural Networks(II). Science in China (Series A), 542–551 (1995)
4. Dorato, P.: Short Time Stability in Linear Time-Varying System. In: Proc. IRE International Convention Record Part, vol. 4, pp. 83–87 (1961)
5. Amato, F., Ariola, M., Abdallah, C.T., Dorato, P.: Dynamic Output Feedback FiniteTimeControl of LTI SystemsSubject to Parametric Uncertainties and Disturbances. In: Proc. European control Conference, Karlsruhe, CA, pp. 1176–1180 (1999)
6. Boyd, S., Ghaoui, E.L., Feron, E., Balakrishnan, V.: Linear Matrix Inequalities in System and Control Theory. SIAM, Philadelphia (1994)

7. Arik, S., Orman, Z.: Global Stability Analysis of CohenCGrossberg Neural Networks with Time Varying Delays. Physics Letters A 341, 410–421 (2005)
8. Ou, O.: Global Robust Exponential Stability of Delayed Neural Networks: An LMI Approach. Chaos. Solutions and Fractals 32(5), 1742–1748 (2007)
9. Song, Q.K., Cao, J.: Stability Analysis of Cohen-Grossberg Neural Network with Both Time-varying and Continuously Distributed delays. Journal of Computational and Applied Mathematics 197, 188–203 (2006)
10. Jiang, H.H., Cao, J., Teng, Z.H.: Dynamics of Cohen-Grossberg Neural Networks with Time-varying Delays. Physics Letters A 354, 414–422 (2006)
11. Liu, B.W., Huang, L.H.: Existence and Exponential Stability of Periodic Solutions for a Class of Cohen-Grossberg Neural Networks with Time-varying Delays. Chaos, Solutions, Fractals 32(2), 95–103 (2007)
12. Cao, J., Liang, J.L.: Boundedness and Stability for Cohen-Grossberg Neural Network with Time-varying Delays. Journal of Mathematical Analysis and Applications 296, 665–685 (2004)
13. Li, C.D., Liao, X.F., Zhang, R.: Dealy-dependent Exponential Stability Analysis of Bi-directional Associative Memory Neural Networks with Time Delay: an LMI approach. Chaos, Solution and Fractals 24, 1119–1134 (2005)
14. Li, C.D., Liao, X.F.: New Algebraic Conditions for Global Exponential Stability of Delayed Recurrent Neural Networks. Neurocomputing 64, 319–333 (2005)
15. Sun, C.Y., Feng, C.B.: Exponential Periodicity and Stability of Delayed Neural Networks. Mathematics and Computers in Simulation 66, 469–478 (2004)
16. Sun, C.Y., Zhang, K.J., Fei, S.M., Feng, C.B.: On Exponential Stability of Delayed Neural Networks with a General Class of Activation Functions. Physics Letters A 298, 122–132 (2002)
17. Chen, B.S., Wang, J.: Global Exponential Periodicity and Global Exponential Stability of a Class of Recurrent Neural Networks. Physics Letters A 329, 36–48 (2004)
18. Cao, J., Ho, W., Daniel, C.: A General Framework for Global Asymptotic Stability Analysis of Delayed Neural Networks Based on LMI Approach Chaos. Solitons and Fractals 24, 1317–1329 (2005)
19. Xiong, W.J., Cao, J.: Absolutely Exponential Stability of CohenCGrossberg Neural Networks with Unbounded Delays. Neurocomputing 68, 1–12 (2005)
20. Chen, A.P., Cao, J.: Periodic Bi-directional Cohen-Grossberg Neural Networks with Distributed Delays. Nonlinear Analysis 66(12), 2947–2961 (2007)
21. Xia, Y.H., Cao, J., Lin, M.R.: New Results on the Existence and Uniqueness of almost periodic solution for BAM neural networks with Continuously Distributed Delays. Chaos, Solitons and Fractals 31, 928–936 (2007)
22. Liu, Y.R., Wang, Z.D., Liu, X.H.: Global Asymptotic Stability of Generalized Bi-directional Associative Memory Networks with Discrete and Distributed Delays. Chaos, Solitons and Fractals 28, 793–803 (2006)
23. Weiss, L.: Finite Time Stability under Perturbing Forces and on Product Spaces. IEEE Transactions on Automatic Control 12, 54–59 (1967)
24. Onori, S., Dorato, P., Galeani, S., Abdallah, C.T.: Finite Time Stability Design via Feedback Linearization. In: 44th IEEE Conference on Decision and Control and the European Control Conference, pp. 4915–4920 (2005)
25. Ebihara, Y., Hagiwara, T.: Robust Controller Synthesis with Parameter-dependent Lyapunov Variables: A Dilated LMI Approach. In: Proceedings of the 41st IEEE conference on decision and control, pp. 4179–4178 (2002)

Evolution Differential Systems with Impulse and Delay

Yan Li

College of Science, Huazhong Agriculture University
Wuhan, Hubei 430079, China
ly@mail.hzau.edu.cn

Abstract. In this paper, we discuss a class of impulsive evolution neutral functional differential inclusions with infinite delay. The existence of mild solutions of these evolution neutral differential systems is obtained by using the new nonlinear alternatives of Leray-Schauder type.

Keywords: Evolution differential systems, Impulses, Delays, Existence.

1 Introduction

In this paper, we shall disscuss the existence of mild solutions of some impulsive evolution neutral functional differential inclusions with infinite delay in Banach spaces described in the form

$$\begin{cases} \dfrac{d}{dt}[x - g(t, x_t)] \in A(t)x(t) + F(t, x_t),\ t \in J, t \neq t_k \\ \qquad \Delta x(t_k) = I_k(x(t_k^-)), \quad k = 1, 2, \ldots, m, \\ \qquad x_0 = \phi(t) \in \mathcal{B}, \end{cases} \tag{1}$$

where $J := [0, b]$, X denotes a separable Banach space with norm $\|\cdot\|$. $A(t)$ generates an evolution system, the function $x_t : (-\infty, 0] \to X$, $x_t(\theta) = x(t + \theta)$, belongs to some abstract phase space $\mathcal{B}$ defined axiomatically; $g : J \times \mathcal{B} \to X$ and $F : J \times \mathcal{B} \to \mathcal{P}(X)$; $\mathcal{P}(X)$ denotes the class of all nonempty subsets of X. $0 = t_0 < t_1 < \cdots < t_m < t_{m+1} = b$, $I_k : X \to X (k = 1, 2, \ldots, m)$, $\Delta x(t_k) = x(t_k^+) - x(t_k^-)$, $x(t_k^+)$ and $x(t_k^-)$ are respectively the right and the left limit of x at $t = t_k$, and $x(t_k^+) = x(t_k)$.

The theory of impulsive differential and partial differential equations has become an more important area of investigation in recent years; see the papers of Erbe et al. [1], Rogochenko [3], Liu [4] and the survey papers of Rogochenko [2], and the references therein.

The problem of the existence and controllability for differential inclusions has been extensively studied [5]–[7]. Benchohra et al.[5] considered the existence of solutions for functional and neutral functional inclusions. Li et al.[6] discussed the controllability of impulsive functional systems. And Benchohra et al.[7] studied the controllability of semilinear evolution and neutral functional differential

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 532–541, 2009.

inclusions. In this paper, we will give the existence of mild solutions for impulsive evolution neutral functional differential inclusions with infinite delay. Our approach will be based on another nonlinear alternatives of Leray-Schauder type for multivalued maps due to D.O'Regan.

The paper is organized as follows. In Section 2, we recall some necessary preliminaries, In Section 3, we prove the existence.

2 Preliminaries

Let $\mathcal{P}(X)$ denote the class of all nonempty of subsets of X. Let $\mathcal{P}_{bd,cl}(X)$, $\mathcal{P}_{cp,cv}(X)$, $\mathcal{P}_{bd,cl,cv}(X)$ and $\mathcal{P}_{cd}(X)$ denote respectively the family of all nonempty bounded-closed, compact-convex, bounded-closed-convex and compact-acyclic [8] subsets of X.

F is called upper semicontinuous(shortly u.s.c.) on X, if for each $x_* \in X$, the set $F(x_*)$ is nonempty, closed subset of X, and if for each open set of V of X containing $F(x_*)$, there exists an open neighborhood N of x_* such that $F(N) \subseteq V$. F is said to be completely continuous if $F(V)$ is relatively compact, for every bounded subset $V \subseteq X$.

If the multivalued map F is completely continuous with nonempty compact values, then F is u.s.c. if and only if F has a closed graph,(i.e. $x_n \to x_*, y_n \to y_*, y_n \in F(x_n)$ imply $y_* \in F(x_*)$).

A point $x_0 \in X$ is called a fixed point of the multivalued map F if $x_0 \in F(x_0)$.

A multivalued map $F : J \to \mathcal{P}_{bd,cl,cv}(X)$ is said to be measurable if for each $x \in X$, the function $t \mapsto D(x, F(t))$ is a measurable function on J.

Let $\{A(t) : t \in J\}$ be a family of linear operators and satisfy:

(A1) The domain $D(A(t)) = D$ of $A(t)$ is dense in X and independent of t, $A(t)$ is a closed linear operator,

(A2) For each $t \in J$, the resolvent $R(\lambda, A(t))$ exists for all λ with $Re\lambda \leq 0$ and there exists $k > 0$ such that $\|R(\lambda, A(t))\| \leq \dfrac{k_0}{|\lambda| + 1}$,

(A3) There exist constants $H > 0$ and $0 < \alpha \leq 1$ such that for $t, s, \tau \in J$, $\|(A(t) - A(s))A^{-1}(\tau)\| \leq H|t - s|^{\alpha}$.

Remark:

1. From (A3), we obtain $\|A(t)A^{-1}(0)\| \leq H|b|^{\alpha} + 1 := M_0$

2. Under the assumptions (A1)–(A3), there is a unique evolution system $U(t, s)$ on $0 \leq s \leq t \leq b$, such that, there exists a positive constant $\widetilde{M}$, $\|U(t, s)\| \leq \widetilde{M}$ for $0 \leq s \leq t \leq b$([11]).

Next, we define the set

$$\mathcal{PC}([\mu, \tau]; X) = \left\{ \varphi : [\mu, \tau] \to X : \begin{array}{l} \varphi(\cdot) \text{ is continuous at } t \neq t_k, \varphi(t_k^+) = \varphi(t_k) \\ \text{and } \varphi(t_k^-) \text{ exist for } k = 1, 2, \ldots, m. \end{array} \right\}$$

The notation $\mathcal{PC}$ stands for the space formed by all functions $u \in \mathcal{PC}([0, b]; X)$. The norm $\|\cdot\|_{\mathcal{PC}}$ of the space $\mathcal{PC}$ is defined by $\|\varphi\|_{\mathcal{PC}} = \sup_{0 \leq s \leq b} \|\varphi(s)\|$. It is

clear that $(\mathcal{PC}, \|\cdot\|_{\mathcal{PC}})$ is a Banach space. We will employ an axiomatic definition of the phase space $\mathcal{B}$ which is similar to that used in [12]. $\mathcal{B}$ will be a linear space of functions mapping $(-\infty, 0]$ to X endowed with a seminorm $\|\cdot\|_{\mathcal{B}}$. We will assume that $\mathcal{B}$ satisfies the following axioms:

(A) If $x : (-\infty, \mu+\sigma] \to X, \sigma > 0$ is such that $x_\mu \in \mathcal{B}$ and $x|_{[\mu,\mu+\sigma]} \in \mathcal{PC}([\mu, \mu+\sigma]; X)$,then for every $t \in [\mu, \mu+\sigma]$ the following conditions hold:
  (1) x_t is in $\mathcal{B}$,
  (2) $\|x(t)\| \le L\|x_t\|_{\mathcal{B}}$,
  (3) $\|x_t\|_{\mathcal{B}} \le K(t-\mu)\sup\{\|x(s)\| : \mu \le s \le t\} + M(t-\mu)\|x_\mu\|_{\mathcal{B}}$,
  where $L > 0$ is a constant, $K, M : [0,\infty) \to [0,\infty)$, K is continuous, M is locally bounded, and L, K, M are independent of $x(\cdot)$.
(B) The space $\mathcal{B}$ is complete.

Definition 1. Let $F : X \to \mathcal{P}_{bd,cl}(X)$ be a multivalued map.Then F is called a multivalued contraction if there exists a constant $k \in (0,1)$ such that for each $x, y \in X$ we have

$$H(F(x), F(y)) \le k\|x - y\|.$$

The constant k is called a contraction constant of F.

The consideration of this paper is based the another nonlinear alternatives of Leray-Schauder type for multivalued maps due to D.O'Regan [9].

Lemma 1. [9] Let X be a Banach space with U an open,convex subset of X and $u_0 \in U$. Suppose

(a) $F : \overline{U} \to \mathcal{P}_{cd}(X)$ has closed graph, and
(b) $F : \overline{U} \to \mathcal{P}_{cd}(X)$ is condensing map with $F(\overline{U})$ a subset of a bounded set in X

hold. Then either

(i) F has a fixed point in $\overline{U}$; or
(ii) There exist $u \in \partial U$ and $\lambda \in (0,1)$ with $u \in \lambda F(u) + (1-\lambda)\{u_0\}$.

Definition 2. A multivalued map $F : J \times \mathcal{B} \to \mathcal{P}_{bd,cl,cv}(X)$ is called L^1-Carathéodory if

(i) $F(t,x)$ is measurable with respect to t for each $x \in \mathcal{B}$,
(ii) $F(t,x)$ is u.s.c. with respect to x for each $t \in J$, and
(iii) for each $q > 0$, there exists a function $h_q \in L^1(J, [0,\infty)$ such that

$$\|F(t,v)\| := \sup\{\|g\| : g \in F(t,v)\} \le h_q(t), \text{a.e. } t \in J$$

for all $v \in \mathcal{B}$ with $\|v\|_{\mathcal{B}} \le q$.

We need the theorem due to Lasota and Opial [10].

Lemma 2. Let X be a Banach space, F be an L^1-Carathéodory multivalued map with $S_{F,\phi} \neq \emptyset$ where $S_{F,\phi} := \{g \in L^1(J,X) : g(t) \in F(t,\phi) \text{ a.e. } t \in J\}$, for each fixed $\phi \in \mathcal{B}$, and $\mathcal{K}$ be a linear continuous map from $L^1(J,X)$ to $C(J,X)$. Then the operator $\mathcal{K} \circ S_{F,\phi} : C(J,X) \to \mathcal{P}_{cp,cv}(C(J,X))$ is a closed graph operator in $C(J,X) \times C(J,X)$.

3 Main Result

Before stating and proving our main result, we give first the definition of mild solution.

Definition 3. A function $x : (-\infty, b] \to X$ is called a mild solution of the system(1) if $x_0 = \phi \in \mathcal{B}$, $x|_J \in \mathcal{PC}$ and for each $s \in [0,t)$, the function $U(t,s)A(s)g(s,x_s)$ is integrable such that

$$\begin{aligned} x(t) = U(t,0)[\phi(0) - g(0,\phi)] + g(t,x_t) + \int_0^t U(t,s)A(s)g(s,x_s)ds \\ + \int_0^t U(t,s)f(s)ds + \sum_{0<t_k<t} U(t,t_k)I_k(x(t_k^-)), \quad t \in J. \end{aligned} \tag{2}$$

where $f \in S_{F,x} = \{f \in L^1(J,X) : f(t) \in F(t,x_t), t \in J\}$.

We consider the following assumptions in the sequel

(H1) The multivalued map $F(t,x)$ is an L^1-Carathéodory multivalued map and has compact and convex values for each $(t,x) \in J \times \mathcal{B}$.

(H2) The impulsive functions I_k are continuous and there exist positive constants β_k such that $\|I_k(x)\| \le \beta_k$, $k = 1,2,\ldots,m$ for each $x \in X$.

(H3) There exist constants $c_1, c_2, L_0 > 0$ such that $A(0)g$ is continuous, and

(i) $\|A(0)g(t,\psi)\| \le c_1\|\psi\|_{\mathcal{B}} + c_2$, $(t,x) \in J \times \mathcal{B}$,

(ii) $\|A(0)g(t,\psi_1) - A(0)g(t,\psi_2)\| \le L_0\|\psi_1 - \psi_2\|_{\mathcal{B}}$, $t \in J, \psi_1, \psi_2 \in \mathcal{B}$, with

$$\|A^{-1}(0)\|L_0K_b + \widetilde{M}M_0K_bL_0b < 1,$$

where $K_b = \sup\{K(t) : 0 \le t \le b\}$, $M_b = \sup\{M(t) : 0 \le t \le b\}$.

(H4) There exists a positive function $p \in L^1(J,[0,\infty))$ such that

$$\|F(t,\phi)\| := \sup\{\|v\| : v \in F(t,\phi)\} \le p(t)\Theta(\|\phi\|_{\mathcal{B}}), \text{ a.e. } t \in J, \phi \in \mathcal{B}$$

where $\Theta : [0,\infty) \to (0,\infty)$ a continuous nondecreasing function.

Theorem 1. Let $\phi \in \mathcal{B}$, If the assumptions (H1)–(H4) are satisfied, then the system (1) has at least one solution on $(-\infty, b]$ provided that there exist a constant N_* with

$$\frac{(1 - c_1K_b\|A^{-1}(0)\| - K_b\widetilde{M}M_0bc_1)\|\chi\|}{N_1 + K_b\widetilde{M}\Theta(\|\chi\|)\int_0^b p(s)ds} > 1, \tag{3}$$

Where $N_1 = K_b\widetilde{M}\|A^{-1}(0)\|(c_1\|\phi\|_{\mathcal{B}} + c_2) + K_b\|A^{-1}(0)\|c_2 + M_b\|\phi\|_{\mathcal{B}} + K_b\widetilde{M}\|\phi(0)\| + K_b\widetilde{M}M_0c_2b + K_b\widetilde{M}\sum_{k=1}^m \beta_k$.

Proof. Let $\mathcal{B}_b$ be the space of all function $x : (-\infty, b] \to X$ such that $x_0 \in \mathcal{B}$ and the restriction $x|_J \in \mathcal{PC}$. For each $x(t) \in \mathcal{B}_b$, let $\|\cdot\|_b$ be the seminorm in $\mathcal{B}_b$ defined by $\|x\|_b = \|x_0\|_{\mathcal{B}} + \sup\{\|x(s)\| : 0 \le s \le b\}$.

The multivalued map $\Phi : \mathcal{B}_b \to \mathcal{P}(\mathcal{B}_b)$ is defined by Φx the set of $h \in \mathcal{B}_b$ such that

$$h(t) = \begin{cases} \phi(t), \quad \text{if } t \in (-\infty, 0], \\ U(t,0)[\phi(0) - g(0,\phi)] + g(t, x_t) + \int_0^t U(t,s)A(s)g(s,x_s)ds \\ + \int_0^t U(t,s)f(s)ds + \sum\limits_{0<t_k<t} U(t,t_k)I_k(x(t_k^-)), \quad t \in J, \end{cases}$$

where $f \in S_{F,x} = \{f \in L^1(J,X) : f(t) \in F(t,x_t), t \in J\}$, and Φ has a fixed point which is then a solution of the system (1).

We define a function

$$y(t) = \begin{cases} \phi(t), \quad \text{if } \ t \in (-\infty, 0], \\ U(t,0)\phi(0), \quad \text{if } t \in J. \end{cases}$$

Set $x(t) = z(t) + y(t), -\infty < t \le b$, It is clear that x satisfies (2) if and only if z satisfies $z_0 = 0$ and

$$z(t) = -U(t,0)g(0,\phi) + g(t, z_t + y_t) + \int_0^t U(t,s)A(s)g(s, z_s + y_s)ds$$

$$+ \int_0^t U(t,s)f(s)ds + \sum_{0<t_k<t} U(t,t_k)I_k(z(t_k^-) + y(t_k^-)),$$

where $f \in S_{F,z} = \{f \in L^1(J,X) : f(t) \in F(t, z_t + y_t), t \in J\}$

Let $\mathcal{B}_b^0$ be the space of all function $z : (-\infty, b] \to X$ such that $z_0 \equiv 0$ and the restriction $z|_J \in \mathcal{PC}$. For each $z(t) \in \mathcal{B}_b$, let $\|\cdot\|_b$ be the norm in $\mathcal{B}_b^0$ defined by

$$\|z\|_b = \sup\{\|z(s)\| : \ 0 \le s \le b\}$$

Thus $(\mathcal{B}_b^0, \|\cdot\|_b)$ is Banach space.

The multivalued map $\Phi_1 : \mathcal{B}_b^0 \to \mathcal{P}(\mathcal{B}_b^0)$ is defined by $\Phi_1 z$ the set of $h_0 \in \mathcal{B}_b$ such that

$$h_0(t) = \begin{cases} 0, \quad \text{if } t \in (-\infty, 0], \\ -U(t,0)g(0,\phi) + g(t, z_t + y_t) + \int_0^t U(t,s)A(s)g(s, z_s + y_s)ds \\ + \int_0^t U(t,s)f(s)ds + \sum\limits_{0<t_k<t} U(t,t_k)I_k(z(t_k^-) + y(t_k^-)), \ t \in J, \end{cases}$$

Step 1. Choose $u_0 = 0$ and a set U is convex and open in $\mathcal{B}_b^0$.

Let $\lambda \in (0,1)$ and let $u \in \lambda\Phi_1 u$, Then exists $f \in S_{F,u}$, such that

$$u(t) = -\lambda U(t,0)g(0,\phi) + \lambda g(t, u_t + y_t) + \lambda \int_0^t U(t,s)A(s)g(s, u_s + y_s)ds$$

$$+\lambda \int_0^t U(t,s)f(s)ds + \lambda \sum_{0<t_k<t} U(t,t_k)I_k(u(t_k^-) + y(t_k^-)),$$

Then

$$\|u(t)\| \le \widetilde{M}\|A^{-1}(0)\|(c_1\|\phi\|_{\mathcal{B}} + c_2) + \|A^{-1}(0)\|(c_1\|u_t + y_t\|_{\mathcal{B}} + c_2)$$
$$+\widetilde{M}M_0\int_0^t (c_1\|u_s+y_s\|_{\mathcal{B}}+c_2)\,ds+\widetilde{M}\sum_{k=1}^{m}\beta_k + \widetilde{M}\int_0^t p(s)\Theta(\|u_s+y_s\|_{\mathcal{B}})\,ds$$

And since

$$\|u_t + y_t\|_{\mathcal{B}} \le K_b \sup\{\|u(s)\| : \ 0 \le s \le t\} + M_b\|\phi\|_{\mathcal{B}} + K_b\widetilde{M}\|\phi(0)\|,$$

set $\chi(t) = K_b \sup\{\|u(s)\| : \ 0 \le s \le t\} + M_b\|\phi\|_{\mathcal{B}} + K_b\widetilde{M}\|\phi(0)\|$, then the functions $\chi(t)$ is nondecreasing in J,

So

$$\chi(t) \le K_b\widetilde{M}\|A^{-1}(0)\|(c_1\|\phi\|_{\mathcal{B}} + c_2) + K_b\|A^{-1}(0)\|(c_1\chi(t) + c_2) + M_b\|\phi\|_{\mathcal{B}}$$

$$+K_b\widetilde{M}\|\phi(0)\|+K_b\widetilde{M}M_0\int_0^t (c_1\chi(s)+c_2)\,ds+K_b\widetilde{M}\sum_{k=1}^{m}\beta_k+K_b\widetilde{M}\int_0^t p(s)\Theta(\chi(s))\,ds$$

Consider the norm of the function $\chi(t)$, $\|\chi\| = \sup\{\chi(t) : 0 \le t \le b\}$, Therefore, we obtain the following inequality

$$\frac{(1 - c_1K_b\|A^{-1}(0)\| - K_b\widetilde{M}M_0bc_1)\|\chi\|}{N_1 + K_b\widetilde{M}\Theta(\|\chi\|)\int_0^b p(s)ds} \le 1$$

Then by (3), there exists N_* such that $\|\chi\| \ne N_*$. Set $U = \{u \in \mathcal{B}_b^0 : \|u\|_b < N_*\}$, From the choice of U, there is no $u \in \partial U$ such that $u \in \lambda\Phi_1 u$ for $\lambda \in (0,1)$.

Step 2. Φ_1 has a close graph.

Let $z^n \to z^*, h_0^n \in \Phi_1 z^n$ and $h_0^n \to h_0^*$, we shall prove that $h_0^* \in \Phi_1 z^*$. Indeed, If $h_0^n \in \Phi_1 z^n$ means that there exists $f_n \in S_{F,z^{(n)}}$, such that

$$h_0^n(t) = -U(t,0)g(0,\phi) + \int_0^t U(t,s)A(s)g(s,z_s^n + y_s)ds + g(t,z_t^n + y_t)$$

$$+\int_0^t U(t,s)f_n(s)ds + \sum_{0<t_k<t} U(t,t_k)I_k(z^n(t_k^-) + y(t_k^-)),$$

We must prove that there exists $f_* \in S_{F,z^*}$, such that

$$h_0^*(t) = -U(t,0)g(0,\phi) + \int_0^t U(t,s)A(s)g(s,z_s^* + y_s)ds + g(t,z_t^* + y_t)$$

$$+\int_0^t U(t,s)f_*(s)ds + \sum_{0<t_k<t} U(t,t_k)I_k(z^*(t_k^-) + y(t_k^-)),$$

Clearly, we have

$$\Big\|\Big\{h_0^n(t)+U(t,0)g(0,\phi)-\int_0^t U(t,s)A(s)g(s,z_s^n+y_s)ds-g(t,z_t^n+y_t)$$

$$-\sum_{0<t_k<t}U(t,t_k)I_k(z^n(t_k^-)+y(t_k^-))\Big\}-\Big\{h_0^*(t)+U(t,0)g(0,\phi)$$

$$-\int_0^t U(t,s)A(s)g(s,z_s^*+y_s)ds-g(t,z_t^*+y_t)-\sum_{0<t_k<t}U(t,t_k)I_k(z^*(t_k^-)+y(t_k^-))\Big\}\Big\|$$

$$\to 0,\ \text{as } n\to\infty,$$

Consider the linear and continuous operator $\mathcal{K}: L^1(J,X)\to C(J,X)$ defined by

$$\mathcal{K}f(t)=\int_0^t U(t,s)f(s)ds$$

From Lemma 2, it follows that $\mathcal{K}\circ S_F$ is a closed graph operator, and

$$h_0^n(t)+U(t,0)g(0,\phi)-\int_0^t U(t,s)A(s)g(s,z_s^n+y_s)ds$$

$$-g(t,z_t^n+y_t)-\sum_{0<t_k<t}U(t,t_k)I_k(z^n(t_k^-)+y(t_k^-))\in\mathcal{K}\circ S_{F,z^{(n)}}$$

Since $z^{(n)}\to z^*$ and $h_0^n\to h_0^*$, it follows from Lemma 2 that , there exist $f_*\in S_{F,z^*}$, such that

$$h_0^*(t)=-U(t,0)g(0,\phi)+\int_0^t U(t,s)A(s)g(s,z_s^*+y_s)ds+g(t,z_t^*+y_t)+\int_0^t U(t,s)f_*(s)ds$$

$$+\sum_{0<t_k<t}U(t,t_k)I_k(z^*(t_k^-)+y(t_k^-)),$$

So we can conclude that Φ_1 has closed graph.

We define two the maps. The map $\mathcal{A}: U\to \mathcal{B}_b^0$ is defined by

$$h_1(t)=\begin{cases}0, & \text{if } t\in(-\infty,0],\\ -U(t,0)g(0,\phi)+g(t,z_t+y_t)+\int_0^t U(t,s)A(s)g(s,z_s+y_s)ds, & t\in J,\end{cases}\tag{4}$$

and $\mathcal{C}: U\to\mathcal{P}(\mathcal{B}_b^0)$ is defined by $\mathcal{C}z$ the set $h_2(t)\in\mathcal{B}_b^0$ such that

$$h_2(t)=\begin{cases}0 & \text{if } t\in(-\infty,0],\\ \int_0^t U(t,s)f(s)\,ds+\sum_{0<t_k<t}U(t,t_k)I_k(z(t_k^-)+y(t_k^-)), & t\in J\end{cases}\tag{5}$$

Then $\Phi_1=\mathcal{A}+\mathcal{C}$.

Step 3. $\mathcal{A}$ is a contraction on $\mathcal{B}_b^0$.

It is easy to prove $\mathcal{A}$ and $\mathcal{C}$ maps bounded sets into bounded sets in $\mathcal{B}_b^0$.

Let $z^1(t), z^2(t) \in U$, for $h_1(t) = \mathcal{A}z^1(t)$, $h_2(t) = \mathcal{A}z^2(t)$, by hypothesis (H3), we have

$$\|h_1(t) - h_2(t)\| \leq \left[\|A^{-1}(0)\|L_0K_b + \widetilde{M}M_0L_0K_bb\right]\|z^1 - z^2\|_b$$

By the assumption (H3) , $\mathcal{A}$ is a contraction map. Also it is obviously that the map $\mathcal{A}$ has closed values.

Step 4. $\mathcal{C}$ is convex for $z \in U$.

In fact, if $h_2^1(t), h_2^2(t)$ belong to $\mathcal{C}z$, then exist $f_1, f_2 \in S_{F,x}$ such that

$$h_2^i(t) = \int_0^t U(t,s)f_i(s)\,ds + \sum_{0<t_k<t} U(t,t_k)I_k(z(t_k^-) + y(t_k^-))$$

Since $F(t,z)$ has convex valued, for $0 \leq \tau \leq 1$, $[\tau f_1 + (1-\tau)f_2](t)$ belongs to $S_{F,x}$,

$$(\tau h_2^1(t) + (1-\tau)h_2^2(t)) = \sum_{0<t_k<t} U(t,t_k)I_k(z(t_k^-) + y(t_k^-))$$

$$+ \int_0^t U(t,s)[\tau f_1(s) + (1-\tau)f_2(s))]\,ds$$

Therefore, $(\tau h_2^1(t) + (1-\tau)h_2^2(t)) \in \mathcal{C}z$ and consequently $\mathcal{C}z$ has convex values.

Step 5. $\mathcal{C}$ maps bounded sets into equicontinuous sets in U.

Let $z(t) \in U$ If $h_2(t) \in \mathcal{C}z(t)$, then exists $f \in S_{F,z}$, such that, for each $t \in J$

$$h_2(t) = \int_0^t U(t,s)f(s)\,ds + \sum_{0<t_k<t} U(t,t_k)I_k(z(t_k^-) + y(t_k^-))$$

Let $\tau_1, \tau_2 \in J, \tau_1 < \tau_2$. Then we have

$$\|h_2(\tau_2) - h_2(\tau_1)\| \leq \int_{\tau_1}^{\tau_2} \|U(\tau_2,s)f(s)\|ds + \sum_{\tau_1<t_k<\tau_2} \widetilde{M}\beta_k$$

$$+ \left\{\int_0^{\tau_1-\epsilon} + \int_{\tau_1-\epsilon}^{\tau_1}\right\} \|U(\tau_2,s) - U(\tau_1,s)\|\|f(s)\|ds + \sum_{0<t_k<\tau_1} \|U(\tau_2,t_k) - U(\tau_1,t_k)\|\beta_k.$$

We see that $\|h_2(\tau_2) - h_2(\tau_1)\|$ tends to zero as $(\tau_2 - \tau_1) \to 0$ with ϵ sufficiently small, Since $U(t,s)$ is a strongly continuous operator and the compactness of $U(t,s)$ for $t > 0$ implies the continuity in the uniform operator topology. Hence, $\mathcal{C}$ maps bounded sets into equicontinuous sets.

The equicontinuity for the cases $\tau_1 < \tau_2 \leq 0$, or $\tau_1 \leq 0 \leq \tau_2 \leq b$ are similar.

Step 6. Next we shall $\mathcal{C}$ maps U into a precompact set.

Let $0 < t \leq b$ be fixed and let ϵ be a real number satisfying $0 < \epsilon < t$. For $z \in U$, we define

$$h_2^{\epsilon}(t) = \int_0^{t-\epsilon} U(t, \epsilon + s)U(\epsilon + s, s)f(s)\, ds + \int_{t-\epsilon}^{t} U(t, s)f(s)\, ds + \sum_{0<t_k<t-\epsilon} U(t - t_k)I_k(z(t_k^-) + y(t_k^-))$$

where $f \in S_{F,x}$, Since $U(t, s)$ is a compact operator, the set $Y_{\epsilon}(t) := \{h_2^{\epsilon}(t) : z \in U\}$ is precompact for every $\epsilon, 0 < \epsilon < t$. Also, for every $z \in U$, we have

$$\|h_2(t) - h_2^{\epsilon}(t)\| \leq \int_{t-\epsilon}^{t} \|U(t, s)\|\|f(s)\|ds + \widetilde{M} \sum_{t-\epsilon<t_k<t} \beta_k$$

The right hand side of the above inequality tends to zero as $\epsilon \to 0$. Since there are precompact sets arbitrarily close to the set $Y(t) := \{h_2(t) : \ z \in U\}$. Hence the set $Y(t)$ is precompact in X. By Arzelá-Ascoli theorem, we conclude that $\mathcal{C}$ $U \to \mathcal{P}(\mathcal{B}_b^0)$ is completely continuous.

By step 5 and 6, we conclude that $\Phi_1 = \mathcal{A} + \mathcal{C}$ is a condensing map.

So, All of the conditions of Lemma 1 are satisfied. Therefore, the system (1) has at least one solution.

References

1. Erbe, L.H., Freedman, H.I., Liu, X.Z., Wu, J.H.: Comparison Principles for Impulsive Parabolic Equations with Applications to Models of Single Species Growths. J. Austral. Math. Soc. Ser. B. 32, 382–400 (1991)
2. Rogovchenko, Y.V.: Impulsive Evolution Systems: Main Results and New Trends. Dyn. Contin. Discrete Impuls. Syst. 3, 57–88 (1997)
3. Rogovchenko, Y.V.: Nonlinear Impulse Evolution Systems and Applications to Population Models. J. Math. Anal. Appl. 207, 300–315 (1997)
4. Liu, J.H.: Nonlinear Impulsive Evolution Equations. Dyn. Contin. Discrete Impuls. Syst. 6, 77–85 (1999)
5. Benchohra, M., Gatsori, E.P., Ntouyas, S.K.: Exitence Results for Functional and Neutral Functional Integrodifferential Inclusions with Lower Semicontinuous Right-hand Side. J. Math. Anal. Appl. 281, 525–538 (2003)
6. Li, M., Wang, M., Zhang, F.: Controllability of Impulsive Functional Differential Systems in Banach spaces. Chao, Solitons and Fractals 29, 175–781 (2006)
7. Benchohra, M., Gatsori, E.P., Ntouyas, S.K.: Controllbility of Semilinear Evolution Inclusions with Nonlocal Conditions. J. Optim. Theory Appl. 118, 493–513 (2003)
8. Fitzpatrick, P.M., Petryshyn, W.V.: Fixed Point Theorems for Multivalued Noncompact Acyclic Mappings. Pacific J. Math. 54, 17–23 (1974)
9. Regan, D.: Nonlinear Alternatives for Multivalued Maps with Applications to Operator Inclusions in Abstract Spaces. Proc. of the Amer. Math. Society 127, 3557–3564 (1999)

10. Lasota, A., Opial, Z.: An Application of the Kakutani-Ky Fan Theorem in the Theory of Ordinary Differential Equations. Bull. Acad. Pol. Sci. Ser. Sci. Math. Astronom. Phys. 13, 781–786 (1965)
11. Pazy, A.: Semigroups of Linear Operators and Applications to Partial Differential Equations. Applied Methematical Sciences. Springer, New York (1983)
12. Hino, Y., Murakami, S., Naito, T.: Functional Differential Equations with Infinite Delay. Lecture Notes in Mathematics. Springer, New York (1991)

Passivity Analysis of Neural Networks with Time-Varying Delays of Neutral Type

Jianqin Wang[1] and Qiankun Song[2]

[1] School of Basic Medical, North Sichuan Medical College, Sichuan 637007, China
[2] Department of Mathematics, Chongqing Jiaotong University,
Chongqing 400074, China
qiankunsong@163.com

Abstract. In this paper, the passivity is investigated for neural networks with time-varying delays of neutral type and generalized activation functions. By using Lyapunov method, Newton-Leibniz formulation and linear matrix inequality (LMI) technique, several delay-independent sufficient conditions in LMI are obtained to guarantee the passivity of the addressed neural networks. The proposed passivity criteria do not require the monotonicity of the activation functions and the differentiability of the time-varying delays, which means that our results generalize and further improve those in the earlier publications. An example is given to show the effectiveness and less conservatism of the obtained conditions.

Keywords: Passivity, Neural networks, Time-varying delays, Neutral type.

1 Introduction

In the past few years, various classes of neural networks have been increasingly studied due to their practical importance and successful applications in many areas such as combinatorial optimization, signal processing and communication [1]-[4]. These applications greatly depend on the stability of the underlying neural networks [5]. As is well known, time delay may occur in the process of information storage and transmission in neural networks. In electronic implementation of neural networks, the time delay is often time-variant, and even varies dramatically with time because of the finite switch speed of amplifiers and faults in the electrical circuits. Up to now, the stability analysis for delayed neural networks has attracted considerable attention, and a large amount of results have been available in the literature, see [1]-[15] and the references therein for some recent publications.

On the other hand, the passivity theory is another effective tool to the stability of nonlinear system [16]. The main idea of passivity theory is that the passive properties of system can keep the system internal stability [17]. Thus, the passivity theory has received a lot of attention from the control community since 1970s [18]. Recently, the passivity theory for delayed neural networks was investigated, some criteria checking the passivity were provided for certain or

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 542–549, 2009.

uncertain neural networks with time-varying delays, see [19]-[21] and references therein. It is worth pointing out that, the given criteria in [19]-[21] have been based on the following assumptions: 1) the time-varying delays are continuously differentiable; 2) the derivative of time-varying delay is bounded and is smaller than one; and 3) the activation functions are bounded and monotonically nondecreasing. However, time delays can occur in an irregular fashion, and sometimes the time-varying delays are not differentiable. In such a case, the methods developed in [19]-[21] may be difficult to be applied, and it is therefore necessary to further investigate the passivity problem of neural networks with time-varying delays under *milder* assumptions. To the best of our knowledge, few authors have considered the passivity problem for neural networks with time-varying delays of neutral type as well as generalized activation functions.

Motivated by the above discussions, the objective of this paper is to study the passivity of neural networks with time-varying delays of neutral type as well as generalized activation functions by employing a combination of Lyapunov method, Newton-Leibniz formulation and the free-weighting matrix technique. The obtained sufficient conditions require *neither* the differentiability of time-varying delays *nor* the monotony of the activation functions, and are expressed in terms of LMIs, which can be checked numerically using the effective LMI toolbox in MATLAB. An example is given to show the effectiveness and less conservatism of the proposed criterion.

2 Problem Formulation and Preliminaries

In this paper, we consider the following neural network of neutral type:

$$\frac{d}{dt}(x(t) - Cx(t-\tau(t))) = -Dx(t) + Af(x(t)) + Bf(x(t-\tau(t))) + u(t) \quad (1)$$

for $t \geq 0$, where $x(t) = (x_1(t), x_2(t), \cdots, x_n(t))^T \in R^n$ is the state vector of the network at time t, n corresponds to the number of neurons; $D = \mathrm{diag}(d_1, d_2, \cdots, d_n)$ is a positive diagonal matrix, $A = (a_{ij})_{n\times n}$, $B = (b_{ij})_{n\times n}$ and $C = (c_{ij})_{n\times n}$ are the interconnection weight matrices; $f(x(t)) = (f_1(x_1(t)), f_2(x_2(t)), \cdots, f_n(x_n(t)))^T$ denotes the neuron activation at time t; $u(t) = (u_1(t), u_2(t), \cdots, u_n(t))^T \in R^n$ is a varying external input vector; $\tau(t) > 0$ is the time-varying delay, and is assumed to satisfy $0 \leq \tau(t) \leq \tau$, where τ is constant.

The initial condition associated with model (1) is given by

$$x(s) = \phi(s), \qquad s \in [-\tau, 0]. \quad (2)$$

Let $x(t, \phi)$ denote the state trajectory of model (1) from the above initial condition and $x(t, 0)$ is the corresponding trajectory with zero initial condition.

Throughout this paper, we make the following assumption:

(H).([6]) For any $j \in \{1, 2, \cdots, n\}$, $f_j(0) = 0$ and there exist constants F_j^- and F_j^+ such that

$$F_j^- \leq \frac{f_j(\alpha_1) - f_j(\alpha_2)}{\alpha_1 - \alpha_2} \leq F_j^+$$

for all $\alpha_1 \neq \alpha_2$.

Definition 1. *([19]) System (1) is called globally passive if there exists a scalar $\gamma > 0$ such that*

$$2\int_0^{t_p} f^T(x(s))u(s)ds \geq -\gamma \int_0^{t_p} u^T(s)u(s)ds$$

for all $t_p \geq 0$ and for all $x(t,0)$.

To prove our result, the following lemma is necessary.

Lemma 1. *([14]) For any constant matrix $W \in R^{m\times m}$, $W > 0$, scalar $0 < h(t) < h$, vector function $\omega : [0,h] \to R^m$ such that the integrations concerned are well defined, then*

$$\Big(\int_0^{h(t)} \omega(s)ds\Big)^T W \Big(\int_0^{h(t)} \omega(s)ds\Big) \leq h(t)\int_0^{h(t)} \omega^T(s)W\omega(s)ds.$$

3 Main Results

For presentation convenience, in the following, we denote

$$F_1 = \mathrm{diag}(F_1^- F_1^+, \cdots, F_n^- F_n^+), \quad F_2 = \mathrm{diag}(\frac{F_1^- + F_1^+}{2}, \cdots, \frac{F_n^- + F_n^+}{2}),$$

Theorem 1. *Under assumptions (**H**), model (1) is passive if there exist a scalar $\gamma > 0$, a symmetric positive definite matrices P, two positive diagonal matrices L and S, and matrices M, N and W such that the following LMI holds:*

$$\Omega = \begin{bmatrix} \Omega_1 & PA + F_2L & PB & P & W - M^T & DPC + N - M^T \\ * & -L & 0 & -I & 0 & A^TPC \\ * & * & -S & 0 & 0 & F_2S + B^TPC \\ * & * & * & -\gamma I & 0 & PC \\ * & * & * & * & -W - W^T & -N - W^T \\ * & * & * & * & * & -F_1S - N - N^T \end{bmatrix} < 0, \qquad (3)$$

where $\Omega_1 = -PD - DP - F_1L + M + M^T$.

Proof. Consider the following Lyapunov functional as

$$V(t) = (x(t) - Cx(t-\tau(t)))^T P(x(t) - Cx(t-\tau(t))), \qquad (4)$$

Calculating the time derivative of $V(t)$ along the trajectories of model (1), we obtain

$$\begin{aligned} \frac{dV(t)}{dt} &= x^T(t)(-PD - DP)x(t) + 2x^T(t)PAf(x(t)) \\ &\quad +2x^T(t)PBf(x(t-\tau(t))) + 2x^T(t)Pu(t) \\ &\quad +2x^T(t-\tau(t))C^TPDx(t) - 2x^T(t-\tau(t))C^TPAf(x(t)) \\ &\quad -2x^T(t-\tau(t))C^TPBf(x(t-\tau(t))) - 2x^T(t-\tau(t))C^TPu(t). \qquad (5) \end{aligned}$$

From assumption (**H**), we have

$$\Big(f_i(x_i(t)) - F_i^- x_i(t)\Big)\Big(f_i(x_i(t)) - F_i^+ x_i(t)\Big) \le 0, \quad i = 1, 2, \cdots, n,$$

which are equivalent to

$$\begin{bmatrix} x_i(t) \\ f_i(x_i(t)) \end{bmatrix}^T \begin{bmatrix} F_i^- F_i^+ e_i e_i^T & -\frac{F_i^- + F_i^+}{2} e_i e_i^T \\ -\frac{F_i^- + F_i^+}{2} e_i e_i^T & e_i e_i^T \end{bmatrix} \begin{bmatrix} x_i(t) \\ f_i(x_i(t)) \end{bmatrix} \le 0, \quad i = 1, 2, \cdots, n,$$

where e_r denotes the unit column vector having 1 element on its rth row and zeros elsewhere. Let

$$L = \mathrm{diag}\{l_1, l_2, \cdots, l_n\}, \quad S = \mathrm{diag}\{s_1, s_2, \cdots, s_n\},$$

then

$$\sum_{i=1}^{n} l_i \begin{bmatrix} x_i(t) \\ f_i(x_i(t)) \end{bmatrix}^T \begin{bmatrix} F_i^- F_i^+ e_i e_i^T & -\frac{F_i^- + F_i^+}{2} e_i e_i^T \\ -\frac{F_i^- + F_i^+}{2} e_i e_i^T & e_i e_i^T \end{bmatrix} \begin{bmatrix} x_i(t) \\ f_i(x_i(t)) \end{bmatrix} \le 0,$$

that is

$$\begin{bmatrix} x(t) \\ f(x(t)) \end{bmatrix}^T \begin{bmatrix} F_1 L & -F_2 L \\ -F_2 L & L \end{bmatrix} \begin{bmatrix} x(t) \\ f(x(t)) \end{bmatrix} \le 0. \tag{6}$$

Similarly, one has

$$\begin{bmatrix} x(t-\tau(t)) \\ f(x(t-\tau(t))) \end{bmatrix}^T \begin{bmatrix} F_1 S & -F_2 S \\ -F_2 S & S \end{bmatrix} \begin{bmatrix} x(t-\tau(t)) \\ f(x(t-\tau(t))) \end{bmatrix} \le 0. \tag{7}$$

From Newton-Leibniz formulation $x(t) - x(t-\tau(t)) - \int_{t-\tau(t)}^{t} \dot{x}(s)ds = 0$, we have

$$\begin{aligned} 0 = 2\Big(x(t) - x(t-\tau(t)) - \int_{t-\tau(t)}^{t} \dot{x}(s)ds\Big)^T \\ \times \Big(Mx(t) + Nx(t-\tau(t)) + W\int_{t-\tau(t)}^{t} \dot{x}(s)ds\Big). \end{aligned} \tag{8}$$

It follows from (5)-(8) that

$$\begin{aligned} &\frac{dV(t)}{dt} - 2f^T(x(t))u(t) - \gamma u^T(t)u(t) \\ &\le x^T(t)(-PD - DP)x(t) + 2x^T(t)PAf(x(t)) \\ &\quad + 2x^T(t)PBf(x(t-\tau(t))) + 2x^T(t)Pu(t) \\ &\quad + 2x^T(t-\tau(t))C^T PDx(t) - 2x^T(t-\tau(t))C^T PAf(x(t)) \\ &\quad - 2x^T(t-\tau(t))C^T PBf(x(t-\tau(t))) - 2x^T(t-\tau(t))C^T Pu(t) \end{aligned}$$

$$
\begin{aligned}
&-\begin{bmatrix} x(t) \\ f(x(t)) \end{bmatrix}^T \begin{bmatrix} F_1L & -F_2L \\ -F_2L & L \end{bmatrix} \begin{bmatrix} x(t) \\ f(x(t)) \end{bmatrix} \\
&-\begin{bmatrix} x(t-\tau(t)) \\ f(x(t-\tau(t))) \end{bmatrix}^T \begin{bmatrix} F_1S & -F_2S \\ -F_2S & S \end{bmatrix} \begin{bmatrix} x(t-\tau(t)) \\ f(x(t-\tau(t))) \end{bmatrix} \\
&+2\Big(x(t)-x(t-\tau(t))-\int_{t-\tau(t)}^{t}\dot{x}(s)ds\Big)^T \\
&\times\Big(Mx(t)+Nx(t-\tau(t))+W\int_{t-\tau(t)}^{t}\dot{x}(s)ds\Big) \\
&-2f^T(x(t))u(t)-\gamma u^T(t)u(t) \\
&=\alpha^T(t)\Omega\alpha(t), \qquad (9)
\end{aligned}
$$

where

$$
\alpha(t)=\Big(x^T(t),f^T(x(t)),f^T(x(t-\tau(t))),u^T(t),\int_{t-\tau(t)}^{t}\dot{x}(s)ds,x^T(t-\tau(t))\Big)^T.
$$

We get from (3) and (9) that

$$
\frac{dV(t)}{dt}-2f^T(x(t))u(t)-\gamma u^T(t)u(t)\le 0. \qquad (10)
$$

From (10) and the definition of $V(t)$, we have

$$
2\int_0^{t_p} f^T(x(s))u(s)ds\ge-\gamma\int_0^{t_p}u^T(s)u(s)ds
$$

for all $t_p\ge 0$. The proof is completed.

When $C=0$, model (1) turns to the following neural network:

$$
\frac{dx(t)}{dt}=-Dx(t)+Af(x(t))+Bf(x(t-\tau(t)))+u(t) \qquad (11)
$$

For model (11), we have the following result.

Corollary 1. *Under assumptions (***H***), model (1) is passive if there exist a scalar $\gamma>0$, a symmetric positive definite matrices P, two positive diagonal matrices L and S, and matrices M, N and W such that the following LMI holds:*

$$
\Pi=\begin{bmatrix}
\Omega_1 & PA+F_2L & PB & P & W-M^T & N-M^T \\
* & -L & 0 & -I & 0 & 0 \\
* & * & -S & 0 & 0 & F_2S \\
* & * & * & -\gamma I & 0 & 0 \\
* & * & * & * & -W-W^T & -N-W^T \\
* & * & * & * & * & -F_1S-N-N^T
\end{bmatrix}<0, \qquad (12)
$$

where $\Pi_1=-PD-DP-F_1L+M+M^T$.

Remark 1. Assumption (**H**) was first proposed in [6] and [7]. The constants F_j^- and F_j^+ $(i = 1, 2, \cdots, n)$ in assumption (**H**) are allowed to be positive, negative or zero. Hence, Assumption (**H**) is weaker than the assumption in [19]-[21]. In addition, the conditions in [19]-[21] that the time-varying delay is differentiable and the derivative is smaller than one have been removed in this paper.

4 An Example

Consider a two-neuron neural network (11), where

$$D = \begin{bmatrix} 3.4 & 0 \\ 0 & 2.8 \end{bmatrix}, \quad A = \begin{bmatrix} -0.1 & -0.2 \\ 0 & 0.2 \end{bmatrix}, \quad B = \begin{bmatrix} 0.1 & 0 \\ 0.3 & -0.1 \end{bmatrix},$$

$$f_1(z) = \tanh(0.2z), \quad f_2(z) = \tanh(-0.1z), \quad \tau(t) = 5|\sin t|.$$

It can be verified that assumption (**H**) is satisfied, and $F_1 = 0$, $F_2 = \text{diag}\{0.1, -0.05\}$.

By the Matlab LMI Control Toolbox, we find a solution to the LMI in (12) as follows:

$$P = \begin{bmatrix} 0.0025 & -0.0007 \\ -0.0007 & 0.0002 \end{bmatrix}, \quad L = \begin{bmatrix} 0.0133 & 0 \\ 0 & 0.0049 \end{bmatrix},$$

$$S = 10^{-5} \begin{bmatrix} 0.0731 & 0 \\ 0 & 0.2975 \end{bmatrix}, \quad M = 10^7 \begin{bmatrix} -1.0257 & -2.2472 \\ 2.3666 & -4.7193 \end{bmatrix},$$

$$N = W = 10^7 \begin{bmatrix} 1.0257 & 2.2472 \\ -2.3666 & 4.7193 \end{bmatrix}, \quad \gamma = 1.5682 \times 10^8.$$

Therefore, by Corollary 1, we know that the considered model (11) is passive. It should be pointed out that the conditions in [19]-[21] can not be applied to this example since it requires the differentiability of the time-varying delay.

5 Conclusions

In this paper, the passivity has been investigated for neural networks with time-varying delays of neutral type as well as generalized activation functions. By employing a combination of Lyapunov method, Newton-Leibniz formulation and LMI technique, a new delay-independent criterion for the passivity of the addressed neural networks has been established in terms of LMI, which can be checked numerically using the effective LMI toolbox in MATLAB. The obtained results generalize and improve the earlier publications, and remove the traditional assumptions on the differentiability of the discrete time-varying delay and the boundedness of its derivative. An example has been provided to demonstrate the effectiveness and less conservatism of the proposed criterion.

Acknowledgments. The authors would like to thank the reviewers and the editor for their valuable suggestions and comments which have led to a much improved paper. This work was supported by the National Natural Science Foundation of China under Grant 10772152.

References

1. Chen, T.P.: Global Exponential Stability of Delayed Hopfield Neural Networks. Neural Networks 14, 977–980 (2001)
2. Arik, S.: An Analysis of Exponential Stability of Delayed Neural Networks with Time Varying Delays. Neural Networks 17, 1027–1031 (2004)
3. Wang, Z.D., Ho, D.W.C., Liu, X.H.: State Estimation for Delayed Neural Networks. IEEE Transactions on Neural Networks 16, 279–284 (2005)
4. Cao, J.D., Song, Q.K.: Stability in Cohen-Grossberg Type BAM Neural Networks with Time-varying Delays. Nonlinearity 19, 1601–1617 (2006)
5. Zeng, Z.G., Wang, J.: Improved Conditions for Global Exponential Stability of Recurrent Neural Networks with Time-varying Delays. IEEE Transactions on Neural Networks 17, 623–635 (2006)
6. Liu, Y.R., Wang, Z.D., Liu, X.H.: Global Exponential Stability of Generalized Recurrent Neural Networks with Discrete and Distributed Delays. Neural Networks 19, 667–675 (2006)
7. Wang, Z., Shu, H., Liu, Y., Ho, D.W.C., Liu, H.: Robust Stability Analysis of Generalized Neural Networks with Discrete and Distributed Time Delays. Chaos, Solitons and Fractals 30, 886–896 (2006)
8. Li, P., Cao, J.D.: Stability in Static Delayed Neural Networks: A Nonlinear Measure Approach. Neurocomputing 69, 1776–1781 (2006)
9. Zhang, H.G., Wang, Z.S.: Global Asymptotic Stability of Delayed Cellular Neural Networks. IEEE Transactions on Neural Networks 18, 947–950 (2007)
10. Shen, Y., Wang, J.: Noise-Induced Stabilization of the Recurrent Neural Networks With Mixed Time-Varying Delays and Markovian-Switching Parameters. IEEE Transactions on Neural Networks 18, 1857–1862 (2007)
11. Song, Q.K., Wang, Z.D.: A delay-dependent LMI Approach to Dynamics Analysis of Discrete-time Recurrent Neural Networks with Time-varying Delays. Physics Letters A 368, 134–145 (2007)
12. Cao, J.D., Feng, G., Wang, Y.Y.: Multistability and Multiperiodicity of Delayed Cohen-Grossberg Neural Networks with A General Class of Activation Functions. Physica D 237, 1734–1749 (2008)
13. Liao, X.X., Luo, Q., Zeng, Z.G., Guo, Y.X.: Global Exponential Stability in Lagrange Sense for Recurrent Neural Networks with Time Delays. Nonlinear Analysis: Real World Applications 9, 1535–1557 (2008)
14. Huang, H., Feng, G., Cao, J.D.: An LMI Approach to Delay-dependent State Estimation for Delayed Neural Networks. Neurocomputing 71, 2857–2867 (2008)
15. Hu, L., Gao, H.J., Zheng, W.X.: Novel Stability of Cellular Neural Networks with Interval Time-varying Delay. Neural Networks 21, 1458–1463 (2008)
16. Yu, W.: Passivity Analysis for Dynamic Multilayer Neuro Identifier. IEEE Transactions on Circuits and Systems I 50, 173–178 (2003)
17. Li, C.G., Zhang, H.B., Liao, X.F.: Passivity and Passification of Fuzzy Systems with Time Delays. Computers and Mathematics with Applications 52, 1067–1078 (2006)

18. Gao, H., Chen, T., Chai, T.: Passivity and Passification for Networked Control Systems. SIAM Journal on Control and Optimization 46, 1299–1322 (2007)
19. Li, C.G., Liao, X.F.: Passivity Analysis of Neural Networks with Time Delay. IEEE Transactions on Circuits and Systems II 52, 471–475 (2005)
20. Park, J.H.: Further Results on Passivity Analysis of Delayed Cellular Neural Networks. Chaos, Solitons and Fractals 34, 1546–1551 (2007)
21. Lou, X.Y., Cui, B.T.: Passivity Analysis of Integro-differential Neural Networks with Time-varying Delays. Neurocomputing 70, 1071–1078 (2007)

Adaptive Exponential Synchronization of Stochastic Delay Neural Networks with Reaction-Diffusion

Birong Zhao[1] and Feiqi Deng[2]

[1] Department of Applied Mathematics , School of Mathematics and Information Science, Guangzhou University,Guangzhou, Guangdong 510006, China
[2] Control of Automation Science and Engineering, South China University of Technology, Guangzhou 510640, China
au_brzhao@163.com

Abstract. In the paper, adaptive exponential synchronization scheme for a class of reaction-diffusion neural networks with continuously distributed delays and stochastic influence are considered. An adaptive synchronization controller is derived to achieve the exponential synchronization of the drive-response structure of neural networks. Lyapunov stability theory, stochastic Fubini theorem and semimartingale theorem are used in our approach. It is shown that the approaches developed here extend and improve the ideas presented in recent literatures.

Keywords: Stochastic neutral networks, Delays, Exponential synchronization.

1 Introduction

Chaos synchronization has attracted increasing attention in both theory and applications since its introduction by Pecora and Carroll in 1990 [1,2]. Research on the synchronization of coupled chaotic systems has received considerable attention in the last decade due to its potential applications in many different areas including secure communication, chaos generators design, chemical reactions, biological systems, information science. A wide variety of approaches have been proposed for the synchronization of chaotic systems which include impulsive control method [3,4], adaptive design control [5,6], feedback control [7-9], and so on. Since artificial neural networks can exhibit some complicated dynamics and even chaotic behaviors, synchronization of chaotic neural networks has also become an important area of study. Nowadays, some authors pay attention to the synchronization of neural networks [10-21].However, strictly speaking, diffusion effects cannot be avoided in the neural networks when electrons are moving in asymmetric electromagnetic fields. So we must consider that the activations vary in space as well as in time. In [15-18], the stability of neural networks with diffusion terms, which are expressed by partial differential equations, has been considered. In this paper we will further consider the stochastic influence in the reaction–diffusion neural networks with continuously distributed delays. In fact a real system is usually affected by external perturbations which in many cases are of great uncertainty and hence may be treated as random, as pointed out by [19] that in real nervous systems synaptic transmission is a noisy process

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 550–559, 2009.

brought on by random fluctuations from the release of neurotransmitters, and other probabilistic causes. To the best of our knowledge, however, there are few results about stochastic effects to the synchronization property of neural networks with delays in the literature today. In this paper, the problem of exponential synchronization is investigated for the class of stochastic neural networks with time varying and distributed delays and reaction-diffusion terms based on the drive-response synchronization concept and Lyapunov stability theory. By using adaptive control approach in stead of the common linear coupling scheme, and by using Lyapunov stability theory, Stochastic Fubini theorem and semimartingale theorem, the suitable parameters update laws are proposed and the analytic results are developed to ensure exponential synchronization of the delayed neural networks with all the parameters unknown. Some comparisons are provided to illustrate that our results improve and general some of existing results.

2 Problem Statements and Preliminaries

In this section, we will give preliminary knowledge for our main results. Since most of the synchronization methods belong to master –slave (drive-response) type by one system driving another we mean that the two systems are coupled so that the behavior of the second is influenced by the behavior of the first one. But the behavior of the first one is independent of the second. The first system will be called the master system or drive system, and the second system will be the slave system or response system. In this paper, the object is to design a controller to let the slave system synchronize with the master system. Now we consider the reaction diffusion neural networks with continuously disturbed delays, which are described by the following partial differential equations:

$$\begin{aligned} du_i(t,x) &= \sum_{k=1}^{l} \frac{\partial}{\partial x_k}\left(D_{ik}(t,x,u)\frac{\partial u_i(t,x)}{\partial x_k}\right)dt + \Bigg[-d_i u_i(t,x) + \sum_{j=1}^{n} a_{ij} f_j(u_j(t,x)) \\ &+ \sum_{j=1}^{n} b_{ij} f_j(u_j(t-\tau_j(t),x)) + \sum_{j=1}^{n} c_{ij}\int_{-\infty}^{t} \kappa_{ij}(t-s) f_j(u_j(s,x))ds + J_i \Bigg] dt \end{aligned} \tag{1}$$

where $x = (x_1, x_2, \ldots, x_n)^T \in G \subset R^l$, G is a bounded compact set with smooth boundary ∂G and $mesG > 0$ in space R^l; $u_i(t,x)$ is the state of the i-th unit at time t and in space x, $f_i(\cdot)$ denotes the signal functions of the i th neurons at time t and in space x; J_i denotes the external inputs on the i th neurons; $D = diag(d_1, d_2, \ldots, d_n)$ is a diagonal matrix with $d_i > 0$, and denotes the rate with which the i-th neuron will reset its potential to the resting state in isolation when disconnected from the networks and external inputs; $A = (a_{ij})_{n\times n}$, $B = (b_{ij})_{n\times n}$ and $C = (c_{ij})_{n\times n}$ are real matrices, which denote the weights of neuron interconnections. $\tau_j(t)$, $j = 1,2,\ldots,n$ are time-varying delays of the neural network satisfying $0 \le \tau_j(t) \le \tau$ and $0 \le \dot{\tau}_j(t) \le \sigma < 1$; $\kappa_{ij}(\cdot), i, j = 1,2,\ldots,n$ are delay kernels. Smooth function $D_{ik} = D_{ik}(t,x,u) \ge 0$ corresponds to the transmission diffusion operator along the i-th neuron.

The boundary conditions and initial conditions are given by

$$\frac{\partial u_i(t,\mathbf{x})}{\partial \mathbf{n}} = 0, \quad i \in \mathcal{N}, (t,\mathbf{x}) \in \mathbb{R}^+ \times \partial G$$

$$u_i(s,\mathbf{x}) = \phi_i(s,\mathbf{x}) \quad i = \mathcal{N}, s \in (-\infty, 0]$$

We assume that the activation functions and delay kernels functions satisfy the following properties:

(H1) the neurons activation functions $f_i(\cdot)$, $i = 1,2,\ldots,n$ and σ_{il} are Lipschitz-continuous, with Lipschitz constants $L_i > 0$ and $L_{il} > 0$, respectively, and

$$\sum_{l=1}^{\infty} L_{il} < \infty, \text{ for } (i = 1,2,\ldots,n, l \in N)$$

(H2) The delay kernels functions $\kappa_{ij} : [0,\infty) \to [0,\infty)$ $(i, j = 1,2,\ldots n)$ are real-valued nonnegative continuous functions that satisfy the following conditions:

(i) $\int_0^{\infty} \kappa_{ij}(s)ds = 1$,and $\int_0^{\infty} s\kappa_{ij}(s)ds < \infty$

(ii) there exists a positive number μ such that $\int_0^{\infty} se^{\mu s}\kappa_{ij}(s)ds < \infty$

In this paper, we consider mode (1) as the master system. For drive system (1), we construct the response system as follows:

$$\begin{aligned} d\tilde{u}_i(t,x) = &\sum_{k=1}^{l} \frac{\partial}{\partial x_k}\left(D_{ik}(t,x,u)\frac{\partial \tilde{u}_i(t,x)}{\partial x_k}\right)dt + \left[-\bar{d}_i\tilde{u}_i(t,x) + \sum_{j=1}^{n} \bar{a}_{ij} f_j(\tilde{u}_j(t,x)) \right. \\ &+ \sum_{j=1}^{n} \bar{b}_{ij} f_j(\tilde{u}_j(t-\tau_j(t),x)) + \sum_{j=1}^{n} \bar{c}_{ij} \int_{-\infty}^{t} \kappa_{ij}(t-s) f_j(\tilde{u}_j(s,x))ds + J_i + v_i(t) \Big] dt \qquad (2) \\ &+ \sum_{l=1}^{\infty} \sigma_{il}(t, e_i(t,x))dw_{il}(t) \qquad i = 1,2,\ldots,n \end{aligned}$$

where the parameters $\bar{D} = diag(d_1, d_2, \ldots, d_n)$, $\bar{A} = (\bar{a}_{ij})_{n\times n}$, $\bar{B} = (\bar{b}_{ij})_{n\times n}$ and $\bar{C} = (\bar{c}_{ij})_{n\times n}$ are completely unknown. $\tilde{u}_i(t,x)$ denotes the state variable of the response system, $v_i(t)$ indicates the external control inputs that will be appropriately design for a control objective. Moreover, $\{w_{il}(t)\}$ are independent scalar standard Wiener processes on the complete probability space $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\in I}, \mathbb{P})$ with the natural filtration $\{\mathcal{F}_t\}_{t\geq 0}$ generated by the standard Wiener process $\{w(s)\}$ which is independent of $w_{il}(t)$,where we associate Ω with the canonical space generated by $w(t)$, and denoteby $\mathcal{F}$ the associated σ -algebra generated by $w(t)$ with the probability measure $\mathbb{P}$ .The initial condition of system (4) is given in the following form: $\tilde{u}_i(t,x) = \varphi_i(s,x), \quad s \in (-\infty, 0], \quad i = 1,2,\ldots,n$, The goal of control is to design and implement appropriate controller $v_i(t)$ for the response system and parameters adaptive estimation laws of $\bar{D}, \bar{A}, \bar{B}$ and $\bar{C}$, such that the controlled response system (4) could be synchronous with the drive system (1)and all the parameters $\bar{D} \to D, \bar{A} \to A, \bar{B} \to B$ and $\bar{C} \to C$ as $t \to +\infty$ .Inspired by the ideas in[21], the

control inputs in the response system are taken as $v_i(t) = \varepsilon_i e_i(t,x)$, where ε_i varies with the synchronization error $e_i(t,x)$,the synchronization error $e(t,x)$ be defined as $e(t,x) = (e_1(t,x), e_2(t,x),\ldots,e_n(t,x))$,Where $e_i(t,x) = \tilde{u}_i(t,x) - u_i(t,x)$ .Therefore the error dynamics between (1) and (2) can be expressed by

$$
\begin{aligned}
de_i(t,x) &= \sum_{k=1}^{m} \frac{\partial}{\partial x_k}\left(D_{ik} \frac{\partial e_i(t,x)}{\partial x_k} \right) dt - \left[d_i e_i(t,x) + \sum_{j=1}^{n} a_{ij} g_j(e_j(t,x)) \right] dt \\
&+ \left[\sum_{j=1}^{n} b_{ij} g_j(e_j(t-\tau_j(t),x)) + \sum_{j=1}^{n} c_{ij} \int_{-\infty}^{t} \kappa_{ij}(t-s) f_j(e_j(s,x)) ds - (\bar{d}_i - d_i)\tilde{u}_i(t,x) \right] dt \\
&+ \left[\sum_{j=1}^{n} (\bar{a}_{ij} - a_{ij}) f_j(\tilde{u}_j(t,x)) + \sum_{j=1}^{n} (\bar{b}_{ij} - b_{ij}) f_j(\tilde{u}_j(t-\tau_j(t),x)) + v_i(t) \right] dt \\
&+ \sum_{j=1}^{n} (\bar{c}_{ij} - c_{ij}) \int_{-\infty}^{t} \kappa_{ij}(t-s) f_j(\tilde{u}_j(s,x)) ds dt + \sum_{l=1}^{\infty} \sigma_{il}(e_i(t,x)) dw_{il}(t)
\end{aligned}
\tag{3}
$$

Where $g_j(e_j(t,x)) = f_j(e_j(t,x)) + u_j(t,x)) - f_j(u_j(t,x))$ .According to the assumption (H1), $g_j(\cdot)$ possesses the following properties:

$\left|g_i(e_i(t,x))\right| \le L_i \left|e_i(t,x\right|$ and $g_i(0) = 0$, $i = 1, 2, \cdots, n$

Throughout this paper, we denote $u = (u_1,\ldots,u_n)^T$ and $L^2(G)$ is the space of scalar value Lebesgue measurable functions on G which is a Banach space for the L_2 -norm $\|v\|_2 = \left(\int_G |v(x)| dx\right)^{1/2}$, $v \in L^2(G)$.

Then we define the norm $\|u\| \triangleq \left(\sum_{i=1}^{n} \|u_i\|_2^2 \right)^{1/2}$

To give our main results in the following section, we need the following definition.

Definition 1. the system(1) and the uncontrolled system(4) are said to be exponentially synchronized in mean square if there exists a pair of positive constants μ and M such that

$$
\mathbb{E}\left\|\tilde{u}(t,x) - u(t,x)\right\|^2 \le M \mathbb{E}\left\|\varphi(s,x) - \phi(s,x)\right\|^2 e^{-\mu t} \qquad t \ge 0 \tag{4}
$$

In this case

$$
\limsup_{t\to\infty} \frac{1}{t} \ln\left(\mathbb{E}\left\|\tilde{u}(t,x) - u(t,x)\right\|^2 \right) \le -\mu \tag{5}
$$

The left-hand side of (5) is called the Lyapunov synchronization exponent of system (1) and system(2).

3 Criteria of Synchronization

In this section, based on the Lyapunov–Krasovskii stability theorem and a stochastic analysis approach, new criteria are presented for the global asymptotical stability of

the equilibrium point of system (3), and thus the drive system (1) synchronize with the response system (3). We have the following theorem:

Theorem 1. Under the assumptions (H1) and (H2), the stochastic response system (4) is globally exponentially synchronized in mean square with the drive system (1), if the feedback strength $\varepsilon = diag(\varepsilon_1, \ldots, \varepsilon_n)$ with the undated law is chosen as

$$\dot{\varepsilon}_i = -\delta_i e^{\mu t} e_i^2$$

and the parameters adaptive laws of

$\overline{D} = diag(\overline{d}_1, \ldots, \overline{d}_n), \overline{A} = (\overline{a}_{ij})_{n\times n}, \overline{B} = (\overline{b}_{ij})_{n\times n}$ and $\overline{C} = (\overline{c}_{ij})_{n\times n}$ are chosen as

$$\overline{d}_i = \rho_i e_i \tilde{u}_i(t,x) e^{\mu t}, \overline{a}_{ij} = -\alpha_{ij} e_i f_j(\tilde{u}_i(t,x)) e^{\mu t},$$

$$\overline{b}_{ij} = -\beta_{ij} e_i f_j(\tilde{u}_i(t-\tau_j(t),x)) e^{\mu t}, \overline{c}_{ij} = -\gamma_{ij} e_i \int_{-\infty}^{t} \kappa_{ij}(t-s) f_j(\tilde{u}_i(t,x)) ds e^{\mu t}$$

where $\mu \geq 0$ a real number is properly selected, $\delta_i > 0, \rho_i > 0, \alpha_{ij} > 0,$

$\beta_{ij} > 0,\ \gamma_{ij} > 0$ are arbitrary constants.

Proof. Define

$$v_i(t,e(t,x)) = \lambda_i \Big\{ e_i^2 e^{\mu t} + \frac{1}{1-\sigma} \sum_{j=1}^{n} L_j \left| b_{ij} \right| \int_{t-\tau_j(t)}^{t} e_j^2(s,x) e^{\mu(s+\sigma)} ds$$

$$+ \sum_{j=1}^{n} \left| c_{ij} \right| \int_0^{\infty} \kappa_{ij}(s) \int_{t-s}^{t} g_j^2(e_j(z,x)) e^{\mu(z+s)} dz ds + \frac{(\overline{d}_i - d_i)^2}{\rho_i}$$

$$+ \sum_{j=1}^{n} \frac{(\overline{a}_{ij} - a_{ij})^2}{\alpha_{ij}} + \sum_{j=1}^{n} \frac{(\overline{b}_{ij} - b_{ij})^2}{\beta_{ij}} + \sum_{j=1}^{n} \frac{(\overline{c}_{ij} - c_{ij})^2}{\gamma_{ij}} + \frac{(\varepsilon_i + m_i)^2}{\delta_i} \Big\}$$

We construct a Lyapunov auxiliary functional as

$$V(t) = V(t,e(t,x)) = \int_{\Omega} \overline{V}(t,e(t,x)) dx = \int_{\Omega} \sum_{i=1}^{n} v_i(t,e(t,x)) dx \tag{6}$$

Obviously, $V(t,e(t,x))$ is positive definite and it is a compound function of stochastic process. By using *Itô* differential formula, we obtain that

$$dV(t,e(t,x)) = \int_{\Omega} \left[\mathcal{L}\overline{V}(t,e(t,x)) dt + \left(\frac{\partial \overline{V}}{\partial e} \right)^T \sigma(e(t,x)) dw(t) \right] dx$$

$$\int_{\Omega} \sum_{i=1}^{n} \lambda_i \Big\{ 2 e_i d e_i(t,x) e^{\mu t} + [\mu e^{\mu t} e_i^2 + \frac{1}{1-\sigma} \sum_{j=1}^{n} L_j \left| b_{ij} \right| e_j^2(t,x) e^{\mu(t+\sigma)}$$

$$- \sum_{j=1}^{n} L_j \left| b_{ij} \right| e_j^2(t-\tau_j(t),x) e^{\mu t} + \sum_{j=1}^{n} \left| c_{ij} \right| \int_0^{\infty} \kappa_{ij}(s) g_j^2(e_j(t,x)) e^{\mu(t+s)} ds$$

$$- \sum_{j=1}^{n} \left| c_{ij} \right| \int_0^{\infty} \kappa_{ij}(s) g_j^2(e_j(t-s,x)) e^{\mu t} ds + \frac{(\overline{d}_i - d_i)\dot{\overline{d}}}{\rho_i} + \sum_{j=1}^{n} \frac{(\overline{a}_{ij} - a_{ij})\dot{\overline{a}}_{ij}}{\alpha_{ij}}$$

$$+ \sum_{j=1}^{n} \frac{(\overline{b}_{ij} - b_{ij})\dot{\overline{b}}_{ij}}{\beta_{ij}} + \sum_{j=1}^{n} \frac{(\overline{c}_{ij} - c_{ij})\dot{\overline{c}}}{\gamma_{ij}}_{ij} + \frac{(\varepsilon_i + m_i)\dot{\varepsilon}_i}{\delta_i}] dt \Big\} dx$$

It is easy to calculate by the boundary condition and Green formula that

$$\int_\Omega \sum_{i=1}^{n} e_i(t,x) \sum_{k=1}^{l} \frac{\partial}{\partial x_k}\left(D_{ik} \frac{\partial e_i(t,x)}{\partial x_k} \right) dx \le -\sum_{i=1}^{n}\sum_{k=1}^{l} \int_\Omega D_{ik} \left(\frac{\partial e_i}{\partial x_k} \right)^2 dx$$

Thus we have

$$dV(t,e(t,x)) \le \int_\Omega \sum_{i=1}^{n} \lambda_i \left\{ 2e_i e^{\mu t} \left[-d_i e_i(t,x) + \sum_{j=1}^{n} a_{ij} g_j(e_j(t,x)) \right.\right.$$

$$\left. + \sum_{j=1}^{n} b_{ij} g_j(e_j(t-\tau_j(t),x)) + \sum_{j=1}^{n} c_{ij} \int_{-\infty}^{t} \kappa_{ij}(t-s) f_j(e_j(s,x)) ds \right]$$

$$+ \mu e^{\mu t} e_i^2 + \frac{1}{1-\sigma} \sum_{j=1}^{n} L_j \left| b_{ij} \right| e_j^2(t,x) e^{\mu(t+\sigma)} - \sum_{j=1}^{n} L_j \left| b_{ij} \right| e_j^2(t-\tau_j(t),x) e^{\mu t}$$

$$+ \sum_{j=1}^{n} \left| c_{ij} \right| \int_0^\infty \kappa_{ij}(s) g_j^2(e_j(t,x)) e^{\mu(t+s)} ds - \sum_{j=1}^{n} \left| c_{ij} \right| \int_0^\infty \kappa_{ij}(s) g_j^2(e_j(t-s,x)) e^{\mu t} ds$$

$$\left. + 2\varepsilon_i e_i^2 e^{\mu t} + \frac{(\varepsilon_i + m_i)\dot{\varepsilon}}{\delta_i} \right\} dt dx + \int_\Omega \sum_{i=1}^{n} \tilde{\sigma}_i(t,e_j(t,x)) \tilde{\sigma}_i^T(t,e_j(t,x))) dt dx$$

$$+ \int_\Omega 2e^{\mu t} \sum_{i=1}^{n} \sum_{l=1}^{\infty} \lambda_i e_i(t,x) \sigma_{il}(e_i(t,x)) dw_{il} dx$$

$$\le \int_\Omega e^{\mu t} \sum_{i=1}^{n} \lambda_i \left(-2d_i e_i^2 + 2\sum_{j=1}^{n} \left| e_i \right| \left| a_{ij} \right| L_j \left| e_j \right| + 2\sum_{j=1}^{n} \left| e_i \right| \left| b_{ij} \right| L_j \left| e_j(t-\tau_j(t),x) \right| \right.$$

$$+ 2\sum_{j=1}^{n} \left| e_i \right| \left| c_{ij} \right| \int_{-\infty}^{t} \kappa_{ij}(t-s) \left| f_j(e_j(s,x)) \right| ds + \mu e_i^2 + \frac{e^{\mu\sigma}}{1-\sigma} \sum_{j=1}^{n} L_j \left| b_{ij} \right| e_j^2(t,x)$$

$$- \sum_{j=1}^{n} L_j \left| b_{ij} \right| e_j^2(t-\tau_j(t),x) + \sum_{j=1}^{n} \left| c_{ij} \right| \int_0^\infty \kappa_{ij}(s) e^{\mu s} g_j^2(e_j(t,x)) ds -$$

$$\left. - \sum_{j=1}^{n} \left| c_{ij} \right| \int_0^\infty \kappa_{ij}(s) g_j^2(e_j(t-s,x)) ds - 2m_i e_i^2 \right) \right\} dt dx$$

$$+ \int_\Omega \sum_{i=1}^{n} \tilde{\sigma}_i(t,e_j(t,x)) \tilde{\sigma}_i^T(t,e_j(t,x))) dt dx$$

$$+ 2\int_\Omega \sum_{i=1}^{n} \lambda_i e_i(t,x) e^{\mu t} \sum_{l=1}^{\infty} \sigma_{il}(e_i(t,x)) \mathbf{d}W_{il}(t) dx$$

By Young's Inequality and (H2), we have

$$dV(t) \le \int_\Omega e^{\mu t} \sum_{i=1}^{n} \lambda_i \Bigg(-2d_i + \sum_{j=1}^{n} |a_{ij}| L_j + \sum_{j=1}^{n} \frac{\lambda_j}{\lambda_i} |a_{ji}| L_i + \sum_{j=1}^{n} |b_{ij}| L_j$$

$$+ \sum_{j=1}^{n} |c_{ij}| + \mu + \frac{e^{\mu\sigma}}{1-\sigma} \sum_{j=1}^{n} \frac{\lambda_j}{\lambda_i} |b_{ji}| L_i + \int_\Omega e^{\mu t} \sum_{i=1}^{n} \sum_{l=1}^{\infty} \lambda_i \sigma_{il}^2 (t, e_i(t,x)) dt dx$$

$$+ \sum_{j=1}^{n} \frac{\lambda_j}{\lambda_i} |c_{ji}| L_i^2 \int_0^\infty \kappa_{ij}(s) e^{\mu s} ds - 2m_i \Bigg) e_i^2 dx dt$$

$$+ \int_\Omega 2e^{\mu t} \sum_{i=1}^{n} \sum_{l=1}^{\infty} \lambda_i e_i(t,x) \sigma_{il}(e_i(t,x)) dw_{il}(t) dx$$

$$\le \int_\Omega e^{\mu t} \sum_{i=1}^{n} \lambda_i \Bigg(-2d_i + \sum_{j=1}^{n} |a_{ij}| L_j + \sum_{j=1}^{n} \frac{\lambda_j |a_{ji}| L_i}{\lambda_i} + \sum_{j=1}^{n} |b_{ij}| L_j + \sum_{j=1}^{n} |c_{ij}|$$

$$+ \mu + \frac{e^{\mu\sigma}}{1-\sigma} \sum_{j=1}^{n} \frac{\lambda_j |b_{ji}| L_i}{\lambda_i} + \sum_{j=1}^{n} \frac{\lambda_j |c_{ji}| L_i^2}{\lambda_i} \int_0^\infty \kappa_{ij}(s) e^{\mu s} ds - 2m_i \Bigg) e_i^2 dx dt$$

$$+ \int_\Omega e^{\mu t} \sum_{i=1}^{n} \sum_{l=1}^{\infty} \lambda_i L_{il}^2 e_i^2 dt dx + \int_\Omega 2e^{\mu t} \sum_{i=1}^{n} \sum_{l=1}^{\infty} \lambda_i e_i(t,x) \sigma_{il}(e_i(t,x)) dw_{il}(t) dx$$

According to (H2), if

$$-2d_i + \sum_{j=1}^{n} |a_{ij}| L_j + \sum_{j=1}^{n} \frac{\lambda_j}{\lambda_i} |a_{ji}| L_i + \sum_{j=1}^{n} |b_{ij}| L_j + \sum_{j=1}^{n} |c_{ij}| + \frac{1}{1-\sigma} \sum_{j=1}^{n} \frac{\lambda_j}{\lambda_i} |b_{ji}| L_i$$

$$+ \sum_{j=1}^{n} \frac{\lambda_j}{\lambda_i} |c_{ji}| L_i^2 - 2m_i + \sum_{l=1}^{\infty} \lambda_i L_{il}^2 < 0$$

We can choose a small number $\mu > 0$ such that

$$-2d_i + \sum_{j=1}^{n} |a_{ij}| L_j + \sum_{j=1}^{n} \frac{\lambda_j}{\lambda_i} |a_{ji}| L_i + \sum_{j=1}^{n} |b_{ij}| L_j + \frac{e^{\mu\sigma}}{1-\sigma} \sum_{j=1}^{n} \frac{\lambda_j}{\lambda_i} |b_{ji}| L_i$$

$$+ \mu + \sum_{j=1}^{n} |c_{ij}| + \sum_{j=1}^{n} \frac{\lambda_j}{\lambda_i} |c_{ji}| L_i^2 \int_0^\infty \kappa_{ij}(s) e^{\mu s} ds - 2m_i + \sum_{l=1}^{\infty} \lambda_i L_{il}^2 < 0$$

Thus, the constants m_i can be properly chosen as

$$m_i = -d_i + \sum_{j=1}^{n} |a_{ij}| L_j + \sum_{j=1}^{n} \frac{\lambda_j}{\lambda_i} |a_{ji}| L_i + \sum_{j=1}^{n} |b_{ij}| L_j + \sum_{j=1}^{n} |c_{ij}|$$

$$+ \frac{1}{1-\sigma} \sum_{j=1}^{n} \frac{\lambda_j}{\lambda_i} |b_{ji}| L_i + \sum_{j=1}^{n} \frac{\lambda_j}{\lambda_i} |c_{ji}| L_i^2 + \sum_{l=1}^{\infty} \lambda_i L_{il}^2$$

Then we have

$$dV(t, e(t,x)) \le \int_\Omega \alpha e^{\mu t} \sum_{i=1}^{n} \lambda_i e_i^2 dx dt$$

$$+ \int_\Omega 2e^{\mu t} \sum_{i=1}^{n} \sum_{l=1}^{\infty} \lambda_i e_i(t,x) \sigma_{il}(e_i(t,x)) dw_{il}(t) dx$$

Integrating the two sides of the above inequality from 0 to t and taking mathematical expectation, we obtain that

$$(\min_{1\le i\le n}\{\lambda_i\})e^{\mu t}\mathbb{E}\left\|\tilde{u}(t,x)-u(t,x)\right\|^2$$

$$\le \mathbb{E}V(0,e(0,x))+\mathbb{E}\int_0^t\int_\Omega 2e^{\mu t}\sum_{i=1}^n\sum_{l=1}^\infty \lambda_i e_i(t,x)\sigma_{il}(e_i(t,x))dxdw_{il}(s)$$

In the sequel, we define that

$$W(t)=\frac{1}{\rho_i}(\bar{d}_i-d_i)^2+\sum_{j=1}^n\frac{1}{\alpha_{ij}}(\bar{a}_{ij}-a_{ij})^2+\sum_{j=1}^n\frac{1}{\beta_{ij}}(\bar{b}_{ij}-b_{ij})^2$$
$$+\sum_{j=1}^n\frac{1}{\gamma_{ij}}(\bar{c}_{ij}-c_{ij})^2+\frac{1}{\delta_i}(\varepsilon_i+m_i)^2 \tag{7}$$

According to (6), we have

$$V(0)=\int_\Omega\sum_{i=1}^n\lambda_i\left\{e_i^2(0,x)+\frac{1}{1-\sigma}\sum_{j=1}^n L_j\left|b_{ij}\right|\int_{-\tau_j(0)}^0 e_j^2(s,x)e^{\mu(s+\sigma)}ds\right.$$
$$\left.+\sum_{j=1}^n\left|c_{ij}\right|\int_0^\infty\kappa_{ij}(s)\int_{-s}^0 g_j^2(e_j(z,x))e^{\mu(z+s)}dzds+W(0)\right\}dx$$
$$\le\int_\Omega\sum_{i=1}^n\lambda_i\left\{\frac{1}{1-\sigma}\sum_{j=1}^n L_j\left|b_{ij}\right|\sigma e_j^2(0,x)e^{\mu\sigma}+e_i^2(0,x)\right.$$
$$\left.+\sum_{j=1}^n\left|c_{ij}\right|L_i^2e_j^2(0,x)\int_0^\infty s\kappa_{ij}(s)e^{\mu s}ds+W(0)\right\}dx \tag{8}$$
$$\le\max_{1\le i\le n}\{\lambda_i\}\left(1+\frac{1}{1-\sigma}\sum_{j=1}^n\frac{\lambda_j\left|b_{ji}\right|L_ie^{\mu\sigma}}{\lambda_i}+\sum_{j=1}^n\frac{\lambda_j\left|c_{ji}\right|L_i^2}{\lambda_i}\int_0^\infty\kappa_{ij}(s)e^{\mu s}ds+W(0)\right)$$
$$\times\left\|\varphi(s,x)-\phi(s,x)\right\|$$

Noticing that

$$\mathbb{E}\int_0^t 2e^{\mu s}\sum_{i=1}^n\sum_{l=1}^\infty\lambda_i\int_\Omega e_i(t,x)\sigma_{il}(e_i(t,x))dxdw_{il}(s)=0$$

Thus we have

$$(\min_{1\le i\le n}\{\lambda_i\})e^{\mu t}\mathbb{E}\left\|\tilde{u}(t,x)-u(t,x)\right\|^2\le\eta\mathbb{E}\left\|\varphi(s,x)-\phi(s,x)\right\|$$

where $\lambda=\max_{1\le i\le n}\{\lambda_i\}$,

$$\eta=\max_{1\le i\le n}\{\lambda_i\}\left(1+\frac{1}{1-\sigma}\sum_{j=1}^n\frac{\lambda_j\left|b_{ji}\right|L_ie^{\mu\sigma}}{\lambda_i}+\sum_{j=1}^n\frac{\lambda_j\left|c_{ji}\right|L_i^2}{\lambda_i}\int_0^\infty\kappa_{ij}(s)e^{\mu s}ds+W(0)\right)$$

Therefore

$$\mathbb{E}\left\|\tilde{u}(t,x)-u(t,x)\right\|^2\le\frac{\eta}{\lambda}\mathbb{E}\left\|\varphi(s,x)-\phi(s,x)\right\|e^{-\mu t} \tag{9}$$

So we conclude that the drive system (1) and the response system (2) are exponentially synchronized in mean square. And the Lyapunov synchronization exponent is μ . This completes the proof.

Now we compare our results with the previous results derive in the literature for the usual continuous distributed delay without diffusion or stochastic perturbation. Set $\sigma_{il} = 0$, then system(1) becomes the deterministic continuously distributed delayed reaction diffusion neutral networks which have been extensively studied in[21]. Set $D_{ik} = 0$, then the system(1) reduces to the following usual delayed neural networks $\dot{u}_i(t) = -d_i u_i(t) + \sum_{j=1}^{n} a_{ij} f_j(u_j(t)) + \sum_{j=1}^{n} b_{ij} f_j(u_j(t-\tau_j(t))) + J_i$ Which have been extensively studied in [5,6,8,12,14]. It is easy to see that our results expanded the models in [5,6,8,12,14]

References

1. Pecora, L.M., Carrol, T.L.: Synchronization in Chaotic Systems. Phys. Rev. Lett. 64, 821–833 (1990)
2. Pecora, L.M., Carrol, T.L., Johnson, G.A.: Fundamentals of Synchronization in Chaotic Systems, Concepts, and Applications. Chaos 7, 520–543 (1998)
3. Khadra, A., Liu, X., Shen, X.: Impulsively Synchronizing Chaotic Systems with Delay and Applications to Secure Communication. Automatica 41, 1491–1502 (2005)
4. Wang, Y., Guan, Z., Xiao, J.: Impulsive Control for Synchronization of a Class of Continuous Systems. Chaos 14, 199–203 (2004)
5. Sun, Y., Cao, J.: Adaptive Lag Synchronization of Unknown Chaotic Delayed Neural Networks with Noise Perturbation. Phys. Lett. A 364, 277–285 (2007)
6. Xiong, W., Xie, W., Cao, J.: Adaptive Exponential Synchronization of Delayed Chaotic Networks. Physica A 370, 832–842 (2006)
7. Sun, J.: Delay-dependent Stability Criteria for Time-delay Chaotic Systems Via Time-delay Feedback Control. Chaos, Solitons & Fractals 21, 143–150 (2004)
8. Cao, J., Li, H., Daniel, W.: Synchronization Criteria of Lure Systems with Time-delay Feedback Control. Chaos, Solitons & Fractals 23, 1285–1298 (2005)
9. Park, J.: A Novel Criterion for Delayed Feedback Control of Time-delay Chaotic Systems. Chaos, Solitons & Fractals 23, 495–501 (2005)
10. Li, Z., Chen, G.: Global Synchronization and Asymptotic Stability of Complex Dynamical Networks. IEEE.Trans. Circuits Syst. 53, 28–33 (2006)
11. Li, P., Cao, J., Wang, Z.: Robust Impulsive Synchronization of Coupled Delayed Neural Networks with Uncertainties. Physica A 373, 261–272 (2007)
12. Chen, G., Zhou, J., Liu, Z.: Global Synchronization of Coupled Delayed Neural Networks with Application to Chaotic CNN models. Int. J. Bifurcat. Chaos 14, 2229–2240 (2004)
13. Lu, W., Chen, T.: Synchronization of Coupled Connected Neural Networks with Delays. IEEE. Trans. Circuit Syst. II 51, 2491–2503 (2004)
14. Lu, H.T., Van, L.C.: Synchronization of Chaotic Neural Networks Via Output or State Coupling. Chaos, Solitons & Fractals 30, 166–176 (2006)
15. Wang, Y., Cao, J.: Synchronization of a Class of Delayed Neural Networks with Reaction-Diffusion Terms. Phys Lett. A 369, 201–211 (2007)

16. Song, Q.K., Zhao, Z.J., Li, Y.M.: Global Exponential Stability of BAM with Distributed Delays and Reaction-diffusion Terms. Phys. Lett. A 335, 213–225 (2005)
17. Lou, X., Cui, B.: Boundedness and Exponential Stability for Nonautonomous Cellular Neural Networks with Reaction-diffusion Terms. Chaos, Solitons & Fractals 33, 653–662 (2007)
18. Yuan, K., Cao, J., Li, H.: Robust Stability of Switched Cohen-Grossberg Neural Networks with Mixed Time-varying Delays. IEEE Trans. Syst. Man Cybernet – Part B 36, 1356–1363 (2006)
19. Luo, Q.I., Deng, F.Q., Zhao, B.R.: Stabilization of Stochastic Hopfield Neural Network with Distributed Parameters. Science in China (F) 7, 752–762 (2004)
20. Lv, Y., Lv, W., Sun, J.H.: Convergence Dynamics of Stochastic Reaction–diffusion Recurrent Neural Networks with Continuously Distributed Delays. Nonlinear Analysis: Real World Applications 9, 1590–1606 (2008)
21. Sheng, L., Yang, H.Z., Lou, X.Y.: Adaptive Exponential Synchronization of Delayed Neural Networks with Reaction-diffusion Terms. Chaos, Solitons and Fractals, 1–10 (2007)
22. Wang, Z., Liu, Y., Liu, X.: On Global asymptotic Stability of Neural Networks with Discrete and Distributed Delays. Phys. Lett. A 345, 299–308 (2005)

Global and Local Synchronization of General Multi-linked Delayed Complex Dynamical Networks

Yongqing Zhao and Minghui Jiang

Institute of Nonlinear Complex Systems, China Three Gorges University, Yichang 443000, China

Abstract. This paper investigates the phenomena of a new and general multi-linked complex network with delayed nodes and couplings. Via constructing Lyapunov-Krasovskii function and using inequality technique, the criteria for locally exponential synchronization and globally asymptotical synchronization are derived. The latter neither requires the outer-coupling configuration matrix to be symmetric and irreducible nor needs the network to be linearized at its synchronization state, it is obtained by Kronecker product method. Finally, a numerical example illustrates the theoretical results.

Keywords: Complex networks, Synchronization, Delayed couplings, Delayed nodes, Multi-links.

1 Introduction

In recent decades, complex networks occur in different fields, such as physics, biology, neuroscience, chemistry and social science, etc. All kinds of dynamics of complex networks have been extensively investigated, among them, synchronization which was discovered by Pecora and Carroll in 1990 [1], have received much attention from the scientific community [1,2,3,4,5,6,7,8,9,10,11,12].

Recently, Wang and Chen introduced an uniform complex network model and also investigated its synchronization phenomena in small-world and scale-free networks [2,3,4]. To a certain extent, this model reflects the complexity of networks, but time delay which exists in real world more commonly was not considered. [5,6,7,8,9,10] studied general complex networks with coupling delay and derived some synchronization conditions respectively. In [11], a general complex dynamical network with delayed nodes was discussed and an adaptive feedback controller for synchronization was designed. Global synchronization of complex dynamical networks with time-varying delayed nodes was investigated in [12].

Note that in most of these works, linearizing the network locally at its synchronization state is the main approach used to analyze the synchronization phenomena of a complex dynamical network. It is why they only obtained local synchronization criteria. We can also find that all the papers above did not consider the more general complex dynamical networks which contain both delayed

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 560–569, 2009.

nodes and delayed couplings. Furthermore, all the network models discussed in the above works just have single link. These simplicities can not reflect the real world very well.

In this paper, a new general multi-linked complex dynamical network model with both delayed nodes and delayed couplings is introduced, and then we investigated its synchronization phenomena. Criteria for exponentially synchronization and globally asymptotical synchronization are derived, and the globally asymptotical synchronization criterion obtained in this paper does not require the coupling configuration matrix to be symmetric and irreducible. Furthermore, the Kronecker product method is used instead of linearizing the network locally at its synchronization state.

Notation: $\lambda_{max}(P)$ and $\lambda_{min}(P)$ denote the maximum and minimum eigenvalues of matrix P respectively. $A \otimes B$ denotes the Kronecker product of matrixes A and B.

2 Mathematical Preparation

In this paper, we consider a new delayed dynamical network with multi-links, this network is consisted of N linearly and diffusively coupled identical nodes, and each node of the network is an n-dimensional time-delayed dynamical system. The state equations of the entire network are

$$\begin{aligned}\dot{x}_i(t) = f_1(x_i(t)) + f_2(x_i(t-\tau_0)) + \varepsilon_1 \sum_{j=1}^{N} a_{(1)ij} \Gamma_1 x_j(t-\tau_1) \\ +\varepsilon_2 \sum_{j=1}^{N} a_{(2)ij} \Gamma_2 x_j(t-\tau_2), \qquad i = 1, 2, \ldots, N, \end{aligned} \tag{1}$$

where $x_i(t) = (x_{i1}(t), x_{i2}(t), \ldots, x_{in}(t))^T \in R^n$ is the state variable of the ith node. Γ_k $(k = 1, 2)$ is the inner-coupling matrix which describes the individual coupling between two connected nodes of the kth subnetwork, $\tau_0 \geq 0$ is the time delay of individual node, $\tau_1 \geq 0$ and $\tau_2 \geq 0$ are the coupling time delays, $f_k : R^n \to R^n$ $(k = 1, 2)$ are vector-valued function describing the dynamics of an individual node, ε_1 and ε_2 describe the time-invariant coupling strength. $\mathbf{A}_k = a_{(k)ij} \in R^{n \times n}$ $(k = 1, 2)$ is the outer-coupling configuration matrix. It is described as follow: from node i to node j $(i \neq j)$, $a_{(k)ij} \geq 0$ and

$$a_{(k)ii} = -\sum_{j=1 j \neq i}^{N} a_{(k)ij}, \quad k = 1, 2, \quad i = 1, 2, \ldots, N. \tag{2}$$

Lemma 1. *[10] Suppose the outer-coupling configuration matrix $A \in R^{N \times N}$ is symmetric and irreducible, then there exists a unitary matrix $U = (u_1, u_2, \ldots, u_N)$ such that $A = U \Lambda U^T$, where $U^T U = I$, $\Lambda = diag(\lambda_1, \lambda_2, \ldots, \lambda_N)$, $0 = \lambda_1 > \lambda_2 \geq \ldots \geq \lambda_N$.*

Lemma 2. *[7] Let $V(t) > 0$ for $t \in R$, $\tau \in [0, \infty)$ and $t_0 \in R$. Suppose that*

$$\dot{V}(t) \leq -aV(t) + b \sup_{t-\tau \leq \alpha \leq t} V(\alpha) \tag{3}$$

for all $t > t_0$. If $a > b > 0$, there exist two constants $\gamma > 0$ and $k > 0$, such that

$$V(t) \leq ke^{-\gamma(t-t_0)}, \quad t > t_0. \tag{4}$$

Lemma 3. *[7] Suppose $P \in R^{n\times n}$ is a real symmetric matrix, and $w \in R^n$ is a real vector, then*

$$\lambda_{Pmin} w^T w \leq w^T P w \leq \lambda_{Pmax} w^T w. \tag{5}$$

Lemma 4. *[13] For any vectors x, $y \in R^n$ and positive definite matrix $Q \in R^{n\times n}$, the following matrix inequality holds:*

$$2x^T y \leq x^T Q x + y^T Q^{-1} y. \tag{6}$$

Definition 1. *[5] The delayed multi-linked dynamical network (1) is said to achieve(asymptotical) synchronization if*

$$x_1(t) = x_2(t) = \ldots = x_N(t) = s(t), \quad as \quad t \to \infty, \tag{7}$$

where $s(t) \in R^n$ is a solution of the individual system, or a periodic orbit, or an orbit of a chaotic attractor, namely,

$$\dot{s}(t) = f_1(s(t)) + f_2(s(t - \tau_0)). \tag{8}$$

Definition 2. *In this paper, the vector-valued function f is called to be DLC (Double-Lipschitz Continuous), if there exist two nonnegative constants α and β such that*

$$\beta\|y(t) - x(t)\| \leq \|f(y(t)) - f(x(t))\| \leq \alpha\|y(t) - x(t)\|$$

holds for any t. Here $x(t)$, $y(t)$ are time-varying vectors, and the norm $\|\cdot\|$ of a vector is defined as $\|x\| = (x^T x)^{1/2}$.

3 Main Results

3.1 Locally Exponential Synchronization

In this part, we suppose the outer coupling configuration matrixes $A_1 = A_2 = A \in R^{N\times N}$ and A is symmetric and irreducible. By Lemma (1), one has that $A^T U = U\Lambda$, where $U^T U = I$, $\Lambda = diag(\lambda_1, \lambda_2, \ldots, \lambda_N), and\ 0 = \lambda_1 > \lambda_2 \geq \ldots \geq \lambda_N$. Under this assumption, the network (1) becomes

$$\begin{aligned} \dot{x}_i(t) = f_1(x_i(t)) + f_2(x_i(t - \tau_0)) + \varepsilon_1 \sum_{j=1}^{N} a_{ij} \Gamma_1 x_j(t - \tau_1) \\ + \varepsilon_2 \sum_{j=1}^{N} a_{ij} \Gamma_2 x_j(t - \tau_2), \qquad i = 1, 2, \ldots, N, \end{aligned} \tag{9}$$

Theorem 1. *Suppose that $f_k(x(t))(k = 1, 2)$ is continuously differentiable at the synchronous state $s(t) \in R^n$, if the following N linear time varying delayed differential equations is exponentially stable about their zero solutions:*

$$\begin{aligned}\dot{\eta}_i(t) &= J_1(t)\eta_i(t) + J_2(t-\tau_0)\eta_i(t-\tau_0) \\ &\quad +\varepsilon_1\lambda_i\Gamma_1\eta(t-\tau_1) + \varepsilon_2\lambda_i\Gamma_2\eta(t-\tau_2), \quad i = 1, 2, \ldots, N. \end{aligned} \tag{10}$$

where $J_1(t) = Df_1(s(t))$ and $J_2(t-\tau_0) = Df_2(s(t-\tau_0))$ are the Jacobians of $f_1(x)$ at $s(t)$ and $f_2(x)$ at $s(t-\tau_0)$ respectively, then the network (9) achieves exponential synchronization.

Proof. Let

$$x_i(t) = \xi_i(t) + s(t), \tag{11}$$

then substitute (8) and (11) to (9), and linearize (9) about its synchronous state $S(t)$, then one has

$$\begin{aligned}\dot{\xi}_i(t) &= Df_1(s(t))\xi_i(t) + Df_2(s(t-\tau_0))\xi_i(t-\tau_0) + \varepsilon_1\sum_{j=1}^{N} a_{ij}\Gamma_1\xi_j(t-\tau_1) \\ &\quad +\varepsilon_2\sum_{j=1}^{N} a_{ij}\Gamma_2\xi_j(t-\tau_2), \qquad i = 1, 2, \ldots, N, \end{aligned} \tag{12}$$

we can write (12) in matrix form

$$\begin{aligned}\dot{\xi}(t) &= Df_1(s(t))\xi(t) + Df_2(s(t-\tau_0))\xi(t-\tau_0) + \varepsilon_1\Gamma_1\xi(t-\tau_1)A^T \\ &\quad +\varepsilon_2\Gamma_2\xi(t-\tau_2)A^T, \end{aligned} \tag{13}$$

Let $\eta(t) = \xi(t)U$, then we have

$$\begin{aligned}\dot{\eta}(t) &= Df_1(s(t))\eta(t) + Df_2(s(t-\tau_0))\eta(t-\tau_0) + \varepsilon_1\Gamma_1\eta(t-\tau_1)\Lambda \\ &\quad +\varepsilon_2\Gamma_2\eta(t-\tau_2)\Lambda, \end{aligned} \tag{14}$$

when $1 \le i \le N$, it hodes

$$\begin{aligned}\dot{\eta}_i(t) &= J_1(t)\eta_i(t) + J_2(t-\tau_0)\eta_i(t-\tau_0) + \varepsilon_1\lambda_i\Gamma_1\eta(t-\tau_1) \\ &\quad +\varepsilon_2\lambda_i\Gamma_2\eta(t-\tau_2), \qquad i = 1, 2, \ldots, N. \end{aligned} \tag{15}$$

Therefore, the stability of these n-dimentional linear time-varying delayed systems about their zero solutions is equivalent to the lolally exponential synchronization of the network (9). Thus, the proof is completed.

Theorem 2. *Suppose that $f_k(x(t))(k = 1, 2)$ is continuously differentiable at the synchronous state $s(t) \in R^n$, if there exist four $n \times n$ real symmetric matrixes $P > 0$, $R_1 > 0$, $R_2 > 0$, $R_2 > 0$, such that*

$$\begin{aligned}&P[J_1(t)^T + J_1(t) + J_2(t-\tau_0)R_1^{-1}J_2(t-\tau_0)^T + \varepsilon_1^2\lambda_i^2\Gamma_1R_2^{-1}\Gamma_1^T \\ &+\varepsilon_2^2\lambda_i^2\Gamma_2R_3^{-1}\Gamma_2^T]P + R_1 + R_2 + R_3 \le -cI_n, \qquad i = 1, 2, \ldots, N, \end{aligned} \tag{16}$$

and

$$\lambda_{min}(P) > \sum_{i=1}^{3} \tau_{i-1}\lambda_{max}(R_i),$$

where c is a positive number, $J_1(t) = Df_1(s(t))$ and $J_2(t) = Df_2(s(t-\tau_0))$ are the Jacobians of $f_1(x)$ at $s(t)$ and $f_2(x)$ at $s(t-\tau_0)$, then the state $s(t) \in R^n$ of (1) is ***delay-dependent*** *exponentially stable.*

Proof. Define a Lyapunov-Krasovskii function for the ith individual time-varying system as follow:

$$V_i(t) = \eta_i^T(t)P\eta_i(t) + \int_{t-\tau_0}^{t} \eta_i(s)R_1\eta_i(s)ds + \int_{t-\tau_1}^{t} \eta_i(s)R_2\eta_i(s)ds + \int_{t-\tau_2}^{t} \eta_i(s)R_3\eta_i(s)ds, \tag{17}$$

where $\eta_i(t) \in R^n$. By Lemma 4, the derivative of V along the trajectory of the ith individual time-varying delayed system in (10) is

$$\begin{aligned}
\dot{V}_i(t) &= \eta_i^T(t)J_1(t)^T P\eta_i(t) + \eta_i^T(t-\tau_0)J_2(t-\tau_0)^T P\eta_i(t) \\
&\quad +\varepsilon_1\lambda_i\eta^T(t-\tau_1)\Gamma_1^T P\eta_i(t) + \varepsilon_2\lambda_i\eta^T(t-\tau_2)\Gamma_2^T P\eta_i(t) \\
&\quad +\eta_i^T(t)PJ_1(t)\eta_i(t) + \eta_i^T(t)PJ_2(t-\tau_0)\eta_i(t-\tau_0) \\
&\quad +\varepsilon_1\lambda_i\eta_i^T(t)P\Gamma_1\eta(t-\tau_1) + \varepsilon_2\lambda_i\eta_i^T(t)P\Gamma_2\eta(t-\tau_2) + \eta_i^T(t)R_1\eta_i(t) \\
&\quad -\eta_i^T(t-\tau_0)R_1\eta_i(t-\tau_0) + \eta_i^T(t)R_2\eta_i(t) - \eta_i^T(t-\tau_1)R_2\eta_i(t-\tau_1) \\
&\quad +\eta_i^T(t)R_3\eta_i(t) - \eta_i^T(t-\tau_2)R_3\eta_i(t-\tau_2) \\
&= \eta_i^T(t)(PJ_1(t) + J_1(t)^T P + R_1 + R_2 + R_3)\eta_i(t) \\
&\quad +2\eta_i^T(t-\tau_0)J_2(t-\tau_0)^T P\eta_i(t) + 2\varepsilon_1\lambda_i\eta^T(t-\tau_1)\Gamma_1^T P\eta_i(t) \\
&\quad +2\varepsilon_2\lambda_i\eta_i^T(t)P\Gamma_2\eta(t-\tau_2) - \eta_i^T(t-\tau_0)R_1\eta_i(t-\tau_0) \\
&\quad -\eta_i^T(t-\tau_1)R_2\eta_i(t-\tau_1) - \eta_i^T(t-\tau_2)R_3\eta_i(t-\tau_2) \\
&\le [P(J_1(t)^T + J_1(t) + J_2(t-\tau_0)R_1^{-1}J_2(t-\tau_0)^T + \varepsilon_1^2\lambda_i^2\Gamma_1R_2^{-1}\Gamma_1^T \\
&\quad +\varepsilon_2^2\lambda_i^2\Gamma_2R_3^{-1}\Gamma_2^T)P + R_1 + R_2 + R_3]\|\eta_i(t)\|^2 \le -c\|\eta_i(t)\|^2.
\end{aligned} \tag{18}$$

From (18), by Lemma 3, we can further get

$$\dot{V}_i(t) \le \frac{-c}{\lambda_{max}(P)}V_i(t) + \frac{c(\sum_{i=1}^{3}\tau_{i-1}\lambda_{max}(R_i))}{\lambda_{min}(P)\lambda_{max}(P)} \sup_{t-max(\tau_0,\tau_1,\tau_2)} V_i(\alpha). \tag{19}$$

Then, by Lemma 2, when

$$\lambda_{min}(P) > \sum_{i=1}^{3} \tau_{i-1}\lambda_{max}(R_i),$$

the ith individual delayed time-varying system is exponetially stable about its zero solution. By theorem 1, the network (9) is exponentially synchronizous about the state $s(t) \in R^n$. The proof is completed.

In particular, if $P = R_1 = R_2 = R_3 = I_n$, Theorem 2 reduced to the following simple and useful corollary.

Corollary 1. *If the following N conditions*

$$J_1(t)^T + J_1(t) + J_2(t)J_2(t)^T + \varepsilon_1^2\lambda_i^2\Gamma_1\Gamma_1^T + \varepsilon_2^2\lambda_i^2\Gamma_2\Gamma_2^T \leq -(c+3)I_n, \qquad i = 1, 2, \ldots, N \tag{20}$$

and

$$\sum_{i=1}^{3} \tau_{i-1} < 1 \tag{21}$$

are satisfied, then the network (9) is ***delay-dependent*** *exponentially stable at the state* $s(t) \in R^n$.

3.2 Globally Asymptotical Synchronization

Consider the network (1), suppose $s(t)$ is a solution of an individual nodes of (1). Let

$$x_i(t) = \omega_i(t) + s(t), \tag{22}$$

then substituting (22) into (1), and from (8), one has

$$\begin{aligned}\dot{\omega}_i(t) = {} & f_1(\omega_i(t) + s(t)) - f_1(s(t)) + f_2(\omega_i(t-\tau_0) + s(t-\tau_0)) - f_2(s(t-\tau_0)) \\ & + \varepsilon_1 \sum_{j=1}^{N} a_{(1)ij}\Gamma_1\omega_j(t-\tau_1) + \varepsilon_2 \sum_{j=1}^{N} a_{(2)ij}\Gamma_2\omega_j(t-\tau_2), \\ & \qquad i = 1, 2, \ldots, N.\end{aligned} \tag{23}$$

Let

$$\begin{aligned} g_1(\omega_i(t)) &= f_1(\omega_i(t) + s(t)) - f_1(s(t)), \\ g_2(\omega_i(t-\tau_0)) &= f_2(\omega_i(t-\tau_0) + s(t-\tau_0)) - f_2(s(t-\tau_0)), \end{aligned} \tag{24}$$

since $f_k(x(t))(k = 1, 2)$ is DLC, it's easy to obtain

$$\beta\|x\| \leq \|g_k(x)\| \leq \alpha\|x\|, \qquad k = 1, 2. \tag{25}$$

Substituting (24) into network (23), we get the following

$$\begin{aligned}\dot{\omega}_i(t) = {} & g_1(\omega_i(t)) + g_2(\omega_i(t-\tau_0)) + \varepsilon_1 \sum_{j=1}^{N} a_{(1)ij}\Gamma_1\omega_j(t-\tau_1) \\ & + \varepsilon_2 \sum_{j=1}^{N} a_{(2)ij}\Gamma_2\omega_j(t-\tau_2), \qquad i = 1, 2, \ldots, N.\end{aligned} \tag{26}$$

By Kronecker product, we can further rewrite system (1) in a compact form as

$$\begin{aligned}\dot{\omega}(t) = g_1(\omega(t)) + g_2(\omega(t-\tau_0) + \varepsilon_1(A_1 \otimes \Gamma_1)\omega(t-\tau_1) \\ +\varepsilon_2(A_2 \otimes \Gamma_2)\omega(t-\tau_2), \qquad i = 1, 2, \ldots, N,\end{aligned} \tag{27}$$

where $\omega(t) = (\omega_1(t), \ldots, \omega_N(t))^T \in R^{nN}$, $\omega(t-\tau_0) = (\omega_1(t-\tau_0), \ldots, \omega_N(t-\tau_0))^T \in R^{nN}$, $g_1(\omega(t-\tau_1)) = (g_1(\omega_1(t-\tau_1)), \ldots, g_1(\omega_N(t-\tau_1)))^T \in R^{nN}$, $g_2(\omega(t-\tau_2)) = (g_2(\omega_1(t-\tau_2)), \ldots, g_2(\omega_N(t-\tau_2)))^T \in R^{nN}$, It is convienient to use (27) than (26) sometimes.

Theorem 3. *Suppose that $f_k(x(t))(k = 1, 2)$ is DLC(Definition 2), if there exists a diag matrix $P > 0$, three real symmetric matrixes $R_1 > 0$, $R_2 > 0$, $R_3 > 0$, such that*

$$\lambda_{max}(\Theta) < -\frac{\lambda_{max}(R_1)\alpha^2 + \lambda_{max}(R_2 + R_3)}{\beta^2},$$

where $R_k \in R^{nN \times nN}(k = 1, 2, 3)$, $P = diag\{p_1, \ldots, p_N\} \in R^N$,

$$\Theta = \Lambda_P(2\Lambda_P^{-1} + R_1^{-1} + \varepsilon_1^2 \Sigma_1 R_2^{-1} \Sigma_1^T + \varepsilon_2^2 \Sigma_2 R_3^{-1} \Sigma_2^T)\Lambda_P, \tag{28}$$

$\Lambda_P = P \otimes I_n$, $\Sigma_k = A_k \otimes \Gamma_k(k = 1, 2)$, Then the state $s(t)$ of the network (1) is asymptotically synchronized.

Proof. Define a Lyapunov-Krasovskii function as follows:

$$V(t) = V_1(t) + V_2(t) + V_3(t) + V_4(t),$$

where

$$V_1(t) = 2\sum_{i=1}^{N} p_i \int_0^{\omega_i(t)} g_1^T(s)ds, \; V_2(t) = \int_{t-\tau_0}^{t} g_2^T(\omega(s))R_1 g_2(\omega(s))ds,$$
$$V_3(t) = \int_{t-\tau_1}^{t} \omega(s)R_2\omega(s)ds, \qquad V_4(t) = \int_{t-\tau_2}^{t} \omega(s)R_3\omega(s)ds,$$

$p_i > 0(i = 1, \ldots, nN)$, $R_1 > 0$, $R_2 > 0$, $R_3 > 0$, are all real symmetrical positive definite matrixes. The time derivative of V along the trajectory of (27) is given by

$$\dot{V}(t) = \dot{V}_1(t) + \dot{V}_2(t) + \dot{V}_3(t) + \dot{V}_4(t). \tag{29}$$

Let $P = diag(p_1, \ldots, p_{nN})$, by Lemma 4, we have

$$\begin{aligned}
\dot{V}_1(t) &= 2\sum_{i=1}^{N} p_i g_1^T(\omega_i(t))\dot{\omega}_i(t) = 2g_1^T(\omega(t))(P \otimes I_n)\dot{\omega}(t) \\
&= 2g_1^T(\omega(t))(P \otimes I_n)g_1(\omega(t)) + 2g_1^T(\omega(t))(P \otimes I_n)g_2(\omega(t-\tau_0)) \\
&\quad +2g_1^T(\omega(t))(P \otimes I_n)\varepsilon_1(A_1 \otimes \Gamma_1)\omega(t-\tau_1) \\
&\quad +2g_1^T(\omega(t))(P \otimes I_n)\varepsilon_2(A_2 \otimes \Gamma_2)\omega(t-\tau_2) \\
&\leq 2g_1^T(\omega(t))(P \otimes I_n)g_1(\omega(t)) + g_2^T(\omega(t-\tau_0))R_1 g_2(\omega(t-\tau_0)) \\
&\quad +g_1^T(\omega(t))(P \otimes I_n)R_1^{-1}(P \otimes I_n)g_1(\omega(t)) + \omega^T(t-\tau_1)R_2\omega(t-\tau_1) \\
&\quad +g_1^T(\omega(t))\varepsilon_1^2(P \otimes I_n)(A_1 \otimes \Gamma_1)R_2^{-1}(A_1 \otimes \Gamma_1)^T(P \otimes I_n)g_1(\omega(t)) \\
&\quad +\omega^T(t-\tau_2)R_3\omega(t-\tau_2) \\
&\quad +g_1^T(\omega(t))\varepsilon_2^2(P \otimes I_n)(A_2 \otimes \Gamma_2)R_3^{-1}(A_2 \otimes \Gamma_2)^T(P \otimes I_n)g_1(\omega(t)), \quad (30) \\
\dot{V}_2(t) &= g_2^T(\omega(t))R_1 g_2(\omega(t)) - g_2^T(\omega(t-\tau_0))R_1 g_2(\omega(t-\tau_0)), \quad (31) \\
\dot{V}_3(t) &= \omega(t)^T R_2\omega(t) - \omega^T(t-\tau_1)R_2\omega(t-\tau_1), \quad (32) \\
\dot{V}_4(t) &= \omega(t)^T R_3\omega(t) - \omega^T(t-\tau_2)R_3\omega(t-\tau_2). \quad (33)
\end{aligned}$$

Let $\Lambda_P = P \otimes I_n$, and $\Sigma_k = A_k \otimes \Gamma_k (k = 1, 2)$, from (30) (31) (32) (33), we can easily get

$$\begin{aligned}
\dot{V}(t) &\leq g_1^T(\omega)(\Lambda_P(2\Lambda_P^{-1} + R_1^{-1} + \varepsilon_1^2\Sigma_1 R_2^{-1}\Sigma_1^T + \varepsilon_2^2\Sigma_2 R_3^{-1}\Sigma_2^T)\Lambda_P)g_1(\omega) \\
&\quad +g_2^T(\omega(t))R_1 g_2(\omega(t)) + \omega(t)^T(R_2 + R_3)\omega(t). \quad (34)
\end{aligned}$$

By Lemma 3, from equation (28) and inequality (25), we can further obtain

$$\dot{V}(t) \leq (\lambda_{max}(\Theta)\beta^2 + \lambda_{max}(R_1)\alpha^2 + \lambda_{max}(R_2 + R_3))\|\omega(t)\|^2. \quad (35)$$

Since

$$\lambda_{max}(\Theta) < -\frac{\lambda_{max}(R_1)\alpha^2 + \lambda_{max}(R_2 + R_3)}{\beta^2},$$

then $\dot{V}(t) < 0$, by Lyapunove stability theorem, the state $s(t)$ of the network (1) is asymptotically stable. The proof is thus completed.

4 Examples

Consider a time-delayed multi-linked network consisting of 4 nodes, with each node is a 3-dimensional nonlinear system. The individual system is described as follow

$$\begin{cases}
\dot{x}_1 = -5x_1 + x_2^2 + x_1(t-\tau_0) \\
\dot{x}_2 = -6x_2 + x_2(t-\tau_0) \\
\dot{x}_3 = -7x_3 + x_1^2 + x_3(t-\tau_0).
\end{cases} \quad (36)$$

Obviously, this system is asymptotically stable at $s(t) = 0$, and the Jacobian at $s(t) = 0$ are

$$\mathrm{J}_1 = \begin{bmatrix} -5 & 0 & 0 \\ 0 & -6 & 0 \\ 0 & 0 & -7 \end{bmatrix}, \quad \mathrm{J}_2 = \begin{bmatrix} 1\,0\,0 \\ 0\,1\,0 \\ 0\,0\,1 \end{bmatrix}.$$

For simplicity, in (9), let the outer-coupling configuration matrix

$$\mathrm{A} = \begin{bmatrix} -3 & 1 & 1 & 1 \\ 1 & -3 & 1 & 1 \\ 1 & 1 & 1 & -3 \\ 1 & 1 & 1 & -3 \end{bmatrix},$$

and $\tau_0 = 0.1$, $\tau_1 = 0.2$, $\tau_2 = 0.3$, $\varepsilon_1 = 0.001$, $\varepsilon_2 = 0.002$, $\Gamma_1 = \Gamma_2 = I_3$, the eigenvalues of A are $\lambda_1 = 0$, $\lambda_2 = \lambda_3 = \lambda_4 = -4$. It is easy to verify that the two inequality conditions of corollary 2 are both satisfied. Fig. 1 shows that this network realizes synchronization.

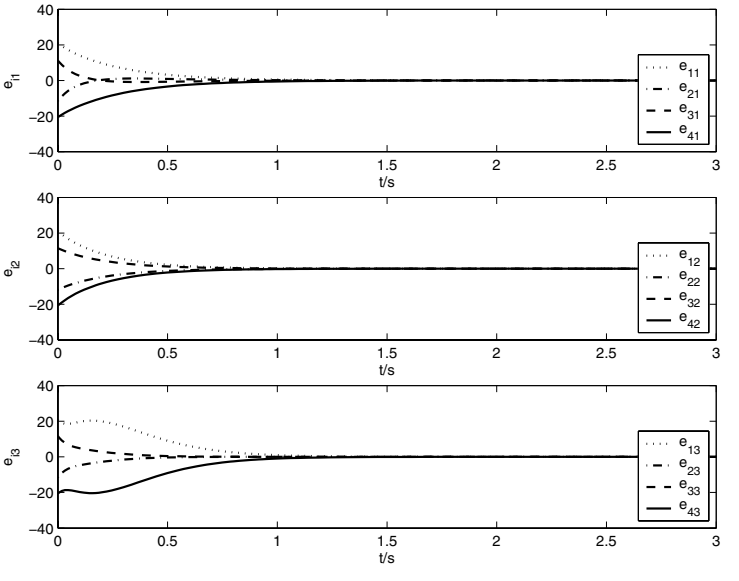


Fig. 1. Exponential synchronization of network (11)

5 Conclusion

Since delays and different kinds of links exist commonly in networks in real world. It is important to research networks with delays and multi-links. In this paper, we introduce a new more general multi-linked complex dynamical networks with both delayed nodes and delayed couplings and then investigate its phenomena. Via constructing Lyapunov-Krasovskii function and using some inequality

technique, we have obtained criteria for locally exponential synchronization and globally asymptotical synchronization. The latter does not require the outer-coupling configuration matrixes to be symmetric and irreducible. Meanwhile, it is obtained by Kronecker product method which is different from other papers' method–linearizing the network at its synchronization state. So the results obtained in this paper are more general.

Acknowledgments. Supported by the National Natural Science Foundation of China (60740430664), the Scientific Innovation Team Project of Hubei Provincial Department of Education (T200809) and partially supported by the Scientific Research Projects of Hubei Provincial Department of Education (Q200713001).

References

1. Pecora, L.M., Carroll, T.L.: Synchronization of Chaotic Systems. phys. Rev. Lett. 64, 821–830 (1990)
2. Wang, X., Chen, G.: Synchronization in Scale-free Dynamical Networks: Robustness and Fragility. IEEE Trans. Circuits Syst. I. 49, 54–62 (2002)
3. Wang, X., Chen, G.: Synchronization in Small-world Dynamical Networks. Int. J. Bifurcat. Chaos. 12, 187–192 (2002)
4. Wang, X., Chen, G.: Pinning Control of Scale-free Dynamical Networks. Phys. A. 310, 521–531 (2002)
5. Yang, H., Wang, F., Zhang, H., Zong, G.: Chaos Synchronization of General Complex Dynamical Networks with Coupling Delays. In: Proceedings of the 6th World Congress on Intelligent Control and Auto, Dalian (China), pp. 21–23 (June 2006)
6. Liu, T., Dimiroviski, G., Zhao, J.: Exponential Synchronization of Complex Delayed Dynamical Networks with General Topilogy. Phys. A. 387, 643–652 (2008)
7. Wu, J., Jiao, L.: Synchronization in Complex Delayed Networks with Nonsymmetric Coupling. Phys. A. 386, 513–530 (2007)
8. Li, P., Zhang, Y.: Synchronization Analysis of Delayed Complex Networks with Time-varying Couplings. Phys. A. 387, 3729–3737 (2008)
9. Wu, X.: Synchronization-based Topology Identification of Weighted General Complex Dynamical Networks with Time-varying Coupling Delay. Phys. A. 387, 997–1008 (2008)
10. Dai, Y., Cai, Y., Xu, X.: Synchronization Criteria for Complex Dynamical Networks with Neutral-type Coupling Delay. Phys A 387, 4673–4682 (2008)
11. Zhang, Q., Lu, J., Lü, J.: Adaptive Feedback Synchronization of a General Complex Dynamical Network with Delayed Nodes. IEEE Trans. Circuits Syst. 55, 183–187 (2008)
12. Li, Z., Lee, J.: Global Synchronization of General Complex Dynamical Networks with Time-varying Delay. In: IEEE International Conference on Robot and Human Interavtive Communication, Jeju (Korea), pp. 26–29 (August 2007)
13. Cao, J., Wang, J.: Global Asymptotic and Robust Stability of Recurrent Neural Networks with Time Delays. IEEE Trans. Circuits and Systems I. 52, 417–426 (2005)

Global Exponential Stability of Impulsive Fuzzy Cellular Neural Networks with Delays and Reaction-Diffusion Terms

Xiaobo Li[1,2] and Haijun Jiang[1]

[1] College of Mathematics and System Sciences, Xinjiang University, Urumqi 830046, P. R. China
[2] Normal College, Shihezi University, Shihezi 832000, China
jianghai@xju.edu.cn, lixiaobo1974@163.com

Abstract. In this paper, we investigate a class of impulsive fuzzy cellular neural networks with delays and reaction-diffusion terms. By applying Lyapunov functional method and using inequality techniques, some sufficient conditions ensuring the global exponential stability of equilibrium point for impulsive fuzzy cellular neural networks with delays and reaction-diffusion terms are established. An illustrative example is given in the end to guarantee the validity of our theory.

Keywords: Cellular neural networks, Equilibrium point, Exponential stability, Reaction-diffusion, Lyapunov functional.

1 Introduction

Since cellular neural networks(CNNs) was firstly proposed by Chua and Yang [1], many researchers have done extensive work on this subject due to their extensive applications in many fields such as pattern recognize, signal and image processing. Another group of fundamental neural networks, fuzzy cellular neural networks(FCNNs), introduced by Yang et al. [2], integrate fuzzy logical into structure of traditional cellular neural networks and maintain local connectedness. Study have shown the potential of FCNNs in image processing and pattern recognize. Such applications heavily depend on the dynamical behaviors. Thus, the analysis of the dynamical behaviors such as stability, especially exponential stability, is a necessary step for practical design of FCNNs. Some result have be reported on dynamical behaviors of FCNNs [3-6]. In the factual operations, however, the diffusion phenomena could not be ignored in neural networks and electric circuits once electrons transport in a nonuniform electromagnetic field. Hence, it is essential to consider the state variables varying with the time and space variables. In [7, 8], the authors have obtain some sufficient conditions for the global exponential stability of the equilibrium point of FCNNs with reaction-diffusion terms.

On the other hand, in the real world, many evolutionary processes are not continuous and often be changed abruptly at certain moment due to man-made control or environmental perturbation, which called impulsive phenomena. We need

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 570–578, 2009.

to consider the effect of impulsion on FCNNs too. To the best of our knowledge, few authors have consider the global exponential stability of impulsive fuzzy cellular neural networks with delays and reaction-diffusion terms. Motivated by the above discussion, the aim of the present paper is to consider a class of impulsive fuzzy cellular neural networks with delays and reaction-diffusion terms, which is more general. Under the hypothesis that the system have unique equilibrium point, by utilizing the method of Lyapunov functional, we shall establish some sufficient conditions for the global exponential stability of the equilibrium point of such neural networks.

2 Model Description and Preliminaries

Firstly, in order to simplify our description, we introduce some notations as follows. Let $\Omega \subset R^m$ is a bounded open set with smooth boundary $\partial\Omega$ and $mes\Omega > 0$, $PC[J \times \Omega, R^n] \triangleq \{u(t,x) : J \times \Omega \to R^n \mid u(t,x)$ is continuous at $t \neq t_k$, $u(t_k^+, x) = u(t_k, x)$ and $u(t_k^-, x)$ exists for $t_k \in J \quad k \in N\}$, $PC[J, R^n] \triangleq \{u(t) : J \to R^n \mid u(t)$ is continuous at $t \neq t_k$, $u(t_k^+) = u(t_k)$ and $u(t_k^-)$ exists for $t_k \in J \quad k \in N\}$, where $J \subset R$ is a interval.

Furthermore, let $PC(\Omega) \triangleq \{\phi(s,x) : [-\tau, 0] \times \Omega \to R^n \mid \phi(s^+, x) = \phi(s,x)$ for $s \in [-\tau, 0)$; $\phi(s^-, x)$ exists for $s \in (-\tau, 0]$, $\phi(s^-, x) = \phi(s,x)$ for all but finite number of points $s \in [-\tau, 0]$ $\}$, $PC \triangleq \{\phi(s) : [-\tau, 0] \to R^n \mid \phi(s^+) = \phi(s)$ for $s \in [-\tau, 0)$; $\phi(s^-)$ exists for $s \in (-\tau, 0]$, $\phi(s^-) = \phi(s)$ for all but finite number of points $s \in [-\tau, 0]\}$.

For $u(t,x) = (u_1(t,x), u_2(t,x), \cdots, u_n(t,x))^T \in R^n$, we define $\|u(t,x)\|_2 = \sqrt{\sum_{i=1}^n \|u_i(t,x)\|_2^2}$, where $\|u_i(t,x)\|_2 = (\int_\Omega |u_i(t,x)|^2 dx)^{\frac{1}{2}}$, $i = 1, 2, \cdots, n$. For $\phi(s,x) = (\phi_1(s,x), \phi_2(s,x), \cdots, \phi_n(s,x))^T \in PC(\Omega)$, we define $\|\phi\|_2 = (\sum_{i=1}^n \|\phi_i\|_2^2)^{\frac{1}{2}}$, where $\|\phi_i\|_2 = (\int_\Omega |\phi_i(s,x)|_\tau^2 dx)^{\frac{1}{2}}$, $|\phi_i(s,x)|_\tau = \sup_{-\tau < s \leq 0} |\phi_i(s,x)|$. Then $PC(\Omega)$ is a Banach space.

For any $A, B \in R^{m \times n}$, we define Schur product by $A \otimes B = (a_{ij} b_{ij})_{m \times n}$. Denote $|A| = (|a_{ij}|)_{m \times n}$, and $A \geq B$ $(A > B)$ if and only if $a_{ij} \geq b_{ij}$ $(a_{ij} > b_{ij})$.

We consider the following system

$$\begin{cases} \frac{\partial u_i(t,x)}{\partial t} &= \sum\limits_{l=1}^m \frac{\partial}{\partial x_l}(D_{il}(\frac{\partial u_i(t,x)}{\partial x_l})) - \alpha_i(u_i(t,x))[\beta_i(u_i(t,x)) \\ &\quad - \sum\limits_{j=1}^n a_{ij} f_j(u_j(t,x)) - \sum\limits_{j=1}^n b_{ij}\mu_j - I_i - \bigwedge\limits_{j=1}^n \alpha_{ij} f_j(u_j(t-\tau_{ij}(t),x)) \\ &\quad - \bigvee\limits_{j=1}^n \beta_{ij} f_j(u_j(t-\tau_{ij}(t),x)) - \bigwedge\limits_{j=1}^n T_{ij}\mu_j - \bigvee\limits_{j=1}^n H_{ij}\mu_j], t \neq t_k \\ u_i(t^+, x) &= u_i(t^-, x) + I_{ik}(u_i(t^-, x)), t = t_k \\ \frac{\partial u_i(t,x)}{\partial n} &: = (\frac{\partial u_i(t,x)}{\partial x_1}, \frac{\partial u_i(t,x)}{\partial x_2}, \cdots, \frac{\partial u_i(t,x)}{\partial x_m})^T = 0, (t,x) \in [-\tau, \infty) \times \partial\Omega \\ u_i(s,x) &= \phi_i(s,x), (s,x) \in [-\tau, 0] \times \Omega, \phi_i(s,x) \in PC(\Omega) \\ &\quad i = 1, 2, \cdots, n \end{cases} \tag{1}$$

where n is the number of neurons in the networks, $x = (x_1, x_2, \cdots, x_m)^T \in \Omega \subset R^m$, $u_i(t,x)$ corresponds to the states of the ith neural unit at time t in

space x ; $D_{il} = D_{il}(t,x) \geq 0$ corresponds to the transmission diffusion operator alone the ith neuron; $\alpha_i(u_i(t,x))$ represents an amplification function at time t; $\beta_i(u_i(t,x))$ is an appropriately behaved function at time t; $f_j(u_j(t,x))$ denote the activation function of the jth neurons at time t and in space x; α_{ij}, β_{ij} are elements of fuzzy feedback MIN template and fuzzy feedback MAX template respectively; T_{ij}, H_{ij} are elements of fuzzy feedforward MIN template and fuzzy feedforward MAX template, respectively; a_{ij}, b_{ij} are elements of feedback template and feedforward template respectively; $\bigvee$ and $\bigwedge$ denote the fuzzy AND and fuzzy OR operation, respectively; $\tau_{ij}(t)$ correspond to transmission delays along the axon of jth unit from the i th unit and satisfy $0 \leq \tau_{ij}(t) \leq \tau_{ij} \leq \tau$, μ_i and I_i denote input and bias of ith unit respectively; t_k satisfy $0 < t_1 < t_2 < \cdots$ and $\lim_{k\to+\infty} t_k = +\infty$.

Throughout this paper, we introduce the following assumptions.

(H_1). Each function $\alpha_i(u)$ is a positive bounded functions, i.e. there exist constants $\underline{\alpha_i} > 0$, $\overline{\alpha_i} > 0$, such that

$$0 < \underline{\alpha_i} \leq \alpha_i(u) \leq \overline{\alpha_i} < +\infty \quad \text{for all } u \in R,\ i = 1, 2, \cdots, n.$$

(H_2). There exist constants $\beta_i > 0$, such that $\frac{\beta_i(u)-\beta_i(v)}{u-v} > \beta_i$ for all $u, v \in R, i = 1, 2, \cdots, n$.

(H_3).There exist positive constants $F_i > 0, i = 1, 2, \cdots, n,$ such that

$$\frac{|f_i(u) - f_i(v)|}{|u - v|} < F_i, \quad \text{for all } u, v \in R \text{ and } u \neq v.$$

(H_4). Let $h_{ik}(u_i) = u_i + I_{ik}(u_i)$ be Lipschitz continuous in R^n, that is, there exist constants $\gamma_{ik} > 0,$ such that

$$|h_{ik}(u_i) - h_{ik}(v_i)| \leq \gamma_{ik}|u_i - v_i|, \quad i = 1, 2, \cdots, n; k = 1, 2, \cdots.$$

Definition 1. *The equilibrium point* $u^* = (u_1^*, u_2^*, \cdots, u_n^*)^T$ *of system (1) is side to be globally exponentially stable, if there exist constants* $\varepsilon > 0$ *and* $M \geq 1$ *such that*

$$\|u(t,x) - u^*\|_2 \leq M\|\phi - u^*\|_2 e^{-\varepsilon t}$$

for all $t \geq 0$. *where* $u(t,x)$ *is a solution of system (1) with initial value* $\phi(s,x) \in PC(\Omega)$.

Lemma 1. [3] *For any* $a_{ij} \in R, x_j, y_j \in R, i, j = 1, 2, \cdots, n,$ *we have the following estimation.*

$$|\bigwedge_{j=1}^{n} \alpha_{ij}x_j - \bigwedge_{j=1}^{n} \alpha_{ij}y_j| \leq \sum_{j=1}^{n}(|a_{ij}| \cdot |x_j - y_j|)$$

$$|\bigvee_{j=1}^{n} \alpha_{ij}x_j - \bigvee_{j=1}^{n} \alpha_{ij}y_j| \leq \sum_{j=1}^{n}(|a_{ij}| \cdot |x_j - y_j|).$$

Lemma 2. [9] *Let* $\tau > 0, a < b \leq +\infty$. *Suppose that* $v(t) = (v_1(t), v_2(t), \cdots, v_n(t))^T \in C[[a,b), R^n]$ *satisfies the following differential inequality*

$$\begin{aligned} D^+v(t) &\leq Pv(t) + (Q \otimes V(t))e_n \qquad t \in [a,b) \\ v(a+s) &\in PC \qquad\qquad\qquad\qquad s \in [-\tau, 0] \end{aligned}$$

where $P = (p_{ij})_{n\times n}$, $p_{ij} \geq 0 \quad i \neq j$, $Q = (q_{ij})_{n\times n} \geq 0$, $V(t) = (v_j(t - \tau_{ij}(t)))_{n\times n}$, $e_n = (1, 1, \cdots, 1)^T \in R^n$. *If the initial condition satisfies* $v(t) \leq K\xi e^{-\lambda(t-a)} \quad K \geq 0, \quad t \in [a-\tau, a]$, *where* $\xi = (\xi_1, \xi_2, \cdots, \xi_n)^T > 0$ *and constants* $\lambda > 0$ *is determined by the following inequality*

$$[\lambda E + P + Q \otimes \varepsilon(\lambda)]\xi < 0$$

where $\varepsilon(\lambda) = (e^{\lambda\tau_{ij}})_{n\times n}$. *Then* $v(t) \leq K\xi e^{-\lambda(t-a)}$, *for* $t \in [a,b)$.

3 Global Exponential Stability of Equilibrium Point

We suppose that system (1) have unique equilibrium point $u^*=(u_1^*, u_2^*, \cdots, u_n^*)^T$, then we have the following result.

Theorem 1. *Under assumptions* (H_1)-(H_4) , *if the following conditions hold:*

(A_1). *There exist a vector* $\xi = (\xi_1, \xi_2, \cdots, \xi_n)^T > 0$ *and a positive number* $\lambda > 0$ *such that*

$$[\lambda E - \underline{\alpha}\beta + \overline{\alpha}|A|F + \overline{\alpha}(|\overline{A}| + |\overline{B}|)F \otimes \varepsilon(\lambda)]\xi < 0,$$

where $\underline{\alpha} = diag(\underline{\alpha_1}, \cdots, \underline{\alpha_n})$, $\overline{\alpha} = diag(\overline{\alpha_1}, \cdots, \overline{\alpha_n})$, $A = (a_{ij})_{n\times n}$, $\overline{A} = (\alpha_{ij})_{n\times n}$, $\overline{B} = (\beta_{ij})_{n\times n}$, $F = diag(F_1, F_2, \cdots, F_n)$, $\beta = diag(\beta_1, \beta_2, \cdots, \beta_n)$, $\varepsilon(\lambda) = (e^{\lambda\tau_{ij}})_{n\times n}$,

(A_2). $\eta = \sup_{k\in N}\{\frac{\ln \eta_k}{t_k - t_{k-1}}\} < \lambda, \quad \eta_k = \max_{1\leq i\leq n}\{1, \gamma_{ik}\}, \quad k \in N,$

then the equilibrium point u^* *is globally exponentially stable, and its exponential convergence rate equals* $\lambda - \eta$.

Proof. Let $w_i(t,x) = u_i(t,x) - u^* \quad i = 1, 2, \cdots, n$, where $u(t,x)$ is a solution of system (1) with initial value $\phi(s,x) \in PC(\Omega)$. Then system (1) can be transformed into

$$\begin{cases} \frac{\partial w_i(t,x)}{\partial t} = \sum\limits_{l=1}^{m} \frac{\partial}{\partial x_l}(D_{il}(\frac{\partial w_i(t,x)}{\partial x_l})) - \widetilde{\alpha}_i(w_i(t,x))[\widetilde{\beta}_i(w_i(t,x)) - \sum\limits_{j=1}^{n} a_{ij}\widetilde{f}_j(w_j(t,x)) \\ \qquad\qquad - \bigwedge\limits_{j=1}^{n} \alpha_{ij}\widetilde{f}_j(w_j(t-\tau_{ij}(t),x)) - \bigvee\limits_{j=1}^{n} \beta_{ij}\widetilde{f}_j(w_j(t-\tau_{ij}(t),x))] \quad t \neq t_k \\ w_i(t^+,x) = \widetilde{h}_{ik}(w_i(t^-,x)) \qquad t = t_k \\ \frac{\partial w_i(t,x)}{\partial n} := (\frac{\partial w_i(t,x)}{\partial x_1}, \frac{\partial w_i(t,x)}{\partial x_2}, \cdots, \frac{\partial w_i(t,x)}{\partial x_m})^T = 0, \qquad (t,x) \in [-\tau, \infty) \times \Omega \\ w_i(s,x) = \phi_i(s,x) \qquad (s,x) \in [-\tau, 0] \times \Omega \quad \phi_i(s,x) \in PC(\Omega) \\ \qquad i = 1, 2, \cdots, n \end{cases} \tag{2}$$

Where $\widetilde{\alpha}_i(w_i(t,x)) = \alpha_i(w_i(t,x)+u_i^*)$, $\widetilde{\beta}_i(w_i(t,x)) = \beta_i(w_i(t,x)+u_i^*) - \beta_i(u_i^*)$, $\widetilde{f}_j(w_j(t,x)) = f_j(w_j(t,x)+u_j^*) - f_j(u_j^*)$, $\widetilde{h}_{ik}(w_i(t,x)) = h_{ik}(w_i(t,x)+u_i^*) - h_{ik}(u_i^*)$.

For all $t \neq t_k$, we multiply both sides of equation by $w_i(t,x)$ and integrate it on Ω, then we have

$$\begin{aligned}\int_\Omega w_i(t,x)\frac{\partial w_i(t,x)}{\partial t}dx &= \int_\Omega w_i(t,x)\sum_{l=1}^m \frac{\partial}{\partial x_l}(D_{il}(\frac{\partial w_i(t,x)}{\partial x_l}))dx \\ &\quad - \int_\Omega w_i(t,x)\widetilde{\alpha}_i(w_i(t,x))\widetilde{\beta}_i(w_i(t,x))dx \\ &\quad + \sum_{j=1}^n a_{ij}\int_\Omega \widetilde{\alpha}_i(w_i(t,x))w_i(t,x)\widetilde{f}_j(w_j(t,x))dx \\ &\quad + \int_\Omega \widetilde{\alpha}_i(w_i(t,x))w_i(t,x)\bigwedge_{j=1}^n \alpha_{ij}\widetilde{f}_j(w_j(t-\tau_{ij}(t),x))dx \\ &\quad + \int_\Omega \widetilde{\alpha}_i(w_i(t,x))w_i(t,x)\bigvee_{j=1}^n \beta_{ij}\widetilde{f}_j(w_j(t-\tau_{ij}(t),x))dx.\end{aligned} \tag{3}$$

By the boundary conditions of system (2), we have

$$\begin{aligned}&\int_\Omega w_i(t,x)\sum_{l=1}^m \frac{\partial}{\partial x_l}(D_{il}(\frac{\partial w_i(t,x)}{\partial x_l}))dx \\ &= \int_\Omega \sum_{l=1}^m \frac{\partial}{\partial x_l}(D_{il}w_i(t,x)\frac{\partial w_i(t,x)}{\partial x_l})dx - \int_\Omega \sum_{l=1}^m D_{il}(\frac{\partial w_i(t,x)}{\partial x_l})^2dx \\ &= \int_{\partial\Omega}\sum_{l=1}^m (D_{il}w_i(t,x)\frac{\partial w_i(t,x)}{\partial x_l})\cos(\overrightarrow{n},x_l)dS - \int_\Omega \sum_{l=1}^m D_{il}(\frac{\partial w_i(t,x)}{\partial x_l})^2dx \\ &= -\int_\Omega \sum_{l=1}^m D_{il}(\frac{\partial w_i(t,x)}{\partial x_l})^2dx.\end{aligned} \tag{4}$$

From assumptions (H_1),(H_3) and Holder inequality, we have

$$\begin{aligned}&a_{ij}\int_\Omega w_i(t,x)\widetilde{\alpha}_i(w_i(t,x))\widetilde{f}_j(w_j(t,x))dx \\ &\leq |a_{ij}|\overline{\alpha_i}F_j\int_\Omega |w_i(t,x)||w_j(t,x)|dx \leq |a_{ij}|\overline{\alpha_i}F_j\|w_i(t,x)\|_2\cdot\|w_j(t,x)\|_2.\end{aligned} \tag{5}$$

Base on Lemma 1, we have

$$\begin{aligned}&\int_\Omega \widetilde{\alpha}_i(w_i(t,x))w_i(t,x)\bigwedge_{j=1}^n \alpha_{ij}\widetilde{f}_j(w_j(t-\tau_{ij}(t),x))dx \\ &\leq \int_\Omega \widetilde{\alpha}_i(w_i(t,x))|w_i(t,x)|\bigwedge_{j=1}^n \alpha_{ij}\widetilde{f}_j(w_j(t-\tau_{ij}(t),x))|dx\end{aligned}$$

$$\begin{aligned}
&\leq \overline{\alpha_i} \int_\Omega |w_i(t,x)| \sum_{j=1}^n |\alpha_{ij}||\widetilde{f_j}(w_j(t-\tau_{ij}(t),x))|dx \\
&\leq \overline{\alpha_i} \int_\Omega \sum_{j=1}^n |\alpha_{ij}|F_j|w_i(t,x)||w_j(t-\tau_{ij}(t),x)|dx \\
&\leq \overline{\alpha_i} \sum_{j=1}^n |\alpha_{ij}|F_j\|w_i(t,x)\|_2\|w_j(t-\tau_{ij}(t),x)\|_2.
\end{aligned} \tag{6}$$

By the same way, we obtain that

$$\begin{aligned}
&\int_\Omega \widetilde{\alpha_i}(w_i(t,x))w_i(t,x) \bigvee_{j=1}^n \beta_{ij}\widetilde{f_j}(w_j(t-\tau_{ij}(t),x))dx \\
&\leq \overline{\alpha_i} \sum_{j=1}^n |\beta_{ij}|F_j\|w_i(t,x)\|_2\|w_j(t-\tau_{ij}(t),x)\|_2.
\end{aligned} \tag{7}$$

Noting $\int_\Omega w_i(t,x)\frac{\partial w_i(t,x)}{\partial t}dx = \frac{1}{2}\frac{d}{dt}\int_\Omega (w_i(t,x))^2dx = \frac{1}{2}\frac{d}{dt}\|w_i(t,x)\|_2^2$, combining (4)-(7) into (3), for all $t \neq t_k$, we have

$$\begin{aligned}
D^+\|w_i(t,x)\|_2 \leq &-\underline{\alpha_i}\beta_i\|w_i(t,x)\|_2 + \overline{\alpha_i}\sum_{j=1}^n |a_{ij}|F_j\|w_j(t,x)\|_2 \\
&+\overline{\alpha_i}\sum_{j=1}^n (|\alpha_{ij}|+|\beta_{ij}|)F_j\|w_j(t-\tau_{ij}(t),x)\|_2
\end{aligned} \tag{8}$$

Let $v_i(t) = \|w_i(t,x)\|_2, \quad i = 1,2,\cdots,n$, then we have

$$D^+v_i(t) \leq -\underline{\alpha_i}\beta_i v_i(t) + \overline{\alpha_i}\sum_{j=1}^n |a_{ij}|F_j v_j(t) + \overline{\alpha_i}\sum_{j=1}^n (|\alpha_{ij}|+|\beta_{ij}|)F_j v_j(t-\tau_{ij}(t)). \tag{9}$$

Let $P = -\underline{\alpha}\beta + \overline{\alpha}|A|F, \quad Q = \overline{\alpha}(|\overline{A}|+|\overline{B}|)F, \quad v(t) = (v_1(t), v_2(t), \cdots, v_n(t))^T$, $V(t) = (v_j(t-\tau_{ij}(t)))_{n\times n}$, then we have

$$D^+v(t) \leq Pv(t) + (Q \otimes V(t))e_n \qquad t \neq t_k. \tag{10}$$

From condition (A_1) , we can obtain

$$[\lambda E + P + Q \otimes \varepsilon(\lambda)]\xi < 0. \tag{11}$$

Let $K = \frac{\|\phi - u^*\|_2}{\min_{1\leq i\leq n}\{\xi_i\}}$, it is easily obtained that $v(t) \leq K\xi e^{-\lambda t} \quad -\tau \leq t \leq t_0 = 0$. Then from Lemma 2, we obtain $v(t) \leq K\xi e^{-\lambda t}$, for $t \in [t_0, t_1)$.

Suppose that

$$v(t) \leq K\eta_0\eta_1\cdots\eta_{l-1}\xi e^{-\lambda t}, \quad 1 \leq l \leq k, \quad t \in [t_{l-1}, t_l) \tag{12}$$

where $\eta_0 = 1, \quad k \geq 2$. When $l = k+1$, from (H_4) and (A_2), we have

$$\begin{aligned} v_i(t_k^+) &= \|w_i(t_k^+, x)\|_2 = \|\widetilde{h}_{ik}(w_i(t_k^-, x))\|_2 \leq \gamma_{ik}\|w_i(t_k^-, x)\|_2 \\ &= \gamma_{ik} v_i(t_k^-) \leq \gamma_{ik} K\eta_0\eta_1\cdots\eta_{k-1}\xi_i e^{-\lambda t_k} \leq K\eta_0\eta_1\cdots\eta_{k-1}\eta_k\xi_i e^{-\lambda t_k}. \end{aligned} \tag{13}$$

By (12), (13) and $\eta_k \geq 1$, we have

$$v(t) \leq K\eta_0\eta_1\cdots\eta_{k-1}\eta_k\xi e^{-\lambda t}, \quad t \in [t_k - \tau, t_k]. \tag{14}$$

Combining (10), (11), (14) and Lemma 2, we have

$$v(t) \leq K\eta_0\eta_1\cdots\eta_{k-1}\eta_k\xi e^{-\lambda t}, \quad t \in [t_k, t_{k+1}). \tag{15}$$

Applying mathematic induction, we conclude that

$$v(t) \leq K\eta_0\eta_1\cdots\eta_{k-1}\eta_k\xi e^{-\lambda t}, \quad t \in [t_{k-1}, t_k),\ k \in N. \tag{16}$$

Form (A_2), we have

$$v(t) \leq Ke^{\eta t_1}e^{\eta(t_2-t_1)}\cdots e^{\eta(t_{k-1}-t_{k-2})}\xi e^{-\lambda t} \leq K\xi e^{\eta t}e^{-\lambda t} = K\xi e^{-(\lambda-\eta)t}$$

for all $t \in [t_{k-1}, t_k)$ and $k \in N$, which implies that

$$\|u(t,x)-u^*\|_2 = (\sum_{i=1}^{n}\|u_i(t,x)-u_i^*\|_2^2)^{\frac{1}{2}} \leq K(\sum_{i=1}^{n}\xi_i^2)^{\frac{1}{2}}e^{-(\lambda-\eta)t} = M\|\phi-u^*\|_2 e^{-(\lambda-\eta)t}$$

where $M = \frac{(\sum_{i=1}^{n}\xi_i^2)^{\frac{1}{2}}}{\min_{1\leq i\leq n}\{\xi_i\}} \geq 1$.

It means that the equilibrium point u^* of system (1) is globally exponentially stable, and its exponential convergence rate equals $\lambda - \eta$.

Corollary 1. *Under assumption $(H_1)-(H_4)$, if the following conditions hold:*

(A_1'). There exist a vector $\xi = (\xi_1, \xi_2, \cdots, \xi_n)^T > 0$ and a positive $\lambda > 0$, such that

$$[\lambda E - \underline{\alpha}\beta + \overline{\alpha}|A|F + \overline{\alpha}(|\overline{A}| + |\overline{B}|)F \cdot e^{\lambda\tau}]\xi < 0,$$

(A_2). $\eta = \sup_{k\in N}\{\frac{\ln\eta_k}{t_k - t_{k-1}}\} < \lambda, \qquad \eta_k = \max_{1\leq i\leq n}\{1, \gamma_{ik}\}, \quad k \in N,$

then the equilibrium point u^* is globally exponentially stable, and its exponential convergence rate equals $\lambda - \eta$.

Proof. *Since $e^{\lambda\tau} \geq e^{\lambda\tau_{ij}}$, by the condition (A_1'), we have condition (A_1) holds. This complete the proof.*

Corollary 2. *Under assumption $(H_1)-(H_4)$, if the following conditions hold:*

(A_1). There exist a vector $\xi = (\xi_1, \xi_2, \cdots, \xi_n)^T > 0$ and a positive $\lambda > 0$, such that

$$[\lambda E - \underline{\alpha}\beta + \overline{\alpha}|A|F + \overline{\alpha}(|\overline{A}| + |\overline{B}|)F \otimes \varepsilon(\lambda)]\xi < 0,$$

(A_2'). $\gamma_{ik} \le 1, \quad i = 1, 2, \cdots, n \quad k \in N,$
then the equilibrium point u^* *is globally exponentially stable, and its exponential convergence rate equals* λ.

Proof. *Since* $\gamma_{ik} \le 1, \ \eta_k = \max_{1 \le i \le n}\{1, \gamma_{ik}\} = 1, \quad i = 1, 2, \cdots, n \quad k \in N.$ *Therefore,* $\eta = 0 < \lambda$ *which implies that* (A_2) *hold.*

4 An Illustrative Example

Example

$$\begin{cases} \frac{\partial u_i(t,x)}{\partial t} = \sum\limits_{l=1}^{3} \frac{\partial}{\partial x_l}(D_{il}(\frac{\partial u_i(t,x)}{\partial x_l})) - \alpha_i(u_i(t,x))[\beta_i(u_i(t,x)) - \sum\limits_{j=1}^{2} a_{ij} f_j(u_j(t,x)) \\ \qquad - \sum\limits_{j=1}^{2} b_{ij}\mu_j - I_i - \bigwedge\limits_{j=1}^{2} \alpha_{ij} f_j(u_j(t-\tau_{ij}(t),x)) \\ \qquad - \bigvee\limits_{j=1}^{2} \beta_{ij} f_j(u_j(t-\tau_{ij}(t),x)) - \bigwedge\limits_{j=1}^{2} T_{ij}\mu_j - \bigvee\limits_{j=1}^{2} H_{ij}\mu_j] \qquad t \ne t_k \\ u_i(t_k^+, x) = u_i(t_k^-, x) - (1 + e^{0.005k})(u_i(t_k^-, x) - 1) \quad t_k = t_{k-1} + 0.5k, t_0 = 0 \\ \frac{\partial u_i(t,x)}{\partial n} = (\frac{\partial u_i(t,x)}{\partial x_1}, \frac{\partial u_i(t,x)}{\partial x_2}, \cdots, \frac{\partial u_i(t,x)}{\partial x_m})^T = 0 \qquad (t,x) \in [-\tau, \infty) \times \partial\Omega \\ u_i(s,x) = \phi_i(s,x) \qquad (s,x) \in [-4, 0] \times \Omega \quad \phi_i(s,x) \in PC(\Omega), \ i = 1, 2 \end{cases} \tag{17}$$

where $\alpha_1(x) = 2 + 0.1\cos x, \alpha_2(x) = 2 + 0.1\sin x, \beta_1(x) = \beta_2(x) = x, f_1(x) = f_2(x) = \frac{1}{2}(|x+1| - |x-1|), \alpha_{ij} = \beta_{ij} = T_{ij} = H_{ij} = a_{ij} = b_{ij} = 0.1 \quad i = 1, 2 \quad j = 1, 2, I = (0.2, 0.2)^T, u = (1,1)^T, (\tau_{ij}(t)) = \begin{pmatrix} \cos^2 t & 2\sin^2 t \\ 3\cos^2 t & 4\sin^2 t \end{pmatrix}$. It is easily obtained that $\underline{\alpha} = \begin{pmatrix} 1.9 & 0 \\ 0 & 1.9 \end{pmatrix}, \quad \overline{\alpha} = \begin{pmatrix} 2.1 & 0 \\ 0 & 2.1 \end{pmatrix}, \quad \beta = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$
$F = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad A = \begin{pmatrix} 0.1 & 0.1 \\ 0.1 & 0.1 \end{pmatrix}, \quad \overline{A} = \begin{pmatrix} 0.1 & 0.1 \\ 0.1 & 0.1 \end{pmatrix}, \quad \overline{B} = \begin{pmatrix} 0.1 & 0.1 \\ 0.1 & 0.1 \end{pmatrix}, \quad \varepsilon(\lambda) = \begin{pmatrix} e^{\lambda} & e^{2\lambda} \\ e^{3\lambda} & e^{4\lambda} \end{pmatrix}$
$\gamma_{1k} = \gamma_{2k} = e^{0.005k}$, $\eta_k = \max\{1, e^{0.005k}\} = e^{0.005k} > 1$, $\eta = \sup_{k \in N}\{\frac{\ln \eta_k}{t_k - t_{k-1}}\} = 0.01$

Let $\lambda = 0.1, \quad \xi = (1,1)^T$. Obviously $\eta < \lambda$. By computing we have

$$[\lambda E - \underline{\alpha}\beta + \overline{\alpha}|A|F + \overline{\alpha}(|\overline{A}| + |\overline{B}|)F \otimes \varepsilon(\lambda)]\xi < 0$$

From theorem 1, we know that the equilibrium point $u^* = (1,1)^T$ of the system (17) is globally exponentially stable, and its exponential convergence rate is not less than $\lambda - \eta = 0.09$.

5 Conclusion

In this paper, we have investigated the global exponential stability for a class of impulsive fuzzy cellular neural networks with delays and reaction-diffusion

terms. Using Lyapunov functional method and inequality technique, we gave a sufficient criterion ensuring the global exponential stability of the equilibrium point.

Acknowledgement

This work was supported by The National Natural Science Foundation of P.R. China (60764003), The Major Project of The Ministry of Education of P.R. China (207130) and The Scientific Research Programmes of Colleges in Xinjiang (XJEDU2007G01, XJEDU2006I05).

References

1. Chua, L.O., Yang, L.: Cellular Neural Networks: Theory. IEEE Trans. Circuits Systems 35, 1257–1272 (1988)
2. Yang, T., Yang, L., Wu, C., Chua, L.: Fuzzy Cellular Neural Networks: Theory. Proc IEEE Int Workshop Cellular Neural Networks Appl., 181–186 (1996)
3. Yang, T., Yang, L.: The Global Stability of Fuzzy Cellular Neural Networks. IEEE Transactions on Circuits Systems I 43, 880–883 (1996)
4. Song, Q., Cao, J.: Dynamical Behaviors of Discrete-Time Fuzzy Cellular Neural Networks with Variable Delays and Impulses. Journal of the Franklin Institute 345, 39–59 (2008)
5. Chen, L., Zhao, H.: Stability Analysis of Stochastic Fuzzy Cellular Neural Networks with Delays. Neurocomputing 72, 436–444 (2008)
6. He, D., Xu, D.: Attracting and Invariant Sets of Fuzzy Cohen-Grossberg Neural Networks with Time-Varying Delays. Physics Letters A 372, 7057–7062 (2008)
7. Wang, J., Lu, J.: Global Exponential Stability of Fuzzy Celluar Neural Networks with Delays ang Reaction-Diffusion Term. Chaos, Solitons and Fractals 38, 878–885 (2008)
8. Huang, T.: Exponential Stability of Delayed Fuzzy Cellular Neural Networks with Diffusion. Chaos, Solitons and Fractals 31, 658–664 (2007)
9. Li, K., Song, Q.: Exponential Stability of Impulsive Cohen-Grossberg Neural Networks with Time-Varying Delays and Reaction-Diffusion Term. Neurocomputing 72, 231–240 (2008)

Nonnegative Periodic Dynamics of Cohen-Grossberg Neural Networks with Discontinuous Activations and Discrete Time Delays*

Xiangnan He, Wenlian Lu, and Tianping Chen

Key Laboratory of Nonlinear Science of Chinese Ministry of Education, School of Mathematical Sciences, Fudan University, Shanghai, 200433, P.R. China
seanhe2007@gmail.com, {042018047,wenlian,tchen}@fudan.edu.cn

Abstract. In this paper, we report the results concerned with the nonnegative periodic dynamics of the delayed Cohen-Grossberg neural networks with discontinuous activation functions and periodic interconnection coefficients, self-inhibitions, and external inputs. Filippov theory is utilized to study the viability, namely, the existence of the solution of the Cauchy problem. The conditions of diagonal dominant type are presented to guarantee the existence and the asymptotical stability of a periodic solution. Numerical examples are provided to illustrate the theoretical results.

1 Introduction

Research on the dynamical behavior of the recurrently connected neural networks is an important topic in neural network theory. Among them, Cohen-Grossberg neural networks were proposed in pioneering works of Cohen and Grossberg [1], and can be modeled by the following differential equations:

$$\frac{dx_i(t)}{dt} = A_i(x_i(t))\Big[- d_i x_i(t) + \sum_{j=1}^{n} a_{ij} g_j(x_j(t)) + J_i\Big],\ i = 1, \cdots, n \qquad (1)$$

where $x_i(t)$ denotes the state variable of the potential of the i-th neuron, d_i represents the self-inhibition with which the i-th neuron will reset its potential to the resting state in isolations when disconnected from the network, a_{ij} denotes the connection strength of j-th neuron on the i-th neuron, $g_i(\cdot)$ denotes the activation function of i-th neuron, J_i denotes the external input to the i-th neuron, and $A_i(\cdot)$ denotes amplification function of the i-th neuron. There are a lot of papers in literature discussing the local and global stability of this system as well as the well-know Hopfield neural networks [2]:

$$\frac{dx_i(t)}{dt} = -d_i x_i(t) + \sum_{j=1}^{n} a_{ij} g_j(x_j(t)) + J_i,\ i = 1, \cdots, n \qquad (2)$$

* This work was supported by the National Natural Sciences Foundation of China under Grant Nos. 60774074 and 60804044, and sponsored by Shanghai Pujiang Program No. 08PJ14019.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 579–588, 2009.

which can be regarded as the special case of the system (1) by letting $A_i(\rho) = 1$, $i = 1, \cdots, n$. For reference, see [1,3,4,5,6] and many others.

In practice, time delays inevitably occur due to the finite switching speed of the amplifiers and communication time. Thus, the neural networks can be modeled by the following delayed differential equations:

$$\frac{dx_i(t)}{dt} = A_i(x_i(t))\Big[- d_i x_i(t) + \sum_{j=1}^{n} a_{ij} g_j(x_j(t)) + \sum_{j=1}^{n} b_{ij} g_j(x_j(t-\tau_{ij})) + J_i\Big], \quad i = 1, \cdots, n, \tag{3}$$

where b_{ij} denotes the delayed feedback of the j-th neuron on the i-th neuron. There are also many papers discussing the stability of delayed neural networks. See [7,8,9,10,11,12,13]for references. In these papers, various conditions based on Lyapunov functionals were given guaranteeing the global stability.

Furthermore, the interconnections may be asynchronous, namely, the interconnection weights a_{ij}, b_{ij}, self-inhibitions d_i and inputs J_i should vary through time. Therefore, we need to study the non-autonomous dynamical systems with time-varying self-inhibitions, connections, and inputs:

$$\frac{dx_i(t)}{dt} = A_i(x_i(t))\Big[- d_i(t) x_i(t) + \sum_{j=1}^{n} a_{ij}(t) g_j(x_j(t)) + \sum_{j=1}^{n} b_{ij}(t) g_j(x_j(t-\tau_{ij})) + J_i(t)\Big], \ i = 1, \cdots, n. \tag{4}$$

Recently, a number of researchers have investigated the existence and global attraction of the periodic solution ([14,15,16,17,18,19])or almost periodic solution([20,21])for these non-autonomous delayed differential systems, assuming that the system is periodic or almost periodic respectively. Among them, several methods (such as the fixed point theorem) have been used to obtain the existence of the solutions.

However, all the results obtained in these papers were based on the assumption that amplifier function $A_i(\cdot)$ is always **positive** (see [23,6]), even greater than some positive number $A_i(\cdot) \geq \underline{A}_i > 0$ (see [22,23]). But in their original paper, [1,24,25] , they proposed this model as a kind of competitive-cooperation dynamical system for decision rules, pattern formation, and parallel memory storage. Hereby, each state of neuron x_i might be the population size, activity, or concentration, etc. of the ith species in the system, which is always nonnegative for all time. To guarantee the positivity of the states, one should assume $A_i(\rho) > 0$ for all $\rho > 0$ and $A_i(0) = 0$ for all $i = 1, \cdots, n$. [1,25] provided the pioneering study on the dynamics of such neural network model with assuming $A_i(\rho) > 0$ for all $\rho > 0$ and $A_i(0) = 0$ for all $i = 1, \cdots, n$ and without considering any time delay. [26,27] studied the global stability of nonnegative equilibrium or nonnegative periodic solution of the Cohen-Grossberge neural networks with assuming $A_i(\rho) > 0$ for $\rho > 0$ and $A_i(0) = 0, i = 1, \cdots, n$.

In addition, the works mentioned above were based on the assumption that the activation functions are continuous even globally Lipshitz. As mentioned by [28] a brief review on some common neural network models reveals that neural networks with discontinuous activations are of importance and do frequently arise in practice.

In the last few years, there arise several papers studying neural networks with discontinuous activations. [28] discussed the absolute stability of Hopfield neural networks (2) with bounded and discontinuous activations. [29] proved the global convergence for Cohen-Grossberg neural networks with unbounded and discontinuous activations. Also, [30,31] studied the dynamics of delayed neural networks. [32] discussed the periodic solution of the periodic delayed neural networks with discontinuous activations and periodic parameters and [33] studied almost periodic dynamical behaviors of delayed neural networks with almost periodic coefficients. However, these works are all concerned with Hopfield neural networks or Cohen-Grossberg networks with always positive amplifiers.

Continuing with our previous work [29,31,33], the aim of this paper is to study the nonnegative periodic dynamical behaviors of the delayed Cohen-Grossberg neural network system:

$$\dot{x}_i = A_i(x_i)\Big[-d_i(t)x_i(t) + \sum_{j=1}^{n} a_{ij}(t)g_j(x_j(t) + \sum_{j=1}^{n} b_{ij}(t)g_j(x_j(t-\tau_{ij})) + J_i(t)\Big], \quad i = 1, \cdots, n. \tag{5}$$

with discontinuous activations, periodic coefficients, and without assuming the positivity of the amplifier functions. Hereby, we focus our study of the dynamical behaviors on the first orthant: $R_+^n = \{(x_1, \cdots, x_n)^\top \in R^n : \ x_i \geq 0, \ i = 1, \cdots, n\}$ and consider all trajectories initiated in the first orthant R_+^n instead of the whole space R^n. We introduce the concept of solutions in the Filippov sense for delayed dynamical system (5) and prove its existence in the first orthant by the idea introduced by Filippov [34]. Then, we present sufficient condition to guarantee the existence of a nonnegative periodic solution and the global exponential stability of this periodic solution. Since a constant can be viewed as a special periodic function with arbitrary period, the results also apply to the stability of nonnegative equilibrium of the systems with constant self-inhibition, connection weights and outer inputs.

2 Preliminaries

In this section, we present the necessary hypotheses for the model description, definitions, and notations, which will be used in the following part of this paper.

2.1 Model Hypotheses

In this paper, we consider the differential equations (5). The amplifier functions $A_i(\cdot)$ is assumed to satisfy the condition below to guarantee the positivity of the solution(see lemma 2).

H_1: For all $i = 1, 2, \ldots, n$, $A_i(s)$ is continuous and for $s \geq 0$ with $A_i(s) > 0$ for $s > 0$ and $A_i(0) = 0$, and

$$\int_0^{\epsilon} \frac{ds}{A_i(s)} = +\infty, \ \ i = 1, \ldots, n$$

where ϵ is an arbitrary positive number.

The self-inhibitions are assumed to be always positive.

H_2: For all $i = 1, 2, \ldots, n$, $d_i(t) \geq D_i$, where D_i is a positive constant for $i = 1, 2, \ldots, n$.

The activation functions have isolated points of discontinuity:

H_3: Let $g(x) = (g_1(x_1), g_2(x_2), \ldots, g_n(x_n))^T$. $g_i(\cdot)$ is non-increasing and in every compact set of $\mathbb{R}$, each $g_i(\cdot)$ has only finite discontinuous points. Therefore, in any compact set in $\mathbb{R}$, except a finite points $\{\rho_k\}$, where there exist finite right and left limit $g_i(\rho^+)$ and $g_i(\rho^-)$ with $g_i(\rho^+) > g_i(\rho^-)$, $g_i(\cdot)$ is continuous. Moreover, we assume $g_i(\cdot)$ is bounded, i.e. there exist a positive number $G > 0$, such that $g_i(\cdot) \leq G$.

The time-varying coefficients, including the self-inhibitions, interconnection coefficients, and the external inputs, are all periodic with the same period.

H_4: $d_i(t)$, $a_{ij}(t)$, $b_{ij}(t)$, and $J_i(t)$ are all continuous function and satisfy

$$d_i(t+\omega) = d_i(t),\ a_{ij}(t+\omega) = a_{ij}(t),\ b_{ij}(t+\omega) = b_{ij}(t),\ J_i(t+\omega) = J_i(t)$$

for all $t \in R$ and $i, j = 1, \cdots, n$.

The following diagonal dominant conditions lie at the key position of the existence and its stability of the periodic solution of the delayed system (5).

H_5: There exist positive constants $\xi_1, \xi_2, \ldots \xi_n$ such that for all $i = 1, 2, \ldots, n$, the following conditions are satisfied:

$$\xi_i a_{ii}(t) + \sum_{j=1, j\neq i}^{n} \xi_j |a_{ji}(t)| + \sum_{j=1}^{n} \xi_j |b_{ji}(t+\tau_{ij})| < 0. \tag{6}$$

2.2 Sense of the Solution and Its Stability

Here, we present the sense of the solution of the Cauchy problem of the delayed differential system (5) according to positive initial conditions $\phi_i(\theta) > 0$, $\theta \in [-\tau_M, 0]$. The existence of the solution of the Cauchy problem is studied with the Filippov theory [34], which is based on the set-valued map analysis.

Definition 1. *Suppose $E \subset \mathbb{R}^n$. Map $x \mapsto F(x)$ is called a set-value map from $E \hookrightarrow \mathbb{R}^n$, if to each point x of a set $E \subset \mathbb{R}^n$, there corresponds to a non-empty set $F(x) \subset \mathbb{R}^n$. A set-value map F with non-empty value is said to be upper semicontinuous at $x_0 \in E$, if for any open set N containing $F(x_0)$, there exists a neighborhood M of x_0 such that $F(M) \subset N$. $F(x)$ is said to have closed (convex, compact) image, if for each $x \in E$, $F(x)$ is closed (convex, compact).*

More details can be found in [35].

The sense of the solution of the dynamical system (5) with delays and discontinuous activations comes from the Filoppov's definition [34]. Denote the convex closure of $g_j(\rho)$ by $K[g_j(\rho)]$. That is, for a point of discontinuity of g_j: ρ^*, $K[g_j(\rho)] = [g_j(\rho^* - 0), g_j(\rho^* + 0)]$. Similar to [31, 33], we define the Filippov solution of Eqs. (5) as follows:

Definition 2. *A solution of Cauchy problem of the system (5) with respect to continuous initial conditions $\phi(\theta) = [\phi_1(\theta), \cdots, \phi_n(\theta)]^\top$, $\theta \in [-\tau_M, 0]$, is an absolutely*

continuous function $x(t)$ *on* $[0,T)$*(*T *might be* $+\infty$*) such that* $x(\theta)=\phi(\theta)$*, for* $\theta\in[-\tau_M,0]$*, and satisfies*

$$\begin{cases} \frac{dx_i}{dt}=A_i(x)[-d_i(t)x_i(t)+\sum_{j=1}^{n}a_{ij}(t)\gamma_j(t) \\ +\sum_{j=1}^{n}b_{ij}(t)\gamma_j(t-\tau_{ij})+J_i(t)] & a.e.t\in[0,T) \\ \\ \gamma_j(t)\in K[g_j(x_j(t))] & a.e.t\in[0,T) \end{cases} \tag{7}$$

Thus, the asymptotical stability of a nonnegative periodic solution $x^*(t)$ (if existing) is defined in the first orthant.

Definition 3. $x^*(t)$ *is said to be asymptotically* R_+^n*-stale if for any positive initial conditions* $\phi_i(\theta)$*,* $\theta\in[-\tau_M,0]$ *and* $i=1,\cdots,n$*, the solution* $x(t)$ *satisfies*

$$\lim_{t\to\infty}|x_i(t)-x_i^*(t)|=0,\ i=1,\cdots,n.$$

A basic condition is used to derived the existence of such solution in Filippov's sense.

3 Main Results

In this section, we firstly study the existence of the positive solution of the Cauchy problem of the system (7) with respect to positive initial conditions. Second, we discuss the existence of nonnegative periodic solution. Finally, we investigate the R_+^n-stability of the nonnegative solution. Due to the limit of spaces, all the proofs are omitted and will be presented somewhere else in our future paper.

Definition 4. *[34] A set valued map* F*:*$\mathbf{R}^n\times\mathbf{R}^+\mapsto\mathbf{R}^n$ *is said to satisfy the* **basic conditions** *in a domain* $G\subset\mathbf{R}^n\times\mathbf{R}^+$*, if for any* $(x,t)\in G$*,* $F(x,t)$ *is non-empty, bounded, closed and convex, and* F *is upper semicontinuous in* (x,t)*.*

Lemma 1. *[34] If a set-valued map* $F(x,t)$ *satisfies the* **basic conditions** *in the interior of the domain G, then for any point* $(x_0,t_0)\in G$*, there exists a solution in* G *of the following differential equation:*

$$\begin{cases} \dot{x}\in F(x,t) \\ x(t_0)=x_0 \end{cases} \tag{8}$$

over an interval $[t_0,t^{'})$ *for some* $t^{'}>t_0$*. Moreover, If* $F(x,t)$ *satisfies the* basic conditions *in a closed bounded domain* D*, then each solution of the system (8) lying within* D *can be continued on both sides up to the boundary of the domain* D*.*

The assumption H_1 can lead that each solution trajectory (if existing) of the system (7) is positive if the initial data are positive.

Lemma 2. *(Positivity) Under the assumption* H_1**,** H_4*, if initial date is positive, i.e.,* $x_i(0)>0$ *for all* $i=1,\cdots,n$*, then each solution of the system (7) is positive within the duration time.*

Also, the boundedness of the activation functions can imply that the solution is bounded in the whole duration time interval.

Lemma 3. *(Boundedness) Under the assumptions H_2, H_3, H_4, any positive solution $x(t)$ of system (7) is bounded on its duration time interval.*

Therefore, we are in the position to give the theorem of viability.

Theorem 1. *(Viability) Under the assumption H_{1-4}, for each positive initial conditions, the system (7) admits one positive bounded solution of which the duration time interval is $[0, +\infty)$.*

Lemma 1 tells us that there exists one solution on $[0, \tau_m]$, lemma 2 and lemma 3 restrict the solution in $\mathbb{R}^n_+$. By regrading the value of $x_i(t)$ on $(-\infty, \tau_m]$ as initial condition of system (7), the solution can be extended to $[\tau_m, 2\tau_m]$. Continuing this phase can lead that the solution of the system (7) exist in the whole time interval $[0, +\infty)$ and belongs to the domain $\mathbb{R}^n_+$.

For convenient citation, we rewrite the solution of the delayed Cohen-Grossberg neural networks with discontinuous activations as follows:

$$\frac{dx_i}{dt} = A_i(x_i(t))\left[-d_i(t)x_j(t) + \sum_{j=1}^{n} a_{ij}(t)\gamma_j(x_j(t)) + \sum_{j=1}^{n} b_{ij}(t)\gamma_j(t-\tau_{ij}) + J_i(t)\right], \quad i = 1, \cdots, n$$
$$\gamma_j(t) \in K[g_j(x_j(t))], \ j = 1, \cdots, n, \tag{9}$$

for almost every $t \geq 0$, where $K[g_j(x_j(t))] = [g(x_j(t) - 0), g_j(x_j(t) + 0)]$, $j = 1, \cdots, n$.

With diagonal dominant condition H_5, we can obtain the existence of a nonnegative periodic solution as well as its R^n_+-stability.

Theorem 2. *Under the assumptions H_{1-5}, the system (9) has a nonnegative ω-periodic solution $x^*(t)$ and this periodic $x^*(t)$ is globally asymptotically stable for the system (9).*

Let $x^m(t) = x(t + m\omega)$. Define a candidate Lyapunov function:

$$L(t) = \sum_{i=1}^{n} \xi_i |z_i(t)| + \sum_{i,j=1}^{n} \xi_i \int_{t-\tau_{ij}}^{t} |b_{ij}(s + \tau_{ij})||w_i(s)| ds, \tag{10}$$

where $z_i(t) = \int_{x_i(t)}^{x_i(t+\omega)} \frac{d\rho}{a_i(\rho)}$ and $w_i(t) = \gamma_i(t+\omega) - \gamma_i(t)$. Differentiating it, we have

$$\frac{dL(t)}{dt} \leq -\sum_{i=1}^{n} D_i \xi_i |x_i(t+\omega) - x_i(t)| + \sum_{i=1}^{n} \Big[\xi_i a_{ii}(t) + \sum_{j \neq i}^{n} \xi_j a_{ji}(t)$$
$$+ \sum_{j=1}^{n} \xi_j |b_{ji}(t + \tau_{ij})|\Big] |w_i(t)| \leq -\alpha \sum_{i=1}^{n} |x_i(t+\omega) - x_i(t)|,$$

(the last inequality holds by H_5) where $\alpha = \min_i \xi_i D_i$. This indicates

$$\int_0^{+\infty} |x_i(t+\omega) - x_i(t)| \leq \frac{L(0)}{\alpha} < +\infty,$$

which implies that $\{x^m(t)\}_{m=0}^{\infty}$ is a Cauchy sequence in the space $L^1([0,\omega], R^n)$. Therefore, there exists an ω-periodic function$x^*(t) \in L^1([0,\omega], R^n)$ such that $\lim_{m\to\infty} \int_0^{\omega} |x(t+m\omega) - x^*(t)|dt = 0$. By Arzela-Ascoli lemma and Mazur convexity theorem, we can proof that $x^*(t)$ is a solution of solution of system (9). The globally asymptotical stability can be obtained by a similar Lyapunov function.

The case of constant coefficients can be regarded as special case of periodic ones.

Corollary 1. *Under the assumptions $H_{1,2,3,5}$, if all d_i, a_{ij}, b_{ij}, J_i, $i,j = 1,\cdots,n$, are constants, the the system (9) admits a positive solution for each positive initial conditions; moreover, there exists a nonnegative equilibrium of the system (9) which is asymptotically R_+^n-stale.*

4 Numerical Illustrations

In this section, we present two examples of two dimensions to verify theoretical results given in previous sections.

Example 1

Consider the following system:

$$\begin{cases} \dot{x_1} = x_1(t)[-x_1(t) + (-3+\cos t)sign(x_1(t)-2) \\ +(1-\cos t)sign(x_1(t-1)-2) \\ +(1+\sin t)sign(x_1(t-1)-2) + 7 + \sin(t)] \\ \\ \dot{x_2} = x_2(t)[-x_2(t) + \cos t sign(x_1(t)-2) \\ +(-5+\cos t)sign(x_2-2) \\ +\sin(t)sign(x_1(t-1)-2) \\ +(1+\cos t)sign(x_2(t-1)-2) + 5 + \sin t] \end{cases} \tag{11}$$

Here, the activation function is picked as $sign(\rho - 2)$, where $sign$ is the sign function. Assumptions H_1-H_4 are obviously satisfied, Thus we only need to verify H_4, and by choosing $\xi_1 = \xi_2 = 1$, assumption H_5 can be rewritten as the inequities:

$$\begin{cases} (-3+\cos t) + |\cos t| + |1 - \cos(t+1)| + |\sin(t+1)| < 0 \\ (-5+\cos t) + |1 + \sin(t+1)| + |(1+\cos(t+1)| < 0, \end{cases}$$

which holds for any $t \geq 0$ via direct algebras. Therefore, from Theorems 2, one can see that the system (11) has a nonnegative 2π-periodic solution which is globally asymptotically stable. As shown in Fig. 1, the two components converge two positive periodic functions respectively.

We can see that the solution of the system (11) is positive, and the next example shows that the solution can be nonnegative under assumptions H_1-H_5.

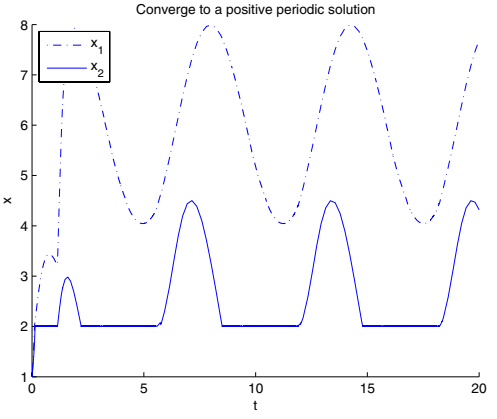


Fig. 1. Dynamics of the system (11) with initial condition $x_i(\theta) = 1$ for $\theta \in [-1, 0]$

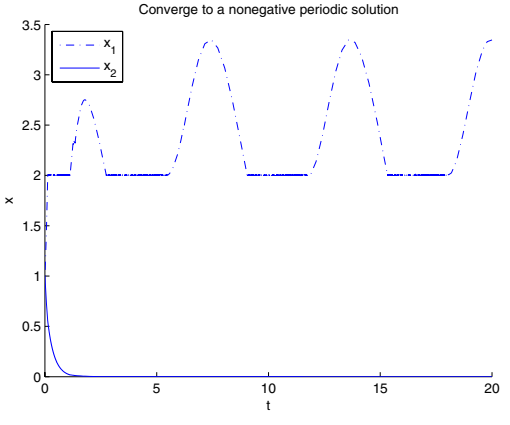


Fig. 2. Dynamics of the system (12) with initial condition $x_i(\theta) = 1$ for $\theta \in [-1, 0]$

Example 2
Consider the other system:

$$\begin{cases} \dot{x_1} = x_1(t)[-x_1(t) + (-3 - \cos t)sign(x_1(t) - 2) \\ +(1 + \cos t)sign(x_2(t)) \\ +(1 + \cos t)sign(x_1(t-1) - 2) \\ +(1 + \sin t)sign(x_2(t-1)) + 2] \\ \\ \dot{x_2} = x_2(t)[-x_2(t) + \cos t sign(x_1(t) - 2) \\ +(-3 - \sin t)sign(x_2(t)) \\ +\sin t sign(x_1(t-1) - 2) + \sin t] \end{cases} . \tag{12}$$

where the assumption H_5 can be rewritten as:

$$\begin{cases} (-3-\cos t)+|\cos t|+|1+\cos(t+1)|+|\sin(t+1)|<0 \\ (-3-\sin t)+|1+\cos t|+|1+\sin(t+1)|<0, \end{cases}$$

which holds for any $t \in \mathbb{R}$. Therefore by Theorem 2, system (15) has a nonnegative 2π-periodic solution which is globally asymptotically stable. As shown in Fig. 2, one component converges to a positive periodic solution but the other converges to zero. This illustrate that even though each solution is positive due to the positive initial conditions, the terminal limit may be zero.

5 Conclusions

In conclusion, we study the nonnegative periodic dynamics of the delayed Cohen-Grossberg neural networks with discontinuous activations, periodic coefficients, and without assuming the positivity of the amplifiers. We use the Filippove theory to prove that the existence of a positive solution with respect to positive initial conditions and Lyapunov methods to investigate the existence and asymptotical R^n_+-stability of the nonnegative periodic solution. That is to say, the all positive solutions asymptotically converge to a periodic solution, which can be positive or zero. We present numerical examples to illustrate that both cases are possible.

References

1. Cohen, M.A., Grossberg, S.: Absolute Stability and Global Pattern Formation and Parallel Storage by Competitive Neural Networks. IEEE Transactions on systems, Man, and Cybernectics 13, 815–826 (1983)
2. Hopfield, J.J., Tank, D.W.: Computing with Neural Circuits: a Model. Science 233, 625–633 (1986)
3. Hirsh, M.: Convergence Activation Dynamics in Continuous Time Networks. Neural Networks 2, 331–349 (1989)
4. Michel, A.N., Gray, D.L.: Analysis and Synthesis of Neural Networks with Lower Block Triangular Interconnecting Structure. IEEE Transactions on Circuits and System 37, 1267–1283 (1990)
5. Forti, M., Tesi, A.: New Condition for Global Stability of Neural Networks with Application to Linear Quadratic Programming Problems. IEEE Transactions on Circuits and Systems-1 42, 354–366 (1995)
6. Lu, W., Chen, T.P.: New Conditions Global Stability of Cohen-Grossberg Neural Networks. Neural Computation 15, 1173–1189 (2003)
7. Bleair, J., Campbell, S.A., Driessche, P.: Stability and Delay-Induced Oscillations in a Neural Network Model. SIAM. Appl. Math. 56, 245–255 (1996)
8. Cao, J., Zhou, D.: Stability Analysis of Delayed Neural Networks. Neural Networks 11, 1601–1605 (1998)
9. Chen, T.P.: Global Exponential Stability of Delayed Hopfield Neural Networks. Neural Networks 14, 977–980 (2001)
10. Civalleri, P.P., Gilli, L.M., Pabdolfi, L.: On Stability of Cellular Neural Networks with Delay. IEEE Transactions on Circuits and Systems 40, 157–164 (1993)

11. Gopalasamy, K., He, X.: Stability in Asymmetric Hopfield Nets with Transmission Delays. Phys. D 76, 344–358 (1991)
12. Joy, M.P.: Results Concerning the Absolute Stability of Delayed Neural Networks. Neural Networks 13, 613–616 (2000)
13. Lu, W., Rong, L., Chen, T.P.: Global Convergence of Delayed Neural Network Systems. International Journal of Neural Systems 13, 193–204 (2003)
14. Cao, J.: New Results Concerning Exponential Stability and Periodic Solutions of Delayed Cellular Neural Networks. Phys. Lett. A 307, 136–147 (2003)
15. Chen, T.P., Lu, W., Chen, G.: Dynamical Behaviors of a Large Class of General Delayed Neural Networks. Neural Computation 17, 949–968 (2005)
16. Gopalsamy, K.S.: Time Delays and Stimulus-Dependent Pattern Formation in Periodic Environments in Isolated Neurals. IEEE Transactions Neural Networks 13, 551–563 (2002)
17. Lu, W., Chen, T.P.: On Periodic Dynamical Systems. Chin. Ann. Math. 25B, 455–462 (2004)
18. Zhou, J., Liu, Z., Chen, G.: Dynamics of Delayed Periodic Neural Networks. Neural Networks 17, 87–101 (2004)
19. Zheng, Y., Chen, T.: Global Exponential Stability of Delayed Periodic Dynamical Systems. Phys. Lett. A 322, 344–355 (2004)
20. Huang, X., Cao, J.: Almost Periodic Solution of Shunting Inhibitory Cellular Neural Networks with Time-Varying Delays. Phys. Lett. A 314, 222–231 (2003)
21. Lu, W., Chen, T.P.: Global Exponential Stability of Almost Periodic Solutions for a Large Class Of Delayed Dynamical Systems. Sci. Chin. Ser. A MAth. 48, 1015–1026 (2005)
22. Wang, L., Zou, X.: Exponential Stability of Cohen-Grossberg Neural Networks. Neural Networks 15, 415–422 (2002)
23. Chen, T., Rong, L.: Robust Global Exponential Stability of Cohen-Grossberg Neural Networks with Time Delays. IEEE Trans. Neural Networks 15, 203–206 (2004)
24. Grossberg, S.: Biological Competition: Decision Rules, Pattern Formation, and Oscillations. Proceedings of the National Academy of Sciences of the United States of Amercia 77(4), 2338–2342 (1980)
25. Grossberg, S.: Nonlinear Neural Networks Principles, Mechanisms, and Architectures. Neural Networks 1, 17–66 (1988)
26. Lin, W., Chen, T.P.: Positive Periodic Solutions of Delayed Periodic Lotka-Volterra Systems. Phys. Lett. A 334, 273–287 (2005)
27. Lu, W., Chen, T.P.: R_n^n-Global Stability of a Cohen-Grossberg Neural Network with Nonnegative Equilibria. Neural Networks 20, 714–722 (2007)
28. Forti, M., Nistri, P.: Global Convergence of Neural Networks with Discontinuous Neuron Activations. IEEE Trans. Circuits Syst.-1 50(11), 1412–1435 (2003)
29. Lu, W., Chen, T.P.: Dynamical Behavious of Cohen-Grossberg Neural Networks with Discontinuous Activation Functions. Neural Networks 18, 231–242 (2005)
30. Forti, M., Nistri, P., Papini, D.: Global Exponential Stability and Global Convergence in Finite Time of Delayed Neural Networks with Infinite Gain. IEEE Transactions on Neural Networks 16, 1449–1463 (2005)
31. Lu, W., Chen, T.P.: Dynamical Behaviors of Delayed Neural Network Systems with Discontinuous Activation Functions. Neural Computation 18, 683–708 (2006)
32. Papini, D., Taddei, V.: Global Exponential Stability of the Periodic Solution of the Delayed Neural Networks with Discontinuous Activations. Phys. Lett. A 343, 117–128 (2005)
33. Lu, W., Chen, T.P.: Almost Periodic Dynamics of a Class Of Delayed Neural Networks with Discontinuous Activations. Neural Computation 20, 1065–1090 (2008)
34. Filippov, A.F.: Differential Equations with Discontinuous Right-Hand Sides. Mathematics and Its Applications (Soviet Series). Kluwer Academic Publishers, Boston (1988)
35. Aubin, J.P., Frankowska, H.: Set-Valued Analysis. Birkhäusera, Boston (1990)

Research on the Application of Neural Network in Diaphragm Icing Sensor Fault Diagnosis

Zhen Zhang, Jie Zhang, Lin Ye, and Ying Zheng

Department of Control Science & Engineering,
Huazhong University of Science & Technology, Wuhan 430074, China
zhangzhen_hust@126.com

Abstract. As the core component of the Icing Detection System of aircrafts, the reliability of Diaphragm Icing Sensor is a key factor for the ice detection system to work normally. This paper makes use of Neural Network and Autoregressive Exogeneous Model (ARX) to set up the output prediction model of the diaphragm icing sensor. Compare the predicted output of the model with the actual output to diagnose sensor faults of the sensor. According to the data acquiring from our experiment platform of Diaphragm Icing Sensor, it has been proved that this method is effective for fault diagnosis of the Diaphragm Icing Sensor.

Keywords: Diaphragm icing sensor, Diaphragm icing sensor, Fault diagnosis.

1 Introduction

As the core component of the Icing Detection System of aircrafts, the Diaphragm Icing Sensor is a kind of safety equipment for airplane flight. However, due to the harsh environment of airplane flight, the sensitivity and linearity of the sensor may decline, severe zero drift also happens, causing frequent fault of the sensor. According to the application of the icing detection system, the reliability of the system mainly depends on the diaphragm sensor and the circuits. Therefore, in order to improve the reliability of detection system and ensure flight safety, it is of great significance to have research on the methods of the diaphragm icing sensor fault diagnosis.

Neural Network is widely applied into sensor fault diagnosis because of its distinguished advantages such as self-adapting, self-learning and dealing with complex model. Literature [1] and [2] investigate the fault diagnosis of sensor based on Neural Network Pattern Recognition and get favorable fault diagnosis through laboratory simulation. Literature [3] studies the fault diagnosis of sensor from the aspect of Neural Network Observer. According to the experimentation results, Neural Network is effective in sensor fault diagnosis.

Different from the sensor operating principle discussed above, since the accurate input and output of the diaphragm icing sensor is available (the operating principle of the diaphragm icing sensor is discussed in the next section), because of the excellent nonlinear mapping ability of Neural Network and the favorable anti-jamming capability of autoregressive exogeneous model(ARX), it is feasible to set up the input-output model of the diaphragm icing sensor based on Neural Network and ARX (NNARX) model. When sensor fault happens, information about it can be obtained from the

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 589–600, 2009.

residual between predicted output and the actual output of the model to have fault diagnosis of the sensor. Because NNARX is good at predicting and fast network convergence, the diaphragm icing sensor model can realize the on-line and off-line fault diagnosis of the sensor so as to provide solid technical support for reliable operation of the airplane ice detection system.

2 Operating Principle and Fault Modes of the Diaphragm Icing Sensor

2.1 Operating Principle of the Diaphragm Icing Sensor

The icing sensor in this study is a diaphragm sensor based on piezoelectric resonance principle. Piezoelectric resonance is a kind of electrical-mechanical transformation. The main element of the piezoelectric resonator component is diaphragm mechanical oscillator made from piezoelectric material. The piezoelectric resonant sensor is based on piezoelectricity. The input voltage from the electrodes is transformed into mechanical stress of the mechanical oscillator through converse piezoelectric effect; on the contrary, the deformation of the mechanical oscillator caused by mechanical stress produces output charge on the electrodes through piezoelectric effect. Electric actuating signals can be inputted to cause mechanical vibration, also electric signals in direct proportion to vibration amplitude is output. Generally, the sensor in this study is based on piezoelectric effect, converse piezoelectric effect and mechanical resonance principle.

This sensor utilizes a three-electrode diaphragm piezoelectric device. As shown in Fig.1,

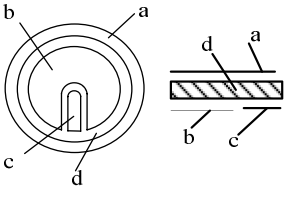


Fig. 1. Structure of the Piezoelectric Element

where electrode a is a metal plate, electrode b is another one, electrode c is a third electrode etched out of electrode b, d is piezoelectric element. When it is used as icing sensor, polar plate a is installed with support leg, the face of the polar plate is upwards for ice detection (see Fig. 2)

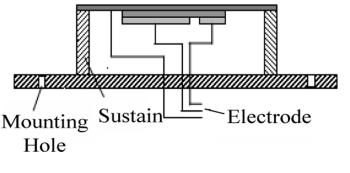


Fig. 2. Structure of the Sensor

When alternating voltage is added on polar plate a and b, the piezoelectric element will experience a radial expansion vibration. This system can be regarded as a signal-order mechanical vibration system.

$$f_0 = C\sqrt{k/m} \tag{1}$$

where the corresponding relation between its resonant frequency, effective mass m, and equivalent stiffness k is prescribed as Eq.(1), where C is a constant.

This kind of piezoelectric device is able to work as icing sensor because when there is no additional mass on plate b, the piezoelectric element does a radial expansion vibration at its own resonant frequency in coordination with certain sort of circuit. When icing occurs on the plate, which will greatly increase the stiffness k of the system, as we know from Eq. (1), the resonant frequency will also be increased. That is, as the thickness of ice increases, the stiffness k increases and eventually the resonant frequency increase. Therefore, the ice thickness will be measured if we can detect the resonant frequency.

Piezoelectric ceramics has both electricity and elasticity, which means it has inherent resonant frequency. As for certain piezoelectric resonant diaphragm, when the frequency of the excitation voltage deviates from resonant frequency, the current in the excitation electrode loop is small, while the frequency of excitation voltage approaching certain resonant frequency of the piezoelectric sensor, the mechanical vibration amplitude increases and reaches the peak value at the resonant frequency because of positive effect, then the charge on electrodes is scaling up.

While viewing the piezoelectric resonant diaphragm as a two-port network, if alternating voltage with frequency of f is inputted, charge will be produced on the electrodes through converse piezoelectric effect and piezoelectric effect, the charge can be amplified into voltage u, which is in proportion to the amplitude of mechanical vibration and the resonance voltage is measured. Thus, the resonance frequency will be obtained according to the vibration principle mentioned above, realizing the measurement of ice thickness.

2.2 Fault Modes of the Diaphragm Icing Sensor

After years of research and observation on the sensor, there are generally several fault modes of the diaphragm icing sensor, as shown in the output voltage u, constant, constant bias and significant mutation, etc.

(1) Constant output u, the sensitive diaphragm of the sensor fails to vibrate and the output of the sensor is constant C, this fault mode is constant output voltage u. Obviously, when constant output happens, the diaphragm icing detection system is almost useless, causing the diaphragm icing sensor out of control among the safety detection system of the airplane.

(2) Constant bias Δy, slow drift of the sensor happens because of the environment factors, that is, in a certain period of time (several hours or more), there is constant bias Δy in the output u of the sensor, and this fault mode leads to false alarm of the detection system.

(3) Significant mutation, if the surface of the diaphragm is polluted, the significant mutation fault mode can happen. In this case, the work of the detection system is extremely unstable, the output u is either far beyond the limits or far less than the allowable value, so the quality of the safe detection is damaged.

3 Neural Network Model of the Diaphragm Icing Sensor

By the discussion above, it is feasible to set up the prediction model between input f and output u of the diaphragm icing sensor. In this study, to adapt the dynamic characteristics of the diaphragm icing sensor under harsh environment during airplane flight, the model between input f and output u of the sensor is set up based on Autoregressive Exogeneous (ARX). ARX Model is one of the "Black-Box" methods; it can set up the sensor model with the input and output information of the system while the internal mechanism is not necessary. The system identification method of combining ARX Model and Neural Network takes advantage of the excellent non-linear mapping capability of Neural Network and the concept of time serials of ARX Model to ensure good dynamic characteristics and anti-jamming capability of the NNARX Model. With NNARX Model, the Diaphragm Icing Sensor Prediction Model is of favorable prediction ability.

3.1 NNARX Model Structure

A dynamic system can be represented by

$$y(t) = q^{-nk}G(q)u(t) + H(q)e(t)\,. \tag{2}$$

where $q^{-nk}G(q)u(t)$ term refers to noise-free output and $H(q)e(t)$ refers to disturbance term. q as an argument of $G(q)$ and $H(q)$ is the negative shift operator, which is equivalent to q^{-1} represented by q^{-nk} and can be demonstrated by $q^{-1}x(t) = x(t-1)$. nk is the time delay in sampling instant between the process input and the output.

The ARX model structure is given by

$$y(t) = q^{-nk}\frac{B(q)}{A(q)}u(t) + \frac{1}{A(q)}e(t)\,. \tag{3}$$

where the polynomials A(q) and B(q) are given by

$$\begin{aligned} A(q) &= 1 + a_1 q^{-1} + \ldots + a_{na} q^{-na} \\ B(q) &= b_0 + b_1 q^{-1} + \ldots b_{nb} q^{-nb} \end{aligned}\,. \tag{4}$$

In estimating the nonlinear counterpart of the ARX structure, the neural network can be utilized. The multilayer perceptron (MLP) is among of the popular neural network

structures especially in identification of a nonlinear system. The neural network version of ARX model structure is denoted as the neural network ARX (NNARX). Assuming the input delay $nk = 1$.The general NNARX model structure is as shown in the Fig.4.

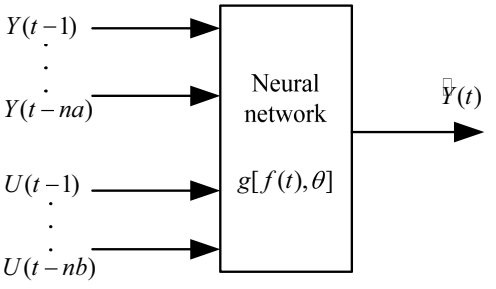


Fig. 3. The general NNARX model structure

The input-output relationship of NNARX model structure can be represented by

$$y(t) = g[\varphi(t), \theta] + e(t). \tag{5}$$

The one-step-ahead-prediction of the NNARX model structure is given by

$$\hat{y}(t \mid \theta) = g[\varphi(t), \theta]. \tag{6}$$

where $\varphi(t)$ is the regression vector, θ is the parameter vector, g is the function realized by the neural network, $e(t)$ is the noise, $y(t)$ is the system output and $\hat{y}(t \mid \theta)$ is the predicted output based on the parameter vector θ.
Considering Eq.(2) and for simplicity, assuming unity time delay, the 1-Step-Ahead-Prediction is given by

$$\hat{y}(t \mid t-1) = H^{-1}(q)G(q)u(t) + [1 - H^{-1}(q)]y(t). \tag{7}$$

3.2 Neural Network Prediction Model of the Diaphragm Icing Sensor

In this paper, diaphragm icing sensor neural network prediction model is a Three-layer neural network as in Fig. 4. The regressors are past frequency inputs and past sensor outputs; the transfer function in the hidden layer is the hyperbolic tangent function, as shown in Fig.5, the transfer function in the output layer is the sigmoid function, as shown in Fig.6. Since all the information data {y(k),(k)} can be made available as inputs to this model any non-linear system can be approximated well.

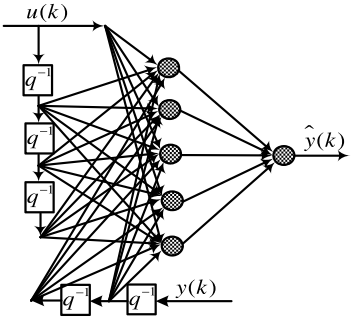


Fig. 4. Three-layer NNARX model of Diaphragm Icing Sensor

The advantageous and distinguishing feature of neural networks is their ability to learn. The network in the adaptive mode abstracts and generalizes the function character in the process of learning from training patterns. The learning algorithm is an optimization method capable of finding weight coefficients and thresholds for a given neural network and a training set. Because Levenberg-Marquardt back propagation(LMBP) method generally has less computational burden and faster convergence, This paper uses LMBP algorithm for training neural network load models in off-line as well as in real-time. Please refer to literature [4] for specific details of the LMBP algorithm.

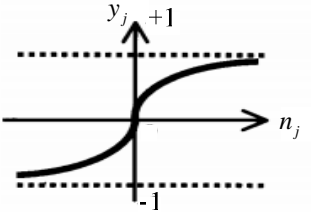


Fig. 5. Hyperbolic tangent transfer function

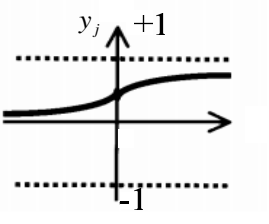


Fig. 6. Sigmoid transfer function

4 Fault Diagnosis of the Diaphragm Icing Sensor

4.1 Principle of Fault Diagnosis of the Diaphragm Icing Sensor

As for the fault modes of diaphragm icing sensor discussed above, the principle of diagnosis of diaphragm icing sensor based on neural network prediction model is as shown in Fig.7. Calculate the difference between the predicted output voltages $\hat{y}(k)$ and actual output $y(k)$ of the sensor at the moment of k and compare the residual deviation with certain detection threshold to decide the sensor fault.

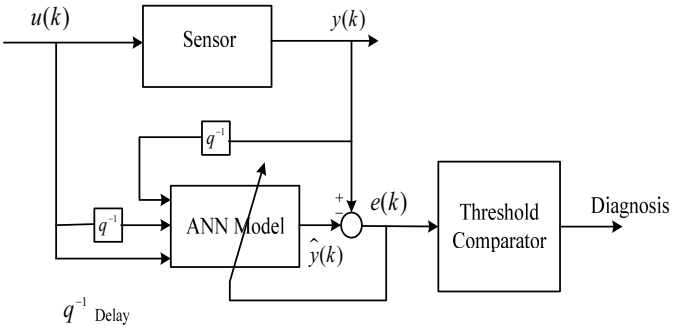


Fig. 7. The structure of Fault Diagnosis of the Diaphragm Icing Sensor

If the residual deviation between the predicted output voltages $\hat{y}(k)$ and actual output $y(k)$ of the sensor at the moment of k is less than the threshold, the sensor works normally, otherwise, sensor fault exists.

4.2 Threshold Design

Both the linear and nonlinear procedures described require a priori knowledge of faults to be detected; following the study of the statistics of the residual signal, a statistical approach has been followed in the design of the thresholds. The residual is a non-stationary stochastic process but does approximate to a Gaussian distribution [5].

Considering the variation of the residual e_i with input amplitudes, the mean and variance of the residual can be expressed according to the stochastic theory:

$$\eta = \frac{1}{n}\sum_{i=1}^{n} e_i \tag{8}$$

$$\delta^2 = \frac{1}{n-1}\sum_{i=1}^{n}(e_i - \eta)^2 \ . \tag{9}$$

From the statistical theory [6], the confidence limits of the mean that represent a confidence of (1-a) is

$$P\{\bar{\eta} - z\sigma < \eta < \bar{\eta} + z\sigma\} = 1 - a \ . \tag{10}$$

where a is the confidence level, and z is the coefficient related to the confidence level. The confidence (1-a) is typically selected to be 95–99% in practice. In this study, the confidence level is 97%, i.e. a=0.03 and then the coefficient z=2.17.

From Eq.(10), the threshold can be calculated as

$$\delta_{th} = \eta \pm 2.17\sigma . \tag{11}$$

In addition, the residual is also influenced by many factors; therefore, enabling the influence of both the model error and the random disturbances to be considered , it is suggested that the basic form of the threshold is

$$\delta_{th} = \Psi(\eta \pm 2.17\sigma) \tag{12}$$

where Ψ are coefficients modifying the mean and variance, in this paper, Ψ ,mean and variance are decided through experiment results.

5 Experiments and Results

5.1 Description of the Experiment Platform

In order to justify the fault diagnosis method of diaphragm icing sensor discussed in this study, we use diaphragm sensor experiment platform to collect experiment data. As shown in the Fig.8, in this experiment platform, the drive circuit and signal processing circuit produce the input excitation voltage *f* and collect and transform into the output voltage *u*, and then these data is transmitted through serial communication to the PC for analysis and display.

As for this experiment platform, one of the most important factors is the simulation of real icing condition during airplane flight. And under this requirement, the High-Low-Temperature Box and the Spray System play the key roles.

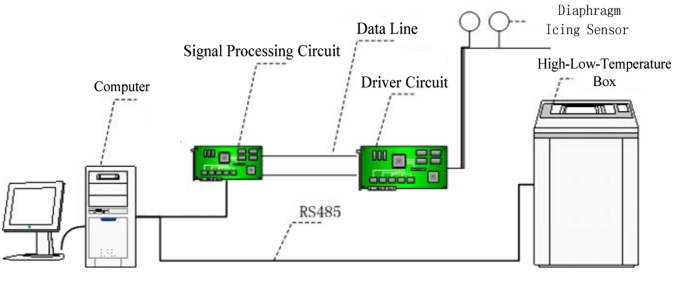


Fig. 8. Experiment Platform of Diaphragm Sensor

The four key elements of airplane icing are: Velocity (V), Temperature (T), Liquid Water Content (LWC) and Liquid Drop Mean Effective Diameter (MVD) [7], which determine the types and geometry of ice and can be regulated according to actual conditions. It is impossible to simulate all of the four elements under current laboratory situation; however, the T and MVD are relatively easy to be controlled. The principle of this experiment platform is to simulate the atmosphere icing through the adjustment of T and MVD while other environmental elements remain constant, which is generally justified by experiments.

Fig. 9. High-Low-Temperature Box

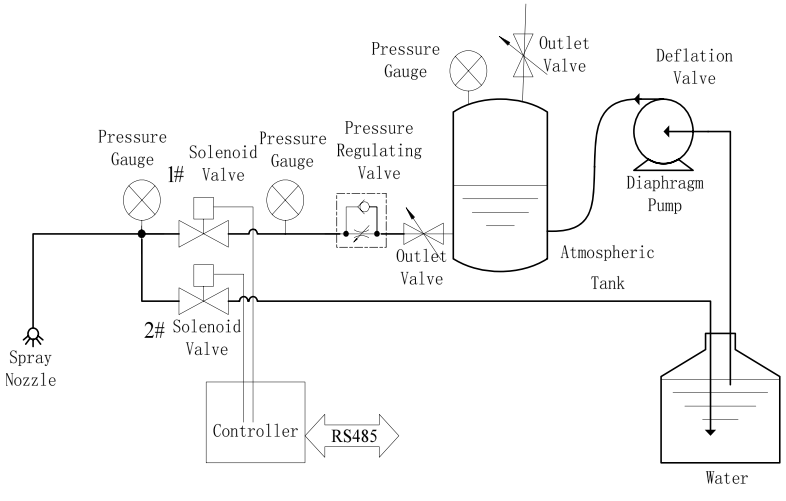


Fig. 10. Structure of the Spray System

On one hand, the control of temperature is realized with High-Low-Temperature Box(see Fig.9), which provides the condition of temperature between -70℃ and +150℃, meeting the temperature requirements. During the experiments, piezoelectric diaphragm icing sensor is placed in the box for low icing temperature.

On the other hand, the Spray System is able to control the MVD, which generally consists of diaphragm pump, pressure storage tank, and electromagnetic valve, pressure regulating valve, sprayer and controller,as shown in Fig.10. This system can produce effective tiny liquid drop for the diaphragm icing sensor to experience icing process.

5.2 Off-Line Learning Neural Network Fault Diagnosis

Take the off-line input and output data of the sensor without faults as input samples for neural network training. In order to research the drift fault diagnosis of the diaphragm icing sensor, add a deviation signal to the sensor's output. The learning

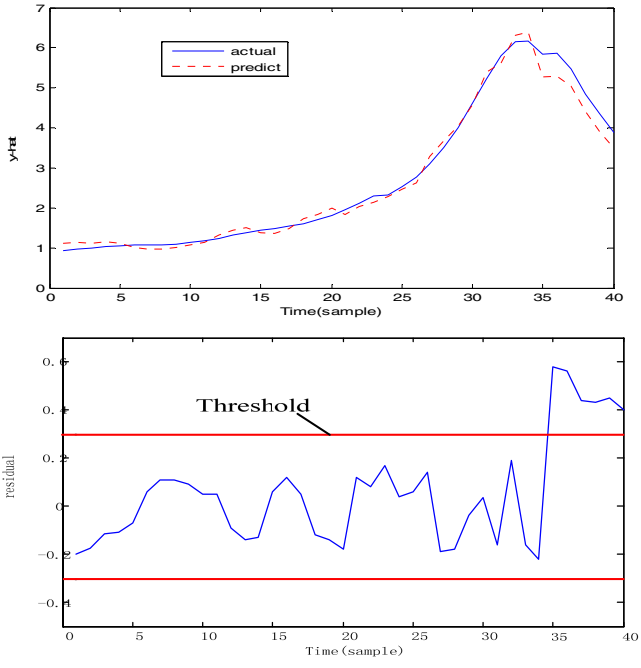


Fig. 11. Predicted Output and Residual Deviation of Off-line Learning Neural Network

precision can be very high because of the neural network off-line learning. According to Eq. (12), the threshold here is -0.3-0.3. As shown in Fig. 11. The residual of the 35th sampling point gets beyond the fault diagnosis threshold, meaning that sensor fault happens from the 35th sampling point.

5.3 On-Line Learning Neural Network Fault Diagnosis

First, take the first m sets of data of the sensor's input and output as input sample of the neural network until certain iteration times or the network converges to the expected precision. Then, predict the (m+1) output $\hat{y}(m+1)$ of the sensor and compare it with the actual output $y(m+1)$ of the sensor. If the residual is less than certain threshold, the sensor works normally; otherwise if the residual gets beyond the threshold, fault exists. The neural network convergence speed is fast enough to meet the real-time requirement. In order to research the drift fault diagnosis of the diaphragm icing sensor, add a deviation signal to the sensor's output after the neural network's training. Because the learning precision of on-line neural network is not as high as off-line neural network, the threshold here is -0.5-0.5 as a matter of experience. Fig.12 shows the sensor's bias fault diagnosis with on-line learning neural network predictor. As shown in the Fig.12, the residual deviation of the 34th sampling point gets beyond the fault diagnosis threshold, meaning that sensor fault happens from the 34th sampling point.

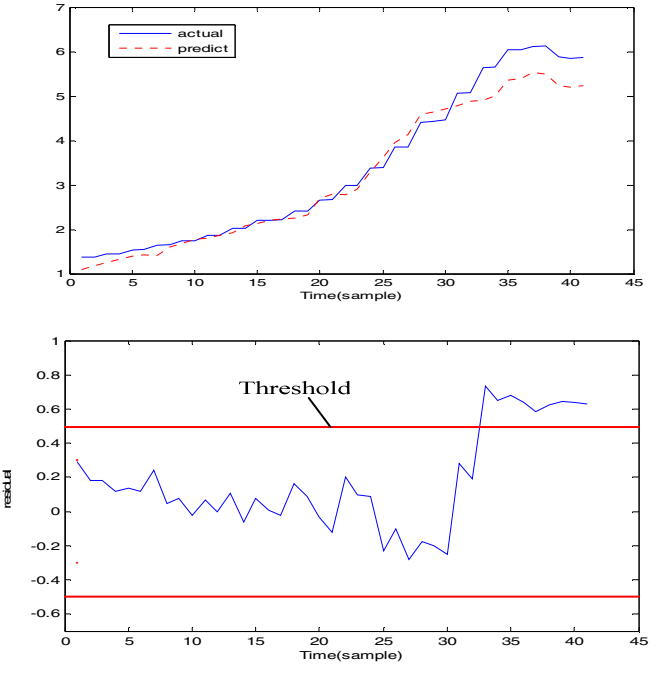


Fig. 12. Predicted Output and Residual Deviation of On-line Learning Neural Network

6 Conclusion

The diagnosis method in this study is proved to be effective with the experiments of the fault diagnosis of diaphragm icing sensor. The sensor fault diagnosis method discuss in this study has the following characteristics: ① only the sensor signal is needed to diagnose the sensor fault; ② not only the constant bias fault can be simulated as in this study, the method can also be applied into the diagnosis of other types of sensor faults such as drift, collision, hardware failure, etc.

Acknowledgements. The authors thank the financial support for this work from Chinese National Natural Science Foundation (10577008 and 60604030).

References

1. Jabbari, A., Jedermann, R., Lang, W.: Application of Computational Intelligence for Sensor Fault Detection and Isolation. J. International Journal of Computer, Information, and Systems Science, and Engineering 1, 142–147 (2007)
2. Chen, Y.M., Lee, M.L.: Neural Networks-Based Scheme for System Failure Detection and Diagnosis. Mathematics and Computers in Simulation 58, 101–109 (2002)

3. Calado, J.M.F., Korbicz, J., Patan, K., Patton, R.J., Sá da Costa, J.M.G.: Soft Computing Approaches to Fault Diagnosis for Dynamic Systems. European Journal of Control 7(2-3), 169–208 (2001)
4. Chen, T., Han, D., Au, F.T.K., Than, L.G.: Acceleration of Levenberg-Marquardt Training of Neural Networks with Variable Decay Rate. IEEE Trans. on Neural Network 3, 1873–1878 (2003)
5. Bhattacharya, R., Waymire, E.C.: Stochastic Processes with Applications. Wiley, New York (1990)
6. Wang, X., Kruger, U., Lennox, B.: Recursive Partial Least Square Algorithms for Monitoring Complex Industrial Processes. Control Engineering Practice 11, 613–632 (2003)
7. Lynch, F.T., Khodadoust, A.: Effects of Ice Accretions on Aircraft Aerodynamics. Progr. Aerospace Sci. 37, 669–767 (2001)

The Periodic Solution of a Class of Two Neurons Hopfield Network with Distributed Delay

Zhaogang Xiong, Wei Xu, and Boshan Chen

School of Mathematics and Statistics, Hubei Normal University, Huangshi, Hubei, 435002, China

Abstract. This paper considers the two neurons Hopfield network with distributed delay, which can be transformed into the three-dimensional differential system. Then we use the result of competitive system to get the periodic solution of the system. We also show that existence and uniqueness of equilibrium point, and if the equilibrium is unstable, then an orbitally asymptotically stable periodic solution exists.

Keywords: Neural network, Equilibrium point, Periodic solution, Competitive system, Distributed delay.

1 Introduction

The study of neural networks began with the early work of Hopfield who used an electronic circuit implementation of such a network. In recent years, to understand better the dynamical behavior of neural networks, various complications have been included [1]-[7]. A typical model can be expressed as the following the two neurons Hopfield network

$$\begin{cases} \dot{x}_1(t) = -a_1 x_1(t) + b_{11} f_1(x_1(t)) + b_{12} f_2(x_2(t)) + I_1 \\ \dot{x}_2(t) = -a_2 x_2(t) + b_{21} \int_{-\infty}^{t} \delta f_1(x_1(\tau)) e^{-\delta(t-\tau)} d\tau + b_{22} f_2(x_2(t)) + I_2 \end{cases} \tag{1}$$

where $x_i(t)$ is the state of the ith neuron, a_i, b_{ij} $(i, j = 1, 2)$ are the connection weight constants, $\delta > 0$, and I_i is an external input(bias) to the ith neuron, $i = 1, 2$. In this letter, we assume that the activation functions $f_i(\cdot) \in C(i = 1, 2)$ satisfy

$$f_i(\xi) = \begin{cases} u_i & -\infty < \xi < p_i \\ \bar{f}'(\xi) & p_i \le \xi \le q_i \\ v_i & q_i < \xi < \infty, \end{cases}$$

where p_i, q_i, u_i, v_i are constant. The initial conditions associated with the model (1) are of the form

$$x_{it_0}(\theta) = \varphi_i(t_0 + \theta) \text{ for } \theta \in (-\infty,\ 0] \text{ and } \varphi_i \in C((-\infty,\ 0], R). \tag{2}$$

Let

$$x_3(t) = \int_{-\infty}^{t} \delta f_1(x_1(\tau)) e^{-\delta(t-\tau)} d\tau.$$

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 601–606, 2009.

The system (1) can be transformed into the following differential equation.

$$\begin{cases} \dot{x}_1(t) = -a_1 x_1(t) + b_{11} f_1(x_1(t)) + b_{12} f_2(x_2(t)) + I_1 \\ \dot{x}_2(t) = -a_2 x_2(t) + b_{21} x_3(t) + b_{22} f_2(x_2(t)) + I_2 \\ \dot{x}_3(t) = \delta f_1(x_1(t)) - \delta x_3(t) \end{cases} \tag{3}$$

We understand the relationship between the systems (1) and (3) as follows.

If $(x_1(t), x_2(t)) : (-\infty, \infty) \to R^2$ is the solution of (1) corresponding initial conditions (2), then $(x_1(t), x_2(t), x_3(t)) : (-\infty, \infty) \to R^3$ is a solution of (3) with initial conditions

$$x_1(t_0) = x_1(t_0),\ x_2(t_0) = x_2(t_0),\ x_3(t_0) = \int_{-\infty}^{t_0} f_1(\varphi_1(s)) e^{\delta(s-t_0)} ds.$$

In this paper, we shall consider the system (3) with the result of competitive system [8]-[15] to get the periodic solution.

2 Positively Invariant Sets and Equilibrium Points

The cube

$$D_1 = \{(x_1, x_2, x_3) : ((b_{11} + b_{12})u_1 + I_1)/a_1 < x_1 < ((b_{11} + b_{12})v_1 + I_1)/a_1;$$

$$((b_{21} + b_{22})u_2 + I_2)/a_2 < x_2 < ((b_{21} + b_{22})v_2 + I_2)/a_2; \quad u_1 < x_3 < v_1\}$$

is positively invariant set of (3), that is, any solution of (3) at time t_0 which lies inside D_1 always remains there for all $t > t_0$, since the vector field points strictly inward on the boundary of D_1.

We first prove that there exists a unique equilibrium point in the invariant set under certain condition, then analysis the local stability of the system. At first, we make a hypothesis:

H(1) : $(b_{ij} v_i + b_{ij} v_i)/a_i + I_i < q_i$, and $(b_{ij} u_i + b_{ij} u_i)/a_i + I_i > p_i$. $i, j = 1, 2$.

Theorem 1. If H(1) holds, then the system (3) has a unique equilibrium point that locates in the invariant set D_1.

3 Asymptotic Stability

In this part, we consider the equilibrium point of the system (3) in the positive invariant set, and we shall analysis the stability of the equilibrium point in the D_1.

If $E^* = (x_1^*, x_2^*, x_3^*)$ is the equilibrium point of system (3) in D_1, then the Jacobian matrix $J^* = J(x_1^*, x_2^*, x_3^*)$ of the system (3) at E^* takes the form of

$$\begin{pmatrix} -a_1 + b_{11} f_1'(x_1^*) & b_{12} f_2'(x_2^*) & 0 \\ 0 & -a_2 + b_{22} f_2(x_2^*) & b_{21} \\ \delta f_1'(x_1^*) & 0 & -\delta \end{pmatrix}$$

The characteristic equation of this Jacobian matrix is

$$\lambda^3 + Q_1\lambda^2 + Q_2\lambda + Q_3 = 0$$

where the coefficients

$Q_1 = a_1 + a_2 + \delta - b_{11}k_1 - b_{22}k_2$,

$Q_2 = (a_1 - b_{11}k_1)(a_2 - b_{22}k_2) + (a_1 + a_2 - b_{11}k_1 - b_{22}k_2)\delta$,

$Q_3 = (a_1 - b_{11}k_1)(a_2 - b_{22}k_2)\delta - b_{12}b_{21}k_1k_2\delta$,

and $k_1 = f_1'(x_1^*)$, $k_2 = f_2'(x_2^*)$.

Note that $Q_1 > 0$, $Q_3 > 0$ if

H(2): $a_1 + a_2 + \delta > b_{11}k_1 + b_{22}k_2$ and $(a_1 - b_{11}k_1)(a_2 - b_{22}k_2) > b_{12}b_{21}k_1k_2$. Furthermore,

$$\begin{aligned}\Delta &= Q_1Q_2 - Q_3 \\ &= (a_1+a_2+\delta-b_{11}k_1-b_{22}k_2)[(a_1-b_{11}k_1)(a_2-b_{22}k_2)+(a_1+a_2-b_{11}k_1-b_{22}k_2)\delta] \\ &\quad - (a_1 - b_{11}k_1)(a_2 - b_{22}k_2)\delta + b_{12}b_{21}k_1k_2\delta\end{aligned}$$

By the Routh-Hurwitzcriterion, we know that E^* is locally asymptotically stable if H(2) holds and

$$(a_1 + a_2 + \delta - b_{11}k_1 - b_{22}k_2)[(a_1 - b_{11}k_1)(a_2 - b_{22}k_2) + (a_1 + a_2 - b_{11}k_1 - b_{22}k_2)\delta]$$

$$> (a_1 - b_{11}k_1)(a_2 - b_{22}k_2)\delta - b_{12}b_{21}k_1k_2\delta \tag{4}$$

4 Main Facts on Three-Dimensional Competitive System

In this section, we will summarize the main facts related to our research. Let us consider the system of differential equations

$$\dot{X} = F(X), X \in D, \tag{5}$$

where D is an open subset on R^3 and F is twice continuously differentiable in D. The solution of (5) satisfying $X(t_0) = X_0$ is denoted as $X(t, X_0)$. The positive (negative) semi-orbit through X_0 is denoted as $\varphi^+(X_0)(\varphi^-(X_0))$, and the orbit through X_0 is denoted as $\varphi(0) = \varphi^+(X_0) \cup \varphi^-(X_0)$. We use the notation $\omega(X_0)(\alpha(X_0)$ to denote the positive (negative) limit set of $\varphi^+(X_0)(\varphi^-(X_0))$.

System (3) is competitive, if for some diagonal matrix $H = diag(\epsilon_1, \epsilon_2, \epsilon_3)$, where ϵ_i is either 1 or -1, H(DF(X))H has nonpostive off-diagonal elements for $X \in D$, where DF(X) is the Jacobian matrix of (3). It is shown in [9] that if D is convex, then the flow of such a system preserves for $t < 0$ the partial order in R^3, which is defined by the orthant

$$K_1 = \{(X_1, X_2, X_3) \in R^3 : \epsilon_i X_i \geq 0\}$$

Hirsch and Smith proved that three-dimensional competitive systems that live in convex sets have the Poincare-Bendixson property; that is, any nonempty compact omega limit set that contains no equilibria must be a closed orbit.

Theorem 2. Let (3) be a competitive system in $D_1 \subset R^3$ and D_1 contains a unique equilibrium point X^*, which is hyperbolic and assume that $DF(X^*)$ is

irreducible. Suppose further that $W^s(X^*)$, the stable manifold of X^*, is one dimensional. If $q \in D \setminus W^s(X^*)$ and $\phi^+(q)$ has compact closure in D, then $\omega(q)$ is a nontrivial periodic orbit.

We introduce the following hypotheses:

(S1) System (3) is dissipative: For each $X \in D_1$, $\varphi^+(X_0)$ has compact closure in D. Moreover, there exists a compact subset B of D with property that for each $\overline{X} \in D$ there exists $T(\overline{X}) > 0$ such that $X(t, \overline{X}) \in B$ for $t \geq T(\overline{X})$.

(S2) System (3) is competitive and irreducible in D_1.

(S3) D_1 is an open, p-convex subset of R^3.

(S4) D_1 contains a unique equilibrium point X, and $det(DF(X^*)) < 0$.

The following result holds [12]:

Theorem 3. Let (S1)-(S4) hold. Then one of following holds:
(a) X^* is stable
(b) there exists a nontrivial orbitally stable periodic orbit in D_1.

In addition, let us assume that F is analytic in D_1. If X^* is unstable, then there is at least one but no more than finitely many periodic orbits for (3) and at least one of these is orbitally asymptotically stable.

Remark 1. It is clear that if the system (3) is a competitive system, the system

$$\frac{dx}{d\tau} = -F(x)$$

where $-\tau = t$, is cooperative system.

5 Existence of a Stable Periodic Orbit

Out main result below gives sufficient conditions that almost every solution is asymptotically periodic.

Theorem 4. Let $\delta > 0$, then the equilibrium point of (3) is locally asymptotically stable if H(1) and H(2) hold. There exists a one-dimensional stable manifold $W^s(E^*)$ if (4) is reversed. Furthermore, there exists an orbitally asymptotically stable periodic orbit, and the omega limit set of every solution $(x_1(t), x_2(t), x_3(t))$ with their initial values and $(x_1(t_0), x_2(t_0), x_3(t_0)) \notin W^s(E^*)$ is a nonconstant periodic orbit.

Proof. We apply Theorems 2 and 3 to the following transform system. By looking at its Jacobian matrix and choosing the matrix H as

$$H = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

we can see that system (3) is competitive in D_1, with respect to the partial order defined by the orthant $K_1 = \{(X_1, X_2, X_3) \in R^3 | X_1 \geq 0, X_2 \geq 0, X_3 \leq 0\}$. (x_1^*, x_2^*, x_3^*) in R^3 is an equilibrium point of system (3), since the inequality H(2) is reversed.

The analysis shows that (x_1^*, x_2^*, x_3^*) is unstable and $det(DF(X^*)) = -Q_3 < 0$. Furthermore, we see that the stable manifold of X^* is one dimensional. The existence of an orbitally asymptotically stable periodic orbit follows from Theorem 3. Note that S(1)-S(4) hold. Theorems 2 and 3 imply the final assertion.

6 Numerical Examples

In this section, we give two examples to illustrate the conditions required in our theorems.

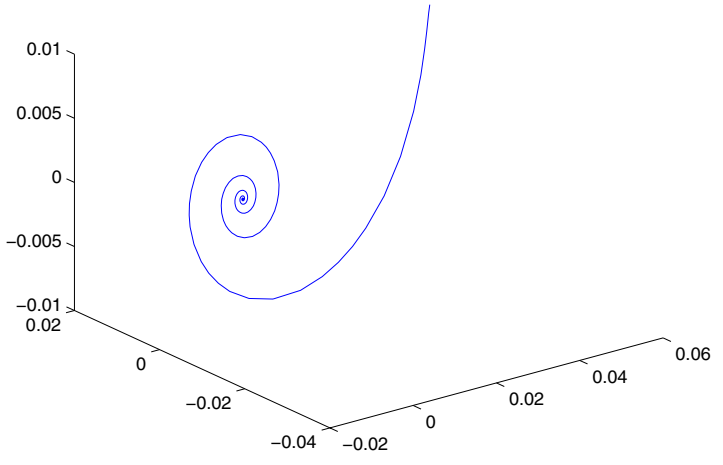


Fig. 1. The positive equilibrium point $p = (0.1601, 0.04, 0.1281)$ of system (1) is locally asymptotically stable in Example 1

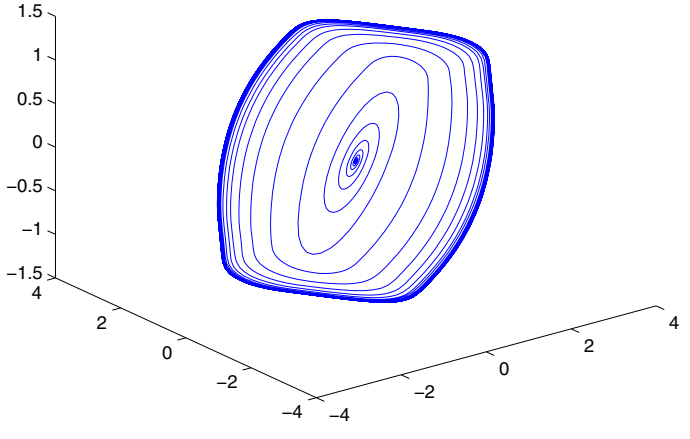


Fig. 2. Periodic orbit in Example 2

Example 1. Consider the system (1) with $a_1 = 5$, $a_2 = 1$, $a_{11} = b_{22} = 0$, $a_{12} = 1$, $a_{21} = -20$, $f(x(t)) = (|(x(t) + 1)| - |(x(t) - 1)|)/2$, $I_1 = 0.000001$, $I_2 = 0.00000002$, $\delta = 1$. Then, the equilibrium point of the system (1) is locally asymptotically stable, see Fig. 1.

Example 2. Consider the system (1) with $a_1 = 0.004$, $a_2 = 0.005$, $a_{11} = b_{22} = 0$, $a_{12} = 0.2$, $a_{21} = -0.3$, $f(x(t)) = (|(x(t) + 1)| - |(x(t) - 1)|)/2$, $I_1 = 0$, $I_2 = 0$, $\delta = 1$. Then, there exists an orbitally asymptotically stable periodic orbits in the system (1), see Fig. 2.

Acknowledgments. We thank Professor Boshan Chen and the referees for their careful reading of the original manuscript and for their many valuable comments and suggestions that improved the presentation of this paper.

References

1. Grossberg, S.: Nonlinear Neural Networks: Principles, Mechanisms and Architectures. Neural Networks 1, 17–61 (1988)
2. Hirsch, M.W.: Convergent Activation Dynamics in Continuous Time Networks. Neural Networks 2, 331–349 (1989)
3. Wang, D.L.: Emergent Synchrony in Locally Coupled Neural Oscillators. IEEE Trans. Neural Networks 6, 941–948 (1995)
4. Cohen, M.A., Grossberg, S.: Absolute Stability and Global Pattern Formation and Parallel Memory Storage by Competitive Neural Networks. IEEE Trans. Syst., Man Cybern. 7, 815–821 (1983)
5. Gedeon, T.: Structure and Dynamics of Artificial Neural Networks: Differential equations with Applications to Biology. Fields Inst. Commun. 21, 217–224 (1999)
6. Arik, S.: Global Asymptotic Stability of A Class of Dynamical Neural Networks. IEEE Trans. Circuits Syst. I 47, 568–571 (2000)
7. Chen, B.: Mixed Monotone Semiflows and Stability for Functional Differential Equations. Acta Mathematica Sinica 38, 267–273 (1995) (in Chinese)
8. Cavani, M., Lizana, M., Smith, H.L.: Stable Periodic Orbit for A Predator-prey Model with Delay. J. Math. Anal. Appl. 249, 324–339 (2000)
9. Hirsch, M.W.: Systems of Differential Equations which are Competitive or Cooperative. SIAM J. Math. Anal. 21, 1225–1234 (1990)
10. Smith, H.L., Zhu, H.R.: Stable Periodic Orbits for A Class of Three-dimensional Competitive Systems. J. Differential Equations 110, 143–156 (1994)
11. Smith, H.L.: Monotone Dynamical Systems: An Introduction to the Theory of Competitive and Cooperative Systems. Trans. Amer. Math. Soc. 41, 143–156 (1995)
12. Tang, S.Y., Chen, L.S.: Global Qualitative Analysis for A Ratio-Dependent Predator-Prey Model with Delay. J. Math. Analysis and Applications 266, 401–419 (2002)
13. LaSalle, J.P.: The Stability of Dynamical System. SIAM, Philadelphia (1976)
14. Chen, B., Wang, J.: Global Exponential Periodicity and Global Exponential Stability of A Class of Recurrent Neural Networks. Physics Letters A 329, 36–48 (2004)
15. Smith, H.: Monotone Semiflows Generated by Functional Differential Equations. J. Diff. Equs. 66, 420–422 (1987)

A Hybrid MPSO-BP-RBFN Model for Reservoir Lateral Prediction

Shiwei Yu, Kejun Zhu, Xiufu Guo, and Jing Wang

School of Economics and Management, China Universiyt of Geosciences,Wuhan, 430074 Hubei, P.R. China
ysw81993@sina.com, zhukejun@cug.edu.cn, xf_Guo@126.com, wj_happy163@163.com

Abstract. The degree of success of many oil and gas drilling, completion, and production activities depends on the accuracy of the models used in the reservoir lateral prediction and description. In this paper, a hybrid MPSO-BP-RBFN model for predicting reservoir from seismic attributes is proposed. The model in which every particle consists of binary and real parts is able to simultaneously search for optimal network topology (the number of hidden nodes) and parameters, as it proceeds. The model has been used to reservoir lateral prediction of a reservoir zone and proved the model's applicability.

Keywords: Reservoir lateral prediction; Seismic attributes; Particle swarm optimization; Multi-encoding; Radial basis function neural networks; Adaptive.

1 Introduction

Reservoir lateral prediction is also one important contents of the reservoir characterization, especially in the cross-cutting nature of the reservoir area; it provides an accurate basis for computing proved reserves. Lateral reservoir prediction can also provide a basis for developing programs. As the seismic data contains very rich reservoir information, and more continuity than drilling information, the reservoir lateral prediction are based mainly on seismic information, which can more accurately reflect the reservoir lithology, physical properties and characterization of oil and gas. In the application of seismic data for reservoir prediction, operators usually use conventional statistics methods to establish the relationship between the seismic attributes [1-3] and reservoir characteristics, and then precede lateral reservoir prediction based on the relationship. Besides statistical methods, artificial neural networks (ANN), or more specifically, multilayer perceptions (MLP), have become increasingly popular in reservoir prediction [4-7]. This intelligent technique is non-linear and non-parametric, and has been applied to reservoir prediction from seismic attributes. Radial basis function networks (RBFN) is a popular technique of ANN. As a simple structure, well established theoretical basis and fast learning speed, RBFN became a popular technique since the 1980s in real applications [8-11]. However, there are still some difficulties with building RBFN. One of the main problems with RBFN is determining the number of radial basis functions. Another problem is how to get the parameters such as the centers, widths and the weights more efficient. Different approaches can be

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 607–616, 2009.

categorized as follows: (1) two-phase approach. This is most popular training technique, which use a hybrid training strategy, using an unsupervised algorithm to pick the centers, followed by a supervised one to obtain the output weights [12-13]. The proper network structure is selected by trial and error, since the k-means technique does not provide any rational means for choosing the number of hidden nodes. (2) Online RBFN self-generate method which the structure of network determines automatically by training process. Most of these methods grow the final net from an initially empty one to which hidden neurons are added until a given condition is reached. One of such methods is orthogonal least squares (OLS) algorithm [14]. Although this algorithm is widely used, Shwrstinsky[15] studied the algorithm from the perspective of the energy compression, found that OLS algorithm would not be able to design a structure with minimal network. (3) Evolutionary computation techniques, in which the network structure and parameters are selected simultaneously by employing optimization methods based on genetic algorithms [16]. In recent years, considerable progress has been made in evolutionary strategies; especially Particle swarm optimization (PSO) algorithm which is used to train artificial neural network as a global search method, and has few parameters to adjust and is easier to implement compared with other stochastic methods.

The objective of this paper is to propose a new methodology for predicting reservoir from seismic attributes using our integrated multi-encoding particle swarm optimization, back propagation, and radial basis function networks (MPSO-BP-RBFN). The proposed method can self-adapt to the network structure and updates its weights according to the performance of the networks. With the advantages of global optimization ability of MPSO and the rapid local approximation of BP algorithm, the new algorithm fully shows the ability of nonlinear approach of multilayer feed forward network, improves the performance of RBFN, and successful for lateral reservoir prediction.

2 Introduction to PSO

2.1 Background of PSO

PSO developed by Kennedy and Eberhart [17] is a stochastic global optimization technique inspired by social behavior of bird flocking. The algorithm models the exploration of a problem space by a population of individuals or particles. In PSO, each single solution is a particle in the search space. During the flight, every particle adjusts its position according to its own experience, as well as the experience of neighboring particles, using the best position encountered by itself and its neighbors. The swarm direction of a particle is defined by its history experience and the experience of its neighbors. A particle status on the search space is characterized by two factors: its position and velocity. Given that a swarm consists of m particles in a D-dimensional problem space, the position and velocity of the i th particle is presented as:

$$x_i = (x_{i1}, x_{i2}, \cdots, x_{iD}), \quad i = 1,2,\cdots,m,$$

$$v_i = (v_{i1}, v_{i2}, \cdots, v_{iD})$$

$P_i^t = (p_{i1}^t, p_{i2}^t, \cdots, p_{iD}^t)$ represents the best previous position of particle i has obtained until iteration t; $P_g^t = (p_{g1}^t, p_{g2}^t, \cdots, p_{gD}^t)$ represents the best position obtained from P_i^t in the swarm (gbest) or local neighborhood (lbest) at iteration t.

The continuous PSO algorithm is formulated as in Eq. (1) and Eq. (2).

$$v_{id}^{t+1} = w \cdot v_{id}^t + c_1 \times rand_1 \times (P_{id}^t - x_{id}^t) + c_2 \times rand_2 \times (P_{gd}^t - x_{id}^t) \tag{1}$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \tag{2}$$

where, v_{id}^{t+1} is the velocity of the i th particle at the t th iteration, c_1 and c_2 are positive constant parameters called acceleration coefficients that control the maximum step size, and w is called the inertia weight that controls the impact of the previous velocity of the particle on its current $rand_1$ and $rand_2$ are two independently distributed random variables with range [0,1]; x_{id}^t is the current position of the i th particle.

To ensure the convergence of the search, Clerc [18] introduced a constriction factor into the standard PSO algorithm. Eq. (1) being converted into:

$$v_{id}^{t+1} = k \cdot \left[v_{id}^t + c_1 \times rand_1 \times (P_{id}^t - x_{id}^t) + c_2 \times rand_2 \times (P_{gd}^t - x_{id}^t) \right] \tag{3}$$

$$k = \frac{2}{\left| 2 - \varphi - \sqrt{\varphi^2 - 4\varphi} \right|} \tag{4}$$

Where $\varphi = c_1 + c_2, \varphi > 4$.

From Eq. (4), k is constricted by c_1 and c_2. Due to k, there is no need of the maximum search velocity $v_{\max}$ and the search convergence is ensured mathematically. In other words, the vibration amplitude of the particle decreases when it is near to the best position. But Eberhart and Shi [19] found constricted $V_{\max} = X_{\max}$ can obtain better performance. Obviously, the constriction factor in the PSO algorithm can produce the solution better than that of the standard PSO.

2.2 Background of PSO

In order to optimize featuring discrete or qualitative distinctions between variables , Kennedy and Eberhart [20] developed a discrete version of PSO. Discrete PSO essentially differs from the original (or continuous) PSO in two characteristics. First, the particle is composed of the binary variable. Second, the velocity must be transformed into the change of probability, which is the chance of the binary variable taking the value one.

By Eq. (4), each particle moves according to its new velocity. Recall that particles are represented by binary variables. For the velocity value of each bit in a particle, Kennedy and Eberhart claim that higher value is more likely to choose 1, while lower value favors the 0 choice. Furthermore, they constrain the velocity value to the interval [0, 1] by using the following sigmoid function:

$$s(v_{id}^t) = \frac{1}{1 + \exp(-v_{id}^t)}, \tag{5}$$

Where $s(v_{id}^t)$ denotes the probability of bit x_{id}^t taking 1. The velocity update formula of discrete PSO is the same with continuous PSO, namely in Eq. (1) or (3); where the position is formulated as:

$$\text{If } (rand() < s(v_{id}^n)) \text{ then } x_{id}^{n+1} = 1; \quad \text{Else } x_{id}^{n+1} = 0 \tag{6}$$

$rand()$ is a random variable with range [0, 1].

Since the stochastic PSO algorithm has been found to be able to find the global optimum with a large probability and high convergence rate [21, 22], it is adopted to train the RBFN in this study.

3 Design of the Model to PSO

The goal of the multi-encoding PSO-BP-RBFN based evolutional learning method is to choose appropriate RBFN as well as to use the fewest number of radial basis functions. First of all, design as followings must be done in this section.

3.1 Multi-encoding

In PSO one particle is corresponding to a solution of a problem to solve. So every particle in this algorithm should include the information of the structure and parameters of the network, namely the number of hidden nod hid_no, the center c_i and the width σ_i of each hidden unit as well as the weight value w_i. Divide the whole particle to two parts $part = [par_1, par_2]$ which binary part par_1 corresponding to each hidden node and real part par_2 corresponding to, c_i, σ_i and w_i. If the max number of hidden nod is $\max\_nod$, the input vector X is n dimensions, the output number is l, then total length of every particle is $m \cdot (n + l + 1)$. If the value coded in binary system, is "1," the neuron is selected, on the contrary, if the value is "0," the neuron is unselected. For example, let the max hidden nod is 7, the masking binary string encoded

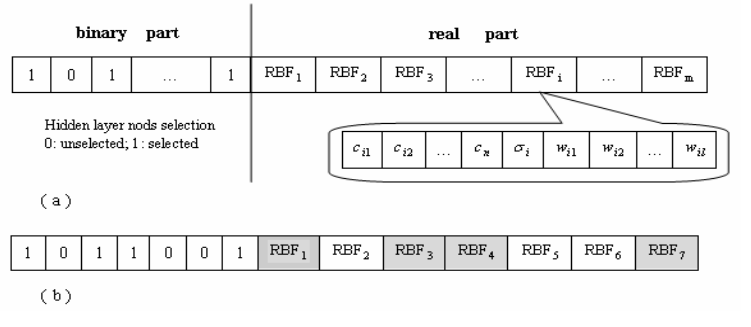


Fig. 1. (a). RBFN particle structure; (b). an example of 4 nods and corresponding RBF parameters selected

as [1 0 1 1 0 0 1] presents that the 1, 3,4and 7 hidden nods should be kept, and the others nods should be removed. The real part codes the weights and bias parameters which may be in values of [-1, 1]. Therefore, the structure of a particle is and an example of 4 hidden nods and corresponding RBF parameters selected are shown in Fig.1.

3.2 Fitness Function

To train the RBFN structure is to make it have the simplest network form at an allowable accuracy. It means that we let the accuracy and complexity of the network attains the minimum. Thus, a fitness value is given by Eq. (7).

$$fitness = E + \alpha \cdot hide\_nod \tag{7}$$

Where $E = \frac{1}{2}\sum_{j=1}^{n}(y^d(x^{(j)}) - f(x^{(j)}))^2$ means the actual error between the desired output $y^d(x^{(j)})$ and the actual output $f(x^{(j)})$ of the RBFN, which is determined by the individual parameter set S. It's should point out, in Eq. (7), E may be instead by other network performance function such as recognition rate of a model recognition problem in some applications. The complexity of the network, which is decided by the number of hidden nodes $hide\_nod$, while $\alpha, (0 < \alpha < 0.01)$ is an influential coefficient of the number of hidden layer nods. Furthermore, one hidden layer nod must be selected at least in every particle, namely, $hide\_nod \geq 1$.In this manner, the proposed MPSO-BP algorithm can find the best RBFN which approach higher accuracy (i.e. smaller E) with fewer number of radial basis functions with the guidance of the proposed fitness function corresponding to the same input. Therefore, the RBFN with very good performance can be automatically generated by the multi-encoding PSO method as compared with the time-consuming, traditional trial-and error methods used to determining appropriate RBFN parameters.

4 The Algorithm Approach of Adaptive Optimize Parameters of RBFN

The simultaneous optimization training method of the topology structure and parameters of the network are used, in which the Gaussian function is adopted as the radial basis function. More details are described as following and illustrated by the flowchart in Fig. 2.

(1) Initialize prior parameters of the model: the size of population pop_size, the max generation $\max\_g$ and $\max\_k$, the max number of hidden nod $\max\_nod$,and the epoch of BP learning BP_epoch.

(2) Randomly generate initial particles $part = [par_1, par_2]$ which randomly initialize particle positions of binary variables in par_1 and continuous real variables in

par_2 .Particle velocities are also initialized by continuous real variables in $[0, v_{\max}]$.

(3) If $g \le \max\_g$, evaluate objective function *ftness* for all particles, else go to step 9

(4) Update the personal best value $par_1\_P_{iD}$, $par_2\_P_{iD}$ for binary part par_1 and real part par_2 of particles, respectively. The global best value P_{gD} is also renewed for all particles.

(5) Renew position and velocity for par_1 using Eq. (4) and (5).

(6) In order to get the optimal network parameters c_i, σ_i, w_i based on a fixed structure, optimize the real part of particle using continuous PSO in further.

(6.1) Let $k = 1$, if $k \le \max\_k$, evaluate objective function *fitness* for all particles, otherwise go to step 7.

(6.2) Update the personal best value $par_2\_P_{iD}$ for real part par_2 of particles and the global best value P_{gD} .

(6.3) Renew position and velocity for par_2 by using Eq. (5) and (8).

(6.4) $k = k + 1$ and go to step (6.1).

(7) Back propagations algorithms are used to adjust the position of par_2 ,namely parameters c_i, σ_i, w_i of RBFN by following formulas:

$$\begin{aligned} c_i(t) &= c_i(t-1) + \eta_1 * \Delta c_i \\ &= c_i(t-1) + \eta_1 \left(-\sum_{j=1}^{n} (y^d(x^{(j)}) - f(x^{(j)})) * w_i * \exp\left[-\frac{\left\|x^{(j)} - c_i\right\|^2}{2\sigma_i^2}\right] * \frac{1}{\sigma_i^2} * (x^{(j)} - c_i)\right) \end{aligned} \tag{10}$$

$$\begin{aligned} \sigma_i(t) &= \sigma_i(t-1) + \eta_2 * \Delta\sigma_i \\ &= \sigma_i(t-1) + \eta_2 \left(-\frac{1}{\sigma_i^3} \sum_{j=1}^{n} (y^d(x^{(j)}) - f(x^{(j)})) * w_i * \exp\left[-\frac{\left\|x^{(j)} - c_i\right\|^2}{2\sigma_i^2}\right] * \left\|x^{(j)} - c_i\right\|^2\right) \end{aligned} \tag{11}$$

$$w_i(t) = w_i(t-1) + \eta_3 * \Delta w_i = w_i(t-1) + \eta_3 \left(\sum_{j=1}^{n} (y^d(x^{(j)}) - f(x^{(j)})) * \exp\left[-\frac{\left\|x^{(j)} - c_i\right\|^2}{2\sigma_i^2}\right]\right) \tag{12}$$

(8) Let $g = g + 1$ and go to step 3.

(9) Stop and the global best solution will be selected to generate the final parameter set $S = \{hid\_no, c_{i1}, c_{i2}, \cdots, c_{in}, \sigma_i, w_{i1}, w_{i2}, \cdots, w_{il}\}$; then, the desired RBFN system can be developed.

From the above descriptions, the only carefully predetermined parameter is max node of RBF which mainly depends on the complexity of the problem. Usually, the more complexity of the problem, the more max nodes of RBF is needed. The other parameters such as max generation and pop size of PSO related to the search range and precision level, which influence only on the utilization of the computational resources.

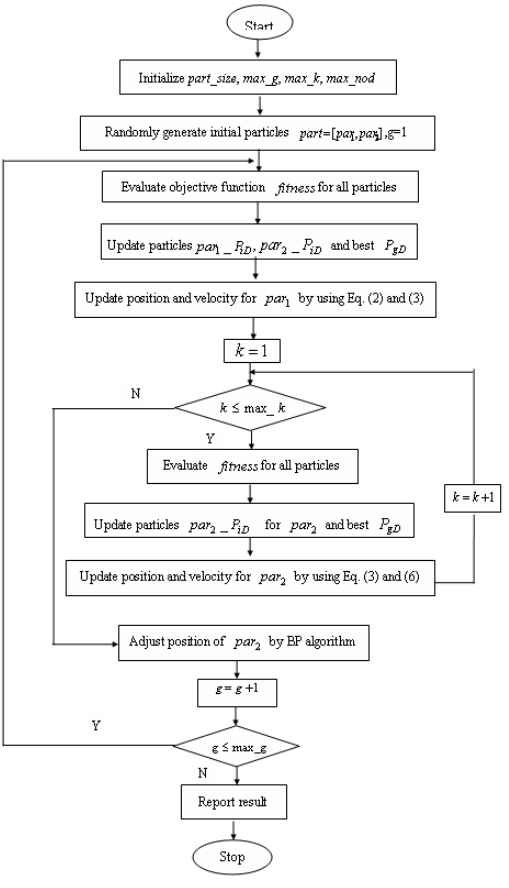


Fig. 2. The flowchart of MPSO-BP-RBFN

5 MPSO-BP-RBFN for Lateral Reservoir Prediction

In this article, we select an oil zone which belongs to JiangHan basin, center of China and extract 15 seismic attributes in certain window length along the line 167 which have 15 wells on or near. The seismic attribute values will be used as inputs sand the thickness of reservoir will be the target. The test line includes 500 CDP points, which every CDP distance is 50 meters. the object layer of the prediction is the top oil group. There are 12 oil well holes near the line. Sample data using the result of seismic attribute optimization in Ref [23], which are six optimized attributes as follows: the average amplitude trough, the summation of positive amplitude, main frequency, instantaneous frequency, instantaneous phase, positive amplitude to negative amplitude ratio from sixteen seismic attributes which are used to networks inputs. The output is the thickness of the reservoir. This case study, we select the seismic attributes of 12 wells and 10 seismic trace nearby as training samples and then the total samples are 120, and the remaining 380 samples, as forecast. The RBF model is 5

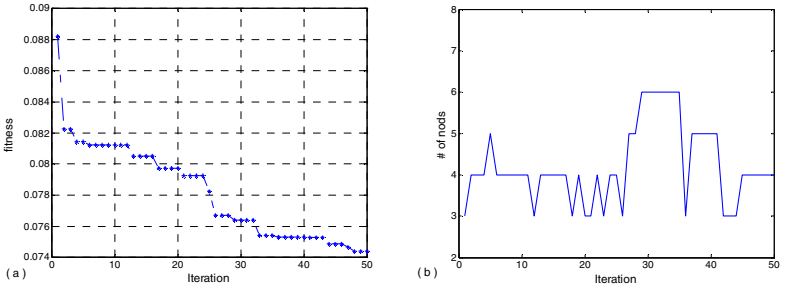


Fig. 3. Average fitness value and the number of hidden nodes

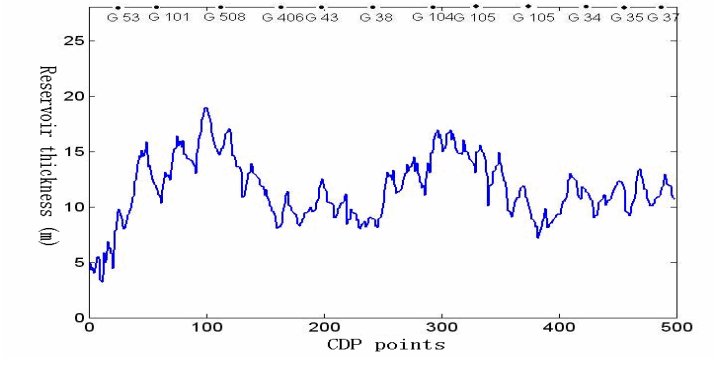


Fig. 4. ML1 test line reservoir thickness

inputs and 1 output. The maximum number of radial basis functions is 6. The length of every particle is 54, in which 48 parameters $\{c_{in}, \sigma_i, w_i, 1 \le i \le 6, 1 \le n \le 6\}$ and 6 selective genes of hidden nods are required to be efficiently chosen from the solution space. The pop size of PSO and BP iterations are 50 and 100, respectively. The influential coefficient of the number of hidden layer nods $\alpha = 0.01$. 10 runs have been implemented randomly using the above parameters. The average fitness value of 10 runs is shown in Fig. 3 (a). Fig.3 (b) depicts the best hidden nodes across iterations. The average forecast results of the entire line are shown Fig. 4. From Fig.3 (a), the best number of hidden nodes is 4. The training MSE is 0.0341 in the well hole when the algorithms terminated.

6 Conclusions

A novel hybrid model, based on a hybrid of MPSO and BP, has been presented and discussed in this paper for training RBFN to reservoir lateral prediction. The combination of the search capabilities of a global and a fast local optimization method has been explored. Adopting multi-encoding, the model can optimize the number of RBFN, centers, widths and weights of RBFN dynamically and adaptively by choosing

a special fitness function. A case study have been conducted using the proposed model and achieved good results.

Acknowledgement

This research was fully supported by National Natural Science Foundation Grant No. 70573101 of the People's Republic of China and the Research Foundation for Outstanding Young Teachers, China University of Geosciences(Wuhan) No. CUGQNW0901.

References

1. Liu, G., Dong, M., Yuan, S., et al.: Multivariate Geologic Statistical Method for Horizontal Reservoir Prediction. Oil Geophysical Prospecting 34(5), 555–559 (1999)
2. Gao, J., Wang, J., Yu, M., et al.: Seismic Attributes Optimization and Application in Reservoir Prediction. Applied Geophysics 3(4), 243–247 (2006)
3. Sullivan, A., Christie, M.: Simulation Error Models for Improved Reservoir Prediction. Reliability Engineering and System Safety 91, 1382–1389 (2006)
4. Herrera, V.M., Russell, B., Flores, A.: Neural Networks in Reservoir Characterization. The Leading Edge 4, 402–411 (2006)
5. West, B.P., May, S.R., Eestwood, J., et al.: Interactive Seismic Facies Classification Using Textural Attributes and Neural Networks. The Leading Edge 10, 1042–1049 (2002)
6. Du, Y., Weiss, W.W., Xu, J., et al.: Obtain an Optimum Artificial Neural Network Model for Reservoir Studies. Society of Petroleum Engineers 84, 23–24 (2002)
7. Wang, Z.: The Predication of Petrochemical Pr-Operties Based on the Artificial Neural Networks with an Early-Res-Tart Algorithm. Computing Techniques for Geophysical and Geochemical Exploration 3, 249–252 (2007)
8. Shahsavand, A., Ahmadpour, A.: Application of Optimal RBF Neural Networks for Optimization and Characterization of Porous Materials. Computers & Chemical Engineering 29, 2134–2143 (2005)
9. Mu, T., Asoke, K.: RBF Neural Networks for Solving the Inverse Problem of Backscattering Spectra. Neural Computing & Applications, doi:10.1007/s00521-007-0138-2
10. Zhang, A., Zhang, L.: RBF Neural Networks for the Prediction of Building Interference Effects. Computers and Structures 82, 2333–2339 (2004)
11. Ram, D., Srivastava, L., Pandit, M., et al.: Corrective Action Planning Using RBF Neural Network. Applied soft computing 7, 1055–1063 (2007)
12. Darken, C., Moody, J.: Fast Adaptive K-means Clustering: Some Empirical Results. In: IEEE INNS International Joint Conference on Neural Networks, pp. 233–238. IEEE Press, New York (1990)
13. Chinrungrueng, C., Sequin, C.H.: Optimal Adaptive K-Means Algorithm with Dynamic Adjustment of Learning Rate. IEEE Transactions on Neural Networks 6(3), 157–168 (1995)
14. Chen, S., Cowan, F.N., Grant, P.M.: Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks. IEEE Trans Neural Networks 2(3), 302–309 (1991)
15. Sherstinsky, A., Picard, R.W.: On the Efficiency of The Orthogonal Least Squares Training Method for Radial Basis Function Networks. IEEE Trans. Neural Networks 7(1), 195–200 (1996)

16. Huang, G., Saratchandran, P., Sundararajan, N.: An Efficient Sequential Learning Algorithm for Growing and Pruning RBF (GAP-RBF) Networks. IEEE Trans. Syst. Man Cybern. Part B 34, 2284–2292 (2004)
17. Eberhart, R., Kennedy, J.: Particle Swarm Optimization. In: Int: IEEE Conf. Neural Networks, vol. IV, pp. 1942–1947. IEEE Press, New York (1995)
18. Clerc, M.: The Swarm and the Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization. In: Int: Proceeding Congress on Evolutionary Computation, Washington, DC, pp. 1951–1957 (1999)
19. Shi, Y.H., Eberhart, R.: A Modified Particle Swarm Optimizer. In: Int: IEEE Conf. Evol. Comput., pp. 69–73. IEEE Press, New York (1998)
20. Kennedy, J., Eberhart, R.: A Discrete Binary Version of the Particle Swarm Algorithm. In: Proceedings of the World Multiconference on Systemic, Cybernetics and Informatics, NJ, Piscataway, pp. 4104–4109 (1997)
21. Gaing, Z.L.: A Particle Swarm Optimization Approach for Optimum Design of PID Controller in AVR System. IEEE Transactions on Energy Conver 19, 384–391 (2004)
22. Kennedy, J., Eberhart, R.C., Shi, Y.: Swarm Intelligence. Morgan Kaufmann, San Francisco (2001)
23. Yu, S.W., Zhu, K.J., Diao, F.Q.: A Dynamic all Parameters Adaptive BP Neural Networks Model and its Application on Oil Reservoir Prediction. Applied Mathematics and Computation 195(1), 66–75 (2008)

Semi-supervised Learning Based on Label Propagation through Submanifold

Jiani Hu, Weihong Deng, and Jun Guo

Beijing University of Posts and Telecommunications, 100876, Beijing, China
cughu@126.com, {whdeng,guojun}@bupt.edu.cn

Abstract. A semi-supervised learning algorithm is proposed based on label propagation through submanifold. The algorithm assumes that samples lying in a local neighborhood share the same labels and the global labels changing among submanifolds is sufficiently smooth. The algorithm firstly introduces a k-nearest neighbor graph to describe local neighborhood among the data set. And then, a cost function and a constraint equation are proposed, which stand for the global smoothness of the class labels' changing and the labeled samples' information respectively. The final semi-supervised learning task is converted to a typical quadratic program, whose optimal solution can minimize the cost function and satisfy the supervised constraint. Experimental results of the algorithm on toy data, digit recognition, and text classification demonstrate the feasibility and efficiency of the proposed algorithm.

Keywords: Semi-supervised learning, K-nearest neighbor graph, Quadratic program, Classification.

1 Introduction

In many practical applications of pattern classification and data mining, unlabeled samples can be obtained easily while the acquisition of labeled training data is costly and time consuming. Semi-supervised learning, which firstly learns from both labeled and unlabeled samples and then classifies unlabeled samples into existing categories, has been the focus of much research in the last few years.

Most recent semi-supervised learning algorithms work by formulating the assumption that "nearby" samples, and samples in the same structure (e.g., a manifold or a cluster), are prone to have the same label [1,2,3,4]. The main differences between the various semi-supervised learning algorithms, such as spectral methods [5,6,7], random walks [4], graph mincuts [8] and transductive SVM [9], lying in their ways of realizing the assumption.

In this paper, a semi-supervised learning algorithm based on Label Propagation through Submanifold (LPS) is proposed. The concept of submanifold derives from manifold, which is a mathematical space where every point belongs to the same class and can be expressed by its neighbors. The LPS algorithm assumes that (1) samples in one submanifold have the same labels, and (2) the class label changing among all the submanifolds is sufficiently smooth. The first assumption

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 617–623, 2009.

is local, which means nearby samples are tend to be in the same class. And the second assumption is global, which describes the smoothness of all submanifolds. The essence of the algorithm is to make the labeled data propagate their class labels smoothly through the submanifold and ensure the resulted label information is smooth along the geodesics in the intrinsic geometry of the labeled data. In brief, the LPS algorithm let every labeled sample propagate its label information to its neighbors until a global optimal state is achieved.

The remainder of this paper will be organized as follows. The analysis of proposed semi-supervised classification algorithm based on Label Propagate through Submanifold (LPS) is proposed in section 2. The algorithm's applications on toy data, digit recognition, and text classification are presented in section 3. Section 4 provides the conclusion.

2 The Semi-supervised Classification Algorithm

Let's first describe the m-class semi-supervised learning task briefly. Suppose there is a set of n data samples $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_l, \mathbf{x}_{l+1}, \cdots, \mathbf{x}_n\}$, $\mathbf{x}_i \in \mathbb{R}^{d\times 1}$, the first l points are labeled while the remaining samples are unlabeled. A data $\mathbf{x}_i$ $(1 \le i \le l)$ is labeled by a binary vector $\mathbf{y}_i = \{y_i^{(1)}, \cdots, y_i^{(c)}, \cdots, y_i^{(m)}\}$, where $y_i^{(c)} = 1$ and $y_i^{(j)} = 0$ for $j \neq c$, indicating that data $\mathbf{x}_i$ belongs to the c th class. Let matrix $\mathbf{Y} = \{\mathbf{y}_1^T, \cdots, \mathbf{y}_l^T, \mathbf{y}_{l+1}^T, \cdots, \mathbf{y}_n^T\}^T$ denote the label indicator matrix of the data set. The object of our semi-supervised classification algorithm is to predict a globally optimized label indicator matrix $\mathbf{Y} \in \mathbb{R}^{n\times m}$ by the data set $\mathbf{X}$ and some labels $\{\mathbf{y}_1^T, \cdots, \mathbf{y}_l^T\}^T$.

The proposed algorithm has two steps. The first step is to construct the local structure of the data, which can be implemented by a weight graph. The second step is to obtain a global optimal label indicator matrix satisfying the global optimal smoothness assuption and the labeled constraints.

The weighted graph is defined as $\mathbf{G} = \{\mathbf{V}, \mathbf{S}\}$, where each vertex $\mathbf{v}_i \in \mathbf{V}$ represents a data sample, and each edge (i, j) is assigned a weight $\mathbf{S}_{ij} \in \mathbf{S}$ to reflect the similarity between the data i and j. The value of $\mathbf{S}_{ij}$ is computed as follows,

$$\mathbf{S}_{ij} = \begin{cases} e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}}, & if\ \mathbf{x}_j \in N_k(\mathbf{x}_i) \vee\ \mathbf{x}_i \in N_k(\mathbf{x}_j) \\ 0, & otherwise. \end{cases} \tag{1}$$

where the set $N_k(\mathbf{x}_i)$ is composed of k nearest neighbors of $\mathbf{x}_i$, and σ is a selected constant. The nearer the two data are, the larger $\mathbf{S}_{ij}$ they share.

LPS algorithm assumes that two nodes with a larger weight edge tend to have the larger probability to be the same label, and overall label changing among samples in the same submanifold is expected to be sufficiently smooth. With this notion, the cost function can be defined as follows,

$$\eta = \sum_{ij} \| \frac{1}{\sqrt{\mathbf{D}_{ii}}} \mathbf{y}_i - \frac{1}{\mathbf{D}_{jj}} \mathbf{y}_j \|^2 \cdot \mathbf{S}_{ij} \tag{2}$$

$\mathbf{D}_{ii}$ is the degree of sample i and $\mathbf{D}_{ii} = \sum_j \mathbf{S}_{ij}$. $\mathbf{D}_{ii}$ measures the similarity among sample $\mathbf{x}_i$ and all the other samples in the data set. η is the total changings of all the data's labels from their k nearest neighbors' label. η reflects the smoothness of the label submanifold structure. Obviously, a smaller value of η denotes a better classification result. At the same time, there is some supervised information, that is the label vector $\mathbf{y}_i$ of $\mathbf{x}_i$ while $1 \leq i \leq l$. This labeled information can be written in the matrix form

$$\mathbf{A} \cdot \mathbf{Y} = \mathbf{b} \tag{3}$$

where $\mathbf{A}$ is a $m \times n$ coefficient matrix, $\mathbf{Y}$ is the $n \times m$ label indicator matrix of the data set, and $\mathbf{b}$ is a $m \times m$ diagonal matrix. To be more exactly, $\mathbf{A} = \{\mathbf{y}_1^T, \mathbf{y}_2^T, \cdots, \mathbf{y}_l^T, \mathbf{0}^T, \cdots, \mathbf{0}^T\}$, $\mathbf{Y} = \{\mathbf{y}_1^T, \mathbf{y}_2^T, \cdots, \mathbf{y}_l^T, \mathbf{y}_{l+1}^T, \cdots, \mathbf{y}_n^T\}^T$, and $\mathbf{b}$ is a diagonal matrix, the value of $\mathbf{b}_{jj}$ equals to the number of labeled samples of the j th class.

Then the semi-supervised classification task is transformed to the following programs,

$$\min_{\mathbf{Y}} \sum_{i,j} \| \frac{1}{\sqrt{\mathbf{D}_{ii}}}\mathbf{y}_i - \frac{1}{\sqrt{\mathbf{D}_{jj}}}\mathbf{y}_j \|^2 \cdot \mathbf{S}_{ij} \tag{4}$$

$$s.t. \quad \mathbf{A} \cdot \mathbf{Y} = \mathbf{b} \tag{5}$$

The object function in (4) can be reduced to

$$\begin{aligned}
&\sum_{i,j} \| \frac{1}{\sqrt{\mathbf{D}_{ii}}}\mathbf{y}_i - \frac{1}{\sqrt{\mathbf{D}_{jj}}}\mathbf{y}_j \|^2 \mathbf{S}_{ij} \\
&= \sum_{i,j} (\frac{1}{\sqrt{\mathbf{D}_{ii}}}\mathbf{y}_i - \frac{1}{\sqrt{\mathbf{D}_{jj}}}\mathbf{y}_j)(\frac{1}{\sqrt{\mathbf{D}_{ii}}}\mathbf{y}_i - \frac{1}{\sqrt{\mathbf{D}_{jj}}}\mathbf{y}_j)^T \mathbf{S}_{ij} \\
&= 2\sum_{i=1}^{n} \mathbf{y}_i \mathbf{y}_i^T - 2\sum_{i,j} \mathbf{S}_{ij} \frac{1}{\sqrt{\mathbf{D}_{ii}}\sqrt{\mathbf{D}_{jj}}} \mathbf{y}_i \mathbf{y}_j^T \\
&= 2\mathrm{trace}\{\mathbf{Y}^T\mathbf{Y} - \mathbf{Y}^T\mathbf{D}^{-\frac{1}{2}}\mathbf{S}\mathbf{D}^{-\frac{1}{2}}\mathbf{Y}\} \\
&= 2\mathrm{trace}\{\mathbf{Y}^T(\mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{S}\mathbf{D}^{-\frac{1}{2}})\mathbf{Y}\}
\end{aligned} \tag{6}$$

where $\mathbf{S}$ is a symmetrical affinity matrix and $\mathbf{D}$ is a diagonal matrix with $\mathbf{D}_{ii}$. If we set matrix $\mathbf{C} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{S}\mathbf{D}^{-\frac{1}{2}}$, the task of semi-supervised learning is the following problem: Finding the global optimal matrix $\mathbf{Y}$ with binary-valued elements that minimizes (7) and satisfies (8).

$$\min_{\mathbf{Y}} \quad \mathrm{trace}\{\mathbf{Y}^T\mathbf{C}\mathbf{Y}\} \tag{7}$$

$$s.t. \quad \mathbf{A} \cdot \mathbf{Y} = \mathbf{b} \tag{8}$$

Generally, solving this optimization problem has been proven to be NP-hard. However, if we relax all the elements of label indicator vector $\mathbf{Y}$ from binary

values to real values, the above optimization problem becomes a typical quadratic program and can be easily solved by Lagrangian method directly.

The Lagrangian function is defined as

$$L(\mathbf{Y}, \lambda) = \text{trace}\{\mathbf{Y}^T\mathbf{C}\mathbf{Y}\} - \lambda(\mathbf{A}\mathbf{Y} - \mathbf{b}) \tag{9}$$

where λ is a constant. Let $\nabla_{\mathbf{Y}} C(\mathbf{Y}, \lambda) = \mathbf{0}$ and $\nabla_{\lambda} C(\mathbf{Y}, \lambda) = \mathbf{0}$. We can get following equation array,

$$\begin{cases} \mathbf{C}\mathbf{Y} - \mathbf{A}^{\mathbf{T}}\lambda = \mathbf{0} \\ -\mathbf{A}\mathbf{Y} + \mathbf{b} = \mathbf{0} \end{cases} \tag{10}$$

These equations can be rewritten to

$$\begin{bmatrix} \mathbf{C} & -\mathbf{A}^{\mathbf{T}} \\ -\mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{Y} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ -\mathbf{b} \end{bmatrix}$$

The inverse matrix of the coefficient matrix can be denoted as,

$$\begin{bmatrix} \mathbf{C} & -\mathbf{A}^{\mathbf{T}} \\ -\mathbf{A} & \mathbf{0} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{Q} & -\mathbf{R}^{\mathbf{T}} \\ -\mathbf{R} & \mathbf{S} \end{bmatrix}$$

If the inverse of matrix $\mathbf{L}$ exists, the matrix $\mathbf{Q}$, $\mathbf{R}$, and $\mathbf{S}$ can be expressed as,

$$\begin{aligned} \mathbf{Q} &= \mathbf{C}^{-\mathbf{1}} - \mathbf{C}^{-\mathbf{1}}\mathbf{A}^{\mathbf{T}}(\mathbf{A}\mathbf{C}^{-\mathbf{1}}\mathbf{A}^{\mathbf{T}})^{-\mathbf{1}}\mathbf{A}\mathbf{C}^{-\mathbf{1}} \\ \mathbf{R} &= (\mathbf{A}\mathbf{C}^{-\mathbf{1}}\mathbf{A}^{\mathbf{T}})^{-\mathbf{1}}\mathbf{A}\mathbf{C}^{-\mathbf{1}} \\ \mathbf{S} &= -(\mathbf{A}\mathbf{C}^{-\mathbf{1}}\mathbf{A}^{\mathbf{T}})^{-\mathbf{1}} \end{aligned} \tag{11}$$

Through the inverse of coefficient matrix, the solution of label indicator matrix $\mathbf{Y}$ is

$$\mathbf{Y} = \mathbf{R}^T\mathbf{b} \tag{12}$$

From above steps, the optimal label indicator matrix $\mathbf{Y}$ can be acquired. However, $\mathbf{Y}$ takes real values for its elements and does not directly indicate the class membership of each data sample. In order to derive the final label information from $\mathbf{Y}$, we have to make $\mathbf{Y}$'s elements binary. For each row of matrix $\mathbf{Y}$, the largest value is set to be "1" and others are set to be "0". After this step, each row $\mathbf{y}_i$ of matrix $\mathbf{Y}$ indicates the class label corresponding to data $\mathbf{x}_i$. The detailed description of the algorithm is proposed in the following Table 1.

3 Performance Evaluations

In this section, three experiments are proposed to evaluate the performance of the proposed algorithm for semi-supervised learning. The tasks include toy data classification, digits recognition, and text classification.

The proposed algorithm is compared with k-NN, Zhou's consistency [1], and harmonic gaussian field method [4]. To best of our knowledge, there is no reliable

Table 1. Description of LPS algorithm

Input: The data matrix $\mathbf{X} \in \mathbb{R}^{d \times n}$ which has n samples in all, and the labeled (class) information of l samples.
Output: The binary-valued matrix $\mathbf{Y} \in R^{n \times m}$, whose row indicates the classification result of each sample.
1. Form the affinity matrix $\mathbf{S}$ defined in equation (11) and set $\mathbf{S}_{ii} = 0$.
2. Construct the matrix $\mathbf{A}$ and $\mathbf{b}$ according to the labeled information.
3. Compute the solution of label indicator matrix $\mathbf{Y}$ through quadratic optimization described in equation(12).
4. Convert the elements of the indicator matrix $\mathbf{Y}$ to be binary. For each row of $\mathbf{Y}$, let $\mathbf{Y}_{ij} = 1$ if $\mathbf{Y}_{ij} = \arg\max_{1 \le q \le c} \mathbf{Y}_{iq}$, and $\mathbf{Y}_{ij} = 0$ otherwise.

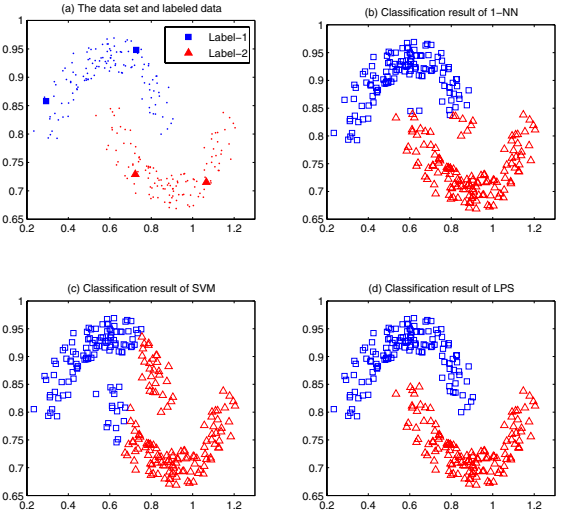
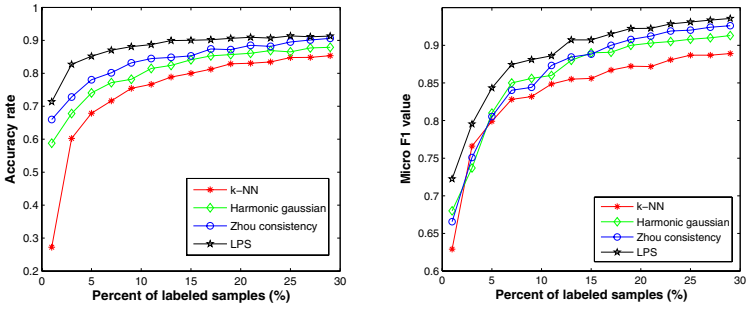


Fig. 1. Classification on the two moons patterns

approach for model selection if very few labeled samples are available. Hence we let all algorithms use their respective optimal parameters. In Zhou's method, α is fixed at 0.99 as in [1]. The parameter σ is 0.15 in Zhou's consistency, Harmonic Gaussian, and LPS algorithm. The parameter k for weighted graph in all three semi-supervised algorithms is set to 3.

An experiment on toy data set is given to illustrate intuitively the power of the LPS algorithm. The data set is shown in Fig.1(a), which looks like two moons. There are only 2 labeled and 127 unlabeled samples in each class. The classification result of 1-nearest neighbor classifier, SVM and LPS are shown in Fig.1(b), (c), and (d). It is clear that LPS is capable of learning the intrinsic structure of the data set and classifing the data correctly with few labeled



(a) (b)

Fig. 2. (a) Accuracy rates of digit recognition with USPS data set, (b) Micro-F_1 value of text classification using 20ng data set

samples. However, traditional classification algorithms, such as 1-nn classifier and SVM, fail to obtain the correct classification result.

Next, we focus on the problem of classifying hand-written digits. The data set we adopt is the USPS handwritten 1616 digits data set [10]. The images of digits "0" to "9" are used in this experiment as ten classes, and there are 100 samples in each class, with a total of 1000. Each image is expressed as a 256 dimensional vector. The evaluation measure is accuracy rate, which is the ratio of correctly classified samples and the total number of samples. The comparison results are illustrated in Fig.2(a), where the horizontal axis represents the percent of randomly labeled samples. The vertical axis is the corresponding accuracy rate which is an average value of 20 independent runs. The results show that LPS outperforms k-NN, Harmonic Gaussian, and Consistency algorithms in general. It can be noted that LPS method is more stable, because it can get a higher accuracy rate when there is only 1% labeled samples.

The task of text classification using 20-newsgroups data set is also proposed. The topic "*rec*", containing "*autos*", "*motorcycles*", "*baseball*" and "hockey" branches, is chosen from the data 20news-18828. The articles were preprocessed by the same procedure as in [1]. The resulted 3970 document vectors are in an 8014-dimensional space. Finally the document vectors are normalized into *TF-IDF* representation. The performance of each algorithm is measured by F_1 value which is based on precision and recall across documents (micro-average)[11]. The comparison results are illustrated in Fig.2(b), where the horizontal axis represents the percent of randomly labeled samples. The vertical axis is the corresponding micro F_1 value which is an average value of 20 independent runs. Experimental results show that the efficiency of LPS algorithm is superior to other algorithms under all the percent of labeled samples.

4 Conclusions

In this paper, a semi-supervised classification algorithm named LPS is proposed. It discovers the structure of the data set through the labeled data propagate their

intrinsic label information to their neighbors smoothly in the submanifold. Experimental results on toy data, digit recognition, text classification demonstrate the feasibility of the proposed algorithm. In our future research, we will focus on theoretical analysis and accelerating issues of our LPS algorithm.

Acknowledgements

This work was supported by National Natural Science Foundation of China under Grant No.60675001.

References

1. Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Scholkopf, B.: Learning with Local and Global Consistency. In: Advances in Neural Information Processing Systems(NIPS), pp. 321–328 (2003)
2. Chapelle, O., Weston, J., Scholkopf, B.: Cluster Kernels for Semi-Supervised Learning. In: Advances in Neural Information Processing Systems(NIPS), pp. 585–592 (2002)
3. Belkin, M., Niyogi, P., Sindhwani, V.: On Manifold Regularization. In: Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (2005)
4. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised Learning Using Gaussian Fields and Harmonic Functions. In: Proceedings of ICML, 20th International Conference on Machine Learning, pp. 912–919 (2003)
5. Belkin, M., Niyogi, P.: Semi-Supervised Learning on Riemannian Manifolds. Machine Learning Journal 56, 209–239 (2004)
6. Joachims, T.: Transductive Learning via Spectral Graph Partitioning. In: Proceedings of ICML, 20th International Conference on Machine Learning, pp. 290–297 (2003)
7. Chung, F.R.K.: Spectral Graph Theory. American Mathematical Society (1997)
8. Blum, A., Chawla, S.: Learning from Labeled and Unlabeled Data Using Graph Mincuts. Proceedings of ICML. In: 18th International Conference on Machine Learning, pp. 19–26 (2001)
9. Vapnik, V.N.: Statistical Learning Theory. Wiley, NY (1998)
10. http://www.kernel-machines.org/data.html
11. Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys 34(1), 1–47 (2002)

BGNN Neural Network Based on Improved *E.Coli* Foraging Optimization Algorithm Used in the Nonlinear Modeling of Hydraulic Turbine

Yijian Liu[1,2] and Yanjun Fang[2]

[1] School of Electrical & Automation Engineering, Nanjing Normal University, Nanjing 210042, China
[2] Department of Automation, Wuhan University, Wuhan 430072, China
Liuyijian_2002@163.com, yjfang@whu.edu.cn

Abstract. A novel Bayesian-Gaussian neural network (BGNN) is proposed in this paper for the nonlinear modeling of hydraulic turbine which is difficult to obtain its mathematical model because of its complex and nonlinear characteristics. The topology and connection weights of BGNN can be set immediately when the training samples are available. The threshold matrix parameters of BGNN are updating based an improved *E.Coli* foraging optimization algorithm (IEFOA) which is an evolutionary optimization algorithm imitating the behaviors of *E.Coli* bacteria. Simulation results for the nonlinear model of hydraulic turbine generating unit are provided and demonstrate the effectiveness and shorter training time and more effective self-tuning compared with the BP neural network for the identification of hydraulic turbine generating unit.

Keywords: Hydraulic turbine, BGNN, Improved *E.Coli* foraging optimization algorithm, Nonlinear modeling.

1 Introduction

The hydraulic turbine generating unit (HTGU) is a complex and essence nonlinear system [1] and it is difficult to obtain its model through mathematical analytical method. Moreover the often change of operating situation also affects the characteristics of the HTGU and the dynamic reposing procedure of the whole hydraulic turbine control system. So it is vital to the reconstruction of HTGU characteristics for the design of HTGU control system.

In the nonlinear modeling techniques, artificial neural network (ANN) plays an important role for the reason that ANN has been proven that it can approximate any nonlinear system [2] and ANN has been successful used in some fields [3][4][5]. In the research of hydraulic turbine generating unit, many artificial neural networks also have been presented to the model identification of the HTGU. In the references [6][7][8], multilayer perception (MLP) network, BP neural network and RBF neural network used for the modeling of the HTGU have been reported and their results show that ANN could obtain the nonlinear model of the HTGU.

Regardless of above inspiring developments for neural networks applications in the nonlinear model of the HTGU, there are the intrinsic vulnerable points of these neural

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 624–634, 2009.

networks. First, the topology needs to be defined by trial and error. Second, the training object required to adjust so many connection weights in error function minimization inevitably results in a complex error surface and a long training time. The lack of self-tuning ability of these networks makes them less helpful in on-line model based applications to some special process, for example, time-variant systems. To overcome the shortcomings of neural networks, a Bayesian-Gaussian neural network (BGNN) [9] is proposed which is a posteriori probability model based on the Bayesian theory and Gaussian hypothesis. Its topology and connection weights can be set immediately when training samples are available. This simplifies the design of BGNN and reduces the training time compared with traditional BP and RBF neural networks. So in this paper, the BGNN is developed to the nonlinear modeling of the HTGU and compared with the BP and RBF neural networks.

Based on the chemotatic behaviors of *E.Coli*, a random optimization algorithm named the *E.Coli* foraging optimization algorithm is first introduced in the reference [10] which can be used for the parameter optimization problems effectively [10][11][12]. In this paper, an improved *E.Coli* foraging optimization algorithm (IEFOA) [13] is presented which only reserves the stage of chemotactic while eliminating other stages. At the same time, a new tracing operator is proposed in the improved algorithm and the optimal positions of individual *E.Coli* are adopted to update the locations of swarm, which enhances the random ability of algorithm and speed to the global optimal goal. The IEFOA is used to update the threshold matrix parameters of the Bayesian-Gaussian network for nonlinear model of the HTGU.

This paper is organized as follows. In section 2 the identified variables relations of hydraulic turbine generating unit (HTGU) is described first. Then section 3 gives the proposed improved *E.Coli* foraging optimization (IEFOA). In section 4, the BGNN and its weights training algorithm are depicted in details and summarizes the procedure of nonlinear model identification of the HTGU based on the BGNN. Simulation results and discussion are given in section 5. The final section 6 contains some conclusions and further work.

2 Identification Problem of Hydraulic Turbine Characteristics

The structure of hydraulic turbine generating unit (HTGU) is shown in Fig.1. Servomechanism, penstock system, water turbine and generator are four main parts of the HTGU of water power plant.

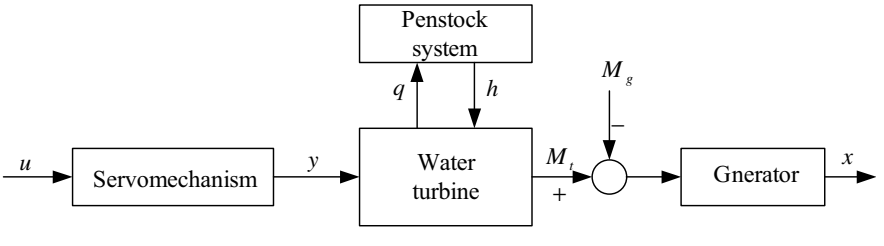


Fig. 1. Illustration structure of hydraulic turbine generating unit

$$y = \frac{1}{1+T_y s} u \,. \tag{1}$$

The servomechanism can be approximately expressed as a first order equation with its translating function model described in formula (1). T_y is inertia time constant of servomechanism and y denotes the output of servomechanism.

In dynamic process, the characteristics of water turbine running vary with the change of operating condition. The movable Francis turbine is nonlinear in essence. The nonlinear characteristics of movable Francis turbine can be depicted as the equation (2) as follows:

$$\begin{cases} m_t = f(y, x, h) \\ q = q(y, x, h) \end{cases}, \tag{2}$$

where q is flow and m_t denotes the movable turbine moment. h is water head and x is the speed of rotation (generally x is expressed as frequency). The function relations of f and q are nonlinear and difficult to obtain through mathematical analysis method. Neural network as a black modeling technique can be used to the nonlinear model identification. So BGNN in this paper is suitable for the above nonlinear characteristics modeling of the HTGU.

The dynamic characteristic of penstock system is complex and nonlinear between the water head h and flow q. Due to the difficulty of establishing the precise nonlinear math model, BGNN is also used to learn the nonlinear relation listed as the following equation (3).

$$h = h(q) \,. \tag{3}$$

Dynamic equation of generator taking account of load characteristics is often simplified as the follows:

$$x = \frac{1}{T_a s + e_n}(m_t - m_g) \,, \tag{4}$$

where T_a is inertia time constant of generator and e_n denotes the adjusting coefficient of load. Those parameters vary with water turbine wok situation and need to be identified.

Therefore the identification problem of HTGU in this paper is to obtain the dynamic nonlinear models described in the equation (2), the equation (3) and the equation (4).

3 Improved *E.Coli* Foraging Optimization Algorithm (IEFOA)

The *E.Coli* foraging optimization algorithm is an evolutionary random optimization algorithm first proposed in the reference [10] which imitates the behaviors of *E.Coli* bacteria. In the previous work [13], we presented an improved *E.Coli* foraging optimization algorithm (IEFOA) and used it to the identification of a nonlinear model parameters obtaining effective results.

3.1 The Operators of IEFOA

The IEFOA adopts three operators listed as follows.

① Swim operator (Swim)

The change of positions of individual *E.Coli* in chemotactic behavior is called "Swim". The role of the swim operator is to update the positions of *E.Coli* individual in optimization field space. The number of swimming step in swim operator is set as N_s, which determines the swimming number of *E.Coli* individual in towards to the direction of optimization values.

② Tumble operator (Tumble)

When the fitness function based on the current positions is less than the former step position, the swimming direction of *E.Coli* individual will be changed, which is called tumbling operator in the improved foraging optimization algorithm. The role of tumbling operator is to ensure the random ability of the improved foraging optimization algorithm in order to escape local minimum to assure the global optimization ability of the *IEFOA*.

③ Trace operator of optimal value (Tracing of optimal value)

In the basic *E.Coli* foraging optimization algorithm, the *E.Coli* individual throws away its history position formation whereas this information is very important for guiding its chemotactic behaviors. In order to accelerate the optimization process, the optimal history position values is utilized to update the position values of the *E.Coli* swarm in *IEFOA*. The details about the tracing operator are listed in the following introduction of the principle of the improved *E.Coli* foraging optimization algorithm in the next section.

To describe expediently, the term of "Chemotactic step" denotes the tumbling procedure closely following the previous tumbling operator or the previous swimming operator.

3.2 Principle of the Improved *E.Coli* Foraging Optimization Algorithm

It is assumed that there is an optimization problem with p dimensions and parameters θ ($\theta = [\theta_1, \theta_2, \cdots, \theta_p]$) to be estimated or optimized. Accordingly, θ are the counterparts of the positions of *E.Coli* in foraging field and all the possible values of θ consist of the values space in the optimization problem, namely the environmental space of *E.Coli* foraging.

S is the size of *E.Coli* swarm. $X_i = (x_{i1}, x_{i2}, \cdots, x_{ip})^{\Gamma} \in R^p$ denotes the positions of *E.Coli* individuals in foraging optimization space, where $i = 1,2 \cdots\cdots S$. $eval_i = f(X_i)$ is the fitness function that denotes the badness or goodness of the individual position of the i_{th} *E.Coli*. Commonly, the fitness function is selected as the function value to be optimized problem.

$P_i^{pbest} = (p_{i1}^{pbest}, p_{i2}^{pbest}, \cdots, p_{ip}^{pbest})^{\Gamma}$ denotes the historical best fitness function value experienced by the i_{th} *E.Coli* individual and its counterpart position.

Assume the step number of *E.Coli* lifecycle is N_c, which is also set as the step number of chemotactic step in the improved *E.Coli* foraging optimization algorithm. In the $n_{th}(n \le N_c)$ chemotactic step, every position of *E.Coli* individual is updated according to the following shown in formula (5a) and (5b).

$$X_i^{n+1} = X_i^n + C(i)\phi(n). \tag{5a}$$

$$X_i^{n+1} = P_i^{pbest} + w(P_e)C(i). \tag{5b}$$

$$w(P_e) = \begin{cases} 0 & if \quad rand(.) < P_e \\ 1 & if \quad rand(.) > P_e \end{cases}, \tag{5c}$$

where, $C(i)$ is the step length of swimming operator in the swimming direction $\phi(n)$; $\phi(n)$ denotes a random number distributing the field of [-1, 1] generated by the tumbling operator. $w(p_e)$ in equation (5c) denotes a constant determined by a probability P_e.

The tracing operator of optimization value is defined as: If the fitness function value $eval_i^{n+1} = f(X_i^{n+1})$ at the position X_i^{n+1} is better than the one $eval_i^n$ at the position X_i^n, the value of $\phi(n)$ is kept and the position of *E.Coli* individuals is updated according to the formula (5a) by the swimming operator until to the maximum swimming step length N_s. Otherwise, the position is updated by the i_{th} *E.Coli* history optimal position value according to the formula (5b).

4 Description of BGNN Based on IEFOA

The Bayesian-Gaussian neural network (BGNN) [9] is proposed which is a posteriori probability model based on Bayesian theory and Gaussian hypothesis.

Compared with the traditional recurrent neural networks and feedforward neural networks such BP and RBF neural networks, the BGNN neural networks can easily determine the topology of neural network and weights of BGNN. The details of the BGNN and some theorems proofs can be found in the reference [10]. The structure and training algorithm of BGNN are illustrated in the following sections.

4.1 The Bayesian-Gaussian Neural Network

The topology and connection weights of a BGNN are shown in Fig. 2.

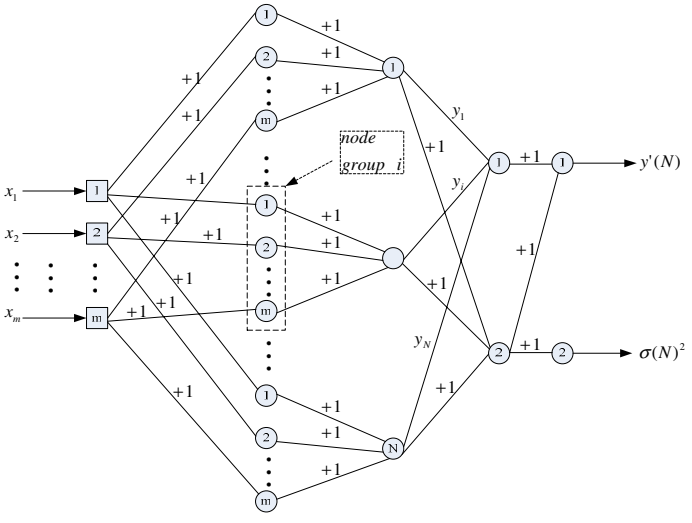


Fig. 2. Topology and connection weights of a BGNN

The main structure of the BGNN is described below. Let (X_i, y_i) i=1, 2,…, N, be the training data set, where N is the number of samples, X_i is the sample input and is represented by $m \times 1$ vector. X_i= $(X_{i1}, X_{i2}, \ldots, X_{im})^{\top}$ and y_i is the sample output. The new output y corresponding to the new input X is generated using the measure of belief view of probability.

Under the Gaussian hypothesis, when the combined information source (X_i, y_i), i=1, 2,…, N, is known, the probability distribution of $Y(X)$ will be approximately

$$p(Y \mid Y_1, Y_2, ..., Y_N) = \frac{c}{\sqrt{2\pi}\sigma(N)} e^{\frac{-1(Y-y'(N))^2}{\sigma(N)^2}}, \tag{6}$$

where c is a normalizing constant.

$$y'(N) = \sigma(N)^2 \sum_{i=1}^{N} \sigma_i^{-2} y_i, \tag{7}$$

$$\sigma(N)^{-2} = \sum_{i=1}^{N} \sigma_i^{-2}. \tag{8}$$

Assume that

$$\sigma_i^{\,2} = \sigma_0^2 e^{(X-X_i)^T D (X-X_i)}, \tag{9}$$

where D is input threshold matrix and d_{11}, d_{12},…, d_{mm} are named the input factors, which will be evaluated through the network training.

$$D = \begin{bmatrix} d_{11}^{-2} & & \\ & d_{jj}^{-2} & \\ & & d_{mm}^{-2} \end{bmatrix}. \tag{10}$$

The criterion of minimize is similar the Prediction Error method:

$$V_N(D) = \frac{1}{2N}\sum_{i=1}^{N}(y_i - y_i^{'})^2 , \tag{11}$$

where N is the net order, while y_i and $y_i^{'}$ are denoted as the desired output and the network output for sample i respectively.

Although there are many choices for the training of the BGNN, in this paper the above IEFOA is employed in this research. In the IEFOA, the evaluation function of each *E.Coli* individual should be selected carefully according to special problem. The fitness function of IEFOA $eval_i^{n+1}$ is set as $V_N(D)$. The parameter D is corresponding to the positions of the i_{th} *E.Coli* individual.

4.2 Procedures of BGNN Based on IEFOA Used for Identification of Hydraulic Turbine

The BGNN combined the IEFOA can be used for the nonlinear characteristics identification of HTGU and the procedures is described as the following steps.

Step 1: Obtain the training sample set and initial the parameters needed in the BGNN and IEFOA, including σ_0, the network order *N,* the threshold matrix parameter *D,* the value range of optimized parameters θ, chemotactic step number N_c, swimming operator step number N_s and step length $C(i)$, the size of swarm S, the probability P_e and the stop criterion of the algorithm.

Step 2: For every *E.Coli* individual, update its position according to the formula (5a) and evaluate its fitness function value $eval_i^{n+1}$.

Step 3: If $eval_i^{n+1} < eval_i^n$, set the swimming counter as zero and keep the swimming direction $\phi(n)$ constant. Keep the swimming operator to the maximum swimming step number N_s or the condition $eval_i^{n+1} < eval_i^n$ dissatisfied. At the same time, update the parameters P_i^{pbest}.

Step 4: If $eval_i^{n+1} > eval_i^n$, update the position of *E.Coli* individual according the formula (5b).

Step 5: If the stop criterion is met, then exit; otherwise, continue the procedure.

Step 6: Loop to Step 2 until the position of every *E.Coli* individual in the swarm updated.

Step 7: Go on the next chemotactic step until the stop criterion is met.

5 Simulation Results and Discussion

The nonlinear characteristics of the HTGU needed to be identified are the three equations (2) (3) and (4). Therefore three BGNNs should be adopted. The first BGNN to identify the nonlinear characteristic in equation (2) has three input variables *y*, *x* and *h*

and two output m_t and q named BGNN_1. The second BGNN called BGNN_2 adopts only one input q and one output h which relationship is shown in equation (3). The third BGNN employs two inputs m_t, m_g and one output x described in equation (4) denoted as BGNN_3.

In the compared BP neural networks, the first structure of BP neural network BP_1 is 3-10-2 which owns the inputs and outputs same as the first BGNN and one hidden layer with 10 neurons. The second BP neural network is the structure of 1-5-1 called BP_2 and the third structure of the BP neural network is 2-6-1 as BP_3. The three BP neural networks all adopt one hidden layer with different neurons number. The training algorithm of the BP neural networks adopts the back-propagation algorithm.

5.1 Training Date Generation

In the simulation experiments, we consider a training set which is generated by the system under close-loop control with a reasonable PID controller shown in Fig.3. Instead of generating random input sequences to the HTGU process, training set obtained from the close-loop control system can excite the dynamic characteristics of the HTGU as soon as possible and this is more practical in the application of the HTGU control.

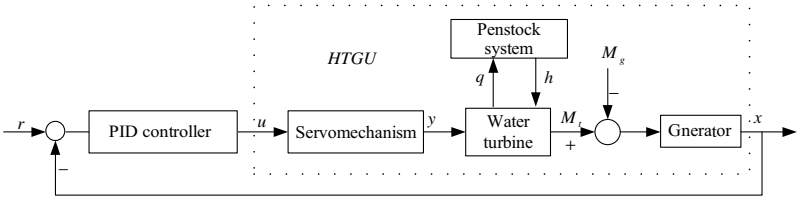


Fig. 3. Generation structure of training set from the control system of the HTGU

where, r is the input exciting signal which takes the normal work situation operating value with random Gaussian white noise. In the control of the HTGU, the training samples u, y, q, h, m_t, m_g, and x are samples in the operation of the HTGU with the typical work situation changed which denotes the nonlinear characteristics of the HTGU. Therefore the nonlinear model of the HTGU can be obtained based on the BGNN and the training samples.

5.2 Off-Line Training of the BGNN and BP Neural Network for the HTGU

From the above generation system of training data, 1000 samples are obtained which will be used to the training of the BGNN and BP neural networks of the HTGU. The first 500 samples are used to the off-line training neural networks and the last 500 samples are employed to test the BGNN and BP neural networks. All the samples are needed to be pretreatment and normalized. Table.1 shows the off-line training performances and iterations.

Table 1. Comparison between BGNN and BP neural network in off-line training

No.	V_N	Iteration number
BGNN_1	5.8e-4	18
BP_1	1.3e-3	1080
BGNN_2	6.2e-5	12
BP_2	4.7e-4	220
BGNN_3	0.6e-4	28
BP_3	1.3 e-3	350

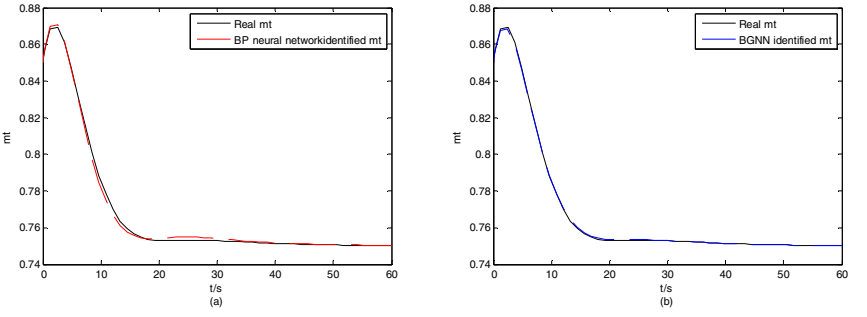


Fig. 4. Identified mt output by BP neural network and BGNN model with 10% decreasing load

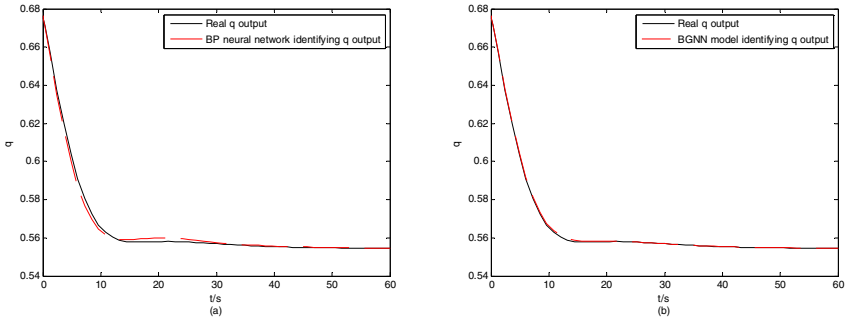


Fig. 5. Identified q output by BP neural network and BGNN model with 10% decreasing load

It can be seen from the table.1 that the BGNN in the off-line training of the nonlinear model of HTGU all saves a large amount of time. After the off-line training, the testing samples are applied to demonstrate the effectiveness of the BGNN and BP neural networks. The testing results are shown in Fig. 4 to Fig. 6. It can be seen that the BGNN and BP neural networks all obtained the nonlinear relations of inputs and outputs of the HTGU. But the BGNN can obtain higher accuracy. So the BGNN provides a more effective method for the identification of nonlinear characteristics of the HTGU compared with the BP neural network.

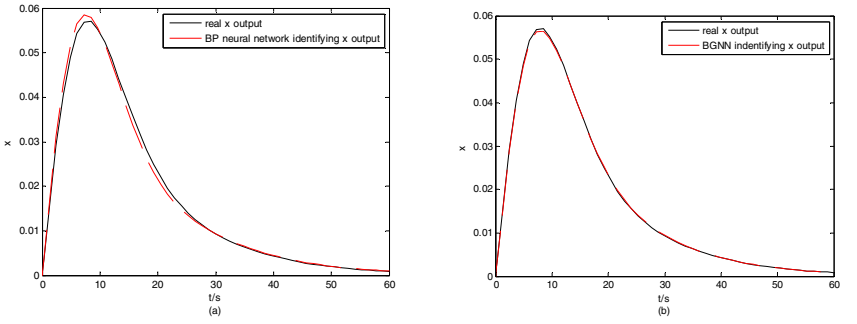


Fig. 6. Identified x output by BP neural network and BGNN model with 10% decreasing load

6 Conclusions

The BGNN is employed to the nonlinear model identification of HTGU in this paper. The random evolutionary algorithm IEFOA is used to update the input factors of BGNN. Compared with the BP neural network, the BGNN shows its predominance such as topology setting and weights tuning time.

Only off-line neural network training method is developed for the modeling of HTGU. The further work will focus on the using of self-tuning BGNN to the nonlinear model of HTGU and to the control strategy design of HTGU based on the BGNN on-line identified model.

Acknowledgments. This work was supported by the National Science Foundation of China (No.60704024, No.60772107). The authors are grateful to Dr. Jingyu Liu for his valuable comments and suggestions.

Reference

1. Jiang, C.: Nonlinear Simulation of Hydro Turbine Governing System Based on Neural Network. IEEE International Conference on System, Man and Cybernetics, 784–787 (1996)
2. Funahashi, K.: On the Approximation of Continuous Mapping by Neural Networks. Neural Netw. 2, 183–192 (1989)
3. Yazdan, S., Mohsen, H., Rostam, M.: Numerical Solution of the Nonlinear Schrodinger Equation by Feedforward Neural Networks. Communications in Nonlinear Science and Numerical Simulation 13(10), 2132–2145 (2008)
4. Guzelbey, I.H., Cevik, A., Erklig, A.: Prediction of Web Crippling Strength of Cold-formed Steel Sheeting Using Neural Networks. Journal of Constructional Steel Research 62(10), 962–973 (2006)
5. Jung, S., Ghaboussi, J.: Neural Network Constitutive Model for Rate-dependent Materials. Computers and Structures 84(15–16), 955–963 (2006)

6. Chang, J., Xiao, Z.H., Wang, S.Q.: Neural Network Predict Control for the Hydro Turbine Generator Set. In: The Second International Conference on Machine Learning and Cybernetics, pp. 2–5 (2003)
7. Cheng, Y., Ye, L., Cai, W.: Modeling of Hydro-turbine Hill Chart by Neural Network. Journal of Huazhong University of Science & Technology (Nature Science Edition) 31(6), 68–70 (2003)
8. Sarimveis, H.: Training Algorithms and Learning Abilities of Three Different Types of Artificial Neural Networks. J. Syst. Anal. Model Simulation 38, 555–581 (2000)
9. Ye, H., Nicolai, R., Reh, L.: A Bayesian-Gaussian Neural Network and Its Application in Process Engineering. Chemical Engineering and Process 38, 439–449 (1998)
10. Passino, K.M.: Biomimicry of Bacterial Foraging for Distributed Optimization and Control. IEEE Control Systems Magazine 22(3), 52–67 (2002)
11. Hanmandlu, M., Nath, A.V., Mishra, A.C.: Fuzzy Model Based Recognition of Handwritten Hindi Numerals Using Bacterial Foraging. In: The 6th IEEE/ACIS International Conference on Computer and Information Science, pp. 309–314 (2007)
12. Mishra, S., Bhende, C.N.: Bacterial Foraging Technique-Based Optimized Active Power Filter for Load Compensation. IEEE Transactions on Power Delivery 22(1), 457–465 (2007)
13. Liu, Y., Fang, Y., Zhang, J.: Simplified E.Coli Foraging Optimization Algorithm and Its Application to Parameter Identification of Nonlinear System Model. Control Theory and Application 24(6), 991–994 (2007)

A Hybrid Recurrent Neural Network for Machining Process Modeling

Xingyu Lai[1], Chunyan Yan[1], Bangyan Ye[2], and Weiguang Li[2]

[1] Department of Mechatronical Engineering, Guangdong Institute of Science and Technology, Guangzhou 510640, China
[2] School of Mechanical Engineering, South China University of Technology, Guangzhou 510640, China

Abstract. A new hybrid recurrent neural network (HRNN) for machining process modeling is presented based on the diagonal recurrent neural network (DRNN). In order to overcome the weakness of back propagation (BP) algorithm, a generalized entropy square error (GESE) criterion is defined and a dynamic recurrent back propagation algorithm is developed to guarantee the global convergence. The HRNN based on the GESE is then used for nonlinear system identification and neural network modeling of the machining process. The numeral experiments results show that the HRNN has better approximate effectiveness, tracking and dynamic performance than traditional BP neural network.

Keywords: Machining process, Modeling, Hybrid recurrent neural network, Generalized entropy square error.

1 Introduction

Machining process has high nonlinearity, time-variability and uncertainty. It is very difficult to be described with accurate mathematical models and conventional control methods based on the plant model may not accordingly yield satisfactory results. The neural network is a new approach for modeling of machining process. The neural network has stronger nonlinear approximating, self-organizing, self-learning, and self-adapting ability. It shows enormous potential in solving high nonlinearity and uncertainty problems and is an effective tool for modeling and controlling of nonlinear system. However, the feedforward network's defects, namely, slow convergence speed and easy getting into local minima are gradually discovered. The feedforward network comprises static mapping, and input-output relations based on it are also static [1]. Therefore, the modeling using the feedforward network cannot truly describe performances of nonlinear dynamic process.

In the last few years the neural network with dynamic structure was introduced in system identification and control [2], [3]. The recurrent neural network comprises dynamic mapping which is more suitable for dynamic systems than the feedforward network. New network architecture, called a hybrid recurrent neural network is presented for the modeling of machining process in this paper. At the same time, in order to overcome the drawback of the mean square error criterion of BP algorithm, the generalized entropy square error (GESE) is defined, and a dynamic back propagation

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 635–642, 2009.

training algorithm is developed to train the HRNN. The simulation results show that the HRNN has better performance than the conventional BP neural network (BPNN) in machining process identification and modeling.

2 Hybrid Recurrent Neural Network and Learning Algorithm

2.1 Hybrid Recurrent Neural Network

Ku and Lee [4] presented the diagonal recurrent neural network for dynamic systems control. But the DRNN only has self-feedback connections among the neurons in the hidden layer; it cannot capture the dynamic behavior of the whole system [5]. In order to make the most of the dynamic information of the system, a hybrid recurrent neural network by adding the output feedback is constructed for machining process modeling.

Figure 1 shows a 3-layer structure of hybrid recurrent neural network that is composed of an input layer, a hidden layer and an output layer. A neuron of the hidden layer is recurrent neuron. Suppose there is a neuron in the output layer, and the output in the discrete time (t-1) feeds back to the hidden. w_{ij}, w_{jk}, w_j^d, and w_k^r represent input, output, recurrent, and feedback weight vectors respectively.

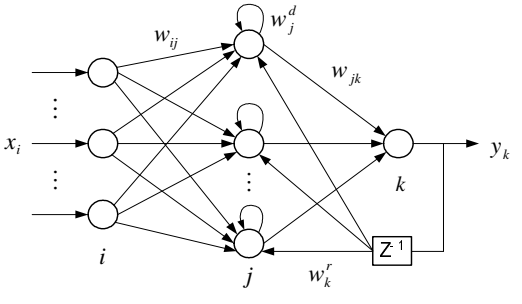


Fig. 1. Hybrid recurrent neural network structure

2.2 Learning Algorithm of HRNN

For each the discrete time t, $x_i(t)$ is the ith input, $S_j(t)$ is the sum of inputs of the jth recurrent neuron, $O_j(t)$ is the output of the jth recurrent neuron, net^0 is the sum of inputs of the output neuron and $y_k(t)$ is the output of the network. The mathematical models of HRNN can be inferred as:

$$S_j(t) = w_j^d O_j(t-1) + w^r y(t-1) + \sum_i w_{ij} x_i(t) \ , \tag{1}$$

$$O_j(t) = f(S_j(t)) \, , \tag{2}$$

$$net^o = \sum_j w_j O_j(t) \, , \tag{3}$$

$$y(t) = f(net^o) \, , \tag{4}$$

where $f(\cdot)$ is the activation function which is often unsymmetrical sigmoid function $f(x)=1/(1+e^{-x})$.

In order to improve the convergent speed, there are a lot of studies on error function structure. The error function of BP algorithm is the mean square error (MSE), namely, $E=(y_d-y_k)^2/2$, but the surface of the MSE is a multidimensional super-surface, it has many flat area and the vale of local minima. So this influences the convergent speed and falls into local minima. Karayiannis proposed the entropy error function to solve the incorrect saturation existing in training course of the conventional error function [6]. The entropy error function was defined as follows:

$$E = -\sum_{k=1}^{K} [\, y_d \ln y_k + (1 - y_d) \ln(1 - y_k)\,], \tag{5}$$

However, the entropy error function suffers from overspecialization for training patterns since the error signal for correctly saturated is too strong. Afterward, Sang-Hoon Oh improved the entropy error function to resolve the incorrect saturation problem [7]. In this sense, the modified function becomes:

$$E = -\sum_{k=1}^{K} [\, y_d \ln \frac{y_k}{y_d} + (1 - y_d) \ln(\frac{1 - y_k}{1 - y_d})\,], \tag{6}$$

Although the entropy error function has active significance on accelerating the convergent speed, it is only one-order convergence. So we propose the modified entropy error function and define the generalized entropy square error to improve the convergent speed and to achieve two-order convergence.

Shannon entropy is defined as $H(x) = -\sum_{k=1}^{n} p_k \ln p_k$. In this paper, the generalized entropy is expressed as $R(x) = -\sum_{k=1}^{n} q_k \ln \frac{q_k}{p_k}$. Then, the generalized entropy square error (GESE) is defined as follows:

$$E = \frac{1}{2}\sum_{k=1}^{K} [e(y_k)]^2 = \frac{1}{2}\sum_{k=1}^{K} [\, y_d \ln \frac{y_d}{y_k} - (1 - y_d) \ln \frac{1 - y_d}{1 - y_k}]^2 \quad , \tag{7}$$

The dynamic recurrent BP algorithm presented can be used to calculate the optimal weight vector w of the HRNN. In each iteration step, the weight vector is updated by:

$$w(t) = w(t-1) - \eta \frac{\partial E}{\partial w} + \alpha(w(t-1) - w(t-2)), \tag{8}$$

where η is the learning rate, α is the momentum factor. The partial derivatives of every parameter are as follows:

$$\frac{\partial E}{\partial w_j} = \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial net^o} \frac{\partial net^o}{\partial w_j} = \delta(y) y(1-y) O_j(t) \ , \tag{9}$$

$$\frac{\partial E}{\partial w_j^d} = \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial net^o} \frac{\partial net^o}{\partial O_j} \frac{\partial O_j}{\partial w_j^d} = \delta(y) y(1-y) w_j P_j(t) \ , \tag{10}$$

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial net^o} \frac{\partial net^o}{\partial O_j} \frac{\partial O_j}{\partial w_{ij}} = \delta(y) y(1-y) w_j Q_{ij}(t) \ , \tag{11}$$

$$\frac{\partial E}{\partial w^r} = \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial net^o} \frac{\partial net^o}{\partial O_j} \frac{\partial O_j}{\partial w^r} = \delta(y) y(1-y) w_j R(t) \ , \tag{12}$$

where $\delta(y)=\frac{\partial E}{\partial y}=[y_d \ln\frac{y_d}{y}-(1-y_d)\ln\frac{1-y_d}{1-y}]\varphi(y)$, $\varphi(y)=-(\frac{y_d}{y}+\frac{1-y_d}{1-y})$,

$$P_j(t)=\frac{\partial O_j}{\partial w_j^d}=\frac{\partial O_j}{\partial S_j}\frac{\partial S_j}{\partial w_j^d}=f'(S_j)O_j(t-1) \quad , \tag{13}$$

$$Q_{ij}(t)=\frac{\partial O_j}{\partial w_{ij}}=\frac{\partial O_j}{\partial S_j}\frac{\partial S_j}{\partial w_{ij}}=f'(S_j)x_i(t) \quad , \tag{14}$$

$$R(t)=\frac{\partial O_j}{\partial w^r}=\frac{\partial O_j}{\partial S_j}\frac{\partial S_j}{\partial w^r}=f'(S_j)y(t-1) \quad , \tag{15}$$

Based on the above discussion, an efficient algorithm of the HRNN can be described by following steps:

Step 1. Initialize weights and all parameters: weights, goal error, learning rate, and momentum factor.

Step 2. Calculate the nodes of hidden and output layers according to (1)~(4).

Step 3. Calculate $P_j(t)$、$Q_{ij}(t)$、$R(t)$ according to (13)~(15).

Step 4. Update the weights of the HRNN using (8).

Step 5. Go back to step 2 cyclically until the error converges.

2.3 Convergence and Stability Analysis

The stability of the hybrid recurrent neural network is relative to the learning rate η. For a small value of η the convergence is guaranteed but the convergent speed is very slow, on the contrary, if η is too big, the network becomes unstable. In order to train neural network effectively, a guideline for proper choice of the learning rate based on the discrete-type Lyapunov function is developed in this section.

Theorem 1. *The convergence is guaranteed as long as the learning rate* η *satisfied* $0<\eta<2$.

Proof. A discrete-type Lyapunov function can be defined as follows:

$$V(t)=\frac{1}{2}e^2(t)=\frac{1}{2}[y_d \ln\frac{y_d}{y}-(1-y_d)\ln\frac{1-y_d}{1-y}]^2 \quad , \tag{16}$$

where $e(t)$ represents the error in the learning process. Then, the change of Lyapunov function can be expressed by

$$\Delta V(t)=V(t+1)-V(t)=\frac{1}{2}e^2(t+1)-\frac{1}{2}e^2(t) \quad , \tag{17}$$

The error difference due to the learning process can be obtained by

$$e(t+1)=e(t)+\Delta e(t)=e(t)+[\frac{\partial e(t)}{\partial w}]^T\Delta w \quad , \tag{18}$$

where $\frac{\partial e(t)}{\partial w} = \varphi(y)\, y(1-y) O_j$, $\Delta w = -\eta \frac{\partial E}{\partial w} = -\eta \delta(y)\, y(1-y) O_j$.

We can obtain

$$\Delta V(t) = \frac{1}{2}[e^2(t+1) - e^2(t)] = \frac{1}{2}\Delta e(t)[2e(t) + \Delta e(t)] = -\frac{1}{2}\eta e^2(t)(2-\eta) \quad , \tag{19}$$

Let $\Delta V(t)<0$, then the convergence is guaranteed if η is chosen as $0 < \eta < 2$.

3 Machining Process Modeling Based on HRNN

3.1 Nonlinear System Identification

Consider a nonlinear system described by

$$y(k) = u(k)^3 + \frac{y(k-1)\, y(k-2)[y(k-1)-1]}{1 + y(k-1)^2} \quad , \tag{20}$$

where the input $u(k) = 0.5\sin(4\pi t)$.

The sampling period is 0.001s. The initial learning rate is 0.35. The output of the plant, the HRNN trained by the GESE and the error are shown in Fig. 2. The simulation results of the BPNN trained by the MSE are shown in Fig. 3. It can be seen that the precision of identification using the HRNN and the proposed method is much superior to that obtained using the BPNN.

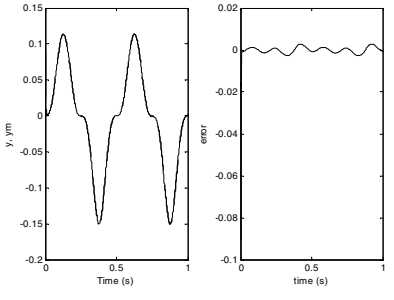


Fig. 2. Identification results of HRNN

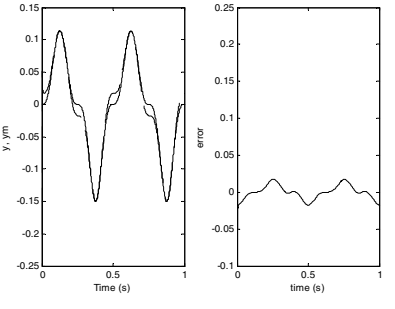


Fig. 3. Identification results of BPNN

3.2 HRNN Modeling for Machining Process

The machining process usually consists of servo system, cutting process, sensor and so on. The input-output can be separately expressed as follows [8]:

$$V_f = \frac{K_n \omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} u \quad , \tag{21}$$

$$f = 60 V_f / pn \quad , \tag{22}$$

$$F_s = K_s a f^m = (K_s a f^{m-1}) f \quad , \tag{23}$$

$$F = K_e F_s \quad , \tag{24}$$

where V_f is feed velocity (mm/s), u is the servo input (V), K_n is the servo gain (mm/(Vs)), ω_n is the servo natural frequency (rad/s), $\hat{t}$ is the servo damping ratio, f is feed rate (mm/r), n is the spindle speed (r/min), K_s is the gain of the cutting feed (N/mm^2), a is the cutting depth (mm), K_e is the conversion factor. F_s is stead-state cutting force, F is the measured cutting force, and m is the index, generally speaking, $0.6<m<1$. So the machining process is non-linear and time-variant.

Total gain of cutting process $K=60K_nK_sK_af^{m-1}/(pn)$, from (21) to (24), the model of machining process can be expressed as follows:

$$\frac{F(s)}{u(s)}=\frac{K\omega_n^2}{s^2+2\xi\omega_n s+\omega_n^2} \quad , \tag{25}$$

Total gain of cutting process K varies with the cutting depth, the spindle speed, and the feed rate, so the cutting process has nonlinearity and time-variability. In this simulation, the parameters of a turning process as follows: n=600r/min, K_n=1mm/(V·s), K_s=1670 N/mm^2, K_e = 1.5, ξ = 0.5, ω_n = 20 rad/s. We have established the non-linear model of the machining process when m=0.7. The machining process is non-linear system if $m\neq1$.

In fact, the force feedback control model of machining process is a high-order system. It may be simplified to a two-order system for the convenience of calculation and simulation. Therefore, there are four inputs and one output. Input vector is defined by

$$x=[F(t-1),F(t-2),u(t-1),u(t-2)]^T \quad , \tag{26}$$

The HRNN model for machining process is shown in Fig. 4. The neuron number of the input, hidden and output layer is 4, 5, and 1 respectively. The activation function of input layer is linear, and the activation functions of the hidden and output layer are sigmoid function.

Input signal is shown in Fig. 5. The initial weights of the networks are random numbers between -0.3 and 0.3. Fig. 6 shows the HRNN tracking results, and Fig. 7 is the BPNN tracking results.

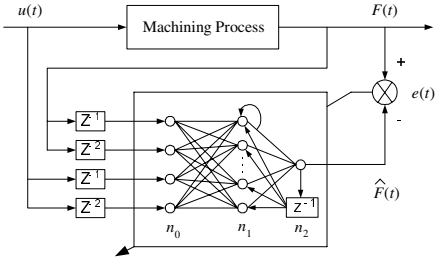


Fig. 4. HRNN model for machining process

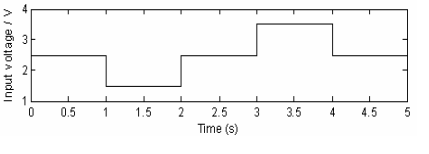


Fig. 5. Input signal

By comparing Fig. 6 with Fig. 7, it can be seen that the HRNN can approximate the plant better accurately than the BPNN. The output of the HRNN is almost the same as that of actual process. However, The output of the BPNN fluctuates when the input voltage changes suddenly.

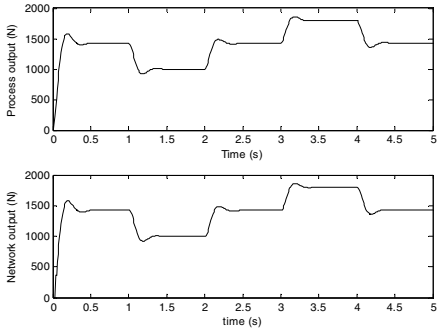


Fig. 6. Outputs of HRNN and actual process

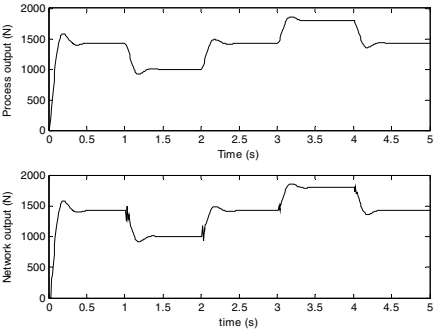


Fig. 7. Outputs of BPNN and actual process

4 Conclusions

This paper constructs a hybrid recurrent neural network with better performance than BP neural network. In order to overcome the drawback of the mean square error criterion of BP algorithm, the generalized entropy square error is defined. A dynamic recurrent back propagation algorithm is presented to guarantee the global convergence. The HRNN based on GESE is then utilized for nonlinear system identification and neural network modeling. Numeral experiments for the force control model of machining process show the HRNN has better approximate effectiveness in the identification and control for dynamic systems.

Acknowledgements. This research was supported by the Natural Science Foundation of Guangdong Province under the grant No. 06025546, Science and Technology Plan Project of Zhuhai under the grant No. PC20061040.

References

1. Tang, F.H., Guo, Y.J., Yang, Y., Kang, J.L.: An Improved Recurrent Neural Network and Its Simulation. Transactions of Beijing Institute of Technology 5, 399–401 (2005)
2. Li, C.J., Huang, T.Y.: Automatic Structure and Parameter Training Methods for Modeling of Mechanical Systems by Recurrent Neural Networks. Applied Mathematical Modeling 23, 933–944 (1999)
3. Sanchez, E.N., Bernal, M.A.: Adaptive Recurrent Neural Control for Nonlinear System Tracking. IEEE Trans. on Systems, Man, and Cybernetics 30, 886–889 (2000)
4. Ku, C.C., Lee, K.Y.: Diagonal Recurrent Neural Networks for Dynamic Systems Control. IEEE Trans on Neural Networks 6, 144–156 (1995)
5. Cho, J.S., Kim, Y.W., Park, D.J.: Identification of Nonlinear Dynamic Systems Using Higher Order Diagonal Recurrent Neural Network. Electronics letters 33, 2133–2135 (1997)

6. Karayiannis, N.B., Venetsanopoulos, A.N.: Fast learning algorithms for neural networks. IEEE Trans. on Circuits and System II: Analog and Digital Signal Processing 39, 453–474 (1992)
7. Oh, S.H.: Improving the Error Backpropagation Algorithm with a Modified Error Function. IEEE Trans. on Neural Networks 8, 799–803 (1997)
8. Yao, X.F., Chang, S.L.: Computer Control of Machining Process. China Machine Press, Beijing (2004)

Alternating Iterative Projection Algorithm of Multivariate Time Series Mixed Models

Zhongcheng Zhang

College of Mathematics and information Science, Huanggang Normal University, Huanggang 438000, China
zzhang63@126.com

Abstract. In this paper, a new parameters estimation method is proposed based on the ill-condition separation method and the alternating iterative projection algorithm for the problem of column multi-collinearity of design matrix in the multiple regression and multivariate time series mixed models. Furthermore, the parameters estimation is improved according to the character of model. Therefore, the improved estimated parameters have better character.

Keywords: Multivariate time series regression, Ill-condition separation, Alternating iterative, Parameters estimation.

1 Introduction

Multivariate time series models were improved more reasonable and more perfect based on the alternating iterative multivariate regression models and the multivariate time series models. Simplex multivariate time series mixed models can only reflect the relation on multivariate random vector and bypast values, and can not reflect the relation on present value of every variable and present values of other variables. Simplex multivariate regression model can only reflect the relation on present values of certain variable and present values of other variables, and can not reflect the relation on present values and bypast values. The improved alternating iterative multivariate regression and the multivariate time series models do not only show a relation between the current value of the one variable and the past value of all variables, but also show a relation between the current value and the past value of one variable. Because the relation of the observed value in the model is very complex and the excellent property of the estimated parameter is difficult to ensure, there are many new problems to estimate the parameters in the model. When the design matrix is column multi-collinear[1], the parameters estimation of model is discussed in this paper through the character of model, based on the ill-condition separation of design matrix X [2] and interative algorithm of interactive projection[3-10].

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 643–650, 2009.

2 A Brief Introduction of Models

Consider the alternating iterative multiple regression and multivariate time series mixed models

$$y_t = \alpha + \Theta_0 y_t + \Theta_1 y_{t-1} + \Theta_2 y_{t-2} + ... + \Theta_p y_{t-p} + \varepsilon_t, \tag{1}$$

where

$$y_{t-s} = \begin{pmatrix} y_{1(t-s)} \\ y_{2(t-s)} \\ \vdots \\ y_{n(t-s)} \end{pmatrix}, \quad \alpha = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_3 \end{pmatrix}, \quad \varepsilon_t = \begin{pmatrix} \varepsilon_{1t} \\ \varepsilon_{2t} \\ \vdots \\ \varepsilon_{3t} \end{pmatrix},$$

$$\Theta_0 = \begin{pmatrix} 0 & a_{012} \cdots a_{01n} \\ a_{021} & 0 \quad \cdots a_{02n} \\ & \ddots \\ a_{0n1} & a_{0n2} \cdots 0 \end{pmatrix},$$

$$\Theta_s = \begin{pmatrix} a_{s11} & a_{s12} \cdots a_{s1n} \\ a_{s21} & a_{s22} \cdots a_{s2n} \\ & \ddots \\ a_{sn1} & a_{sn2} \cdots a_{snn} \end{pmatrix}, \quad s = 1, 2, \cdots, p.$$

ε_t is an n dimensional random variable with independent and identical distribution, Where $E(\varepsilon_t) = 0$, $D(\varepsilon_t) = (\varepsilon_t \varepsilon_t{}') = \Omega$.

Obviously, model (1) can be expanded to the following equation:

$$\begin{cases} y_{1t} = & \alpha_1 + 0\, y_{1t} + a_{012} y_{2t} + \cdots + a_{01n} y_{nt} \\ & + a_{111} y_{1(t-1)} + \cdots + a_{11n} y_{n(t-1)} \\ & + a_{211} y_{1(t-2)} + \cdots + a_{21n} y_{n(t-2)} + \cdots \\ & + a_{p11} y_{1(t-p)} + \cdots + a_{p1n} y_{n(t-p)} + \varepsilon_{1t} \\ y_{2t} = & \alpha_2 + a_{021} y_{1t} + 0\, y_{2t} + \cdots + a_{02n} y_{nt} \\ & + a_{121} y_{1(t-1)} + \cdots + a_{12n} y_{n(t-1)} \\ & + a_{221} y_{1(t-2)} + \cdots + a_{22n} y_{n(t-2)} + \cdots \\ & + a_{p21} y_{1(t-p)} + \cdots + a_{p2n} y_{n(t-p)} + \varepsilon_{2t} \\ & \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ y_{nt} = & \alpha_n + a_{0n1}\, y_{1t} + a_{0n2} y_{2t} + \cdots + 0\, y_{nt} \\ & + a_{1n1} y_{1(t-1)} + \cdots + a_{1nn} y_{n(t-1)} \\ & + a_{2n1} y_{1(t-2)} + \cdots + a_{2nn} y_{n(t-2)} + \cdots \\ & + a_{pn1} y_{1(t-p)} + \cdots + a_{pnn} y_{n(t-p)} + \varepsilon_{nt} \end{cases}$$

Suppose that we continue to observe y_i for T times. The number of the sample that actually is used to regression algorithm in every equation above is only $T-p$ because the time is late for p. For equation $i(i = 1, 2, ..., n)$, let

$$y^{(i)} = (y_{i(p+1)}, y_{i(p+2)}, \cdots, y_{i(T)})'$$

$$X = \begin{pmatrix} 1 & y_{1(p+1)} & \cdots & y_{n(p+1)} & y_{1p} & \cdots & y_{np} \\ 1 & y_{1(p+2)} & \cdots & y_{n(p+2)} & y_{1(p+1)} & \cdots & y_{n(p+1)} \\ 1 & y_{1(p+3)} & \cdots & y_{n(p+3)} & y_{1(p+2)} & \cdots & y_{n(p+2)} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & y_{1(T)} & \cdots & y_{nT} & y_{1(T-1)} & \cdots & y_{n(T-1)} \end{pmatrix}$$

$$\begin{pmatrix} y_{1(p-1)} & \cdots & y_{n(p-1)} & \cdots & y_{11} & \cdots & y_{n1} \\ y_{1(p)} & \cdots & y_{n(p)} & \cdots & y_{12} & \cdots & y_{n2} \\ y_{1(p+1)} & \cdots & y_{n(p+1)} & \cdots & y_{13} & \cdots & y_{n3} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ y_{1(T-2)} & \cdots & y_{n(T-2)} & \cdots & y_{1(T-p)} & \cdots & y_{n(T-p)} \end{pmatrix},$$

$$\beta^{(i)} = (\alpha_i, a_{0i1}, \cdots, a_{0i(i-1)}, 0, a_{0i(i+1)}, \cdots, a_{0in}, a_{1i1}, a_{1i2}, \cdots, a_{1in}, a_{2i1}, a_{2i2}, \cdots, a_{2in}, \cdots, a_{pi1}, \cdots, a_{pin})',$$

$$\varepsilon^{(i)} = (\varepsilon_{i(p+1)}, \varepsilon_{i(p+2)}, \cdots, \varepsilon_{i(T)})'.$$

Hence, the equation i in the model above can be expressed by

$$y^{(i)} = X\beta^{(i)} + \varepsilon^{(i)}. \tag{2}$$

3 Parameter Estimation of the Models

3.1 The Order of the Models

In order to estimate parameters of the model, we can get the order of the model through the following equations

$$AIC(p) = \ln \det(\hat{\Sigma}_n) + 2n^2 p/T,$$
$$BIC(p) = \ln \det(\hat{\Sigma}_n) + n^2 p \ln T/T,$$

where n is the number of the variable in the model, T is the time to observe the sample, $p = 1, 2, 3, \ldots$. $\hat{\Sigma}_n$ is the estimate of the residual matrix and its element is

$$\hat{\sigma}_{ij} = \frac{(y^{(i)} - X\hat{\beta}_i) - (y^{(j)} - X\hat{\beta}_j)}{T}.$$

The order p is the integer p when $AIC(p)$ or $BIC(p)$ reaches least.One the one hand, in order to determine the order number of model, we need to calculate the regression coefficient. On the other hand, in order to determine the regression coefficient, we need to calculate the order number of model. So we set $p = 1, 2, 3, \cdots$, and calculate the regression coefficient, then calculate $AIC(p)$ and $BIC(p)$. When the least value is got, the calculation is over.

3.2 Column Multi-collinearity Parameters Estimation of Design Matrices

The analysis of the models'character. According to the character of the models, the two sides of the equation have y_t. If we directly estimate the parameter $\alpha, \Theta_0, \Theta_1, \cdots, \Theta_p$ with OLS, the estimate is not associative and valid. In order to improve the better property of the parameters estimate, we can use $2LSE, 3LSE$. But the design matrix is column multi-collinear, we can not be easy to get the better property of the estimated parameters with the method showed above. Hence, we can improve the parameters estimation with extended ridge estimate. What's more, because there exists y_t in the left and the right of the equation, we can not prove that the parameter estimated by extended ridge estimate is better than the parameter estimated by $2LSE, 3LSE$.

We will estimate the parameters by the iterative algorithm of ill-condition separation method for the problem of column multi-collinearity of design matrices in the multiple regression and multivariate time series mixed models.

Column multi-collinearity parameters estimation of design matrices. For the equation i,

$$y^{(i)} = X\beta^{(i)} + \varepsilon^{(i)},$$

it can be expressed as

$$\|y^{(i)} - X\beta^{(i)}\| \to \min,$$

which can be considered as an projection from a point $y^{(i)}$ to an $np+1$ dimensional linear subspace $X\beta^{(i)}, \beta^{(i)} \in R^{(np+1)}$. If the design matrix is column multi-collinear, we can separate the design matrix by ill-condition and split X and $\beta^{(i)}$ into $X = (X_1|X_2)$, $\beta^{(i)'} = (\beta_1^{(i)'}|\beta_2^{(i)'})$, such that the linear correlation column separated. Hence, we can get

$$\|y^{(i)} - X_1\beta_1^{(i)} - X_2\beta_2^{(i)}\| \to \min. \tag{3}$$

$X_2\beta_2^{(i)}$ is a linear subspace ($\beta_2^{(i)} \in R^{p_2}, p_1 + p_2 = np + 1$. Let $p_1 = p_2$ when $np+1$ is even). It's easy to prove that $y^{(i)} - X_1\beta_1^{(i)} (\beta_1^{(i)} \in R^{p_1})$ is a close and convex set. So (3) can be considered as the distance between a close convex set and a linear subspace by alternating projection algorithm method.

Put initial value $\beta_{11}^{(i)}$. One has $y_1^{(i)} = y^{(i)} - X_1\beta_{11}^{(i)}$. So the model can be transformed to

$$\|y_1^{(i)} - X_2\beta_2^{(i)}\| \to \min. \tag{4}$$

With common method to compute the least squares estimator of the parameter, we can obtain the projection $X_2\beta_{21}^{(i)}$ of the point $y^{(i)} - X_1\beta_{11}^{(i)}$ from the close convex $y^{(i)} - X_1\beta_1^{(i)}$ to the subspace $X_2\beta_2^{(i)}$.

From (4), we can get $\beta_{21}^{(i)}$. Substituting into (3) yields, we can obtain $y_2^{(i)} = y^{(i)} - X_2\beta_{21}^{(i)}$. So the model (3) can be transformed to

$$\|y_2^{(i)} - X_1\beta_1^{(i)}\| \to \min.$$

With the same methods, we can get $\beta_{12}^{(i)}$. Hence, one has the projection $y^{(i)} - X_1\beta_{12}^{(i)}$ of the point $X_2\beta_{21}^{(i)}$ from the linear subspace $X_2\beta_2^{(i)}$ to the close convex set $y^{(i)} - X_1\beta_1^{(i)}$.

By analogy, we can obtain the sequence

$$\beta_{11}^{(i)}, \beta_{21}^{(i)}, \beta_{12}^{(i)}, \beta_{22}^{(i)}, \cdots, \beta_{1k}^{(i)}, \beta_{2k}^{(i)} \cdots$$

Given a parameter $\varepsilon > 0$, one has the asymptotic solution of the equation i in the model when $\|\beta_{1(k+1)}^{(i)} - \beta_{1k}^{(i)}\| \leq \varepsilon, \|\beta_{2(k+1)}^{(i)} - \beta_{2k}^{(i)}\| \leq \varepsilon$, where the solution is

$$\beta_k^{(i)'} = (\beta_{1k}^{(i)'} | \beta_{2k}^{(i)'}).$$

So we can get the estimation $\hat{\alpha}, \hat{\Theta}_0, \hat{\Theta}_1, \cdots, \hat{\Theta}_p$ of the parameters $\alpha, \Theta_0, \Theta_1, \cdots, \Theta_p$ in model (1). But from (1), we know that the left and the right in the equation have y_t. Obviously, the property of the obtained estimation $\hat{\alpha}, \hat{\Theta}_0, \hat{\Theta}_1, \cdots, \hat{\Theta}_p$ is very poor. Hence, the parameter estimation needs improvement.

The improvement of parameter estimation. The first improvement: put $\hat{\alpha}, \hat{\Theta}_0, \hat{\Theta}_1, \cdots, \hat{\Theta}_p$ into model (1) and get the estimation $\hat{y_t}$ of y_t and then does the second regression. At that time, in model (1), the left is the observed value y_t and the right is the estimation $\hat{y_t}$. With the method OLS, we can get $\hat{\hat{\alpha}}, \hat{\hat{\Theta}}_0, \hat{\hat{\Theta}}_1, \cdots, \hat{\hat{\Theta}}_p$. If the design matrix is column multi-collinear, we can get $\hat{\hat{\alpha}}, \hat{\hat{\Theta}}_0, \hat{\hat{\Theta}}_1, \cdots, \hat{\hat{\Theta}}_p$ by ill-condition separation method mentioned above.

The second improvement: put the value $\hat{\hat{\alpha}}$, $\hat{\hat{\Theta}}_0$, $\hat{\hat{\Theta}}_1$,$\cdots$, $\hat{\hat{\Theta}}_p$ which are obtained in the first improvement into model (1), we can obtain the covariance matrix estimation $\hat{\Sigma}^{(i)}, i = 1, 2, \cdots, n$ of the equation i and then get $\hat{\beta}_{GL}^{(i)} = (X'\hat{\Sigma}^{(i)}X)^{-1}X'y^{(i)}$ by using the extended least squares estimation with $\hat{\Sigma}^{(i)}$.

4 Properties of the Parameter Estimation in the Model

Property 1. The sequence $\{\beta_k^{(i)}\}$ is convergent to the least square solution $\hat{\beta}_L^{(i)}$ of the model (1).

In order to prove this property, the proof process of convergence for interactive projection are given firstly.

Let Ω be a inner product space, d be the distance in Ω which defined from inner product . Then the definition of projection from one point to a closed set is given as follows.

Definition 1. Suppose that $b \in \Omega$,closed set $A \subset \Omega$,$a_0 \in A$,$a \in A$.If

$$d(b, a_0) = \inf_{a \in A} d(b, a) = d(b, A).$$

Then point a_0 is called a projection from point b to a closed set A.

Obviously, if A is a subspace, the above definition is just the same definition of projection in generic inner product space. Next the definition of local distance between two closed sets.

Definition 2. Let A and B be two closed sets, and $A \subset \Omega$,$B \subset \Omega$.For any point $a \in A$,$b \in B$,if they satisfy

$$d(a,b) = d(a,B) = d(b,A),$$

then $d(a,b)$ is called the local distance between A and B.

Furthermore, some conclusions are given.

Theorem 1. Let A and B be two closed convex set in inner product space, and $d(a,b)$ be the local distance between A and B.Then $d(A,B) = d(a,b)$.

Proof. If $d(a,b) = 0$,then the Theorem 1 is obvious. Suppose that $d(a,b) \neq 0$, based on a and b,two hyperplane π_A and π_B are plotted , and they are perpendicular to line segment ab. A and B are situated in the exterior of π_A and π_B respectively. $d(a,b)$is the distance between π_A and π_B,and the same time,$d(a,b)$ is also the distance between A and B. The theorem is proved.

Theorem 2. The process of interactive projection for two closed convex set A and B is convergent

Proof. Suppose that A and B be two closed convex set in inner product space. For any $a_0 \in A$,determining the projection b_0 at B of a_0, then determining the projection a_1 at A of b_0,$\cdots$. By analogy, when we get the projection b_i at B of a_i, then we determine projection a_{i+1} at A of b_i. Finally, we can obtain the sequence

$$a_0, b_0, a_1, b_1, \cdots, a_k, b_k, \cdots$$

Obviously, we have

$$d(a_0,b_0) \geq d(a_1,b_0) \geq d(a_1,b_1) \geq \cdots$$

$$\geq d(a_k,b_k) \geq d(a_{k+1},b_k) \geq \cdots$$

There exists infimum for the monotone decreasing bounded sequence of number. Suppose that it converge to $d(a^*,b^*)$. On the one hand, because A and B are closed set, so the distance is accessible, $a^* \in A, b^* \in B$.On the other hand, because A and B are convex set, so $d(a^*,b^*)$is also the global distance between A and B,namely,there exists $a^* \in A, b^* \in B$,when $k \to \infty$, we have $a_k \to a^*, b_k \to b^*$, and $d(a^*,b^*) = d(A,B)$.The theorem is proved.

Next the proof of Property 1 is given.

According to formula (3), we can conclude that to determine the distance from $Y^{(i)}$ to subspace $X\beta^{(i)}$ can be transform into determining the distance from one closed convex set to a subspace, and this subspace can be regarded as a closed convex set. By Theorem 2, we know that the process of interactive projection for them is convergent. Because $X, X_1^{(i)}, X_2^{(i)}$ are all column nonsingular, so the

least squares estimate is unique. Thus, the sequence $\left\{\beta_k^{(i)}\right\}$ is convergent to the least square solution $\left\{\hat{\beta}_L^{(i)}\right\}$ of the model (1).

Property 2. $\hat{\hat{\alpha}}, \hat{\hat{\Theta}}_0, \hat{\hat{\Theta}}_1, \cdots, \hat{\hat{\Theta}}_p$ have asymptotic associativity.

By property 1, we know $\beta_k^{(i)} \to \beta_L^{(i)}$ as $k \to \infty$. By the first method, one has $\hat{\hat{\alpha}}, \hat{\hat{\Theta}}_0, \hat{\hat{\Theta}}_1, \cdots, \hat{\hat{\Theta}}_p$. Obviously, they are approximate two-stage least square solution of $\alpha, \Theta_0, \Theta_1, \cdots, \Theta_p$.

In the second regression of the first method, we can get $\hat{\hat{\alpha}}, \hat{\hat{\Theta}}_0, \hat{\hat{\Theta}}_1, \cdots, \hat{\hat{\Theta}}_p$ with common least square method if the design matrix has not column multi-collinearity. Because $\beta_k^{(i)} \to \beta_L^{(i)}$, $\beta_k^{(i)}$ is approximately equal to $\beta_L^{(i)}$ as $k \to \infty$, the second regression is obtained by least square method. So $\hat{\hat{\alpha}}, \hat{\hat{\Theta}}_0, \hat{\hat{\Theta}}_1, \cdots, \hat{\hat{\Theta}}_p$ has asymptotic associativity.

In the second regression of the first method, design matrix still has column multi-collinearity, we can get $\hat{\hat{\alpha}}, \hat{\hat{\Theta}}_0, \hat{\hat{\Theta}}_1, \cdots, \hat{\hat{\Theta}}_p$ still with the ill-condition separation method mentioned above. The two solution are, respectively, convergent to least square solution. According to the analysis to the condition above, we know that $\hat{\hat{\alpha}}, \hat{\hat{\Theta}}_0, \hat{\hat{\Theta}}_1, \cdots, \hat{\hat{\Theta}}_p$ are approximate two-stage least square solution of $\alpha, \Theta_0, \Theta_1, \cdots, \Theta_p$. Hence, it has asymptotic associativity. In a word, Property 2 holds.

Property 3. $\hat{\beta}_{GL}^{(i)}$ has asymptotic validity.

The second method is said to a improvement to parameters estimation again based on the first method. Because the parameter $\hat{\hat{\alpha}}, \hat{\hat{\Theta}}_0, \hat{\hat{\Theta}}_1, \cdots, \hat{\hat{\Theta}}_p$ obtained by the first method are approximately equal to two-stage least square solution of $\alpha, \Theta_0, \Theta_1, \cdots, \Theta_p$, together with the second method, it's easy to know $\hat{\beta}_{GL}^{(i)}$ is approximate three-stage least square solution of $\beta^{(i)}$. According to validity of three-stage least square, we can know $\hat{\beta}_{GL}^{(i)}$ has asymptotic validity more than $\hat{\beta}_L^{(i)}$.

5 Conclusions

In the alternating iterative multiple regression and multivariate time series mixed models, there are many variables and date is complex, so it is very more complex if the design matrix is column multi-collinear. In this paper, a new parameters estimation method is proposed by the iterative algorithm of ill-condition separation for the problem of column multi-collinearity of design matrix. Furthermore, the properties of the parameters estimation are discussed such that the method of parameter estimation more perfect.

Acknowledgments

This work was supported by the Significant Project Grant No. 20082704 of Hubei Provincial Department of Education, China.

References

1. Tong, H.Q.: Descent Computation and Separation of Ill-Condition in Linear Model. Numerical Computation and Computer Appication 4, 241–245 (1997)
2. Friedman, J., Tukey, J.: A Projection Pursuit Algorithm for Exploratory Data Nalysis. IEEE Transactions on Computers 889, 881–889 (1974)
3. Jin, J.L.: A Projection Pursuit Method for Solution on Uncertainty Decision-Making. System Engineering Theory & Practice 4, 42–46 (2003)
4. Zhang, X.L., Ren, R.Q.: A Projection Pursuit Model for Judgement of Enterprise Competition. Mathematical Statistics and Management 7, 53–55 (2005)
5. Li, S.L.: A Nonlinear System Modeling Based on Projection Pursuit and Genetic Algorithms. System Engineering Theory & Practice 4, 22–28 (2005)
6. Gu, A.L.: Multidimensional Project Algorithm for Solving Linear Equations. Operations Research and Management Science 15, 17–21 (2006)
7. Li, S.L.: A Nonlinear System Modeling Method Based on Projection Pursuit and Genetic Algorithm. System Engineering Theory & Practice 4, 22–28 (2005)
8. Pu, Y.F.: Two-Step Matrix Projection Algorithm for Data Classification. J. Tsinghua Univ. (Sci.) & Tech. 44, 311–314 (2004)
9. Xu, J.W., Lu, Y.: Local Projective Method and It's Application on Nonlinear Time Series. em Chinese Journal of Mechanical Engineering 39, 146–150 (2003)
10. Wu, H.J.: Data Minig Based on Fluctuation Feature in Time Series. Control and Decision-making 22, 160–163 (2007)

Semi-supervised Learning with Multimodal Perturbation

Lei Su[1], Hongzhi Liao[1], Zhengtao Yu[2], and Jiahua Tang[1]

[1] School of Software, Yunnan University, Kunming 650091, China
[2] School of Information Engineering and Automation,
Kunming University of Science and Technology, Kunming 650051, China
sulei@mail.ynu.edu.cn, hzliao@ynu.edu.cn,
jhtang@mail.ynedu.net.cn, ztyu@bit.edu.cn

Abstract. In this paper, a new co-training style semi-supervised algorithm is proposed, which employs Bagging based multimodal perturbation to label the unlabeled data. In detail, through perturbing the training data, input attributes and learning parameters together, the algorithm generates accurate but diversity k-nearest neighbor classifiers. These classifiers are refined using unlabeled examples which are labeled if the other classifiers agree on the labeling. Experimental results show that the semi-supervised algorithm could effectively improve the classification generalization by utilizing the unlabeled data.

Keywords: Machine Learning, Semi-Supervised Learning, Co-training, Ensemble Learning, Multimodal Perturbation.

1 Introduction

The traditional supervised learning employs a large amount of labeled data to train the model. However, in many practical learning scenarios such as nature language processing, image retrieval and web-page classification, labeled data are often difficult, expensive, as they require the experienced human effort. Meanwhile the unlabeled may be relatively easy to collect. Therefore, semi-supervised learning that exploits the unlabeled data to reduce the need for expensive labeled data has become a hot topic [1].

Various semi-supervised learning methods have been proposed [2]: EM with generative mixture methods, self-training, transductive support vector machines, and graph-based methods. Blum and Mitchell [3] introduced another semi-supervised learning paradigm named co-training, which trains two separate classifiers with the labeled data and lets them label the unlabeled examples for each other. Each classifier is retrained with the additional training examples given by the other classifier. In the standard co-training algorithm, the attributes should be split into two redundant views both of which are sufficient for perfect classification. Co-training makes strong assumptions on the splitting of features. Goldman and Zhou [4] proposed an algorithm to relax the conditions. It used two learners of different type but both takes the whole feature set, and essentially employing time-consuming statistic test to identify one learner's high confidence data points.

Zhou and Li [5] proposed the tri-training algorithm which uses three classifiers. If two of them agree on the classification of an unlabeled example, the classification is used to teach the third classifier. The approach thus avoids the need of explicitly

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 651–660, 2009.

measuring label confidence of any classifiers. It can be applied to dataset without different views, or different types of classifiers. Moreover, it is possible to exploit ensemble learning to help improve generalization.

In co-training style semi-supervised learning, it is obvious that the generalization ability degrades as the unlabeled examples are misclassified. Li and Zhou [6] performed the random forest ensemble learning algorithm to label the unlabeled examples and obtained better generalization. Therefore, the ensemble learning algorithm to determine the most confident examples by two or more classifiers might help improve the generalization ability of the learned hypothesis.

Ensemble learning algorithm succeeds in improving the accuracy of the whole when the component learners are with high accuracy as well as high diversity. Breiman [7] indicated that although Bagging with the help of bootstrap sampling could work well on unstable base learners such as decision trees and neural networks, it could hardly improve nearest neighbor classifiers. This is because nearest neighbor classifiers are robust with respect to perturbation of the training data. Zhou and Yu [8] proposed the FASBIR algorithm which builds ensembles through multimodal perturbation to obtain accurate and diverse *k*-nearest neighbor (*k*-NN) classifiers.

In this paper, we tackle the problem of how to classify the unlabeled examples through the multimodal perturbation ensemble method. In practice, the ensemble of *k*-NN classifiers with perturbation on training data, input attributes and learning parameters determines the confidence of unlabeled examples in each iteration and produces the final hypothesis. Experiments on UCI data sets show that the co-training style algorithm Co-BRSRP (**B**agging based **R**andom **S**ubspaces and **R**andom **P**arameters) for semi-supervised learning could effectively exploit unlabeled data to enhance the learning performance.

The remaining of this paper is organized as follows. Section 2 introduces the multimodal perturbation methods. Section 3 presents the Co-BRSRP algorithm. Section 4 reports the experimental study on UCI data set. Finally, an outlook and a discussion in section 5 conclude the paper.

2 Ensemble Method through Multimodal Perturbation

Krogh and Vedelsby [9] derived a famous equation $E = \overline{E} - \overline{A}$ that clearly demonstrates that the generalization ability of the ensemble is determined by the average generalization ability and the average ambiguity of the component learners. It means that the ambiguity in combination can be used to select new training data to be labeled in the semi-supervised learning schema. Although *k*-NN classifiers are very robust with respect to variation of the training data, the methods are sensitive to features, and to the chosen distance function. Therefore, we utilize multimodal perturbation method [8] to construct the ensembles of *k*-NN classifiers.

2.1 Random Subspace in Input Attributes

The random subspace method, introduced by Ho [10], randomly selects different feature dimension and constructs ensembles on multiple smaller subsets. For the *k*-NN algorithm, it means that only a randomly selected subset of the whole feature space

contributes towards the distance computation. Geometrically this is equivalent to projecting all the instances to the selected subspace, and finding the nearest neighbors in the projected distances. The label of new instances are classified via majority voting through the ensemble, where the *k*-NN classifiers are combined in each randomly selected subspace.

Given a *d*-dimension feature space, the feature vector can be represented as $X = \{x_1, x_2, \cdots, x_d\}$. Then the random subspace $X^s = \{(x_1, x_2, \cdots, x_s) \mid s < d\}$ with size *s* is randomly selected. For base learner $C_i (i = 1, \cdots, N)$, all instances are projected onto the chosen subspace X_i^s, where N is the number of nearest neighbor classifiers. By combining the *k*-NN classifiers which are trained in the subspace X_i^s respectively, discriminative power on the unlabeled examples is improved.

2.2 Random Parameters in Distance Metrics

We use the heterogeneous distance function HVDM [11] for the nearest neighbor classifiers as in FASBIR algorithm [8]. The distance between two input vectors x and y can be computed as follows:

$$HVDM_p(x, y) = (\sum_{a=1}^{m} d_a^p(x_a, y_a))^{1/p} \quad , \tag{1}$$

where *m* is the number of attributes. The function $d_a(x, y)$ returns a distance between the values *x* and *y* on attribute *a* and is defined as:

$$d_a(x, y) = \begin{cases} Minkowsky_a(x, y), & \text{if a is continuous .} \\ VDM_a(x, y), & \text{if a is nominal .} \end{cases} \tag{2}$$

The Minkowsky function shown in (3) is used to measure the distance between different instances described by continuous attributes. Here *x* and *y* are *d*-dimensional continuous attribute vectors.

$$Minkowsky_a(x, y) = (\sum_{n=1}^{d} \left| x_{a,n} - y_{a,n} \right|^p)^{1/p} \quad . \tag{3}$$

A simple version of the Value Difference Metric (VDM) shown in (4) is defined to deal with the nominal attributes. Let $N_{a,x}$ denote the number of instances that have value *x* for the nominal attribute *a*, $N_{a,x,c}$ denote the number of instances holding value x on *a* and outputting class*c*, and C is the number of output classes.

$$VDM_a(x, y) = (\sum_{c=1}^{C} \left| \frac{N_{a,x,c}}{N_{a,x}} - \frac{N_{a,y,c}}{N_{a,y}} \right|^p)^{1/p} \quad . \tag{4}$$

Note that HVDM can be adapted to different distance metrics through setting different values to p. For instance, an alternative function, the Manhattan distance

function, is obtained where $p=1$. Co-BRSRP achieves the diversity of $k-$NN learners by random selection of p values.

3 Co-BRSRP

In co-training style semi-supervised learning, base learners label the unlabeled instances for each other. Co-BRSRP combines the semi-supervised learning with ensemble learning to classify the unlabeled instances.

Let L and U denote the labeled instances set and unlabeled instances set respectively. In order to construct the ensemble C, the k-NN classifier $C_i(i=1,\cdots,N)$ can be trained through multimodal perturbation, where N is the number of classifiers. At first, we get labeled instances set L_i from L via bootstrap sampling. Then, random subspace X_i^s is selected from input feature space. Finally, we randomly set p_i value of the distance function. The initial classifier C_i with the random subspace X_i^s and the random parameter p_i is trained on the L_i data set.

C_i is retrained by the new labeled data set $L_i \cup L_i^{'}$, where $L_i^{'}$ is chosen from the unlabeled data set U and labeled by the ensemble C^* which denotes that all individual classifiers are combined except C_i. When each example in U is checked, C^* labels the new data for C_i if the number of component classifiers in C^* voting for a particular label exceeds a pre-set threshold θ.

If all of examples are added into the training set, the classification noise rate may be increased because of the mislabeled data. Fortunately, the condition for refreshing the training set has been obtained in the co-forest algorithm. According to Li and Zhou [6], in the two iterations ($(t-1)$-th and t-th, $t>1$), the requirement (5) should be satisfied, which is used by the ensemble C^* to determine whether an unlabeled example could be labeled and placed into the $L_i^{'}$ or not. It means that the classification error rate $\xi^t < \xi^{t-1}$ after C_i is trained on $L_i \cup L_{i,t-1}^{'}$ and $L_i \cup L_{i,t}^{'}$.

$$\frac{\hat{e}_{i,t}}{\hat{e}_{i,t-1}} < \frac{W_{i,t-1}}{W_{i,t}} < 1 \ . \tag{5}$$

Let $\hat{e}_{i,t}$ denote the error rate of C^* on $L_{i,t}^{'}$ and $\hat{e}_{i,t}W_{i,t}$ is the weighted number of mislabeled examples by C^* on $L_{i,t}^{'}$. Under the assumptions $\hat{e}_{i,t} < \hat{e}_{i,t-1}$ and $W_{i,t-1} < W_{i,t}$, $W_{i,t} < \frac{\hat{e}_{i,t-1}W_{i,t-1}}{\hat{e}_{i,t}}$ should be satisfied via getting the subsample from $L_{i,t}^{'}$ to make (5) hold.

Co-BRSRP algorithm repeats the following iterations until all of the component classifiers on $L_i^{'}$ do not change. Then the ensemble of the refined component classifiers produces the final hypothesis. Detailed pseudo-code for our Co-BRSRP algorithm is given in table 1.

Table 1. The Co-BRSRP algorithm

Algorithm: Co-BRSRP (L, U, T, N, P, θ)
Input: the labeled set L, the unlabeled set U, the test set T, the number of component learners N, the distance order set P, the confidence threshold θ

```
for i ∈ {1,⋯,N} do
    L_i ← BootstrapSample(L) , L_i^* ← RandomSubspace(L_i) ,
    p_i ← RandomSelect(P)
    C_i ← Learn(L_i^*, HVDM_{p_i}) %construct initial k-NN
end of for
t ← 0
Repeat until none of the learner component changes
    t ← t+1
    for i ∈ {1,⋯,N} do
        U'_{i,t} ← SubSampled(U)
        for x_u ∈ U'_{i,t} do
            if ( Confidence(C_i, x_u) > θ )
                L'_{i,t} ← L'_{i,t} ∪ x_u , W_{i,t} ← W_{i,t} + Confidence(C_i, x_u)
        end of for
    end of for
    for i ∈ {1,⋯,N} do
        if ( ê_{i,t} W_{i,t} < ê_{i,t-1} W_{i,t-1} )
            C_i ← Learn(L_i^* ∪ L'_{i,t}, HVDM_{p_i})  %refresh k-NN
    end of for
end of repeat
```

Output: $H^*(x) \leftarrow \arg\max_{y \in label} \sum_{i:C_i(x)=y} 1$ %majority voting

4 Experiments

For our experiments we used 12 data sets from UCI machine learning repository [12]. The detailed information of these data sets is tabulated in table 2. 10 runs of 10-fold cross validation is performed on each data set, and the results are averaged. In each fold, the training data are randomly separated into two groups, i.e. labeled set L and unlabeled set U. The unlabeled rates are set to 80%, 60%, 40% and 20% respectively. For instance, if training data consists of 1000 examples, the unlabeled rate 80% will produce 200 labeled examples while remaining 800 examples will be put into U without label.

In the experiments, ten k-NN classifiers are contained in the ensemble used by Co-BRSRP. Moreover, the parameter k is set to 1. The distance metric is randomly selected from {1, 2, 3} as done in [8]. Here the random subspace rate is set to 0.66, i.e.

Table 2. Experimental data sets

Data set	Size	Attributes		Class
		Continuous	Nominal	
colic	368	7	15	2
diabetes	768	8	0	2
hepatitis	155	6	13	2
vote	435	0	16	2
sonar	208	60	0	2
ionosphere	351	34	0	2
segment	2310	19	0	7
credit-a	690	6	9	2
anneal	898	6	32	6
heart-c	303	6	7	5
breast-w	699	9	0	2
soybean	683	0	35	19

each component learner trains on about 2/3 attributes. The confidence threshold is set to θ=0.75 as done in [6], i.e. an example in U can be confidently labeled if more than 3/4 of all learners agree on a particular label.

For comparison, two prevailing semi-supervised algorithms, i.e. *co-training* and *self-training*, are conducted. The self-training is an incremental algorithm that the most confidently predicted unlabeled data is converted and added as a training example by itself. The standard co-training algorithm applies the setting where two disjoint sets with almost equal size are randomly spited. Note that the termination criteria of self-training and co-training are modified to be similar to that of Co-BRSRP. Two degenerated variants of Co-BRSRP are also compared. *Co-BRS* (Bagging based Random Subspaces) perturbs the training data and the input attributes without perturbation on the learning parameters. *Co-BRP* (Bagging based Random Parameters) perturbs the training data and the learning parameters without perturbation on the input attributes. *BRSRP* is used as the baseline for comparison with multimodal perturbation, just like what is done in Co-BRSRP, but the component learners are trained only on labeled data set L without unlabeled examples. Moreover, the performance of single k-NN classifier and standard ensemble algorithm *Bagging* [7] trained on L are also evaluated.

Table 3 to Table 6 present the average error rates of Co-BRSRP and the other compared algorithms under different unlabeled rates, where the best performances on each data set have been boldfaced. Note that the experimental results have been truncated. The results show that Co-BRSRP achieves the best performance on 67% (32/48) data sets under different unlabeled rates. In detail, BRSRP employing the multimodal perturbation on labeled data set L achieves an overall 0.140 average error rate. It is obvious that BRSRP is better than Single k-NN and Bagging in supervised learning. However, Co-BRSRP uses the unlabeled data to reduce the average error rate down to 0.133 under different unlabeled rates. These observations indicate that Co-BRSRP

with multimodal perturbation can effectively use the unlabeled data to improve classification generalization.

Moreover, Table 3 to Table 6 compare the performance improvements of Co-BRSRP and the degenerated variants algorithms, i.e. Co-BRS and Co-BRP. For comparison, BRSRP whose component classifiers are trained only on L is used as the baseline. In the columns of three semi-supervised learning algorithms, *Final* shows the average error rates with the help of the unlabeled data. The performance improvement is denoted by *Improv.*, which presents the reduction of the average error rates over that of the baseline. The highest improvement on each data set has been boldfaced. From these results, we certainly conclude that Co-BRSRP successfully utilizes the unlabeled data to improve the classification accuracy, as the performance improvement achieved by Co-BRSRP is greater than the others on most of the data sets. In detail, Co-BRSRP achieves an overall 6.7% performance improvement while Co-BRS obtains only 1.5% and Co-BRP is even worse than BRSRP.

Table 3. Comparison on average error rates under the unlabeled rate of 80%

Data set	Supervised			Self-training	Co-training	Co-BRS		Co-BRP		Co-BRSRP	
	Single	Bagging	BRSRP			Final	Improv.	Final	Improv.	Final	Improv.
colic	.224	.204	.201	.218	.299	.191	5.0%	**.189**	**6.0%**	**.189**	**6.0%**
diabetes	.319	.307	.300	.319	.336	.285	4.7%	.304	-1.5%	**.284**	**5.0%**
hepatitis	.210	.209	.201	.210	.281	.193	4.6%	.204	-1.5%	**.192**	**4.8%**
vote	.072	.068	.072	.070	.117	.066	9.4%	.078	-8.0%	**.063**	**12.4%**
sonar	.273	.281	.276	.271	.312	**.270**	**2.3%**	.279	-0.9%	.273	1.2%
ionosphere	.178	.178	.142	.193	**.139**	.155	-8.9%	.169	-18.5%	.145	**-2.0%**
segment	.073	.076	.061	.085	.116	.056	7.4%	.060	1.0%	**.055**	**8.9%**
credit-a	.198	.168	.176	.203	.311	.176	-0.1%	.179	-1.8%	**.167**	**5.5%**
anneal	.056	.057	.052	.066	.132	.053	-0.6%	.053	-1.5%	**.050**	**3.8%**
heart-c	.261	.246	.221	.262	.307	.214	3.4%	.228	-3.3%	**.208**	**6.1%**
breast-w	.056	.054	.045	.055	.080	.036	20.7%	.051	-15.5%	**.035**	**21.1%**
soybean	.302	.330	.203	.295	.335	.244	-20.0%	.193	-4.9%	**.171**	**16.1%**
Average	.185	.181	.163	.187	.230	.161	2.3%	.166	-4.2%	**.153**	**7.4%**

Table 4. Comparison on average error rates under the unlabeled rate of 60%

Data set	Supervised			Self-training	Co-training	Co-BRS		Co-BRP		Co-BRSRP	
	Single	Bagging	BRSRP			Final	Improv.	Final	Improv.	Final	Improv.
colic	.210	.189	.194	.214	.294	.191	1.6%	.189	2.4%	**.183**	**5.5%**
diabetes	.311	.306	.276	.314	.331	**.271**	**1.8%**	.295	-6.7%	**.271**	**1.9%**
hepatitis	.218	.210	.193	.210	.285	.188	2.6%	.194	-0.9%	**.185**	**4.2%**
vote	.071	.065	.063	**.057**	.113	.063	0.3%	.077	-21.8%	.060	**5.7%**
sonar	.195	**.204**	.214	.203	.236	.218	-2.0%	.212	**1.2%**	.216	-0.9%
ionosphere	.147	.146	.116	.160	**.109**	.129	-10.9%	.136	-17.3%	.120	**-3.6%**
segment	.051	.052	.058	.059	.096	**.046**	**19.4%**	.050	13.7%	**.046**	**20.0%**
credit-a	.196	.170	.173	.200	.304	.174	-0.3%	.186	-7.4%	**.165**	**4.6%**
anneal	.035	.036	.030	.044	.106	.029	2.3%	.033	-9.0%	**.028**	**8.0%**
heart-c	.244	.238	.207	.246	.295	.203	1.8%	.219	-5.7%	**.198**	**4.4%**
breast-w	.051	.051	.040	.049	.087	**.034**	**17.1%**	.047	-15.1%	**.034**	**16.3%**
soybean	.206	.234	.115	.254	.270	.138	-19.3%	.114	0.9%	**.097**	**16.2%**
Average	.161	.158	.140	.168	.210	.140	1.2%	.146	-5.5%	**.133**	**6.9%**

Table 5. Comparison on average error rates under the unlabeled rate of 40%

Data set	Supervised			Self-training	Co-training	Co-BRS		Co-BRP		Co-BRSRP	
	Single	Bagging	BRSRP			Final	Improv.	Final	Improv.	Final	Improv.
colic	.200	**.179**	.189	.205	.281	.186	2.0%	.188	0.9%	.184	**3.0%**
diabetes	.299	.293	**.278**	.309	.325	.285	-2.4%	.303	-8.8%	.282	**-1.3%**
hepatitis	.197	.195	.181	.218	.273	.188	-3.9%	.191	-5.2%	**.171**	**5.5%**
vote	.068	.063	.060	**.055**	.109	.056	**6.9%**	.069	-15.6%	.056	5.9%
sonar	**.169**	.175	.192	.191	.191	.189	1.6%	.189	**1.7%**	.191	0.6%
ionosphere	.137	.135	.109	.145	**.098**	.112	-3.0%	.126	-15.9%	.107	**2.2%**
segment	.042	.043	.053	.049	.084	.039	26.0%	.038	27.9%	**.036**	**31.3%**
credit-a	.191	.167	.180	.193	.296	**.160**	**10.8%**	.179	0.7%	**.160**	**10.8%**
anneal	.023	.026	.027	.030	.088	.022	16.1%	.025	6.7%	**.020**	**23.6%**
heart-c	.243	.227	**.185**	.243	.283	.191	-3.2%	.202	-9.0%	.190	**-2.8%**
breast-w	.050	.049	.039	.049	.081	**.033**	15.0%	.043	-10.3%	**.033**	**15.8%**
soybean	.165	.185	.084	.199	.226	.105	-24.6%	.097	-14.2%	**.081**	**4.4%**
Average	.149	.145	.131	.157	.194	.131	3.4%	.137	-3.43%	**.126**	**8.2%**

Table 6. Comparison on average error rates under the unlabeled rate of 20%

Data set	Supervised			Self-training	Co-training	Co-BRS		Co-BRP		Co-BRSRP	
	Single	Bagging	BRSRP			Final	Improv.	Final	Improv.	Final	Improv.
colic	.196	**.169**	.175	.200	.272	.187	-6.9%	.185	-5.7%	.178	**-1.5%**
diabetes	.298	.295	.299	.303	.322	.280	6.5%	.308	-2.9%	**.279**	**6.8%**
hepatitis	.198	.198	.174	.212	.272	.172	1.0%	.185	-6.8%	**.167**	**3.6%**
vote	.075	.066	.057	.070	.112	**.055**	**4.5%**	.068	-18.5%	.056	2.4%
sonar	.150	**.146**	.159	.163	.161	.160	**-0.4%**	.169	-5.9%	.169	-6.2%
ionosphere	.129	.129	.098	.136	**.090**	.107	-9.4%	.117	-19.4%	.094	**4.1%**
segment	.034	.036	.031	.042	.078	.034	-9.7%	.035	-12.9%	**.030**	**3.3%**
credit-a	.191	.169	.177	.195	.277	.169	4.5%	.182	-3.0%	**.164**	**7.1%**
anneal	**.015**	.018	.024	.022	.080	.021	12.7%	.020	18.9%	.020	**20.1%**
heart-c	.233	.222	.197	.234	.276	.194	1.7%	.202	-2.3%	**.190**	**3.8%**
breast-w	.049	.049	.037	.049	.075	.037	1.3%	.047	-27.0%	**.035**	**6.4%**
soybean	.144	.160	.078	.168	.201	.092	-18.7%	.078	0.0%	**.077**	**0.8%**
Average	.143	.138	.126	.149	.185	.126	-1.1%	.133	-7.1%	**.121**	**4.3%**

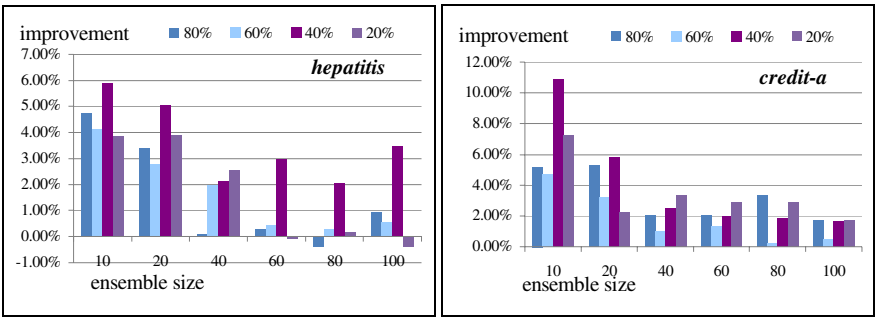


Fig. 1. Performance improvements over different ensemble sizes varying from 10 to 100

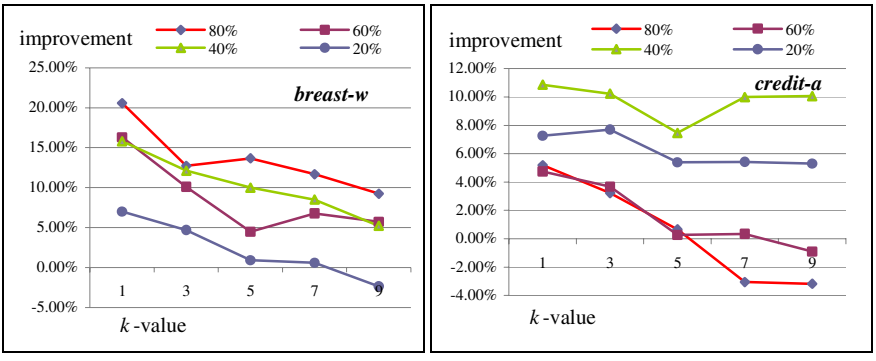


Fig. 2. Performance improvement over different k values varying from 1 to 9

Figure 1 shows the average performance improvements of Co-BRSRP on the two data sets, i.e. *hepatitis* and *credit-a* with different ensemble sizes, where the value of N is set from 10 to 100. In the figure, although costly, Co-BRSRP cannot perform well on the large ensemble size. However, the highest improvements are always achieved on the small ensemble sizes. For instance, the negative improvement even appears in the hepatitis data set when the ensemble size is 80 under the unlabeled rate 80%. The results indicate that, with large size of the ensemble, the diversity between individual classifiers may drops down and hence it can result in a decrease in the classification accuracy on the unlabeled examples.

Figure 2 shows that the average performance improvements of Co-BRSRP on the two data sets, i.e. *breast-w* and *credit-a* with different k values, where the number of k-NN classifiers is fixed to 10. It can be seen that Co-BRSRP obtains the best performance while the value of k is not too big. However, the average performance improvements degrade with the rising of the value of k, e.g. when the value of k is set to 9 on *credit-a*, the average classification accuracy reduces by 3.18% under the unlabeled rate 80% and by 0.91% under the unlabeled rate 60% respectively. These results show that the generalization ability is better under smaller k values.

5 Conclusions

This paper proposes to use the ensemble method with multimodal perturbation to label the unlabeled data in the co-training style semi-supervised algorithm. In detail, Co-BRSRP employs the perturbation on the training data, the input attributes and the learning parameters together to build the ensemble of k-NN classifiers with high diversity, which labels the unlabeled instances for each other. Experiments show that such an approach could effectively utilize the unlabeled instances to improve generalization ability.

Ensemble learning technology has been successfully introduced into semi-supervised learning. It seems that different ensemble methods might contribute to semi-supervised learning. Exploring more powerful ensemble methods for co-training style semi-supervised learning is an interesting issue for future work.

Acknowledgements. This work was support by the National Natural Science Foundation of China under the Grant Nos. 60663004, 60863011, the Yunnan Province Natural Science Foundation Key Project under the Grant No. 2008CC023, and the Yunnan Province Society Development Technology Project under the Grant No. 2008-16.

References

1. Olivier, C., Bernhard, S., Alexander, Z.: Semi-Supervised Learning. MIT Press, Cambridge (2006)
2. Zhu, X.: Semi-supervised Learning Literature Survey. Technical Report, University of Wisconsin, Madison (2005)
3. Blum, A., Mitchell, T.: Combining Labeled and Unlabeled Data with Co-training. In: 11th Annual Conference on Computational Learning Theory, Wisconsin, MI, pp. 92–100 (1998)
4. Goldman, S., Zhou, Y.: Enhancing Supervised Learning with Unlabeled Data. In: 17th International Conference on Machine Learning, San Francisco, CA, pp. 327–334 (2000)
5. Zhou, Z.H., Li, M.: Tri-training: Exploiting Unlabeled Data Using Three Classifiers. IEEE Transactions on Knowledge and Data Engineering 17, 1529–1541 (2005)
6. Li, M., Zhou, Z.H.: Improve Computer-aided Diagnosis with Machine Learning Techniques Using Undiagnosed Samples. IEEE Transactions on Systems, Man and Cybernetic - Part A 37, 1088–1098 (2007)
7. Breiman, L.: Bagging Predictors. Machine Learning 24, 123–140 (1996)
8. Zhou, Z.H., Yu, Y.: Ensembling Local Learners through Multimodal Perturbation. IEEE Transactions on Systems, Man, and Cybernetics - Part B 35, 725–735 (2005)
9. Krogh, A., Vedelsby, J.: Neural Network Ensembles, Cross Validation and Active Learning. In: Tesauro, G., Touretzky, D.S., Leen, T.K. (eds.) Advances in Neural Information Processing Systems, vol. 7, pp. 231–238. MIT Press, Cambridge (1995)
10. Ho, T.K.: Nearest Neighbors in Random Subspaces. In: Amin, A., Dori, D., Pudil, P., Freeman, H. (eds.). LNCS, pp. 640–648. Springer, Berlin (1998)
11. Wilson, D.R., Martinez, T.R.: Improved Heterogeneous Distance Functions. Journal of Artificial Intelligence Research, 1–34 (1997)
12. Blake, C., Keogh, E., Merz, C.J.: UCI Repository of Machine Learning Databases, `http://www.ics.uci.edu/~mlearn/MLRepository.html`

Weights Updated Voting for Ensemble of Neural Networks Based Incremental Learning

Jianjun Liu, Shengping Xia, Weidong Hu, and Wenxian Yu

State Lab of Automatic Target Recognition, National University of Defense Technology, Changsha, 410073, China

Abstract. In this paper, an ensemble of neural networks based incremental learning algorithm with weights updated voting is described. The algorithm defines the class kernel function of the training database of the component neural network in the ensemble. The voting weights are updated based on the distance between the test instance and the kernel function. This method can adaptively update the voting weights according to the classification performance of the component neural network on the test pattern and it is more optimal than the stable weights voting strategy. Experimental results show that the ensemble of neural networks based incremental learning algorithm with weights updated voting is more promising than that with stable weights voting rule.

Keywords: Machine learning, Neural networks, Neural network ensemble, Incremental learning.

1 Introduction

It is well known that the performance of neural network classifiers relies heavily on the availability of the representative set of training examples. But acquisition of a representative training data set is difficult and time consuming in many pattern recognition applications. As a result, it is common to get such representative data in small batches over a period of time. In such settings, it is necessary to learn incrementally. Incremental learning[1] accommodates new data without compromising classification performance on preceding data. Incremental learning in neural network classifiers can be achieved by modifying the classifier weights or growing/pruning the classifier architecture[1,2]. Also, it can be achieved by combining neural network classifiers[3]. Due to the promising potency of incremental learning, it is worth to research on neural network ensemble.

Neural network ensemble is a learning paradigm where a collection of a finite number of neural networks is trained for the same task[4]. The generalization ability of a neural network system can be significantly improved through ensembling a number of neural networks[5]. Inspired by the AdaBoost[6] algorithm, one of the most successful implementations of the ensemble approach, Polikar et al. propose an ensemble of neural networks based incremental learning algorithm, named learn++[3]. The algorithm generates a set of neural networks and combines them through weighted majority voting of the classes predicted by the individual hypotheses. The hypotheses are generated by training weak neural network classifiers, using instances

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 661–669, 2009.

drawn from iteratively updated distributions of the training database. The distribution updating rule used by learn++ is designed to accommodate additional data sets, in particular those that introduce previously unseen classes. Each classifier is trained using a subset of examples drawn from a weighted distribution that gives higher weights to examples misclassified by the previous ensemble.

The voting weights are stable in learn++ algorithm[3], which are determined on the performances of the component neural network classifiers on their own training instances. This voting strategy is suboptimal. A more optimal voting rule is that if a component classifier in the ensemble is likely to correctly classify the test instance, the component classifier would have higher voting weight. Otherwise, the voting weight would be lower. In pattern recognition theory, the class kernel function distance, which is the distance between the test instance and the class kernel function of the training database of the classifier, can be used to estimate the classification performance of the classifier on the test instance. In this paper, we define the class kernel functions of the training data sets of the component classifiers, and update the voting weights by computing the kernel function distances between the test instance and the class kernel functions. Experimental results show that the weights updated voting rule is more promising than the stable weights voting strategy.

2 Neural Network Ensemble Based Incremental Learning

2.1 Neural Network Ensemble

Neural network ensemble has been widely researched due to its excellent generalization ability. Recently research on neural network ensemble is concentrated on training a number of component neural networks and combining the component predictions.

The most prevailing approaches for training component neural networks are Boosting[8] and Bagging[9]. Boosting[8] generates a series of component neural networks whose training sets are determined by the performance of former ones. Training instances that are wrongly predicted by former neural networks will have more opportunity to be selected in the training of later networks. Freund et al propose AdaBoost[6], which is improved from Boosting and widely used in real applications. Bagging[9] is based on bootstrap sampling. It generates several training sets from the original training set and then trains a component neural network from each of those training sets.

The most prevailing approaches for combining the predictions of the component neural networks are plurality voting and majority voting[5] for classification tasks, and simple averaging and weighted averaging for regression tasks. In addition, Jimenez[10] uses dynamic weights determined by the confidence of the component networks to combine the predictions.

Research in neural networks ensemble is predominantly concentrated on improving the generalization performance in complex problems. Feasibility of neural networks ensemble in incremental learning has been largely unexplored. Inspired by AdaBoost algorithm, Polikar et al[3] propose a neural networks ensemble based incremental learning approach, named learn++.

2.2 Learn++ Algorithm

Learn++[3] generates an ensemble of weak neural network classifiers to incrementally learn new information that may later become available. Each of the weak classifier trained with different subsets of the data. For each database $D_k, k = 1, \cdots, K$ that becomes available, the inputs to learn++ are (1) m_k training data instances $S_k = \{(x_i, y_i) \mid i = 1, \cdots, m_k\}$ (2) a weak classification algorithm to generate weak classifiers, (3) an integer T_k specifying the number of classifiers to be generated for that database.

Learn++[3] initializes a set of weights for the training data, and a initial distribution obtained from the weights. At each iteration t, $t = 1, \cdots, T_k$, a training subset TR_t and a test subset TE_t are drawn according to the current distribution, where $S_k = TR_t \bigcup TE_t$. The weak classification algorithm is trained with the training subsets. A hypothesis h_t is obtained as the t^{th} classifier, whose error ε_t is computed on the database S_k simply by adding the distribution weights of the misclassified instances

$$\varepsilon_t = \sum_{i:h_t(x_i) \neq y_i} D_t(i). \tag{1}$$

If $\varepsilon_t > \frac{1}{2}$, h_t is discarded and a new TR_t and TE_t are selected. Else the error is normalized and computed as

$$\beta_t = \varepsilon_t / (1 - \varepsilon_t), \ 0 \leq \beta_t \leq 1. \tag{2}$$

Hypotheses generated in all previous iterations are then combined using weighted majority voting[11] to form a composite hypothesis H_t

$$H_t = \arg\max_{y \in Y} \sum_{t:h_t(x)=y} \log \frac{1}{\beta_t}, \tag{3}$$

where the sum of weights associated with each classifier is computed for every class present in the classification task. A higher weight is given to classifiers that perform better on their specific training sets. The composite hypothesis H_t is obtained by assigning the class label to an instance x_i that receives the largest total vote. The composite error made by H_t is then computed as

$$E_t = \sum_{i=1}^{m} D_t(i)[|H_t(x_i) \neq y_i|]. \tag{4}$$

where $[|\bullet|]$ evaluates to 1, if the predicate holds true. A normalized composite error B_t is computed as:

$$B_t = E_t /(1-E_t),\ 0 \le B_t \le 1. \tag{5}$$

The weights of instances are updated to obtain D_{t+1}, which is used for the selection of the next training and testing subsets, TR_{t+1} and TE_{t+1}, respectively. The distribution update rule which comprises the heart of the algorithm is given by

$$w_{t+1}(i) = w_t(i) \times \begin{cases} B_t, H_t(x_i) = y_i \\ 1, otherwise \end{cases}. \tag{6}$$

This rule reduces the weights of those instances that are correctly classified by the composite hypothesis H_t, so that their probability of being selected into the next training subset is reduced. When normalized during iteration $t+1$, the weights of misclassified instances are increased relative to the rest of the dataset. After T_k hypotheses are generated for each database D_k, the final hypothesis is obtained by weighted majority voting[11] of all composite hypotheses

$$H_{final} = \arg\max_{y \in Y} \sum_{k=1}^{K} \sum_{t:H_t(x)=y} \log \frac{1}{B_t}. \tag{7}$$

2.3 Weights Updated Voting Rule

From section 2.2, we have seen, in learn++, the voting weights are stable and determined on individual performances of hypotheses on their own training data subset. This voting rule is sub optimal. A more optimal voting rule is that which hypotheses in the ensemble are likely to correctly classify the test instance, the voting weights for those hypotheses would be higher. Otherwise the voting weights would be lower. Gangardiwala et. al[7] propose an algorithm to dynamically update the voting weights based on the Mahalanobis distance between a test instance and the training data used to train the component hypotheses.

In pattern recognition theory, the class kernel function distance can be well used to estimate the classification performances of the component classifiers in the ensemble on the test instance. In this paper, we dynamically update the voting weights based on the class kernel function distance. We define the class kernel functions of the training database of the component neural networks in the ensemble system and determine the voting weights by computing the class kernel function distances. If the kernel function distance between the test instance and the class kernel function of the training database of a component neural network is small, the component neural network would be likely to correctly classify the test instance, the voting weight of the component neural network should be higher. Otherwise the voting weight should be lower.

Assumed that there are c classes of training data in TR_t^k, where TR_t^k is the training subset of the t^{th} iteration of database S_k, $k = 1, \cdots, K$, $t = 1, \cdots, T_k$, then $TR_t^k = \bigcup_{i=1}^{c} TR_{ti}^k$. The kernel function of the i^{th} class training data in TR_t^k is defined as

$$K_i(x, V_i) = \frac{1}{(2\pi)^{n_i/2} \left| C_{ti}^k \right|^{1/2}} \exp[-\frac{1}{2}(x - m_{ti}^k)'(C_{ti}^k)^{-1}(x - m_{ti}^k)]. \tag{8}$$

where m_{ti}^k is the mean of TR_{ti}^k. C_{ti}^k is the covariance matrix of TR_{ti}^k.

$$m_{ti}^k = \frac{1}{n_i} \sum_{x_i \in w_i} x_i . \tag{9}$$

$$C_{ti}^k = \frac{1}{n_i - 1} \sum_{x_i \in w_i} (x_i - m_{ti}^k)(x_i - m_{ti}^k)'. \tag{10}$$

where n_i is the number of instances in the i^{th} class of the training subset. Then the kernel function distance between instance x and the class kernel function of the i^{th} class of the training database TR_t^k is

$$D_{ti}^k = \frac{1}{2}(x - m_{ti}^k)'(C_{ti}^k)^{-1}(x - m_{ti}^k) + \frac{1}{2}\log \left| C_{ti}^k \right|. \tag{11}$$

After the c kernel function distances of the training database TR_t^k have been computed. We get the voting weight W_t^k of the component neural network that is generated in the t^{th} iteration of the database S_k

$$W_t^k = \frac{1}{\min(D_{ti}^k)}, \ i = 1, \cdots, c. \tag{12}$$

From equation (12), we can see that the voting weight of the t^{th} component neural network is the reciprocal of the minimum kernel function distance. If the distance is small, the component neural network is likely to correctly classify the test instance. Its voting weight is high. Otherwise if the distance is large, its voting weight is low. This is more optimal than the fixed weights voting rule. Also the computing of the distance only need the means and the covariance matrices, the original training database of the component neural networks need not be stored in the memory, which is in accordance with the definition of the incremental learning.

Note that in the definition of the kernel function, we assume the training data sets of the component neural networks are drawn from a normal distribution, which may not be the case in general. However, the experimental results in section 3 show that the weights updated voting rule is more favorable than the weights fixed voting rule.

The steps of the neural network ensemble based incremental learning algorithm with weights updated voting are as follows.

Inputs: (1) $S_k = \{(x_i, y_i) \mid i = 1, \cdots, m_k\}$, a sequence of m_k training instances. (2) a weak classification algorithm to generate weak classifiers, (3) an integer T_k specifying the number of classifiers to be generated for that database.

Do for each of the database $S_k, k = 1, \cdots, K$:

For $t = 1, \cdots, T_k$, do step1 to step5, where t is the t^{th} iteration.

Step1. Initialize the weights $w(i)$ for the database S_k, and a distribution D_t is obtained

Step2. Draw a training subset TR_t and a test subset TE_t according to D_t.

Step3. Train the weak classification algorithm with TR_t and a hypothesis h_t is obtained. Compute the error ε_t of h_t on the database S_k using equation (1). If $\varepsilon_t > 1/2$, h_t is discarded. Go to sep2 and a new TR_t and TE_t are selected. Compute normalized error β_t using equation (2).

Step4. Combine the hypotheses generated in previous iterations to form a composite hypothesis H_t using equation (3), the voting weights are obtained using equation (12). Compute the composite error E_t using equation (4). If $E_t > 1/2$, go to step2.

Step5. Compute normalized composite error B_t using equation (5). Update the distribution D_t of the weights for the database using equation (6).

After T_k hypotheses are generated for each S_k, the final hypothesis is obtained using equation (7), where the voting weights are calculated using equation (12).

3 Simulation Results

We compare the performances of the neural network ensemble based incremental learning with weights updated voting algorithm to the original learn++ on IRIS database and a real radar database. To simulate incremental learning, the training is done in sessions, where only the most recently available database is shown to the algorithm during the current training session. We use back-propagation neural network training algorithm as the weak classification algorithm, whose weakness can be easily controlled via network size and error goal.

3.1 IRIS Database

IRIS database[12] was created by Fisher. R.A in 1936, which is widely used in pattern recognition experiments. The database consists of 4 features and a total of 150 instances from 3 classes. Each class has 50 instances. 90 instances were used for training and all remaining instances were used for validation. IRIS database was used to evaluate the performance on incremental learning without introducing new classes. The training database of 90 instances was divided into three subsets, $S_1 \sim S_3$, each with 30 instances containing all 3 classes to be used in three sessions. In each training session, only one of these datasets was used. Table 1 and 2 illustrate the training and generalization performances of neural network ensemble based incremental learning with weights updated voting algorithm and the original learn++ algorithm. Each column indicates the performance on the current and previous training datasets as additional data were introduced. Previous datasets were not used for training in subsequent training sessions, but they were only used to evaluate the algorithm performance on previously seen instances. The last rows show the classification performance on the validation dataset.

Table 1. Neural network ensemble based incremental learning with weights updated voting performance on IRIS database

Dataset	Training 1	Training 2	Training 3
S1	96.83%	95.90%	94.79%
S2	---	95.87%	93.66%
S3	---	---	92.57%
Test	82.58%	86.42%	93.94%

Table 2. Lean++ performance on IRIS database

Dataset	Training 1	Training 2	Training 3
S1	96.32%	94.98%	92.84%
S2	---	95.23%	92.14%
S3	---	---	90.91%
Test	80.17%	83.39%	91.72%

From table 1 and 2 we can see that the two algorithms achieved incremental learning ability. Neural network ensemble based incremental learning with weights updated voting performed better than the original learn++ algorithm. Due to stability-plasticity dilemma, there were some decline in training performances over three training sessions in the two algorithms.

3.2 Real Radar Database

The radar database consists of 41 features which are the video features extracted from the target return and the skeleton features extracted from the target micro B image. The real radar database was used to test the performances of the algorithms on incremental learning when new classes were introduced. The database consisted of 3839 instances from 3 classes, which was divided into three training datasets, S_1 through

Table 3. Neural network ensemble based incremental learning with weights updated voting performance on real radar database

Dataset	Training 1	Training 2	Training 3
S1	90.63%	87.26%	86.57%
S2	---	89.43%	88.73%
S3	---	---	88.28%
Test	64.85%	71.57.%	87.46%

Table 4. Lean++ performance on real radar database

Dataset	Training 1	Training 2	Training 3
S1	87.24%	83.19%	82.53%
S2	---	85.24%	83.21%
S3	---	---	84.64%
Test	64.09%	70.17.%	83.93%

S_3, and a validation dataset. S_1 had 1800 instances from each of the classes 1 and 2. S_2 had 1800 instances from each of the classes 1 and 3. S_3 had 1800 instances from all three classes. The validation set had 1200 instances from all three classes. Table 3 and 4 illustrate the training and generalization performances of neural network ensemble based incremental learning with weights updated voting algorithm and the original learn++ algorithm.

Table 3 and table 4 show that the two algorithms were able to learn the new information and the new classes. Neural network ensemble based incremental learning with weights updated voting algorithm performed better than the original learn++ algorithm for the real radar database, and it was able to retain more of its previously acquired information than the original learn++ algorithm along the stability-plasticity dilemma.

4 Conclusions

This paper introduces a neural network ensemble based incremental learning with weights updated voting algorithm. The voting weights updating rule is based on the distance between the test instance and the class kernel function of the training database of the component neural network in the ensemble. If the distance is small, the component neural network is likely to correctly classify the test instance. Then its voting weight is high. Otherwise if the distance is large, its voting weight is low. It is more optimal than the fixed weights voting rule in the original learn++ algorithm. The effectiveness of the weights updating voting rule is demonstrated in two experiments on IRIS dataset and real radar dataset.

References

1. Fu, L.: Incremental Knowledge Acquisition in Supervised Learning Networks. IEEE Transactions on Systems, Man and Cybernetics-Part A: System and Humans 6, 801–809 (1996)

2. Zhang, B.: An Incremental Learning Algorithm That Optimizes Network Size and Sample Size in One Trial. In: Proc. IEEE International Conference on Neural Network, pp. 215–220 (1994)
3. Robi, P.: Learn++: an Incremental Learning Algorithm for Supervised Neural Networks. IEEE Transactions on Systems, Man, and Cybernetics-part C: Applications and Reviews 31, 497–508 (2001)
4. Sollich, P., Krogh, A.: Learning with Ensembles: How Overfitting Can be Useful. In: Touretzky, D., Mozer, M., Hasselmo, M. (eds.) Advances in Neural Information Processing Systems 8, pp. 190–196. MIT Press, Cambridge (1996)
5. Hansen, L., Salamon, P.: Neural Network Ensembles. IEEE Transactions on Pattern Analysis and Machine Intelligence 12, 993–1001 (1990)
6. Freund, Y., Schapire, R.: A Decision-theoretic Generalization of On-line Learning and an Application to Boosting. Journal of Computer and System Sciences 55, 119–139 (1997)
7. Aliasgar, G., Robi, P.: Dynamically weighted majority voting for incremental learning and comparison of three Boosting based approached. In: Proceedings of International Joint Conference on Neural Networks, Montreal, Canada, pp. 1131–1136 (2005)
8. Schapire, R.: The Strength of Weak Learnability. Machine Learning 5(2), 197–227 (1990)
9. Breiman, L.: Bagging Predictors. Machine Learning 24(2), 123–140 (1996)
10. Jimenez, D., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptively mixtures of local experts. Neural Computation 3(1), 79–87 (1991)
11. Littlestone, N., Warmuth, M.: Weighted Majority Algorithm. Inform. Comput. 108, 212–261 (1994)
12. Fisher, R.A.: The Use of Multiple Measurements in Taxonomic Problems. Annual Eugenics, Part II 7, 179–188 (1936)

Numerical Learning Method for Process Neural Network

Tianshu Wu, Kunqing Xie, Guojie Song *, and Xingui He

Key laboratory of Machine Perception, Ministry of Education,
Peking University, Beijing 100871, China
gjsong@pku.edu.cn

Abstract. Process neural network (PNN) dealing with process inputs is widely used. Currently, the learning method of PNN is mainly based on base functions expansion. However, selecting base functions and their parameters is much difficult, and moreover, the corresponding learning method is time consuming due to integral with numbers of base functions. A numerical learning method (NL) for PNN was proposed in this study. It represented PNN's inputs and weights functions in numerical forms and trained the network in a numerical way so that NL avoided the selections of base functions and their parameters. Experiments showed that NL based PNN was more accurate and had lower computation complexity.

Keywords: Process neural network, Base function expansion, Numerical learning method.

1 Introduction

Research of neural networks has been lasting for several decades and there are many kinds of neural network models, such as perceptron [1], self organization mapping networks [2]. In recent years, more neural network models with new structures, new computing or training methods have been proposed, including RBF networks [3], fuzzy networks [4] etc. All these models are widely used in scientific fields and practical applications, such as machine learning [5], pattern recognition [6], industrial control [7].

For all the above models, the inputs of the networks are constant independent of time. However, the trends of inputs are usually meaningful because systems outputs are related to inputs accumulation for a period of time. He Xingui et al. proposed process neural network (PNN) dealing with process inputs [8]. The inputs and weights of PNN can be time-varying functions, representing the inputs processes. Further, some new forms of PNN arise, e.g. multi-aggregation PNN [9] and complex number PNN [10]. PNN has been applied in many real scenarios, such as simulation of oil reservoir exploitation [11], worm harm prediction [12], and churn prediction in mobile communication[13].

* Corresponding author.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 670–678, 2009.

Currently, the widely used learning method of PNN expands the weights functions to coefficients by base functions; the coefficients are gained after training[8]. We call this learning method base function learning method (BFL). However, there are difficulties in applying BFL:

- Base functions and parameters: It is hard to choose suitable basic functions and their parameters before knowing the concrete forms of weights functions.
- Computational complexity: The computational complexity of BFL depends on the number of base functions. The number is usually large and will grow by exponential order with the increase of dimension and resolution of the problem.

To tackle the shortcoming of the existing learning method, we proposed a numerical learning method (NL) for PNN in this study. In a numerical way, NL represented the inputs and weights functions and trained the network, so that selections of base functions and their parameters were avoided. Computational complexity of NL mainly depended on inputs dimension increasing linearly with problems' dimension. Experiments showed that NL based PNN was more accurate than those based on BFL and the time complexity of NL was low.

The rest of this paper was organized as follows. Section 2 introduced PNN theory and the base function learning method. Section 3 presented the numerical learning method. Experiments were reported in section 4. Section 5 was conclusion.

2 Preliminaries

Before introducing the proposed learning method, we first review PNN theory and base function learning method (BFL) briefly.

2.1 Process Neural Network

A process neuron (Fig. 1) has similar structure with a traditional artificial neuron. They are different mainly in two aspects. First, inputs and weights of the process neuron could be time-varying functions, which enables PNN to deal with process inputs. Second, the process neuron has both time accumulation and space aggregation operators(see below), whereas the traditional artificial neuron has only the space aggregation operator. Relation between inputs and outputs of the process neuron can be expressed as

$$y = f\left(\sum\left(\int (W(t), X(t)) - \theta\right)\right) \tag{1}$$

where $X(t)$ is the vector of inputs functions and $W(t)$ is the vector of weights functions; y is the output; θ is the threshold; $f(\cdot)$ is the excitation function; $\int$ is a time accumulation operator, such as integral; $\sum$ is a space aggregation operator such as sum, max, min. Consider the most familiar form where $\int$ is integral

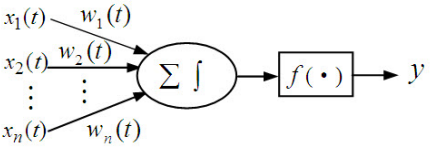


Fig. 1. A process neuron

and $\sum$ is weight sum. Then (1) can be rewritten as (2), where T is the time interval for t. The net structure of PNN can be the same with traditional neural networks. There are some property theorems of PNN [14], including continuity, function approximation ability and computational capacity.

$$y = f\left(\sum_{i=1}^{n} \int_{t\in T} w_i(t)x_i(t)dt - \theta\right) \tag{2}$$

The process functions $x(t)$ and $w(t)$ in PNN do not necessarily rely on time, they can be functions with one or more other arguments, such as spatial location, temperature. Process neural network with multidimensional arguments is a more generalized model named multi-aggregation process neural network [9], whose neuron expressed by (3), and we will use multi-aggregation PNN in the experiments. All the theories and learning algorithms of PNN can be extended to multi-aggregation PNN. For convenience, models and algorithms are demonstrated based on single argument PNN.

$$y = f\left(\sum_{i=1}^{n} \int \cdots \int_{t_k\in T_k} w_{ij}(t_1,\cdots,t_p)x_i(t_1,\cdots,t_p)dt_1\cdots dt_p - \theta\right) \tag{3}$$

2.2 Base Function Learning Method for Process Neural Network

As $w_{ij}(t)$ can be any kind of functions, it is almost impossible to obtain their analytic forms without assumptions. According to BFL, $w_{ij}(t)$ is assumed to be composed of a series of base functions, that is, $w_{ij}(t) = \sum_{l=1}^{L} w_{ij}^{(l)} b_l(t)$ where $b_l(t)$ is a base functions, and L is the number of base functions. Then (2) can be rewritten as

$$y = f\left(\sum_{i=1}^{n}\sum_{l=1}^{L} w_{ij}^{(l)} \int_{t\in T} b_l(t)x_i(t)dt - \theta\right) \tag{4}$$

Then $\int_{t\in T} b_l(t)x_i(t)dt$ can be computed, and the network can be trained by back propagation algorithm used in [11]. The most difficult part of BFL is selecting base functions and their parameters without the concrete forms of weights functions. Then we proposed the numerical learning method for PNN, which avoided the difficult selections.

3 Numerical Learning Method for Process Neural Network

Numerical learning method is illustrated in this section. Avoiding base function selections, NL is more simple and accurate than BFL.

3.1 Structure of NL Based PNN

Unlike BFL representing functions in analytical forms, NL does it in a discrete numerical way so that NL does not expand the weight functions. Considering a NL based feedforward process neural network with a hidden layer showed in Fig. 2, the relation between inputs and outputs is

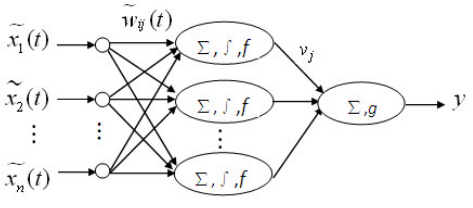


Fig. 2. A NL based feedforward process neural network

$$y = g\left(\sum_{j=1}^{m} v_j f\left(\sum_{i=1}^{n} \int_{t\in T} \tilde{w}_{ij}(t)\tilde{x}_i(t)dt - \theta_j\right) - \theta\right) \tag{5}$$

where $\tilde{x}(t)$ and $\tilde{w}(t)$ are series of discrete values and $\int_{t\in T} \tilde{w}_{ij}(t)\tilde{x}_i(t)dt$ can be computed by numerical integration. In most real-world scenarios, the inputs obtained are usually discrete values rather than functions, so NL is more practical.

NL represents functions in numerical forms. It does not change PNN's net structure nor conflict with PNN theories, so NL based PNN maintain all the properties of PNN.

3.2 Numerical Learning Method (NL)

BP learning algorithm is widely used in feedforward neural networks. It also can be used in PNN in different ways from traditional neural networks.

Assume the training data set being like:

$$\begin{bmatrix} x_{11}(t), & x_{12}(t), & ..., & x_{1n}(t), & d_1 \\ x_{21}(t), & x_{22}(t), & ..., & x_{2n}(t), & d_2 \\ \vdots & & \vdots & & \\ x_{K1}(t), & x_{K2}(t), & ..., & x_{Kn}(t), & d_K \end{bmatrix}$$

where $x_{ij}(t)$ is a function or a sequence; d_i is the i-th desired output. The numerical learning method is based on the basic idea of BP algorithm. The error function is

$$E=\sum_{k=1}^{K}(y_k-d_k)^2=\sum_{k=1}^{K}\left(g\left(\sum_{j=1}^{m}v_j f\left(\sum_{i=1}^{n}\int_{t\in T}\tilde{w}_{ij}(t)\tilde{x}_i(t)dt-\theta_j\right)-\theta\right)-d_k\right)^2 \tag{6}$$

The modifying rules are

$$v_j = v_j + \alpha\Delta v_j, j = 1, 2, \cdots, m \tag{7}$$

$$w_{ij} = w_{ij} + \beta\Delta w_{ij}, i = 1, 2, \cdots, n; j = 1, 2, \cdots, m; \tag{8}$$

$$\theta_j = \theta_j + \gamma\theta_j, j = 1, 2, \cdots, m \tag{9}$$

For convenience, denote $u_{kj}=\sum_{i=1}^{n}\int_{t\in T}\tilde{w}_{ij}(t)\tilde{x}_i(t)dt-\theta_j$, $z_k=\sum_{j=1}^{m}v_j f(u_{kj})-\theta$ so that

$$\Delta v_j = -\frac{\partial E}{\partial v_j} = -2\sum_{k=1}^{K}(g(z_k) - d_k)g'(z_k)f(u_{kj}) \tag{10}$$

$$\Delta\tilde{w}_{ij}(t) = -\frac{\partial E}{\partial\tilde{w}_{ij}(t)} = -2\sum_{k=1}^{K}\left((g(z_k) - d_k)g'(z_k)v_j f'(u_{kj})\int\tilde{x}_{ki}(t)dt\right) \tag{11}$$

$$\Delta\theta_j = -\frac{\partial E}{\partial\theta_j} = -2\sum_{k=1}^{K}\sum_{j=1}^{m}v_j f(u_{kj} - d_k)v_j f'(u_{kj}) \tag{12}$$

$\int\tilde{x}_{ki}(t)dt$ in (11) is computed by numerical integration. If f, g are sigmoid functions, $f(u) = (1 + e^{-u})^{-1}$ then $f'(u) = f(u)(1 - f(u))$.

The main steps of the learning algorithm are:

step 1 Set precision ε, maximum number of iterations M, accumulated learning iterations s initialized as zero.

step 2 Initialize weights w_{ij}, v_j and thresholds θ_j.

step 3 Compute error E, if $E < \varepsilon$ or $s > M$, go to step 5, else go to step 4.

step 4 Update weights w_{ij}, v_j and thresholds θ_j; $s = s + 1$.

step 5 Test the model or apply it to real scenarios.

4 Experiments

We compared our method with the existing learning methods in traffic flow prediction scenario on a real data set; the accuracy and time cost were compared.

4.1 Application Scenario and Data

Traffic flow forecast was important to many applications in intelligent transportation system. Traffic flow showed a continuous process in both time and

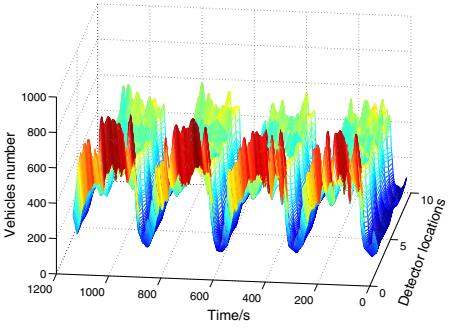


Fig. 3. Time process of the flow

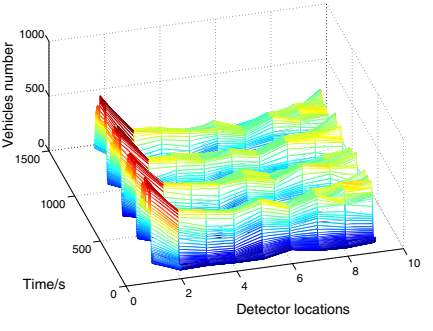


Fig. 4. Spatial process of the flow

space, as in Fig. 3 and Fig. 4. We tried to forecast future traffic flow on basis of past flow. The forecast model can be expressed as

$$flow(t_{pre}, s_k) = F_{forecast}(flow([t_a, t_b], [s_i, s_j])) \tag{13}$$

where $flow(t_{pre}, s_k)$ was the traffic flow at time t_{pre} and on the detector s_k, while $flow([t_a, t_b], [s_i, s_j])$ represented the flow matrix during time interval $[t_a, t_b]$ and on the detector from s_i to s_j.

We used real traffic volume data in the State of California, USA; the data was from the project Freeway Performance Measurement System (PeMS) [17]. We chose a two-way six lanes freeway labeled "I605-S" crossing the seventh district with five consecutive loop detectors numbered 716810, 717898, 717896, 717894 and 717887. All the detectors recorded the volume travel though them in the south-east three lines every 30 seconds and PeMS provided 5 minutes volume by aggregation.

We used flow of all five detectors to forecast future flow on the third detector. That was, in (13), $t_{pre} = t_b + 1$, $s_k = s_3$, $[s_i, s_j] = [s_1, s_5]$. From experience obtained from the experiments, flow during five consecutive time intervals, 25 minutes, was enough to represent the traffic condition at that time, so $t_b - t_a = 5$. We focused on noontime traffic flow, from 10:00 to 15:00 when the traffic was heavy and fluctuant. Take March 28, 2007, Wednesday as an example. We used noontime flow data of March 7, 14, 21, three Wednesdays, in the past three weeks to train the network, and forecasted the noontime traffic flow of March 28.

4.2 Experiments Settings

On basis of the scenario and data above, four methods, including traditional artificial neural network (ANN), PNN based on polynomial BFL, PNN based on wavelet BFL and NL, were carried out in the contrast experiments. The accuracy, time cost and convergent speed of the methods were compared through five groups of experiments done for five days, March 26 to 30, Monday to Friday.

For ANN, because it lacked the ability of handling processes, $t_b - t_a = 1$, and five detectors' data was input into five independent neurons. Suitable numbers of neural networks' hidden neurons were set based on repeated trials, so did all the other methods.

All the PNN in the experiments has one input node for the process input matrix and one output node for the forecast result.

For PNN based on polynomial BFL, bivariate cubic polynomial base functions were used as they had been used in previous work [8].

For PNN based on wavelet BFL, two-dimension Haar Wavelet base functions $\{\phi_{jn}(x)\psi_{jm}(y), \psi_{jn}(x)\phi_{jm}(y), \psi_{jn}(x)\psi_{jm}(y)\}_{j,m,n\in Z}$ were used, where parameter j determined the resolution and m, n determined the functions translations.

NL did not have to do base function expension and added no extra parameters to the model.

Mean Absolute Percentage Error (MAPE) was used to estimate the accuracy of the forecast, defined as

$$MAPE(y, y') = \frac{1}{N}\sum_{i=1}^{N}\frac{|y_i - y_i'|}{y_i} \tag{14}$$

where y was the real flow value, and y' was the forecasted one.

4.3 Experimental Results

Comparisons of accuracy, time cost and convergent speeds of four methods were showed in Fig. 5, Fig. 6 and Fig. 7. Fig. 8 showed the influence of parameters settings on PNN based on wavelet BFL.

Polynomial base functions had a simple form. However, PNN based on polynomial BFL was not accurate in traffic flow forecast (Fig. 5) mainly because the base functions were too simple in such a complex environment like transportation.

PNN based on wavelet BFL was more accurate compared to PNN based on polynomial BFL since wavelet base functions had more powerful approximation

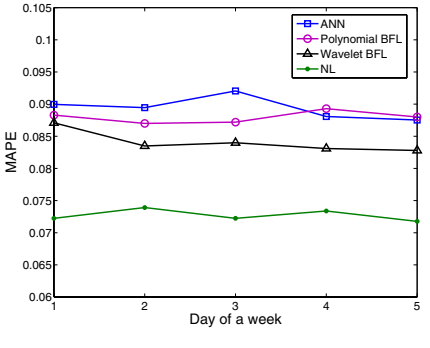


Fig. 5. Accuracy comparision

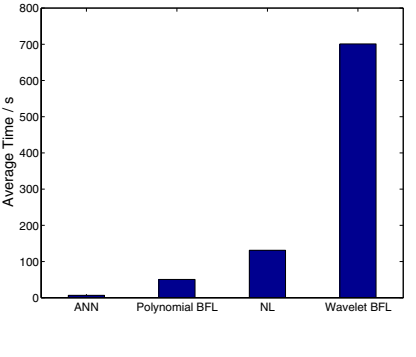


Fig. 6. Computing time comparision

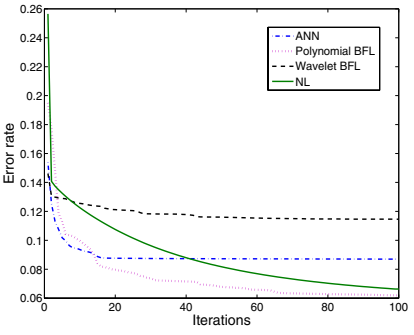


Fig. 7. Convergent speeds

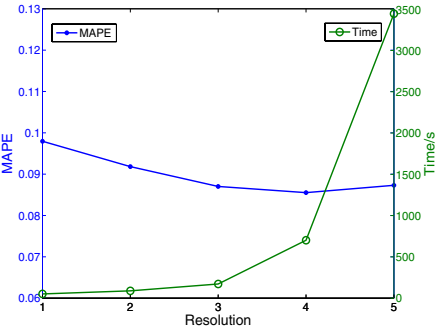


Fig. 8. Forecast of PNN based on wavelet BFL with different parameters

ability with a proper resolution. Fig. 8 shows the accuracy and time cost of wavelet BFL with different resolutions; in the experiment showed in Fig. 5, the resolution of wavelet base functions was set as 4, with which the model performance was the best. However, the number of base functions would increase exponentially with the increase of resolution, and the computational complexity would increase at the same speed, leading to high computational complexity (Fig. 6). On the other hand, it introduced more parameters, making the network more complicated, which limited the accuracy.

NL based PNN were more accurate than the other three models (Fig. 5), as it avoided the error introduced by selections of base functions and their parameters. NL's computational complexity was acceptable (Fig. 6), higher than the most simple BFL (with poor performance, however), much lower than wavelet BFL. All the neural networks had fast convergent speeds (Fig. 7) as their learning algorithms were all belonged to back propagation.

5 Conclusion

A numerical learning method (NL) for process neural network (PNN) was proposed. Different from usually used base function learning methods (BFL), NL represented PNN's input and weight functions in a numerical way, so that it avoided selections of base functions and their parameters. Experiments showed that NL based PNN had a higher accuracy and lower computational complexity.

Process neural network enhanced information processing capability of traditional neural network, and could be applied in many scenarios with processes. NL provided a practicable learning method for PNN, which would extend PNN's potential usage. In the future work, we will focus on how to define the most effective process inputs and time accumulation operators, which would be helpful to both PNN theories and applications.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. 60703066 and No. 60874082; supported by the National High-Tech Research and Development Plan of China (863) under Grant No.2006AA12-Z217.

References

1. Rosenblatt, F.: The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. Psychol. Rev. 65, 386–408 (1958)
2. Kohonen, T.: Data Management by Self-Organizing Maps. In: Zurada, J.M., Yen, G.G., Wang, J. (eds.) Computational Intelligence: Research Frontiers. LNCS, vol. 5050, pp. 309–332. Springer, Heidelberg (2008)
3. Park, J.: Approximation and Radial-Basis-Function Networks. Neural Comput. 5, 305–316 (1993)
4. Buckley, J.J., Hayashi, Y.: Fuzzy Neural Networks: a Survey. Fuzzy Set Syst. 66, 1–13 (1994)
5. Xu, X., Zhang, Y.X., Liang, Y.C., Yang, X.W., Hao, Z.F.: An Improved Fuzzy Neural Network for Ultrasonic Motors Control. In: Liu, D., Fei, S., Hou, Z.-G., Zhang, H., Sun, C. (eds.) ISNN 2007. LNCS, vol. 4491, pp. 8–13. Springer, Heidelberg (2007)
6. Wang, N.N., Liu, X.D., Yin, J.C.: A Hybrid Model of Partial Least Squares and RBF Neural Networks for System Identification. In: 5th International Symposium on Neural Networks, pp. 204–211. Springer, Berlin (2008)
7. Meng, Y.M., Yan, S.J., Tang, Z.H., Chen, Y.L., Liu, J.N.: Data Fusion Based on Neural Networks and Particle Swarm Algorithm and Its Application in Sugar Boiling. In: 5th International Symposium on Neural Networks, pp. 176–185. Springer, Berlin (2008)
8. He, X.G., Xu, S.H.: Process Neural Networks. Science Press, Beijing (2007) (in Chinese)
9. Xu, S.H., He, X.G.: The Multi-Aggregation Process Neural Networks and Learning Algorithm. J. Comput. Sci. Technol. 30, 48–56 (2007) (in Chinese)
10. Liang, J.Z., Han, J.M.: Complex Number Procedure Neural Networks. In: Wang, L., Chen, K., S. Ong, Y. (eds.) ICNC 2005. LNCS, vol. 3610, pp. 336–339. Springer, Heidelberg (2005)
11. He, X.G., Xu, S.H.: Process Neural Network with Time-Varied Input and Output Functions and Its Applications. J. Softw. 14, 764–769 (2003)
12. Liang, J.Z., Wu, X.H.: Worm Harm Prediction Based on Segment Procedure Neural Networks. In: Wang, G.-Y., Peters, J.F., Skowron, A., Yao, Y. (eds.) RSKT 2006. LNCS, vol. 4062, pp. 383–388. Springer, Heidelberg (2006)
13. Song, G.J., Yang, D.Q., Wang, T.J., Tang, S.W.: A Mixed Process Neural Network and Its Application to Churn Prediction in Mobile Communications. In: IEEE International Conference on Data Mining-Workshops, pp. 798–802 (2006)
14. He, X.G., Liang, J.Z.: Some Theoretical Issues on Procedure Neural Network. Eng. Sci. 2, 40–43 (2000) (in Chinese)
15. Liang, J.Z., Zhou, J.Q.: Procedure Neural Networks with Supervised Learning. In: Proceedings of the 9th International Conference on Neural Information Processing, pp. 523–527. IEEE Press, New York (2002)
16. He, X.G., Liang, J.Z., Xu, S.H.: Learning and Applications of Procedure Neural Networks. Eng. Sci. 3, 31–35 (2001) (in Chinese)
17. UC Berkeley, PeMS, http://pems.eecs.berkeley.edu

Specialized Affine Approximation for Nonlinear Systems Output Tracking Using Neural Networks

Tsurng-Jehng Shen, Chorng-Shyr Jou, Meng-Jey Youh, and Chia-Tang Chen

Hsing Wu College, No. 101, Sec.1, Fenliao Rd., LinKou,
Taipei County 244, Taiwan, China
shentj@ms2.hinet.net

Abstract. A special model of artificial neural networks has been developed for the purpose of output tracking of a class of nonlinear systems and an original training structure basing on the error back-propagation algorithm is introduced. This approach which, on the ground of input-output observations, turns feasible the use of inverse control techniques appears much simpler than other existing neural control approaches for inverse control problem. Simulation results reveal that the proposed approach presents rapid training and good tracking performances.

Keywords: Multilayered feedforward neural network, Affine approximation, Inverse control, Trajectory tracking.

1 Problem Statement

Consider a class of n[th] order single input single output (SISO) nonlinear system defined in a bounded input bounded output (BIBO) compact domain $U \subset \Re^{2(\nu-1)}$, and assume that the external dynamic[1] of this plant can be precisely described by following affine difference equation :

$$y(k+1) = f\left[Y(k)\right] + g\left[Y(k)\right]u(k) \tag{1.1}$$

where $f, g : \Re^{(\nu-1)} \to \Re$ and are smooth functions with arguments $Y(k) = \left[y(k), y(k-1), ..., y(k-\nu+1)\right]^T$ being past and present measurements of the plant's output and input at stage k, while $\nu \le n$ is observation index of the plant.

Here the considered control problem is to make the output of system (1.1) asymptotically track a desired trajectory $y_d(k+1)$ that been assumed uniformly bounded in the domain U. If the functions f and g are both known, it follows directly that at stage k, $u(k)$ can be computed from a knowledge of $Y(k)$ as follows:

$$u(k) = \frac{-f\left(Y(k)\right) + w(k)}{g\left(Y(k)\right)} \tag{1.2}$$

with $w(k) = y_d(k+1) - \sum_{i=1}^{\nu-1} a_i\left[y(k-i) - y_d(k-i)\right]$, then with this control input can assure a stable error dynamic for the desired output tracking:

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 679–688, 2009.

$$e(k+1)+\sum_{i=1}^{v-1} a_i e(k-i)=0 \tag{1.3}$$

where $e(k) \equiv y(k)-y_d(k)$.

Even the mentioned nonlinear control technique proposes a convenient output tracking solution to affine nonlinear systems, since, in practical case, f and g are normally not well known, the inverse control formulation can hardly applied directly to any practical affine systems.

For sometime, researches are interested in using neural networks to approximate the input/output mapping of affine nonlinear systems. In this kind of applications, neural networks are mostly used to independently approximate f and g of not well known affine systems[2,3]. When necessary accuracy of system's input/output mapping is obtained by suitable training, the mentioned control law formulation can be again applied to output tracking.

Definition 1. A *Specialized Affine Model* for the external dynamics of the nonlinear system (1.1) defined in a compact neighborhood (V_i) of an operating condition $Y(k_i)$ is defined to take the form:

$$\hat{y}(k+1)=\hat{F}(Y(k))+\hat{G}(Y(k))u(k) \quad \forall Y(k)\in V_i \subset U \tag{1.4}$$

such that for a required precision value $\varepsilon>0$, different from other neural input/output modelization approaches, the following two expressions should be simultaneously made to be satisfied:

$$\begin{cases} \left\| y(k+1)-\hat{y}(k+1) \right\| \le \varepsilon \\ \left\| u(k)-\hat{u}(k) \right\| \le \varepsilon \qquad \forall Y(k)\in V_i \end{cases} \tag{1.5}$$

where $\| \ \|$ denotes a vector norm and $\hat{u}(k) \equiv \dfrac{\left[y(k+1)-\hat{F}(Y(k)) \right]}{\hat{G}(Y(k))}$ with $\hat{F}$ and $\hat{G}$ two continuous functions of $Y(k)$ defined in V_i and

$$\begin{aligned} & k_1 > \hat{G}(Y(k)) > k_2 \qquad \forall Y(k)\in U \\ & k_1 k_2 > 0 \end{aligned} \tag{1.6}$$

Condition (1.5) will assure that (1.4) can not only approximate the system's input/output relation, but also its inverse (output/input) relation in a stable affine form,.

Basing on the specialized affine model of the considered system and supposing that system's initial external states are well known, the tracking of a desired output trajectory, $y_d(k+1)$, can be obtained with the following inverse control law:

$$c(k)=\frac{-\hat{F}_i(Y(k))+y_d(k+1)}{\hat{G}_i(Y(k))} \qquad \forall Y(k)\in V_i \subset U \tag{1.7}$$

Substitute $u(k)$ by $c(k)$ in (1.4), we have $\hat{y}(k+1)=y_d(k+1)$, which implies:

$$\left\| y(k+1)-y_d(k+1) \right\| < \varepsilon \tag{1.8}$$

In the case asymptotic tracking is to be performed, following control law can be applied:

$$c(k)=\frac{-\hat{F}_i\left(Y(k)\right)+y_d(k+1)-\left[\sum_{i=1}^{v-1}a_i\left[\hat{y}(k-i)-y_d(k-i)\right]\right]}{\hat{G}_i\left(Y(k)\right)},\forall Y(k)\in V_i\subset U \quad (1.9)$$

In both cases desired output tracking can be either directly or asymptotically obtained while its final precision is related to the resulted precision of the specialized affine modelization. Notice that in (1.9) $\hat{y}(k-i)$ can be replaced by output measurement of the plant:, then the tracking precision being a consequence of the specialized affine modelization error can be improved.

2 Quasi-Affine Modelization Using Neural Networks

Definition 2. A Neural Network who takes $Y(k)$ as input vector, $\left(\hat{F},\hat{G}\right)^T$ as output vector (so $\hat{F}$, $\hat{G}$ are continuous functions of $Y(k)$ and W which denotes connection weights and the bias of each neuron of the neural network) and is trained to verify (1.5) for $\forall Y(k)\in U,\varepsilon\in\Re^+$ is called a *Specialized Affine Neural Model of degree* ε of the considered systems and is noted as N_{sys}

We propose N_{sys} to be a L-layer ($L\geq 2$) feedforward neural network of type $N^L_{2(v-1),i_1,i_2,\cdots i_{L-1},2}$ (Narendra's notation) with hyperbolic tangent activation functions in hidden layers, and the following sigmoid function in output layer :

$$\begin{aligned}\hat{F}(X)&=a+\frac{(b-a)}{1+e^{-X}}\\ \hat{G}(X)&=c+\frac{(d-c)}{1+e^{-X}}\end{aligned} \quad (2.1)$$

When sufficient large range is chosen for $\hat{F}$ and $\hat{G}$, *Specialized Affine Neural Network* can be taken as an universal approximator[4,5] of any continuous mapping.

2.1 Distal Teachers Group for Quasi-Affine Neural Networks

For the considered class of systems, there is no way to directly train a quasi-affine neural network by minimizing the error between $(\hat{F},\hat{G})^T$ and its target functions (F,G). The neural networks can only be trained on the basis of available examples from the past and present input/output measurements of the plant. For this reason, a distal teacher[6] N^i_{DT} is needed to be designed to train N_{sys} in a local operation domain V_i .

The distal teacher N^i_{DT}, who takes $(\hat{F},\hat{G})^T$ as input and $(\bar{y}(k_i+1),\hat{u}(k_i))^T$ as output vector, is designed here to be an additional neural layer to N_{sys} with two linear

neurons. Actually N_{DT}^{i} represents a 2×2 matrix whose elements (connection weights) having no need to be tuned by training are systematically calculated from available values as :

$$N_{DT}^{i}=\begin{bmatrix} 1 & u(k_i) \\ -1/\hat{G}(W,Y(k_i)) & y(k_i+1)/\hat{G}(W,Y(k_i))^2 \end{bmatrix} \tag{2.2}$$

here $Y(k_i)$ denotes the i^{th} input pattern of the trained neural network.

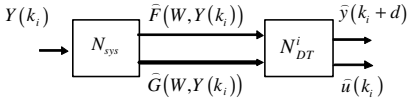


Fig. 1. $\left(N_{sys}\mapsto N_{DT}^{i}\right)$; N_{sys} trained by specific distal teacher N_{DT}^{i}

Now basing only on available input/output examples of the plant, a quasi-affine neural network can be trained by connecting N_{sys} and N_{DT}^{i} in series (Fig.1) and using the following performance index :

$$E_i=\frac{1}{2}\left[y(k_i+1)-\bar{y}(k_i+1))^2+(u(k_i)-\hat{u}(k_i))^2 \right] \tag{2.3}$$

Result 1. The distal teacher N_{DT}^{i} can train N_{sys} to be a local affine neural approximator of system (1.1), the neural structure is then noted as $\left(N_{sys}\mapsto N_{DT}^{i}\right)$

Suppose the neural network is trained over a enough small neighborhood V_i of some operation point $Y(k_i)$, such that only the input-output pair $Y(k_i)$ is required for the neural network training.

Since N_{DT}^{i} is simply a linear layer (a 2×2 matrix), the potential of N_{sys} to be an universal approximator is not lost. To proof that N_{sys} can be a local affine neural approximator, it is sufficient to show that $\left(N_{sys}\mapsto N_{DT}^{i}\right)$ has a suitable input-output structure which can be trained to satisfy the condition (1.5), and that the forward mapping of the connected neural network is not divergent during training ($\hat{G}\neq 0$).
The output vector of $\left(N_{sys}\mapsto N_{DT}^{i}\right)$ is calculated as follows :

$$\begin{bmatrix} \hat{y}(k_i+1) \\ \hat{u}(k_i) \end{bmatrix}=W_{DT}^{i}\begin{bmatrix} \hat{F}(W,Y(k_i)) \\ \hat{G}(W,Y(k_i)) \end{bmatrix}=\begin{bmatrix} \hat{F}(W,Y(k_i))+\hat{G}(W,Y(k_i))\hat{u}(k_i) \\ \hat{G}(W,Y(k_i))^{-1}\left[y(k_i+1)-\hat{F} \right] \end{bmatrix} \tag{2.4}$$

Since by choosing the range parameters c and d, $\hat{G}_i$ can be limited to a range which excludes zero, $(\bar{y}(k_i+1),\hat{u}(k_i))$ which approximates the input-output pairs for a BIBO system can be kept finite. Besides, the chosen error index (2.3) and relation (2.4) assure that when the neural network is satisfactorily trained, condition (1.5) is met, i.e. :

$$\left\| \begin{bmatrix} \hat{F}(W, Y(k_i)) + \hat{G}(W, Y(k_i))\hat{u}(k_i) \\ \hat{G}(W, Y(k_i))^{-1}\left[y(k_i+1) - \hat{F} \right] \end{bmatrix} - \begin{bmatrix} y(k_i+1) \\ u(k_i) \end{bmatrix} \right\| < \varepsilon \tag{2.5}$$

However, a neural approximator trained by using only one single input-output pattern $Y(k_i)$ will have a very poor generalization performance. So, to have a more general quasi-affine neural approximator (valid for a larger domain V_i) or a global neural approximator (valid for the whole BIBO domain U), a complete training data set should be used for batch training.

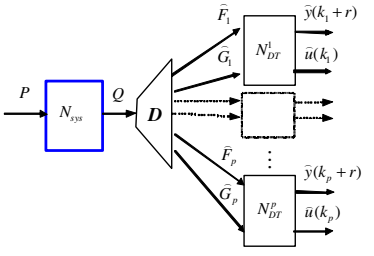


Fig. 2. The global training structure for N_{sys} in domain U

Consider now batch training is applied using a data set contains p well excited (irregularly placed) input-output pairs ($Y(k), i=1,2\ldots,p$) of the continuous system in U, such that the union of the p neighborhoods covers U ($V_1 \cup V_2 \cup \cdots \cup V_p = U$), then for each training data pair $Y(k_i)$ we use a specific ($\left(N_{sys} \mapsto N^i_{DT}\right)$) training structures. This is shown in (Figure 02), with D represents decomposition function of a p columns matrix into p separate vectors, where:

$$P = \left[Y_1, Y_2, \ldots, Y_p \right] \tag{2.6}$$

with $Y_i = \left\{ Y(k_i) \equiv (y(k_i), \ldots, y(k_i - v + 1), u(k_i), \ldots, u(k_i - v + 1))^T \mid \forall Y(k_i) \in V_i \subset U \right\}$, and

$$Q = \begin{bmatrix} \hat{F}_1 & \hat{F}_2 & \cdots & \hat{F}_p \\ \hat{G}_1 & \hat{G}_2 & \cdots & \hat{G}_p \end{bmatrix} \quad \text{with} \quad \begin{cases} \hat{F}_i \equiv \hat{F}(W, Y_i) \\ \hat{G}_i \equiv \hat{G}(W, Y_i) \end{cases} \quad i = 1,2,\ldots,p \tag{2.7}$$

Result 2. Let U be an admissible BIBO domain for the nonlinear system (1.1). Let V_i, $i=1,2...,p$, be compact neighborhoods of p well excited input-output pairs defined over U with $V_1 \cup V_2 \cup \ldots \cup V_p = U$. Let N_{sys} be a quasi-affine Neural Network which has $\left[Y_1, Y_2, \ldots, Y_p \right]$ as batch training input matrix, $[(\hat{F}_1, \hat{G}_1)^T, (\hat{F}_2, \hat{G}_2)^T, \ldots, (\hat{F}_p, \hat{G}_p)^T]$ as batch output matrix. Let $E_G = \sum_{i=1}^{p} E_i$ be a global error index. Then, since the system is suppose smooth and the training set is well excited, a good multilayer neural network will

generalize input/output relation which are not explicitly specified in the training data set, that means, it is possible to find and train a N_{sys} to be a global quasi-affine neural approximator of (1.1) through p distal teachers, N_{DT}^{i}, $i=1,2,...,p$. The corresponding neural structure is written $(N_{sys} \mapsto \begin{matrix} N_{DT}^{1} \\ N_{DT}^{2} \\ \vdots \\ N_{DT}^{p} \end{matrix})$.

2.2 Modified Error Backpropagation for Quasi-Affine Neural Networks

Since the traditional structure of the error back-propagation training algorithm can only use one distal teacher to train one input-output mapping, to train N_{sys} to globally satisfy the condition (1.5), the error backpropagation algorithm is modified as follow to adapt the structure which applies, in the same time, p distal teachers.

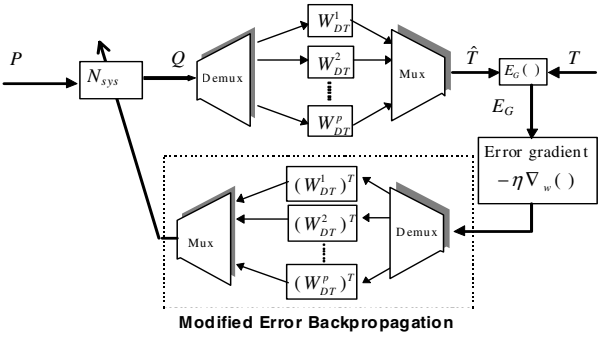


Fig. 3. Architecture of *Modified Error Back-propagation* for the training of Quasi-affine neural networks

Defining *Demux* a decomposition function of a p columns matrix into p separate vectors and *Mux* its inverse function, a *Modified Error Back-propagation* architecture is proposed in graphical representation (Fig.3). All the connections drawn in thick lines represent the standard neural network signals forward propagation and the error back-propagation paths, with:

$$T = \begin{bmatrix} y(k_1+1) & y(k_2+1) \cdots y(k_p+1) \\ u(k_1) & u(k_2) \quad \cdots \quad u(k_p) \end{bmatrix} \tag{2.8}$$

$$\hat{T} = \begin{bmatrix} \hat{t}_1 & \hat{t}_2 & \cdots & \hat{t}_p \end{bmatrix} \tag{2.9}$$

and

$$\hat{t}_i = W_{DT}^{i} \begin{bmatrix} \hat{F}_i \\ \hat{G}_i \end{bmatrix} \qquad i = 1,2\ldots, p \tag{2.10}$$

The global error index used for training is defined as $E_G = \sum_{i=1}^{p} E_i$, where E_i is defined by (2.3).

3 Asymptotic Output Tracking

Once the system (1.1) is represented by a quasi-affine model and all the initial values of the system states can be exactly set to required initial condition, the inverse controller (1.7) can be applied to perform output tracking while provides a tracking precision of $\|y(k+1) - y_d(k+1)\| < \varepsilon$ with ε the quasi-affine modelization error.

Since in practical case, the system states can hardly set to arbitrary condition, the following asymptotic tracking could be applied to convert both the modelization and initial state errors:

$$c(k) = \frac{-\hat{F}\left(W, Y(k)\right) + \left[y_d(k+r) + \sum_{m=1}^{r} a_{r-m}\left(y_d(k+r-m) - y(k+r-m)\right)\right]}{\hat{G}\left(W, Y(k)\right)} \tag{3.1}$$

with $a_{r-m}, m = 1,2,\ldots,r$ the Hurwitz coefficients [7].

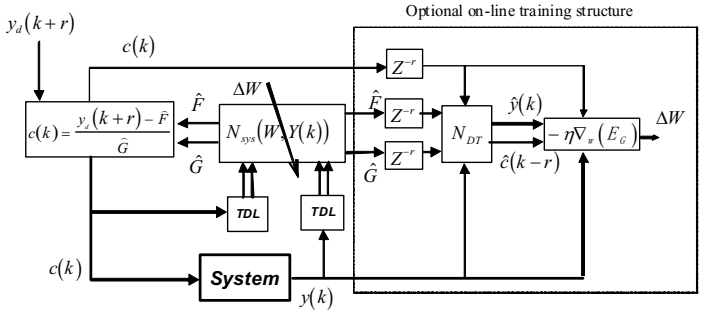


Fig. 4. Output Tracking control loop using affine neural approximator

4 Simulation Example

In this paragraph a simulation study using example already treated in [3] is presented. The considered plant is described by :

$$y(k+1) = \frac{y(k)}{1 + y(k)^2} + u^3(k) \tag{4.1}$$

The goal is to track the following desired trajectory :

$$y_d(k) = \sin(2\pi k / 25) + \sin(2\pi k / 10)$$

In [3] the internal structure of the plant (4.1) was considered exactly known ($F = y(k)/\left[1+y^2(k)\right], G = u^3(k)$) while input-output measurements were also available. To track the desired trajectory Narendra used two neural networks of type $N^3_{1,20,10,1}$ to separately approximate F and G . Then an additional neural controller of type $N^3_{1,20,10,1}$ was trained to control the system. Besides, to get a precise tracking, on-line adaptation techniques were also necessary.

In this example, we use only one off-line trained quasi-affine neural network of type $N^3_{2,32,10,2}$ which modelizes the plant under the form :

$$\hat{y}(k+1) = \hat{F}(W, u(k)) + \hat{G}(W, y(k), u(k))v(k)$$
$$\hat{v}(k) = (y(k+1) - \hat{F}(W, y(k), u(k)))/\hat{G}(W, y(k), u(k))$$

where $v(k) \equiv u^3(k)$.

For training phase, the plant is excited with random $u(k)$ and $y(k)$ which are uniformly distributed over $U = \{u(k), y(k) \mid \forall u(k) \in [-2,2], \forall y(k) \in [-10,10]\}$, and *100* input-output patterns (*p=100*) was generated. After a training of *5000* iterations, a very accurate tracking has been obtained by using the following tracking controller:

$$c(k) = \sqrt{\hat{G}^{-1}(y_d(k+1) - \hat{F}} \tag{4.2}$$

it can be revealed from the simulation results that the effect of a difference between $\hat{G}$ and G is dynamically compensated by $\hat{F}$. This implies that *a precise output tracking can be realized by using the proposed approach, even if the parameter convergence ($(\hat{F}, \hat{G})^T \to (F, G)^T$) cannot be obtained.*

Actually, it is clear that (4.1) is not a standard affine system, but (4.1) can anyway be globally represented into a quasi-affine form as:

$$y(k+1) = F(y(k)) + G(u(k)) \cdot u(k) \text{ with } \begin{cases} F(k) = y(k)/1 + y^2(k)) \\ G(k) = u^2(k) \end{cases} \tag{4.3}$$

Note that, when *u(k)* is zero *G* is also zero. This means that when the control input is zero the representation (4.3) is not invertible.

To modelize the system in a quasi-affine form, a N_{sys} of type $N^3_{2,40,20,2}$ and 200 (*p=200*) uniformly distributed input-output pairs in $U = \{u(k), y(k) \mid \forall u(k) \in [-15,15], \forall y(k) \in [-5,5]\}$ were used. Then the proposed quasi-affine neural approximator would modelize the plant in following form:

$$\hat{y}(k+1) = \hat{F}(W, y(k), u(k)) + \hat{G}(W, y(k), u(k))u(k)$$
$$\hat{u}(k) = \left[y(k+1) - \hat{F}(W, y(k), u(k)) \right]/\hat{G}(W, y(k), u(k))$$

After a training 10,000 iterations a satisfying result has been carried out. Simulation results are shown in (Figure 0), where the system is made to track a programmed output trajectory using the inverse control law:

$$c(k) = \hat{G}^{-1}(W,\ y(k), u(k))\Big[\ y_d(k+1) - \hat{F}(W,\ y(k), u(k))\ \Big] \tag{4.4}$$

Here, the neural controller provides a satisfactory output tracking. But since $u(k)=0.0$ is a singular point of the exact quasi-affine model of the system, the plant output presents a small tracking error when a near zero input is needed. However this simulation has also revealed that ***parameter convergence is not a necessary condition for the proposed tracking controller to be effective***.

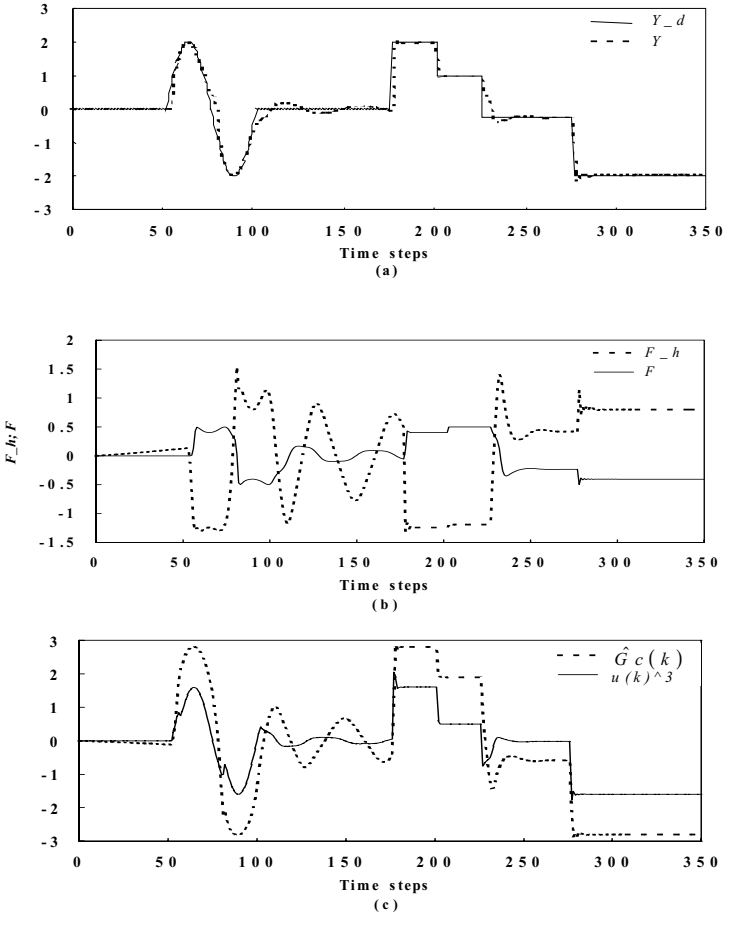


Fig. 5. Trajectory simulation result under the control of a globally trained affine neural networks

(a)**.** Desired trajectory (*Y_d*) and system output (*Y*)

(b). Plot of *F* and $\hat{F}$ (*F_h*).

(c). Plot of $u^3(k)$ and $\hat{G}(k)c(k)$.

5 Conclusion

In this paper a quasi-affine modelization using a special neural network training structure for generating a global affine neural tracking controller are suggested. The neural approximator learns with the help of parallel distal teachers to capture the mapping relating input and output variables of the controlled system. Since the outputs of the proposed neural approximator can be directly used to construct inverse tracking control laws, there is no more need to train another neural network devoted to control. With the new approach very good output tracking performance has been found in simulation studies.

References

1. Isidori, A.: Nonlinear Control System, An Introduction. Springer, New York (1989)
2. Narendra, K.S., Parthasarathy, K.: Identification and Control of Dynamical Systems Using Neural Networks. IEEE Transactions on Neural Networks 1, 4–27 (1990)
3. Narendra, K.S., Mukhopadhyay, S.: Adaptive Control of Nonlinear Multivariable Systems Using Neural Networks. Neural Networks 7, 737–752 (1994)
4. Barron, A.R.: Universal Approximation Bounds for Superpositions of a Sigmodal function. IEEE Transaction on Information Theory 39, 930–945 (1993)
5. Hornik, K.: Multilayer Feedforward Networks Are Universal Approximators. Neural Networks 2, 359–366 (1989)
6. Jordan, M.I., Rumelhart, D.E.: Forward Models: Supervised learning with a distal teacher. Cognitive Science 16, 307–354 (1992)
7. Behtash, S.: Robust Output Tracking for Nonlinear Systems. Int. J. Control. 51, 1381–1407 (1990)

Personalized SCORM Learning Experience Based on Rating Scale Model

Ayad R. Abbas and Liu Juan

School of Computer, Wuhan University, Wuhan 430079, china
ayad_cs@yahoo.com, liujuan@whu.edu.cn

Abstract. Sharable Content Object Reference Model (SCORM) is the most popular suite of technical standard among existing international standards for e-Learning; although it has been designed to provide accessibility, adaptability, interoperability, and reusability, it still suffers from lack of personalization, which may lead to inappropriate learning experience; In other words, learner may suffer from distraction or restriction, when it comes to interact with large or restricted amounts of information respectively, resulting in reduced learning efficiency and performance. However, this can be avoided by providing personalized services. In this paper, we propose a personalized SCORM learning experience based on Rating Scale Model (RSM), which takes into account both the difficulty of learning activity and the learner's ability considering responses from individual learner's understanding and characteristics. To obtain more accurate estimation of learner's ability, polytomous Item Response Model (IRT) is used rather than dichotomous IRT. Experimental results show that the proposed system can exactly provide the closer learning resource to the learner's ability, resulting in increased the learning efficiency and learning performance.

Keywords: Personalize e-learning, SCORM, Item response theory, Rating scale model, Polytomous information function.

1 Introduction

SCORM Sequencing refers to the behaviors that Learning Management System (LMS) follows to deliver a specific learning experience to the learners as intended by content developer [1]. This learning experience may be free choice, in which indicates that the learner is free to choose any activity in any order without restriction, or it may be guided by flow through the structure of the content organization. Besides these capabilities, it suffers from the lack of personalization. For these reasons, rather than redefining SCORM, this work presents an alternative way to reinforce the personalized SCORM without any changes to the existing SCORM. Item Response Theory (IRT) [2], [3], [4] have facilitated the development of Computerized Adaptive Tests (CAT) to construct an optimal test for each examinee and estimate examinee's ability during test administration. Items are selected to match the examinee's estimated ability according to IRT, which is assumed to describe an examinee's response behavior. In some studies, the assessment of person fit has investigated for CAT using dichotomously scored items [5], [6], [7]. Other studies have appeared involving real life CAT

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 689–698, 2009.

applications in attitude or personality testing using polytomous scored items [8], [9], [10]. This study proposes a personalized SCORM learning experience tacking into account learner's ability. allowing more information about ability level to be extracted from a fixed set of learning resources, the estimation of this ability does not depend only on explicit feedback response (i.e., understanding response), but also depend on implicit feedback responses (i.e., learning time, learning attempt, learning score), because these responses have a direct impact on the learner's ability. Furthermore, to obtain more accurate estimation of learner's ability, these responses are classified into three polytomous response categories at each item (learning resource). We have assumed that each learner has a three understanding levels, ''low understanding'', ''moderate understanding'' and ''high understanding'', and we have assigned them into 0, 1 and 2 respectively. The Rating Scale Model (RSM) [11], [12] is applied as a power tool to extract learner's ability because of using polytomous response items. Experimental results show that the proposed system can recommend suitable learning experience, resulting in increased the learning efficiency and learning performance.

2 Related Works

In the last decade, several efforts provided personalized e-learning within SCORM learning environment in accordance with an individual learner's preferences, aptitudes, background, and browsing behaviors. In this section, we have reviewed some of these attempts to emphasize the differences between them and our recent work. An adaptive personalized recommendation model [13] has been presented in order to help recommend SCORM-compliant learning objects from repositories in the Internet. To provide an appropriate associated sequencing definition for different learners according to individual learning characteristics and capability, [14] proposed an algorithm to create personalized activity tree which could be used in SCORM compliant learning environment. Unfortunately, previous studies neglected learner ability as an important factor in implementing personalization mechanisms. Recently many studies have given considerable attention to personalize e-learning system. Including [15], [16] have presented a prototype of personalized Web-based instruction system based on dichotomous IRT to perform personalized curriculum sequencing through simultaneously considering courseware difficulty level, learner's ability and the concept continuity of learning pathways during learning. [17] Extended the dichotomous IRT using fuzzy set in order to recommend courseware with appropriate difficult level to the learner. These later surveys estimated learner's ability using only learner's understating responses regardless the learner's characteristics information. Whereas, our work can estimate learner ability using explicit feedback response and implicit feedback responses.

3 System Architecture

Fig. 1 illustrates the proposed system architecture, which provides various intelligence services, such as learning interface service, SCORM course management service, and personalized SCORM service. Learner interface service controls required for the presentation of a learning activity and the transition from that learning activity to another learning activity. SCORM course management service allows teacher to

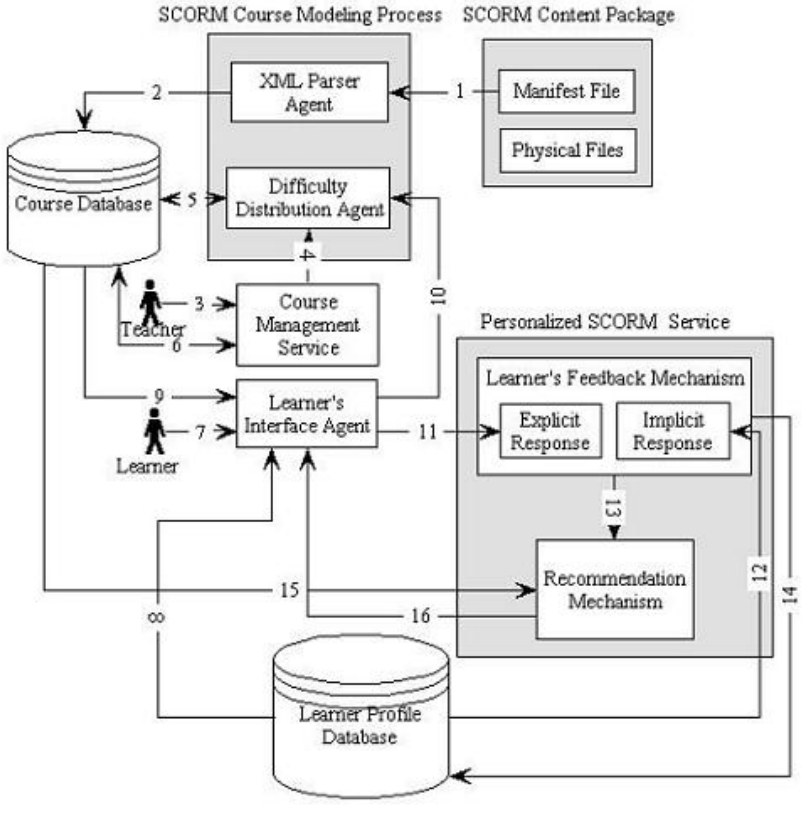


Fig. 1. The proposed system architecture

access and manage courses, modules and other units of learning. Finally, Personalized SCORM service composes of learner's feedback mechanism and learning sequence recommendation mechanism, the learner's feedback mechanism intends to assemble both explicit feedback response and implicit feedback responses, and update learner ability. However, learning sequence recommendation mechanism decides on the next activity to be delivered to the learner.

The system operation procedure of the proposed system architecture above is summarized as follow:

Step 1-2: The structure of the SCORM course is parsed and saved into course database.

Step 3-6: Teacher accesses to the SCORM course management service to read, update and delete units of learning. Moreover, teacher can assign and update learning resource difficulty.

Step 7-8: Learner accesses to the learner's interface agent then the system will get his or her learning profile from learner profile database.

Step 9-10: Learner's interface agent gets the SCORM course form course database and presents a list of recommended learning paths to the learner, and then learner can update learning resource difficulty levels.

Step 11-12: Learner's feedback mechanism collects explicit feedback responses from the learner's interface agent, and it collects implicit feedback responses from learner's profile database, and re-evaluates learner's ability according to these feedback responses by using RSM.

Step 13-14: If learner's ability has estimated, then Learner's feedback mechanism will send learner's ability to the learning experience recommendation mechanism, and all learner feedback responses and his or her ability will save into learner's profile database.

Step 15-16: Learning experience recommendation mechanism calculates information value for each node in the activity tree according to estimated learner's ability and node difficulty parameters, and sends a list of ranking learning paths to the learner's interface agent.

4 Difficulty Distribution Parameters

In [15] a voting approach was proposed to assess difficulty levels parameters by learners and teachers. The course materials' difficulty level was classified into five points ''very hard'', ''hard'', ''moderate'', ''easy'' and ''very easy''. Once those parameters are determined by learners/ teachers, there is a need to distribute and propagate those parameters over activity tree in order to assign difficulty parameter for each node in the activity tree. We propose an easy way to distribute the difficulty parameters, where the difficulty information of an activity in the activity tree is determined by taking the average value of its children's difficulty information. Thus, the difficulty information about the parent activity is updated based on its children's difficulty information.

5 Personalized SCORM Service

5.1 Learner's Feedback Mechanism

Learner's feedback mechanism collects explicit feedback response from the learner's interface agent, whereas it collects implicit feedback responses from learner's profile database. Each of these responses is explained below.

Explicit Feedback Response: A direct learner's response (multiple choice), which contains only one question associated with understanding learning recourse.
Implicit Feedback Responses: An indirect learner's response that associated with three learner's characteristic responses. Below is a description for each implicit response.

1. Learning time response: is the sum of all of the learner's session times.
2. Learning score response: is the learner's score for the SCO.
3. Learning attempt: is the number of attempts on the activity.

During learner activity (take lesson), the learner experiences two types of learning recourses: *Learning Material* or *Assessment* (e.g., Pre-Test, Post Test), if learner experiences *Learning Material*, learner will give three responses , understanding response, learning time response and learning attempt response. If learner experiences

Assessment, also learner will give three responses, learning time response, learning score response and learning attempt response.

The main functions of learner's feedback mechanism are explained in the following subsections.

5.1.1 Implicit Feedback Parameters

The learner's ability estimation does not depend only on understanding level, but also depend on implicit feedback responses, because these responses have a direct impact on the learner's ability. For example, if learner has completed the learning recourse at short time with small number of attempts and obtained high score then his or her ability will be increased or, if learner has completed the learning recourse at long time with high attempts and obtained low score then his or her ability will be decreased.

However, the learning time rate (*LTR*), learning attempt rate (*LAR*), and learning score rate (*LSR*) are proposed by using the following relationships:

$$LTR = \frac{\text{learning time}}{\text{maximum time allowed}} \tag{1}$$

$$LAR = \frac{\text{activity attemtpt count}}{\text{maximum attempt allowed}} \tag{2}$$

$$LSR = \frac{\text{learner score}}{\text{maximum score allowed}} \tag{3}$$

5.1.2 Learner's Ability Estimation

Learner's ability algorithm estimates the level of a latent trait of a learner demonstrated in an observed polytomous response pattern to a learning resource. The term of polytomous means that the learner can give more than two response categories in a specific learning resource. Therefore, polytomous IRT is appropriate for analyzing these responses.

We suggest that each item (learning resource) has three polytomous responses, and each response has three response categories, then the total number of responses for all items equal to N*3, where N denote the number of items in the activity tree. Randomly chosen learner responds to a set of N learning resources with array of polytomous response patterns $P = (I_1, I_2, \ldots, I_n)$, where I_i is an item, each item contains response categories.

For example, assume learner has completed five learning resources (I_1, I_2, I_3, I_4 and I_5), each learning resource has a numerical response categories labeled (0, 1, 2). Consequently, learner will give an array of polytomous response patterns: P = ((2, 2,1),(0,0,1),(2,1,2),(1,2,1),(1,1,1)).

For the polytomously scored items, the probability of a learner reaching a specific score category can be described by estimating measures with RSM [12], this model can be described as follow:

$$P_{nij} = \frac{e^{j(M-D_i)-\sum_{k=b}^{J} F_k}}{\sum_{h=b}^{t} e^{h(M-D_i)-\sum_{k=b}^{h} F_k}} \tag{4}$$

Where:

P_{nij} is the probability of scoring in category j for item i by learner n, M is the learner's ability, D is the learning resource difficulty and F is the item category threshold parameters.

Because of using three responses in the each learning resource, so the total probabilities for each learning resource are presented in the proposed formula (5):

$$P_{ij} = P_{iTR} \times P_{iAR} \times P_{iSR} \tag{5}$$

Formulas (6) and (7) are used to compute expected score and variance for M.

$$\text{Expected Score} = \sum_{i=1}^{L} \sum_{j=b}^{t} jP_{nij} \tag{6}$$

$$\text{Variance} = \sum_{i=1}^{L} \left[\left(\sum_{j=b}^{t} j^2 P_{nij} \right) - \left(\sum_{j=b}^{t} jP_{nij} \right)^2 \right] \tag{7}$$

The initial estimate of learner's ability M can be any finite value, as shown in the formula (8).

$$M = D_{mean} + \log\left(\frac{R - R_{Min}}{R_{Max} - R} \right) \tag{8}$$

Where: M is the initial estimate of learner's ability, D_{Mean} is the average item difficulty, R is the raw score, R_{Min} is the minimum possible score and R_{Max} is the maximum possible score. D_{mean} can be calculated by the following:

$$D_{mean} = \frac{1}{L} \sum_{i=1}^{L} D_i \tag{9}$$

Where, L is the calibrated polytomous items.

Equation (10) used for estimating better of learner's ability.

$$M' = M + \frac{R - Expected\ \ Score}{Variance} \tag{10}$$

Once responses and difficulties have been tuned, some or all of the calibrated items can administer to further learners and measure them as shown in the Fig. 2.

Algorithm: Learner's ability estimation
Input: I, J, D, F, R, R_{Min}, R_{Max}
Output: M': Learner's ability
Step 1: Collect responses by learner n to the desired subset of L.
If $R = R_{Min}$, then set $R = R_{\mathrm{Min}} + 0.3$
If $R = R_{Max}$, then set $R = R_{Max} - 0.3$
Step 2: Each item I has a calibration D_i, and each step J a calibration F_j, in user-scaled units. If not already in logits, convert these to logits.
Step 3: Compute the average item difficulty D_{mean}.
Step 4: Compute the initial estimate of learner's ability M.
Step 5: Compute expected score and variance for M.
Step 6: Obtain a better estimate M' of the measure M.
Step 7: if $|M'-M|>0.01$, then do the following steps:
1. Set $M=M'$.
2. Go back to step (5).
Step 8: Report final ability M'.
Step 9: End.

Fig. 2. Estimating measures with known polytomous item difficulties

5.2 Learning Sequence Recommendation Mechanism

Based on polytomous IRT, Information function computation provides valuable information about the precision measurement at a specific learner's ability level for each node in the activity tree. This method emphasizes that each learning activity with the corresponding difficulty parameter shows different information to learners. Learning experience with higher information value is more suitable to be recommended for the learner.

We suggest a further device for identifying the maximum information. The item information is the expected variance of scoring functions based on probability along the observation ability M, so the expected value $E(M)$ can be expressed as:

$$E(M)=\sum_{j=b}^{t} jP_{ij} \tag{11}$$

Let $I(M)$ be the information function represents the information contributed by specific learning activity i across the range of M and corresponds to the slope of the item characteristic curve, the item's model variance, as shown below:

$$I_I(M)=\left(\sum_{j=b}^{t} j^2 P_{ij}\right)-\left(E(M)\right)^2 \tag{12}$$

6 Experimental Results and Evaluation

In order to provide the proof of feasibility study and verify the learning performance for the proposed system, we used a SCORM 2004 Photoshop examples version 1.1 released by [18], it contains a collection of learning resources about Adobe Photoshop. In this system, the range of both learner's ability and learning resource difficulty are limited from -3, which is very weak ability or very easy material, to 3, which is very high ability or very difficult material. To evaluate the performance of the proposed system, three experiences are performed with three sequence modes: *A recommended choice mode* means learner can click on 19 recommended learning resources. *A linear choice mode*, the learner must experience and complete the "introduction" first, then all modules and lessons in a linear order, directed by the proposed system. Finally *a freely choice mode*, in this mode, the learner can 'jump' (select) individual lessons and specific modules in any order regardless any system recommendation. Paired t-test was used to investigate the statistical difference between learning resource difficulty and the learner ability for three experiences at significant level (α) of 0.05.

Table 1. Paired t-Test for recommended choice mode

Data	Mean	SD^2	Std.Deviation	Std. Error	Minimum
Difficulty	0.74894	2.40916	1.55215	0.35608	-1.50000
Ability	0.33839	2.11821	1.45541	0.33389	-1.73690

t = 3.9007.
p = 0.0010 (lower significant level) = higher correlation coefficient.

Table 2. Paired t-Test for sequential mode

Data	Mean	SD^2	Std.Deviation	Std. Error	Minimum
Difficulty	0.07550	2.89231	1.70068	0.39016	-2.25000
Ability	0.38362	0.38672	0.62187	0.14267	-0.69556

t = -0.7912.
p = 0.43913 (lower to medium significant level) = lower to medium correlation.

Table 3. Paired t-Test for freely choice mode

Data	Mean	SD^2	Std.Deviation	Std. Error	Minimum
Difficulty	0.07556	2.89204	1.70000	0.39014	-2.2500
Ability	0.15268	0.46331	0.68067	0.15616	-1.91428

t = -0.15874.
p = 0.87564 (higher significant level) = lower correlation coefficient.

The results are shown in Tables 1 through 3. From these Tables, it can be seen that the P-value for experiments 1, 2 and 3 are 0.0010, 0.43913 and 0.87564, respectively. This indicates that a higher correlation was found with experiment 1, whereas lower one obtained with examples 2 and 3. The results also revealed that the standard deviation between the analysis groups (i.e. difficulty & ability) in experiment 1 is much close. That is, the experimental results show that the proposed system can recommend

suitable learning experience to the learner with high correlation degree between learner's ability and recommended learning recourse, resulting in increased the learning efficiency and learning performance.

7 Conclusions

This work proposes personalized SCORM learning experience based on RSM. First, the difficulty parameters of the learning resource can be correctly determined and distributed over activity tree by the proposed difficulties distributed process. Second, both non-crisp explicit feedback response and implicit feedback responses can be collected by proposed learner's feedback mechanism and learner's ability can also be correctly estimated by using proposed RSM. Finally, comprehensive learning experiences are provided to the learners by generating list of ranking learning paths.

Furthermore, A Paired t-test has successfully applied to investigate the statistical difference between leaner's abilities curve and learning resources curve difficulty. Experimental results show that the proposed system can exactly provide the closer learning resource to the learner's ability, resulting in increased the learning efficiency and learning performance.

References

1. Sharable Content Object Reference Model (SCORM) 2004, Advanced Distributed Learning, http://www.adlnet.org/
2. Embretson, S.E., Reise, S.P.: Item Response Theory for Psychologists. Lawrence Erlbaum, Mahwah (2000)
3. Baker, Frank, B.: Item Response Theory: Parameter Estimation Techniques. Marcel Dekker, New York (1992)
4. Hambleton, R.K.: Item Response Theory: Principles and Applications. Kluwer Nijhoff Publisher, Boston (1985)
5. Nering, M.L.: The Distribution of Indexes of Person Fit within the Computerized Adaptive Testing Environment. Applied Psychological Measurement 21, 115–127 (1997)
6. Krimpen-Stoop, V., Edith, M.L.A., Meijer, R.R.: The Null Distribution of Person-fit Statistics for Conventional and Adaptive Tests. Applied Psychological Measurement 23, 327–345 (1999)
7. Krimpen-Stoop, V., Edith, M.L.A., Meijer, R.R.: Detecting Person Misfit in Adaptive Testing Using Statistical Process Control Techniques. Kluwer-Nijhoff, Boston (2000)
8. Baek, S.G.: Computerized Adaptive Attitude Testing Using the Partial Credit Model. Dissertation Abstracts International 55(7-A), 1922 (1995)
9. Koch, W.R., Dodd, B.G., Fitzpatrick, S.J.: Computerized Adaptive Measurements of Attitudes. Measurement and Evaluation in Counseling and Development 23, 20–30 (1990)
10. Dodd, B.G., De Ayala, R.J., Koch, W.R.: Computerized Adaptive Testing with Polytomous Items. Applied Psychological Measurement 19, 5–22 (1995)
11. Andrich, D.: Application of a Psychometric Rating Model to Ordered Categories Which Are Scored with Successive Integers. Applied Psychological Measurement 2, 581–594 (1978)
12. John, M.L.: Estimating Measures with Known Polytomous Item Difficulties. Rasch Measurement Transactions 12, 638 (1998)

13. Wang, T.I., Tsai, K.H., Lee, M.C., Chiu, T.K.: Personalized Learning Objects Recommendation based on the Semantic-Aware Discovery and the Learner Preference Pattern. Educational Technology & Society 10, 84–105 (2007)
14. Su, J.M., Tseng, S.S., Wang, W., Weng, J.F., Yang, J.T.D., Tsai, W.N.: Learning Portfolio Analysis and Mining for SCORM Compliant Environment. Educational Technology & Society 9, 262–275 (2006)
15. Chen, C.M., Lee, H.M., Chen, Y.H.: Personalized E-learning system using item response theory. Computers & Education 44, 237–255 (2005)
16. Chen, C.M., Liu, C.Y., Chang, M.H.: Personalized Curriculum Sequencing Using Modified Iitem Response Theory for Web-based Instruction. Expert Systems with Applications 30, 378–396 (2006)
17. Chen, C.M., Duh, L.J., Liu, C.Y.: A Personalized Courseware Recommendation System based on Fuzzy Item Response Theory. IEEE 2004, 305–308 (2004)
18. ADL Technical Team. SCORM Photoshop Examples Version 1.0 (2004), http://www.adlnet.gov/downloads/index.cfm

The Key Theorem of Learning Theory on Uncertainty Space

Shujing Yan, Minghu Ha*, Xiankun Zhang, and Chao Wang

College of Mathematical and Computer Sciences, Hebei University,
Baoding 071002, China
yanshujing1@163.com, mhha@mail.hbu.edu.cn,
zhangxiankun08@163.com, wang222chao@163.com

Abstract. Statistical Learning Theory is commonly regarded as a sound framework within which we handle a variety of learning problems in presence of small size data samples. However, since the theory is based on probability space, it hardly handles statistical learning problems on uncertainty space. In this paper, the Statistical Learning Theory on uncertainty space is investigated. The Khintchine law of large numbers on uncertainty space is proved. The definitions of empirical risk functional, expected risk functional and empirical risk minimization principle on uncertainty space are introduced. On the basis of these concepts, the key theorem of learning theory on uncertainty space is introduced and proved.

Keywords: Uncertain measure, Expected risk functional, Empirical risk functional, Empirical risk minimization principle, the key theorem.

1 Introduction

Statistical Learning Theory (SLT) [1-4] was introduced in the late 1960's by Vapnik et al. and developed maturely in mid 1990's, concerning itself mainly with the statistic principles when samples are limited. In the period between 1992 and 1995, a novel method of pattern recognition--Support Vector Machine (SVM) was provided based on SLT, showing the unique advantages of solving the small-sample, nonlinear and high-dimensional pattern recognition problems. Many scholars believe that SLT and SVM are becoming a new hot area in the field of machine learning, and will promote the development of machine learning theory and technology [1-10].

The key theorem is an important part of SLT. It replaces the problem of consistency with the problems of uniform convergence, and it asserts that the conditions of consistency of the empirical risk minimization principle (ERM) are necessarily and sufficiently determined by the "worst" function of the set of functions [1-3].

It is accepted by the academic that SLT is a good theory in dealing with the small-sample learning theory problem, but there are still some shortcomings such as the fact that SLT is established on probability space. As the condition of additivity of probability is too strong, sometimes it can not be satisfied in many applications. Additivity

* Corresponding author.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 699–706, 2009.

was challenged by non-additive measure. In recent years, some academicians such as Minhu Ha, Yan Li [5], noticed this problem and extended probability space into Sugeno space. This proved the key theorem of SLT and bounds on the rate of convergence on Sugeno space. Minghu Ha, and Yunchao Bai [7], extended probability space into credibility space developing the key theorem on credibility space. In order to deal with general uncertainty, Liu [11] founded uncertainty theory. Uncertainty theory provides the commonness of probability theory, credibility theory and chance theory [11-13].

In this paper, we define empirical risk functional, expected risk functional and empirical risk minimization principle and prove the key theorem of learning theory on uncertainty space. We expand the study of SLT to uncertainty space.

2 Preliminaries

In this section, we review some basic notions, which will be of interest in the ensuing investigation. The reader can refer to [11, 12] for further detail.

Let Γ be a nonempty set, and let L be a σ-algebra over Γ. Each element $\Lambda \in \mathrm{L}$ is called an event. In order to present an axiomatic definition of uncertain measure, it is necessary to assign to each event Λ a number $\mathrm{M}\{\Lambda\}$ which indicates the level that Λ will occur. In order to ensure that the number $\mathrm{M}\{\Lambda\}$ has certain mathematical properties, Liu [11] proposed the following four axioms.

Axiom 1. (Normality) $\mathrm{M}\{\Gamma\}=1$.

Axiom 2. (Monotonicity) $\mathrm{M}\{\Lambda_1\} \le \mathrm{M}\{\Lambda_2\}$ whenever $\Lambda_1 \subset \Lambda_2$.

Axiom 3. (Self-Duality) $\mathrm{M}\{\Lambda\} + \mathrm{M}\{\Lambda^c\}=1$ for any event Λ.

Axiom 4. (Countable Subadditivity)For every countable sequence of events $\{\Lambda_i\}$, we have

$$\mathrm{M}\left\{\bigcup_{i=1}^{\infty} \Lambda_i\right\} \le \sum_{i=1}^{\infty} \mathrm{M}\left(\Lambda_i\right).$$

Definition 1. [11] The set function M is called an uncertain measure if it satisfies the normality, monotonicity, self-duality, countable subadditivity axioms. The triplet $(\Gamma, \mathrm{L}, \mathrm{M})$ is called an uncertainty space.

Throughout this paper, unless otherwise stated, $(\Gamma, \mathrm{L}, \mathrm{M})$ is an uncertainty space.

Definition 2. [11] An uncertain variable ξ is a measurable function from an uncertainty space to the set of real number, i.e., for any Borel set B of real numbers, the set

$$\{\xi \in B\} = \{\gamma \in \Gamma \,|\, \xi(\gamma) \in B\}$$

is an event.

Definition 3. [11] The uncertainty distribution $\Phi : R \to [0,1]$ of an uncertain variable ξ is defined by

$$\Phi(x) = \mathrm{M}\{\gamma \in \Gamma \,|\, \xi(\gamma) \le x\}.$$

Definition 4. [11] The uncertainty density function $\phi: R \to [0,+\infty]$ of an uncertain variable ξ is a function such that

$$\Phi(x) = \int_{-\infty}^{x} \phi(y)dy$$

$$\int_{-\infty}^{+\infty} \phi(y)dy = 1$$

where Φ is the uncertainty distribution of ξ.

Definition 5. [11] Let ξ be an uncertain variable. Then the expected value of ξ is defined by

$$E[\xi] = \int_0^{+\infty} \mathrm{M}\{\xi \geq r\}dr - \int_{-\infty}^{0} \mathrm{M}\{\xi \leq r\}dr$$

provided that at least one of the two integrals is finite.

Proposition 1. [11] Let ξ be an uncertain variable with finite expected value. Then for any real number a and b, we have

$$E[a\xi + b] = aE[\xi] + b.$$

Definition 6. [11] Let ξ be an uncertain variable with finite expected value e. Then the variance of ξ is defined by

$$V[\xi] = E[(\xi - e)^2].$$

Proposition 2. [11] If ξ is an uncertain variable with finite expected value, a and b are real numbers, then

$$V[a\xi + b] = a^2 V[\xi].$$

Definition 7. [11] The uncertain variables $\xi_1, \xi_2, \cdots, \xi_n$ are said to be independent if

$$E\left[\sum_{i=1}^{n} f_i(\xi_i)\right] = \sum_{i=1}^{n} E[f_i(\xi_i)]$$

for any measurable functions $f_1, f_2, \cdots, f_n$ provided that the expected values exist and are finite.

Proposition 3. [11] If ξ and η are independent uncertain variables with finite expected values, then we have

$$E[a\xi + b\eta] = aE[\xi] + bE[\eta]$$

for any real numbers a and b.

Definition 8. [11] The uncertain variables ξ and η are identically distributed if

$$\mathrm{M}\{\xi \in B\} = \mathrm{M}\{\eta \in B\}$$

for any Borel set B of real numbers.

3 The Key Theorem of Learning Theory on Uncertainty Space

In this section, we first give Khintchine Law of Large Numbers on uncertainty space.

Theorem 1. (Khintchine Law of Large Numbers) Let $\xi_1, \xi_2, \cdots, \xi_n$ be uncertain random variables sequence with independent identically distribution, ξ_n $(n = 1, 2, \cdots)$ have the same finite expected value a and variance M . Then for any $\varepsilon > 0$, the following inequality holds true.

Proof. Let

$$\xi = \frac{1}{n}\sum_{i=1}^{n}\xi_i \, , V[\xi_i] = M \ .$$

Then we have

$$E\xi = E\left[\frac{1}{n}\sum_{i=1}^{n}\xi_i\right] = \frac{1}{n}E\left[\sum_{i=1}^{n}\xi_i\right] = \frac{1}{n}\left[\sum_{i=1}^{n}E\xi_i\right] = \frac{1}{n}na = a \ .$$

According to Chebyshev Inequality [11] and the equation above, we have

$$\mathrm{M}\left\{\left|\frac{1}{n}\sum_{i=1}^{n}\xi_i - a\right| \geq \varepsilon\right\} = \mathrm{M}\left\{\left|\xi - E\xi\right| \geq \varepsilon\right\} \leq \frac{V[\xi]}{\varepsilon^2} = \frac{V[\frac{1}{n}\sum_{i=1}^{n}\xi_i]}{\varepsilon^2}$$

$$= \frac{\sum_{i=1}^{n}V\xi_i}{n^2\varepsilon^2} = \frac{M}{n\varepsilon^2} \to 0 \ .$$

The theorem is proved.

Definition 9. Let $\Phi(\mathrm{z})$ be an uncertainty distribution on uncertainty space $(\Gamma, \mathrm{L}, \mathrm{M})$, $z_1, z_2 \cdots, z_l$ be independent identically distributed samples. We introduce a function set $Q(z,\alpha), \alpha \in \Lambda$, expected risk functional and empirical risk functional are defined as follow

$$R_U(\alpha) = E\left[Q(z,\alpha)\right] \tag{1}$$

$$R_{Uemp}(\alpha) = \frac{1}{l}\sum_{i=1}^{l}Q(z_i,\alpha) \tag{2}$$

Definition 10. (Empirical Risk Minimization Principle) Let $Q(z,\alpha_0)$ minimize the expected risk, $Q(z,\alpha_l)$ minimize the empirical risk. We take $Q(z,\alpha_l)$ as an approximation of $Q(z,\alpha_0)$ .The principle of how to solve uncertain risk minimization problem is called Empirical Risk Minimization Principle on uncertainty space (UERM).

Definition 11. We say that the UERM method is nontrivially consistent for the set of $\{Q(z,\alpha), \alpha \in \Lambda\}$ if for any nonempty subset $\Lambda(c)$, $c \in (-\infty, +\infty)$, of this set of functions is defined as

$$\Lambda(c) = \{\alpha : E\left[Q(z,\alpha)\right] \geq c, c \in (-\infty, +\infty)\}$$

the convergence

$$\inf_{\alpha\in\Lambda(c)} R_{U\,emp}(\alpha) \overset{\mathrm{M}}{\underset{l\to\infty}{\to}} \inf_{\alpha\in\Lambda(c)} R_U(\alpha)$$

is valid.

Definition 13. We say that empirical risk converge uniformly one-side to expected risk for the set of $\{Q(z,\alpha),\alpha\in\Lambda\}$ and the uncertainty distribution $\Phi(z)$ if for any $\varepsilon>0$,

$$\lim_{l\to\infty}\mathrm{M}\left\{\sup_{\alpha\in\Lambda}\left(R_U(\alpha)-R_{Uemp}(\alpha)\right)>\varepsilon\right\}=0$$

is valid.

Theorem 2. (The Key Theorem) Let $\{Q(z,\alpha),\alpha\in\Lambda\}$ be a set of functions, and $\Phi(z)$ be all the uncertainty distribution that satisfy the condition

$$a\le E\left[Q(z,\alpha)\right]\le A,\alpha\in\Lambda$$

where the constant *a* and *A* exist. Then for the UEMR principle to be nontrivially consistent, it is necessary and sufficient that the empirical risk converge uniformly one-side to the expected risk.

Proof. Necessary: Suppose empirical risk minimization method is nontrivially consistent for the set of $\{Q(z,\alpha),\alpha\in\Lambda\}$. According to the definition of nontrivially consistency, $\forall c\in(-\infty,+\infty)$ such that

$$\Lambda(c)=\{\alpha: E\left[Q(z,\alpha)\right]\ge c\}$$

is valid, then the convergence

$$\inf_{\alpha\in\Lambda(c)} R_{U\,emp}(\alpha) \overset{\mathrm{M}}{\underset{l\to\infty}{\to}} \inf_{\alpha\in\Lambda(c)} R_U(\alpha)$$

is valid.

$\forall\varepsilon>0$, we consider a finite sequence $a_1,a_2,\cdots,a_n$ such that

$$\left|a_{i+1}-a_i\right|<\frac{\varepsilon}{2}\qquad i=1,2,\cdots,n-1$$

where $a_1=A, a_n=B$, let T_k be the event,

$$\inf_{\alpha\in\Lambda(\alpha_k)} R_{U\,emp}(\alpha)<\inf_{\alpha\in\Lambda(\alpha_k)} R_U(\alpha)-\frac{\varepsilon}{2}\quad . \tag{3}$$

According to (3), $\mathrm{M}(T_k)\underset{l\to\infty}{\to}0$.

Let

$$T=\bigcup_{k=1}^{n}T_k$$

as *n* is finite, and for any *k*, (3) is valid, so

$$\mathrm{M}(T)\underset{l\to\infty}{\to}0\quad .$$

We denote *A* by the event

$$\sup_{\alpha\in\Lambda}(R_U(\alpha)-R_{U\,emp}(\alpha))>\varepsilon\quad .$$

Suppose A occurs, then there exists $\alpha^* \in \Lambda$, such that

$$R_U(\alpha^*) - \varepsilon > R_{U\,emp}(\alpha^*) \ .$$

From α^* we can find a k, such that $\alpha^* \in \Lambda(\alpha_k)$ and

$$R_U(\alpha^*) - \alpha_k < \frac{\varepsilon}{2}$$

holds true, by

$$\inf_{\alpha \in \Lambda(\alpha_k)} R_U(\alpha) \geq \alpha_k$$

for the chosen set $\Lambda(\alpha_k)$, the inequality

$$\mathrm{R}(\alpha^*) - \inf_{\alpha \in \Lambda(\alpha_k)} R(\alpha) < \frac{\varepsilon}{2}$$

is valid, so for the chosen set α^* and $\Lambda(\alpha_k)$, the follow inequality holds true:

$$\inf_{\alpha \in \Lambda(\alpha_k)} R_U(\alpha) - \frac{\varepsilon}{2} > R_{U\,emp}(\alpha^*) \geq \inf_{\alpha \in \Lambda(\alpha_k)} R_{U\,emp}(\alpha)$$

i.e. T_k occurs, so T occurs, then

$$\mathrm{M}\ (A) < \mathrm{M}\ (T) \underset{l \to \infty}{\to} 0$$

then

$$\mathrm{M}\ \{\sup_{\alpha \in \Lambda}(R_U(\alpha) - R_{U\,emp}(\alpha)) > \varepsilon\} \underset{l \to \infty}{\to} 0$$

the necessity is valid.

Sufficiency: We need to prove:

$$\lim_{l \to \infty} \mathrm{M}\ \{\left| \inf_{\alpha \in \Lambda(\alpha)} R_U(\alpha) - \inf_{\alpha \in \Lambda(\alpha)} R_{U\,emp}(\alpha) \right| > \varepsilon\} = 0$$

We denote A by the event

$$\left| \inf_{\alpha \in \Lambda(\alpha)} R_U(\alpha) - \inf_{\alpha \in \Lambda(\alpha)} R_{U\,emp}(\alpha) \right| > \varepsilon$$

then A is the union of two events

$$A = A_1 \cup A_2$$

where

$$A_1 = \{\inf_{\alpha \in \Lambda(c)} R_U(\alpha) + \varepsilon < \inf_{\alpha \in \Lambda(c)} R_{U\,emp}(\alpha)\}$$

$$A_2 = \{\inf_{\alpha \in \Lambda(c)} R_U(\alpha) - \varepsilon > \inf_{\alpha \in \Lambda(c)} R_{U\,emp}(\alpha)\} \ .$$

Suppose A_1 occurs, we can find a function $Q(z, \alpha^*), \alpha^* \in \Lambda(c)$ such that

$$R_U(\alpha^*) < \inf_{\alpha \in \Lambda(c)} R_U(\alpha) + \frac{\varepsilon}{2}$$

then

$$R_U(\alpha^*) + \frac{\varepsilon}{2} < \inf_{\alpha \in \Lambda(c)} R_U(\alpha) + \varepsilon < \inf_{\alpha \in \Lambda(c)} R_{U\,emp}(\alpha) < R_{U\,emp}(\alpha^*)$$

that is

$$R_U\left(\alpha^*\right)+\frac{\varepsilon}{2}<R_{U\,emp}\left(\alpha^*\right)$$

is valid.
By Theorem 1, we get

$$\mathrm{M}\,(A_1)\le \mathrm{M}\,\{(R_{U\,emp}\left(\alpha^*\right)-R_U\left(\alpha^*\right))>\frac{\varepsilon}{2}\}\underset{l\to\infty}{\to}0\ .$$

On the another hand, if A_2 occurs, there exists a function $Q\left(z,\alpha^{**}\right)$, $\alpha^{**}\in\Lambda(c)$ such that

$$R_{U\,emp}\left(\alpha^{**}\right)+\frac{\varepsilon}{2}<\inf_{\alpha\in\Lambda(c)}R_{U\,emp}\left(\alpha\right)+\varepsilon<\inf_{\alpha\in\Lambda(c)}R_U\left(\alpha\right)<R_U\left(\alpha^{**}\right)$$

hence

$$\mathrm{M}\,\left(A_2\right)\le \mathrm{M}\,\left\{(R_U\left(\alpha^{**}\right)-R_{U\,emp}\left(\alpha^{**}\right))>\frac{\varepsilon}{2}\right\}<\mathrm{M}\,\left\{\sup_{\alpha\in\Lambda}(R_U\left(\alpha\right)-R_{U\,emp}\left(\alpha\right))>\frac{\varepsilon}{2}\right\}\underset{l\to\infty}{\to}0$$

By

$$\mathrm{M}\,\left(A\right)\le \mathrm{M}\,\left(A_1\right)+\mathrm{M}\,\left(A_2\right)$$

we have

$$\mathrm{M}\,(A)\underset{l\to\infty}{\to}0\ .$$

The theorem is proved.

4 Conclusions

This paper introduces and proves the key theorem of SLT on uncertain space which is broader than probability space, credibility space and chance space. It lays the theoretical foundation for establishing the key theorem of SLT and SVM on uncertainty space. Further investigations might focus on such fundamental issues as structural risk minimization, VC dimension theory and application aspects such as support vector machines on uncertainty space.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (No. 60773062), the Natural Science Foundation of Hebei Province of China (No. 2008000633), the Key Scientific Research Project of Education Department of Hebei Province of China (No.2005001D) and the Key Scientific and Technical Research Project of the Ministry of Education of China (No. 206012).

References

1. Vapnik, V.N.: Statistical Learning Theory. A Wiley-Interscience Publication, New York (1998)
2. Vapnik, V.N.: The Nature of Learning Theory. New York, Heidelberg (1995)
3. Vapnik, V.N.: An Overview of Statistical Learning Theory. In: 8th IEEE Transaction on Neural Network, pp. 988–999. IEEE Press, New York (1999)

4. Su, C., Yang, C.H.: Feature Selection for the SVM: an Application to Hypertension Diagnosis. Expert Systems with Applications 34, 754–763 (2008)
5. Ha, M.H., Li, Y., Li, J., Tian, D.Z.: Key Theorem of Learning Theory and Bounds on the Rate of Convergence Based on Sugeno space. Science in China Ser. E Information Sciences 36(4), 398–410 (2006)
6. Zhang, X.G.: Introduction to Statistical Learning Theory and Support Vector Machines. Acta Automatica Sinica 26(1), 32–42 (2000)
7. Ha, M.H., Bai, Y.C., Tang, W.G.: The Sub-key Theorem on Credibility Measure Space. In: Proceeding of 2003 International Conference on Machine Learning and Cybernetics, Xi'an, China, vol. 5, pp. 3264–3268 (2003)
8. Huang, C.L., Chen, M.C., Wang, C.J.: Credit Scoring with a Data Mining Approach Based on Support Vector Machines. Expert System with Applications 33, 847–856 (2007)
9. Jin, B., Tang, Y.C., Zhang, Y.Q.: Support Vector Machines with Genetic Fuzzy Feature Transformation for Biomedical Data Classification. Information Sciences 177, 476–489 (2007)
10. Ha, M.H., Tian, J.: The Theoretical Foundations of Statistical Learning Theory Based on Fuzzy Number Samples. Information Sciences 178(16), 3240–3246 (2008)
11. Liu, B.: Uncertainty Theory, 2nd edn. Springer, Berlin (2007)
12. Liu, B.: Uncertainty Theory, 3rd edn. (2008), http://orsc.edu.cn/liu/ut.pdf
13. Liu, B., Zhao, R.Q.: Uncertain Programming with Application. Tsinghua University Press, Beijing (2003)

A New Instance-Based Label Ranking Approach Using the Mallows Model

Weiwei Cheng and Eyke Hüllermeier

Mathematics and Computer Science
University of Marburg, Germany
{cheng,eyke}@mathematik.uni-marburg.de

Abstract. In this paper, we introduce a new instance-based approach to the label ranking problem. This approach is based on a probability model on rankings which is known as the Mallows model in statistics. Probabilistic modeling provides the basis for a theoretically sound prediction procedure in the form of maximum likelihood estimation. Moreover, it allows for complementing predictions by diverse types of statistical information, for example regarding the reliability of an estimation. Empirical experiments show that our approach is competitive to start-of-the-art methods for label ranking and performs quite well even in the case of incomplete ranking information.

Keywords: Instance-based learning, Label ranking, Classification, Maximum likelihood estimation.

1 Introduction

The topic of learning preferences has attracted increasing attention in the recent machine learning literature [1]. Label ranking, a particular preference learning scenario, studies the problem of learning a mapping from instances to rankings over a finite number of predefined labels. It can be considered as a natural generalization of the conventional classification problem, where only a single label is requested instead of a ranking of all labels.

Various approaches for label ranking have been proposed in recent years. Typically, these are extensions of learning algorithms used in binary classification problems. Ranking by pairwise comparison (RPC) is a natural extension of pairwise classification, in which binary preference models are learned for each pair of labels, and the predictions of these models are combined into a ranking of all labels [1]. Two other approaches, constraint classification (CC) and log-linear models for label ranking (LL), seek to learn linear utility functions for each individual label instead of preference predicates for pairs of labels [2,3].

In this paper, we are interested in an alternative to model-based approaches, namely the use of an *instance-based* approach. Instance-based or case-based learning algorithms have been applied successfully in various fields, such as machine learning and pattern recognition, for a long time [4]. These algorithms simply store the training data, or at least a selection thereof, and defer the

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 707–716, 2009.

processing of this data until an estimation for a new instance is requested, a property distinguishing them from typical model-based approaches. Instance-based approaches therefore have a number of potential advantages, especially in the context of the label ranking problem.

As a particular advantage of delayed processing, these learning methods may estimate the target function *locally* instead of inducing a global prediction model for the entire input domain (instance space) $\mathbb{X}$. Predictions are typically obtained using only a small, locally restricted subset of the entire training data, namely those examples that are close to the query $\boldsymbol{x} \in \mathbb{X}$ (hence $\mathbb{X}$ must be endowed with a distance measure). These examples are then *aggregated* in a reasonable way. As aggregating a finite set of objects from an output space Ω is often much simpler than representing a complete $\mathbb{X} \to \Omega$ mapping in an explicit way, instance-based methods are especially appealing if Ω has a complex structure.

In label ranking, Ω corresponds to the set of all rankings of an underlying label set $\mathcal{L}$. To represent an Ω-valued mapping, the aforementioned model-based approaches encode this mapping in terms of conventional binary models, either by a large set of such models in the original label space $\mathcal{L}$ (RPC), or by a single binary model in an expanded, high-dimensional space (CC, LL). Since for instance-based methods, there is no need to represent an $\mathbb{X} \to \Omega$ mapping explicitly, such methods can operate on the original target space Ω directly.

The paper is organized as follows: In Section 2, we introduce the problem of label ranking in a more formal way. The core idea of our instance-based approach to label ranking, namely maximum likelihood estimation based on a special probability model for rankings, is discussed in Section 4. The model itself is introduced beforehand in Section 3. Section 5 is devoted to experimental results. The paper ends with concluding remarks in Section 6.

2 Label Ranking

Label ranking can be seen as an extension of the conventional setting of classification. Roughly speaking, the former is obtained from the latter through replacing single class labels by complete label rankings. So, instead of associating every instance $\boldsymbol{x}$ from an instance space $\mathbb{X}$ with one among a finite set of class labels $\mathcal{L} = \{\lambda_1 \dots \lambda_n\}$, we now associate $\boldsymbol{x}$ with a total order of the class labels, that is, a complete, transitive, and asymmetric relation $\succ_{\boldsymbol{x}}$ on $\mathcal{L}$ where $\lambda_i \succ_{\boldsymbol{x}} \lambda_j$ indicates that λ_i precedes λ_j in the ranking associated with $\boldsymbol{x}$. It follows that a ranking can be considered as a special type of preference relation, and therefore we shall also say that $\lambda_i \succ_{\boldsymbol{x}} \lambda_j$ indicates that λ_i is *preferred* to λ_j given the instance $\boldsymbol{x}$. To illustrate, suppose that instances are students (characterized by attributes such as sex, age, and major subjects in secondary school) and $\succ$ is a preference relation on a fixed set of study fields such as Math, CS, Physics.

Formally, a ranking $\succ_{\boldsymbol{x}}$ can be identified with a permutation $\pi_{\boldsymbol{x}}$ of the set $\{1 \dots n\}$. It is convenient to define $\pi_{\boldsymbol{x}}$ such that $\pi_{\boldsymbol{x}}(i) = \pi_{\boldsymbol{x}}(\lambda_i)$ is the position of λ_i in the ranking. This permutation encodes the (ground truth) ranking:

$$\lambda_{\pi_{\boldsymbol{x}}^{-1}(1)} \succ_{\boldsymbol{x}} \lambda_{\pi_{\boldsymbol{x}}^{-1}(2)} \succ_{\boldsymbol{x}} \dots \succ_{\boldsymbol{x}} \lambda_{\pi_{\boldsymbol{x}}^{-1}(n)} \ ,$$

where $\pi_{\boldsymbol{x}}^{-1}(j)$ is the index of the label at position j in the ranking. The class of permutations of $\{1 \dots n\}$ (the symmetric group of order n) is denoted by Ω. By abuse of terminology, though justified in light of the above one-to-one correspondence, we refer to elements $\pi \in \Omega$ as both permutations and rankings.

In analogy with the classification setting, we do not assume that there exists a deterministic $\mathbb{X} \longrightarrow \Omega$ mapping. Instead, every instance is associated with a *probability distribution* over Ω. This means that, for each $\boldsymbol{x} \in \mathbb{X}$, there exists a probability distribution $\Pr(\cdot \,|\, \boldsymbol{x})$ such that, for every $\pi \in \Omega$,

$$\Pr(\pi \,|\, \boldsymbol{x}) \tag{1}$$

is the probability that $\pi_{\boldsymbol{x}} = \pi$.

The goal in label ranking is to learn a "label ranker" in the form of an $\mathbb{X} \longrightarrow \Omega$ mapping. As training data, a label ranker uses a set of instances $\boldsymbol{x}_k$, $k = 1 \dots m$, together with information about the associated rankings $\pi_{\boldsymbol{x}_k}$. Ideally, complete rankings are given as training information. From a practical point of view, however, it is also important to allow for incomplete information in the form of a ranking

$$\lambda_{\pi_{\boldsymbol{x}}^{-1}(i_1)} \succ_{\boldsymbol{x}} \lambda_{\pi_{\boldsymbol{x}}^{-1}(i_2)} \succ_{\boldsymbol{x}} \dots \succ_{\boldsymbol{x}} \lambda_{\pi_{\boldsymbol{x}}^{-1}(i_k)} \ ,$$

where $\{i_1, i_2 \dots i_k\}$ is a subset of the index set $\{1 \dots n\}$ such that $1 \leq i_1 < i_2 < \dots < i_k \leq n$. For example, for an instance $\boldsymbol{x}$, it might be known that $\lambda_2 \succ_{\boldsymbol{x}} \lambda_1 \succ_{\boldsymbol{x}} \lambda_5$, while no preference information is given about the labels λ_3 or λ_4.

To evaluate the predictive performance of a label ranker, a suitable loss function on Ω is needed. In the statistical literature, several distance measures for rankings have been proposed. One commonly used measure is the number of discordant pairs,

$$D(\pi, \sigma) \ = \ \{ \, (i,j) \,|\, i < j, \, \pi(i) > \pi(j) \text{ and } \sigma(i) < \sigma(j) \, \} \ , \tag{2}$$

which is closely related to the Kendall's tau coefficient. In fact, the latter is a normalization of (2) to the interval $[-1, 1]$ that can be interpreted as a correlation measure (it assumes the value 1 if $\sigma = \pi$ and the value -1 if σ is the reversal of π). Kendall's tau is a natural, intuitive, and easily interpretable measure [5]. We shall focus on (2) throughout the paper, even though other distance measures could of course be used. A desirable property of any distance $D(\cdot)$ is its invariance toward a renumbering of the elements (renaming of labels). This property is equivalent to the *right invariance* of $D(\cdot)$, namely $D(\sigma\nu, \pi\nu) = D(\sigma, \pi)$ for all $\sigma, \pi, \nu \in \Omega$, where $\sigma\nu = \sigma \circ \nu$ denotes the permutation $i \mapsto \sigma(\nu(i))$. The distance (2) is right-invariant, and so are most other commonly used metrics on Ω.

3 The Mallows Model

So far, we did not make any assumptions about the probability measure (1) despite its existence. To become more concrete, we resort to a distance-based

probability model introduced by Mallows [5]. The standard Mallows model is a two-parameter model that belongs to the exponential family:

$$\Pr(\sigma \,|\, \theta, \pi) = \frac{\exp(\theta D(\pi, \sigma))}{\phi(\theta, \pi)}, \tag{3}$$

where the two parameters are the location parameter (modal ranking, center ranking) $\pi \in \Omega$ and the spread parameter $\theta \leq 0$. For right-invariant metrics, it can be shown that the normalization constant does not depend on π and, therefore, can be written as a function $\phi(\theta)$ of θ alone. This is due to

$$\begin{aligned}\phi(\theta, \pi) = \sum_{\sigma \in \Omega} \exp(\theta D(\sigma, \pi)) &= \sum_{\sigma \in \Omega} \exp(\theta D(\sigma \pi^{-1}, e)) \\ &= \sum_{\sigma' \in \Omega} \exp(\theta D(\sigma', e)) = \phi(\theta) \ ,\end{aligned}$$

where $e = (1 \ldots n)$ is the identity ranking. More specifically, it can be shown that the normalization constant is given by [6]

$$\phi(\theta) = \prod_{j=1}^{n} \frac{1 - \exp(j\theta)}{1 - \exp(\theta)}, \tag{4}$$

and that the expected distance from the center is

$$\mathbb{E}\left[D(\sigma, \pi) \,|\, \theta, \pi\right] = \frac{n \exp(\theta)}{1 - \exp(\theta)} - \sum_{j=1}^{n} \frac{j \exp(j\theta)}{1 - \exp(j\theta)} \ . \tag{5}$$

Obviously, the Mallows model assigns the maximum probability to the center ranking π. The larger the distance $D(\sigma, \pi)$, the smaller the probability of σ becomes. The spread parameter θ determines how quickly the probability decreases, i.e., how peaked the distribution is around π. For $\theta = 0$, the uniform distribution is obtained, while for $\theta \to -\infty$, the distribution converges to the one-point distribution that assigns probability 1 to π and 0 to all other rankings.

4 Learning and Inference

Coming back to the label ranking problem and the idea of instance-based learning, consider a query instance $\boldsymbol{x} \in \mathbb{X}$ and let $\boldsymbol{x}_1 \ldots \boldsymbol{x}_k$ denote the nearest neighbors of $\boldsymbol{x}$ (according to an underlying distance measure on $\mathbb{X}$) in the training set, where $k \in \mathbb{N}$ is a fixed integer. Moreover, let $\sigma_1 \ldots \sigma_k \in \Omega$ denote the rankings associated, respectively, with $\boldsymbol{x}_1 \ldots \boldsymbol{x}_k$.

In analogy to the conventional settings of classification and regression, in which the nearest neighbor estimation principle has been applied for a long time, we assume that the probability distribution $\Pr(\cdot \,|\, \boldsymbol{x})$ on Ω is (at least

approximately) *locally constant* around the query $\boldsymbol{x}$. By furthermore assuming independence of the observations, the probability to observe $\boldsymbol{\sigma} = \{\sigma_1 \ldots \sigma_k\}$ given the parameters (θ, π) becomes

$$\begin{aligned} \Pr(\boldsymbol{\sigma} \,|\, \theta, \pi) &= \prod_{i=1}^{k} \Pr(\sigma_i \,|\, \theta, \pi) = \prod_{i=1}^{k} \frac{\exp\left(\theta D(\sigma_i, \pi)\right)}{\phi(\theta)} \\ &= \frac{\exp\left(\theta \sum_{i=1}^{k} D(\sigma_i, \pi)\right)}{\left(\prod_{j=1}^{n} \frac{1-\exp(j\theta)}{1-\exp(\theta)}\right)^k}. \end{aligned} \tag{6}$$

The maximum likelihood estimation (MLE) of (θ, π) is then given by those parameters that maximize this probability. It is easily verified that the MLE of π is given by

$$\hat{\pi} \;=\; \arg\min_{\pi} \sum_{i=1}^{k} D(\sigma_i, \pi), \tag{7}$$

i.e., by the (generalized) median of the rankings $\sigma_1 \ldots \sigma_k$. Moreover, the MLE of θ is derived from the average observed distance from $\hat{\pi}$, which is an estimation of the expected distance $\mathbb{E}\left[D(\sigma, \pi)|\theta, \pi\right]$:

$$\frac{1}{k} \sum_{i=1}^{k} D(\sigma_i, \hat{\pi}) = \frac{n \exp(\theta)}{1 - \exp(\theta)} - \sum_{j=1}^{n} \frac{j \exp(j\theta)}{1 - \exp(j\theta)}. \tag{8}$$

Since the right-hand side of (8) is monotone increasing, a standard line search quickly converges to the MLE [6].

Now, consider the more general case of incomplete preference information, which means that a ranking σ_i does not necessarily contain all labels. The probability of σ_i is then given by

$$\Pr(E(\sigma_i)) \;=\; \sum_{\sigma \in E(\sigma_i)} \Pr(\sigma \,|\, \theta, \pi) \;\;,$$

where $E(\sigma_i)$ denotes the set of all *consistent extensions* of σ_i: A permutation $\sigma \in \Omega$ is a consistent extension of σ if it ranks all labels that also occur in σ_i in the same order.

The probability of observing the neighbor rankings $\boldsymbol{\sigma} = (\sigma_1 \ldots \sigma_k)$ then becomes

$$\begin{aligned} \Pr(\boldsymbol{\sigma} \,|\, \theta, \pi) &= \prod_{i=1}^{k} \Pr(E(\sigma_i) \,|\, \theta, \pi) = \prod_{i=1}^{k} \sum_{\sigma \in E(\sigma_i)} \Pr(\sigma \,|\, \theta, \pi) \\ &= \frac{\prod_{i=1}^{k} \sum_{\sigma \in E(\sigma_i)} \exp\left(\theta D(\sigma, \pi)\right)}{\left(\prod_{j=1}^{n} \frac{1-\exp(j\theta)}{1-\exp(\theta)}\right)^k} \;\;. \end{aligned} \tag{9}$$

Computing the MLE of (θ, π) by maximizing this probability now becomes more difficult. For label sets of small to moderate size, say up to 7, one can afford a simple brute force approach, namely an exhaustive search over Ω to find the center ranking π, combined with a numerical procedure to optimize the spread θ. For larger label sets, this procedure becomes too inefficient. Here, we propose an approximation algorithm that can be seen as an instance of the EM (Expectation-Maximization) family of algorithms.

The algorithm works as follows. Starting from an initial (complete) center ranking $\hat{\pi}$, each incomplete neighbor ranking σ_i is replaced by the most probable consistent extension given $\hat{\pi}$. Regardless of θ, this extension is obviously given by a ranking in $\arg\min_{\sigma \in E(\sigma_i)} D(\sigma, \hat{\pi})$. It can be found by (minimally) re-ranking the center $\hat{\pi}$ so as to make it consistent with the incomplete ranking σ_i. Having replaced all neighbor rankings by their most probable extensions, an MLE (θ, π) can be derived as described for the case of complete information above. The center ranking $\hat{\pi}$ is then replaced by π, and the whole procedure is iterated until the center does not change any more. In the following, we discuss two subproblems of the algorithm in more detail, namely the solution of the median problem (7), which needs to be solved to find an MLE π, and the choice of an initial center ranking.

Solving the (generalized) median problem (7) is known to be NP-complete for Kendall's tau, i.e., if the distance D is given by the number of rank inversions [7]. To solve this problem approximately, we make use of the fact that Kendall's tau is well approximated by Spearman's rank correlation [8], and that the median can be computed for this measure (i.e., for D given by the sum of squared rank differences) by a procedure called *Borda count* [9]: Given a (complete) ranking σ_i of n labels, the top-label receives n votes, the second-ranked $n-1$ votes, and so on. Given k rankings $\sigma_1 \ldots \sigma_k$, the sum of the k votes are computed for each label, and the labels are then ranked according to their total votes.

The choice of the initial center ranking in the above algorithm is of course critical. To find a good initialization, we again resort to the idea of solving the problem (7) approximately using the Borda count principle. At the beginning, however, the neighbor rankings σ_k are still incomplete. To handle this situation, we make the simplifying assumption that the completions are uniformly distributed in $E(\sigma_i)$. Again, this is an approximation, since we actually proceed from the Mallows and not from the uniform model. On the basis of this assumption, we can show the following result (proof omitted due to space restrictions).

Theorem 1. *Let a set of incomplete rankings $\sigma_1 \ldots \sigma_k$ be given, and suppose the associated complete rankings $S_1 \ldots S_k$ to be distributed, respectively, uniformly in $E(\sigma_1) \ldots E(\sigma_k)$. The expected sum of distances $D(\pi, S_1) + \ldots + D(\pi, S_k)$, with D the sum of squared rank distances, becomes minimal for the ranking π which is obtained by a generalized Borda count, namely a Borda count with a generalized distribution of votes from incomplete rankings: If σ_i is an incomplete ranking of $m \leq n$ labels, then the label on rank $i \in \{1 \ldots m\}$ receives $(m-i+1)(n+1)/(m+1)$ votes, while each missing label receives a vote of $(n+1)/2$.*

Table 1. Statistics for the semi-synthetic and real datasets

dataset	#examples	#classes	#features
iris	150	3	4
wine	178	3	13
glass	214	6	9
vehicle	846	4	18
dtt	2465	4	24
cold	2465	4	24

5 Experimental Results

5.1 Methods

In this section, we compare our instance-based (nearest neighbor, NN) approach with existing methods for label ranking, namely ranking by pairwise comparison (RPC), constraint classification (CC), and log-linear models for label ranking (LL). Since space restrictions prevent from a detailed review, we refer to the original literature and [1] for a short review of these methods. Regarding the concrete implementation and parameterization of these methods, we also follow [1].

To fit the Mallows model, we test the two previously discussed variants, namely the exhaustive search which guarantees an optimal solution (NNE) and the approximation algorithm outlined in Section 4 (NNH). The parameter k (neighborhood size) was selected through cross validation on the training set. As a distance measure on the instance space we used the Euclidean distance (after normalizing the attributes).

5.2 Data

We used two real-world data sets, dtt and cold, from the bioinformatics field. These data sets contain two types of genetic data, namely phylogenetic profiles and DNA microarray expression data for the Yeast genome.[1] The genome consists of 2465 genes, and each gene is represented by an associated phylogenetic profile of length 24. Using these profiles as input features, we investigated the task of predicting a "qualitative" representation of an expression profile; see [1] for a detailed description and motivation of this task.

In addition to the real-world data sets, the following multiclass datasets from the UCI repository of machine learning databases and the Statlog collection were included in the experimental evaluation: iris, wine, glass, vehicle. For each of these datasets, a corresponding ranking dataset was generated in the following manner: We trained a naive Bayes classifier on the respective dataset. Then, for each example, *all* the labels present in the dataset were ordered with respect to decreasing predicted class probabilities (in the case of ties, labels with lower index are ranked first). Thus, by substituting the single labels contained in the

[1] This data is publicly available at `http://www1.cs.columbia.edu/compbio/`

Table 2. Experimental results in terms of Kendall's tau (mean and standard deviation) for different missing label rates (parameter p)

iris	*0%*	*10%*	*20%*	*30%*	*40%*	*50%*	*60%*	*70%*
RPC	.885±.068	.888±.064	.886±.060	.871±.074	.854±.082	.837±.089	.779±.110	.674±.139
CC	.836±.089	.825±.095	.815±.088	.807±.099	.788±.105	.766±.115	.743±.131	.708±.105
LL	.818±.088	.811±.089	.805±.087	.806±.087	.800±.091	.788±.087	.778±.096	.739±.186
NNE	.960±.036	**.956±.041**	**.941±.044**	**.934±.049**	**.915±.056**	**.882±.085**	**.859±.082**	**.812±.107**
NNH	**.966±.034**	.948±.036	.917±.051	.863±.072	.822±.088	.802±.084	.767±.122	.733±.104
wine								
RPC	.921±.053	.900±.067	.886±.073	.902±.063	.910±.065	.882±.082	.864±.097	.822±.118
CC	.933±.043	.918±.057	.929±.058	.911±.059	.922±.057	.885±.074	.853±.078	.802±.123
LL	.942±.043	.944±.046	.939±.051	**.944±.042**	.933±.062	.918±.065	.906±.072	**.864±.094**
NNE	.952±.048	.945±.051	.943±.055	.940±.054	**.941±.050**	**.930±.058**	**.910±.061**	.677±.173
NNH	**.953±.042**	**.949±.041**	**.949±.041**	.933±.048	.899±.075	.709±.186	.591±.210	.587±.180
glass								
RPC	**.882±.042**	**.875±.046**	**.867±.044**	**.851±.052**	**.840±.053**	**.813±.062**	.799±.054	.754±.076
CC	.846±.045	.848±.053	.838±.059	.835±.054	.833±.051	.807±.066	.789±.052	.747±.061
LL	.817±.060	.815±.061	.813±.063	.819±.062	.819±.060	.809±.066	**.806±.065**	**.807±.063**
NNE	.875±.063	.866±.059	.840±.059	.803±.062	.750±.071	.677±.066	.598±.082	.500±.078
NNH	.865±.059	.847±.062	.810±.056	.754±.069	.691±.063	.633±.061	.550±.069	.484±.079
vehicle								
RPC	.854±.025	.848±.025	.847±.024	.834±.026	.823±.032	.803±.033	.786±.036	.752±.041
CC	.855±.022	.848±.026	**.849±.026**	**.839±.025**	**.834±.026**	**.827±.026**	**.810±.026**	**.791±.030**
LL	.770±.037	.769±.035	.769±.033	.766±.040	.770±.038	.764±.031	.757±.038	.756±.036
NNE	**.863±.030**	**.859±.031**	.847±.029	.834±.031	.822±.030	.795±.033	.766±.034	.723±.036
NNH	.862±.025	.852±.024	.845±.030	.828±.029	.798±.031	.776±.033	.748±.032	.701±.047
dtt								
RPC	.174±.034	.172±.034	.168±.036	.166±.036	.164±.034	.153±.035	.144±.028	.125±.030
CC	.180±.037	.178±.034	.176±.033	**.172±.032**	.165±.033	.158±.033	.149±.031	.136±.033
LL	.167±.034	.168±.033	.168±.034	.168±.034	**.167±.033**	**.167±.036**	**.162±.032**	**.156±.034**
NNE	.182±.036	.179±.036	.173±.036	.169±.036	.162±.036	.161±.037	.154±.036	.136±.035
NNH	**.191±.034**	**.183±.037**	**.176±.036**	.168±.038	.163±.034	.146±.036	.145±.033	.128±.035
cold								
RPC	.221±.028	.217±.028	.213±.030	.212±.030	.208±.030	.201±.030	.188±.030	.174±.031
CC	.220±.029	.219±.030	.212±.030	.212±.028	.205±.024	.197±.030	.185±.031	.162±.035
LL	.209±.028	.210±.031	.206±.030	.210±.030	.203±.031	.203±.031	**.202±.032**	**.192±.031**
NNE	.230±.028	.226±.029	.220±.030	.213±.031	.199±.029	.195±.033	.190±.035	.188±.035
NNH	**.244±.026**	**.237±.028**	**.235±.031**	**.226±.024**	**.220±.029**	**.214±.029**	.199±.030	.192±.032

original multiclass datasets with the complete rankings, we obtain the label ranking datasets required for our experiments. A summary of the data sets and their properties is given in Table 1.

5.3 Experiments and Results

Results were derived in terms of the Kendall's tau correlation coefficient from five repetitions of a ten-fold cross-validation. To model incomplete preferences, we modified the training data as follows: A biased coin was flipped for every label in a ranking in order to decide whether to keep or delete that label; the probability for a deletion is specified by a parameter p.

The results are summarized in Table 2. As can be seen, NN is quite competitive to the model-based approaches and often outperforms these methods. In any case, it is always close to the best result. It is also remarkable that NN seems to be quite robust toward missing preferences and compares comparably well in this regard. This was not necessarily expected, since NN uses only local information, in contrast to the other approaches that induce global models. Our approximation algorithm NNH gives very good approximations of NNE

throughout and is especially appealing for large label sets: It dramatically reduces the runtime (not shown due to space restrictions) without any significant decrease of the performance.

A nice feature of our approach, not shared by the model-based methods, is that it comes with a natural measure of the reliability of a prediction. In fact, the smaller the parameter θ, the more peaked the distribution around the center ranking and, therefore, the more reliable this ranking becomes as a prediction. To test whether (the estimation of) θ is indeed a good measure of uncertainty of a prediction, we used it to compute a kind of *accuracy-rejection* curve: By averaging over five 10-fold cross validations (with NNE), we computed an accuracy degree $\tau_{\boldsymbol{x}}$ (the average Kendall's tau) and a reliability degree $\theta_{\boldsymbol{x}}$ for each instance $\boldsymbol{x}$. The instances are then sorted in decreasing order of reliability. Our curve plots a value p against the mean τ-value of the first p percent of the instances. Given that θ is indeed a good indicator of reliability, this curve should be decreasing, because the higher p, the more instances with a less strong θ-value are taken into consideration. As can be seen in Fig. 1, the curves obtained for our data sets are indeed decreasing and thus provide evidence for our claim that θ may serve as a reasonable indicator of the reliability of a prediction.

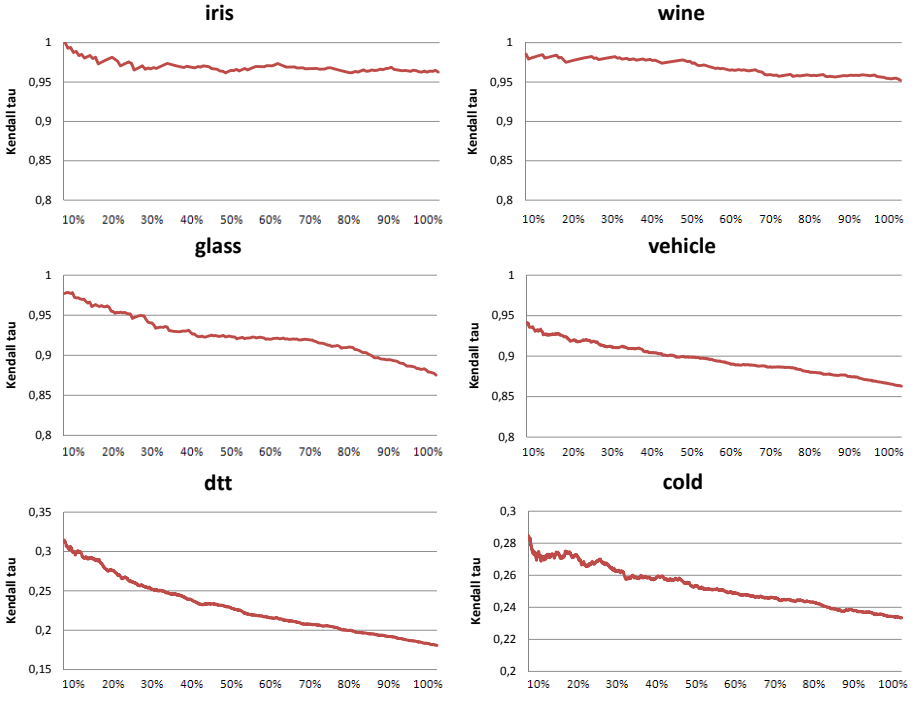


Fig. 1. Accuracy-rejection curves computed on the basis of the parameter θ

6 Conclusions and Future Work

In this paper, we have introduced an instance-based (nearest neighbor) approach to the label ranking problem that has recently attracted attention in the field of machine learning. Our basic inference principle is a consistent extension of the nearest neighbor estimation principle, as used previously for well-known learning problems such as classification and regression: Assuming that the conditional (probability) distribution of the output given the query is locally constant, we derive a maximum likelihood estimation based on the Mallows model, a special type of probability model for rankings. Our first empirical results are quite promising and suggest that this approach is fully competitive, in terms of predictive accuracy, to (model-based) state-of-the-art methods for label ranking. Besides, it has some further advantages, as it does not only produce a single ranking as an estimation but instead delivers a probability distribution over all rankings. This distribution can be used, for example, to quantify the reliability of the predicted ranking.

Currently, we are working on extensions and variants of the label ranking problem, such as calibrated label ranking and multi-label classification [10]. In fact, we believe that the approach proposed in this paper can be extended to a solid framework that not only allows for solving the label ranking problem itself but also variants thereof.

References

1. Hüllermeier, E., Fürnkranz, J., Cheng, W., Brinker, K.: Label Ranking by Learning Pairwise Preferences. Artificial Intelligence 172(16-17), 1897–1916 (2008)
2. Har-Peled, S., Roth, D., Zimak, D.: Constraint Classification for Multiclass Classification and Ranking. In: Becker, S., Thrun, S., Obermayer, K. (eds.) Advances in Neural Information Processing Systems, vol. 15, pp. 785–792 (2003)
3. Dekel, O., Manning, C., Singer, Y.: Log-Linear Models for Label Ranking. In: Touretzky, D.S., Thrun, S., Saul, L.K., Schölkopf, B. (eds.) Advances in Neural Information Processing Systems, vol. 16, pp. 497–504 (2004)
4. Aha, D., Kibler, D., Albert, M.: Instance-Based Learning Algorithms. Machine Learning 6(1), 37–66 (1991)
5. Mallows, C.: Non-Null Ranking Models. Biometrika 44(1), 114–130 (1957)
6. Fligner, M., Verducci, J.: Distance Based Ranking Models. Journal of the Royal Statistical Society 48(3), 359–369 (1986)
7. Alon, N.: Ranking Tournaments. SIAM Journal on Discrete Mathematics 20(1), 134–142 (2006)
8. Diaconis, P., Graham, R.: Spearman's Footrule as a Measure of Disarray. Journal of the Royal Statistical Society 39(2), 262–268 (1977)
9. Saari, D.: Chaotic Elections!: A Mathematician Looks at Voting. American Mathematical Society (2001)
10. Brinker, K., Hüllermeier, E.: Case-based Multilabel Ranking. In: Proc. IJCAI 2007, 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, pp. 701–707 (January 2007)

Learning Performance of Tikhonov Regularization Algorithm with Strongly Mixing Samples

Jie Xu[1] and Bin Zou[1,2]

[1] Faculty of Mathematics and Computer Science,
Hubei University, Wuhan 430062, China
[2] Institute for Information and System Science, Faculty of Science
Xi'an Jiaotong University, Xi'an 710049, China

Abstract. The generalization performance is the main purpose of machine learning theoretical research. The previous bounds describing the generalization ability of Tikhonov regularization algorithm are almost all based on independent and identically distributed (i.i.d.) samples. In this paper we go far beyond this classical framework by establishing the bound on the generalization ability of Tikhonov regularization algorithm with exponentially strongly mixing observations. We then show that Tikhonov regularization algorithm with exponentially strongly mixing observations is consistent.

Keywords: Learning performance; Tikhonov regularization; Strongly mixing observations; Regularization error; Sample error.

1 Introduction

Recently there has been a large increase of the interest for theoretical issues in the machine learning community. It is mainly due to the fact that statistical learning theory has demonstrated its usefulness by providing the ground for developing successful and well-founded learning algorithms such as Support Vector Machines (SVMs)[1]. Besides their good performance in practical applications they also enjoy a good theoretical justification in terms of both universal consistency and learning rates, if the training samples come from an i.i.d. process. This renewed interest for theory naturally boosted the development of performance bounds ([2]). However often this i.i.d. assumption cannot be strictly justified in real-world problems. For example, many machine learning applications such as market prediction, system diagnosis, biological sequence analysis, and speech recognition are inherently temporal in nature, and consequently not i.i.d. processes ([3]). Relaxations of the independence assumption have been considered for quite a while in both machine learning and statistical literature. For example, Modha and Masry [4] established the minimum complexity regression estimation with m-dependent observations and strongly mixing observations respectively. Vidyasagar [5] considered the notions of mixing and proved that most of the desirable properties (e.g. PAC property or UCEMUP property) of i.i.d. sequence

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 717–727, 2009.

are preserved when the underlying sequence is mixing sequence. Steinwart, Hush and Scovel [3] proved that the SVMs algorithm for both classification and regression are consistent if the samples of processes satisfying the law of large numbers. Zou, Li and Xu [6] established the bounds on the rate of relative uniform convergence of learning machines with strongly mixing observations.

To study the generalization performance of Tikhonov regularization algorithm with strongly mixing observations, we first establish the bound on the generalization error of Tikhonov regularization algorithm with exponentially strongly mixing observations, and then we prove that Tikhonov regularization algorithm with exponentially strongly mixing observations is consistent. The rest of this paper is organized as follows: In Section 2, we introduce the definitions of strongly mixing sequence and Tikhonov regularization algorithm. In Section 3 we establish the bound on the generalization ability of Tikhonov regularization algorithm in Section 4. We present some significance conclusions in Section 5.

2 Preliminaries

In this section we introduce the definitions and notations used in the paper.

2.1 Strongly Mixing Sequence

Let $\underline{Z} = \{\mathbf{z}_i\}_{i=-\infty}^{\infty}$ be a stationary real-valued sequence on a probability space $(\Omega, \mathcal{B}, P)$. For $-\infty < i < \infty$, let σ_i^{∞} and $\sigma_{-\infty}^{i}$ denote the σ-algebra events generated by the random variables $\mathbf{z}_j, j \geq i$ and $\mathbf{z}_j, j \leq i$ respectively. With these notations, we have the definition of α-mixing in this literature [5].

Definition 1. ([5]) *The sequence $\underline{Z}$ is called α-mixing, or strongly mixing, if*

$$\sup_{A \in \sigma_{-\infty}^{0}, B \in \sigma_{k}^{\infty}} \{|P(A \cap B) - P(A)P(B)|\} = \alpha(k) \to 0 \quad as \quad k \to \infty.$$

Here $\alpha(k)$ is called the α-mixing coefficient.

Assumption 1. ([4]) *Assume that the α-mixing coefficient of sequence $\underline{Z}$ satisfies*

$$\alpha(k) \leq \overline{\alpha} \exp(-ck^{\beta}), \quad k \geq 1,$$

for $\overline{\alpha} > 0$, $\beta > 0$, and $c > 0$, the constants $\overline{\alpha}$, β and c are assumed to be known.

Remark 1. ([4]) *Assumption 1 is satisfied by a large class of processes, for example, certain linear processes(certain ARMA processes) satisfy the assumption with $\beta = 1$, and certain aperiodic, Harris-recurrent Markov processes (nonlinear ARX processes, and ARH processes) satisfy this assumption. As a trivial example, i.i.d. random variables satisfy this assumption with $\beta = \infty$.*

Denote by $S = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_n\}$ the sample set of size n observations drawn from the exponentially strongly mixing sequence $\underline{Z}$. The goal of machine learning from

random sampling is to find a function f that assigns values to objects such that if new objects are given, the function f will forecast them correctly. Let

$$\mathcal{E}(f) = \mathrm{E}[\ell(f, \mathbf{z})] = \int \ell(f, \mathbf{z}) dP$$

be the expected risk (or expected error) of function f, where the function $\ell(f, \mathbf{z})$, which is integrable for any f and depends on f and $\mathbf{z}$, called loss function. Because our purpose in the present research is to discuss general learning problems, we consider the loss function of general form $\ell(f, \mathbf{z})$ in the following. Define

$$M = \sup_{f \in \mathcal{F}} \max_{\mathbf{z} \in \underline{Z}} |\ell(f, \mathbf{z})|, \ L = \sup_{g_1, g_2 \in \mathcal{F}, g_1 \neq g_2} \max_{\mathbf{z} \in \underline{Z}} \frac{|\ell(g_1, \mathbf{z}) - \ell(g_2, \mathbf{z})|}{|g_1 - g_2|}.$$

We assume that L and M are finite in this paper. Given a hypothesis space $\mathcal{F}$, the learning task is to find the minimizer $\tilde{f}$ of the expected risk $\mathcal{E}(f)$ over the hypothesis space $\mathcal{F}$

$$\tilde{f} = \arg\min_{f \in \mathcal{F}} \mathcal{E}(f) = \arg\min_{f \in \mathcal{F}} \int \ell(f, \mathbf{z}) dP.$$

Since one knows only the set S of random samples instead of the distribution P, the minimizer of the expected risk can not be computed directly. According to the law of large numbers, for a fixed function, the empirical risk (or empirical error) $\mathcal{E}_n(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(f, \mathbf{z}_i)$ converges to the expected risk in probability as the sample size n increases. Then it is a natural idea to use $\hat{f}$, $\hat{f} = \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \ell(f, \mathbf{z}_i)$ as an approximation of the target function $\tilde{f}$. This is called the Empirical Risk Minimization Principle (ERM) (see [1]).

To study the generalization ability of Tikhonov regularization algorithm with strongly mixing sequences, we will apply the following Bernstein's inequality.

Lemma 1. ([4]) *Let $\underline{Z}$ be a stationary α-mixing sequence with the mixing coefficient satisfying Assumption 1. Let an integer $n \geq 1$ be given. For each integer $i \geq 1$, let $U_i = \psi(\mathbf{z}_i)$, where ψ is some real-valued Borel measurable function. Assume that $|U_i| \leq d_1$ a.s. and that $\mathrm{E}[U_1] = 0$. Set $n^{(\alpha)} = \lfloor n \lceil \{8n/c\}^{1/(\beta+1)} \rceil^{-1} \rfloor$. Here n denotes the number of observations drawn from $\underline{Z}$ and $\lfloor u \rfloor (\lceil u \rceil)$ denotes the greatest (least) integer less (greater) than or equal to u. Then for all $\varepsilon > 0$,*

$$P\Big\{\frac{1}{n} \sum_{i=1}^{n} U_i \geq \varepsilon\Big\} \leq (1 + 4e^{-2}\overline{\alpha}) \exp\Big\{\frac{-\varepsilon^2 n^{(\alpha)}}{2(\mathrm{E}|U_1|^2 + \varepsilon d_1/3)}\Big\}.$$

2.2 Tikhonov Regularization Algorithm

Since the problem of solving ERM algorithm is usually ill-posed and overfitting may happen [7]. Thus regularization techniques are introduced [8]. In this paper we shall be interested in Tikhonov regularization algorithm.

Let $\Omega : \mathcal{F} \to \mathbb{R}_+$ be a penalty functional over the hypothesis space $\mathcal{F}$. the ERM with Tikhonov regularization solves the problem

$$f_{\mathbf{z},\lambda} = \arg\min_{f\in\mathcal{F}}\{\mathcal{E}_n(f) + \lambda\Omega(f)\} \tag{1}$$

with $\lambda > 0$ a constant. The functional $\Omega(f)$ is called the regularizer and the constant λ is called the regularization parameter, it often depend on the sample size n: $\lambda = \lambda(n)$.

To bound the generalization ability of Tikhonov regularization algorithm, we should estimate the difference $\mathcal{E}(f_{\mathbf{z},\lambda}) - \mathcal{E}(\tilde{f})$. Since there holds $\mathcal{E}_n(f_{\mathbf{z},\lambda}) + \lambda\Omega(f_{\mathbf{z},\lambda}) \le \mathcal{E}_n(f) + \lambda\Omega(f)$ for any $f \in \mathcal{F}$. Hence we have ([7])

$$\mathcal{E}(f_{\mathbf{z},\lambda}) - \mathcal{E}(\tilde{f}) \le \big\{\mathcal{E}(f_{\mathbf{z},\lambda}) - \mathcal{E}_n(f_{\mathbf{z},\lambda}) + \mathcal{E}_n(f) - \mathcal{E}(f)\big\} + \big\{\mathcal{E}(f) - \mathcal{E}(\tilde{f}) + \lambda\Omega(f)\big\}.$$

The second term in inequality above depends on the choice of $\mathcal{F}$, but is independent of sampling, we will call it the regularization error. For the sake of simplicity, we denote the regularization error by $r(\lambda, f)$ in the sequel, the first term is called the sample error. Since the regularization error is essential and independent of the learning samples, it should be estimated by the knowledge from approximation theory, our aim in this paper is to estimate the sample error.

3 New Refined Concentration Inequalities

Since there is not a satisfactory inequality fitting our purpose, we need to obtain some new refined probability inequalities and give some basic assumptions on the hypothesis space $\mathcal{F}$. We assume that the uniform covering number of the ball $B_\Omega(R) = \big\{f \in \mathcal{F} :\ \Omega(f) \le R^\theta\big\},\quad R > 0,\quad \theta \ge 1$ has complexity exponent $0 < p < 2$, that is there exists some $c_p > 0$ such that

$$\mathcal{N}(B_\Omega(R), \varepsilon) \le \exp\Big\{c_p\Big(\frac{R}{\varepsilon}\Big)^p\Big\}. \tag{2}$$

By Lemma 1, we can obtain the following theorem on the difference between the empirical risks and their expected risks for the strongly mixing sequence $\underline{Z}$.

Theorem 1. *Let $\underline{Z}$ be a stationary α-mixing sequence with the mixing coefficient satisfying Assumption 1. Assume the variance $\mathrm{D}[\ell(f,\mathbf{z})] \le \sigma^2$ for all $\mathbf{z} \in \underline{Z}$ and for all $f \in \mathcal{F}$. Then for any $\varepsilon > 0$,*

$$\mathrm{P}\big\{|\mathcal{E}(f) - \mathcal{E}_n(f)| > \varepsilon\big\} \le 2(1 + 4e^{-2}\overline{\alpha})\exp\Big\{\frac{-\varepsilon^2 n^{(\alpha)}}{2(\sigma^2 + \varepsilon M/3)}\Big\}. \tag{3}$$

Proof. Let $X_i = \mathrm{E}[\ell(f,\mathbf{z}_1)] - \ell(f,\mathbf{z}_i),\quad i = 1, 2, \cdots, n$. Then $\mathcal{E}(f) - \mathcal{E}_n(f) = \frac{1}{n}\sum_{i=1}^n X_i$. We can get easily that for all $i \in \{1, 2, \cdots, n\}$

$$|X_i| = |\ell(f,\mathbf{z}_i) - \mathrm{E}[\ell(f,\mathbf{z}_1)]| \le M$$

and $\mathrm{E}(X_i) = \mathrm{E}\{\mathrm{E}[\ell(f, \mathbf{z}_1)] - \ell(f, \mathbf{z}_i)\} = 0$. Replacing d_1 and $\mathrm{E}|U_1|^2$ by M and σ^2 respectively in Lemma 1, we obtain

$$\mathrm{P}\big\{\mathcal{E}(f) - \mathcal{E}_n(f) \geq \varepsilon\big\} \leq (1 + 4e^{-2}\overline{\alpha}) \exp\Big\{\frac{-\varepsilon^2 n^{(\alpha)}}{2(\sigma^2 + \varepsilon M/3)}\Big\}.$$

By symmetry we also can get

$$\mathrm{P}\big\{\mathcal{E}(f) - \mathcal{E}_n(f) \leq -\varepsilon\big\} \leq (1 + 4e^{-2}\overline{\alpha}) \exp\Big\{\frac{-\varepsilon^2 n^{(\alpha)}}{2(\sigma^2 + \varepsilon M/3)}\Big\}.$$

Combining these two bounds leads to the desired inequality (3). Then we complete the proof of Theorem 1.

From Theorem 1, we can obtain the following corollary.

Corollary 1. *With all notations as in Theorem 1, for any $u > 0$, with probability at least $1 - e^{-u}$, the inequality*

$$\mathcal{E}(f) - \mathcal{E}_n(f) \leq \frac{2BM}{3n^{(\alpha)}} + \sqrt{\frac{2B\sigma^2}{n^{(\alpha)}}}$$

holds, where $B = \ln(1 + 4e^{-2}\overline{\alpha}) + u$. The same bound holds true for $\mathcal{E}_n(f) - \mathcal{E}(f)$.

Proof. For any positive value $u > 0$, Let

$$e^{-u} = (1 + 4e^{-2}\overline{\alpha}) \exp\Big\{\frac{-\varepsilon^2 n^{(\alpha)}}{2(\sigma^2 + \varepsilon M/3)}\Big\}. \tag{4}$$

Solving equation (4) with respect to ε, we obtain

$$\begin{aligned}\varepsilon &= \frac{1}{3n^{(\alpha)}}\Big\{MB + \sqrt{B[M^2(t)B + 18n^{(\alpha)}\sigma^2]}\Big\} \\ &\leq \frac{1}{3n^{(\alpha)}}\Big\{2MB + 3\sqrt{2n^{(\alpha)}B\sigma^2}\Big\} \leq \frac{2BM}{3n^{(\alpha)}} + \sqrt{\frac{2B\sigma^2}{n^{(\alpha)}}}.\end{aligned}$$

By Theorem 1, we can finish the proof of Corollary 1.

Theorem 2. *With all notations as in Theorem 1, then for any $\varepsilon > 0$,*

$$\mathrm{P}\Big\{\sup_{f\in\mathcal{F}} |\mathcal{E}(f) - \mathcal{E}_n(f)| > \varepsilon\Big\} \leq 2C\mathcal{N}\Big(\mathcal{F}, \frac{\varepsilon}{4L}\Big) \exp\Big\{\frac{-\varepsilon^2 n^{(\alpha)}}{8(\sigma^2 + \varepsilon M/6)}\Big\}, \tag{5}$$

where $C = (1 + 4e^{-2}\overline{\alpha})$.

Proof. Define $L_S(f) = \mathcal{E}(f) - \mathcal{E}_n(f)$, $m = \mathcal{N}(\mathcal{F}, \varepsilon)$ and let the disks D_j, $j \in \{1, 2, \cdots, m\}$ be a cover of $\mathcal{F}$ with center at f_j, and radius ε. For any $S \in \underline{Z}^n$ and all $f \in D_j$,

$$\begin{aligned}|L_{\mathbf{z}}(f) - L_{\mathbf{z}}(f_k)| &\leq |\mathrm{E}_z[\ell(f, z)] - \mathrm{E}_z[\ell(f_k, z)]| + \Big|\frac{1}{n}\sum_{i=1}^n \ell(f, z_i) - \frac{1}{n}\sum_{i=1}^n \ell(f_k, z_i)\Big| \\ &\leq 2L \cdot \|f - f_k\|_\infty \leq 2L\varepsilon.\end{aligned}$$

It follows for any $S \in \underline{Z}^n$ and all $f \in D_j$ that $\sup_{f\in D_j} |L_S(f)| \geq 4L\varepsilon \Longrightarrow |L_S(f_j)| \geq 2L\varepsilon$. We conclude that for any $j \in \{1, 2, \cdots, m\}$, $\mathrm{P}\Big\{ \sup_{f\in D_j} |L_S(f)| \geq 4L\varepsilon \Big\} \leq \mathrm{P}\big\{|L_S(f_j)| \geq 2L\varepsilon\big\}$. By Theorem 1, we get

$$\mathrm{P}\Big\{ \sup_{f\in D_j} |\mathcal{E}(f) - \mathcal{E}_n(f)| \geq 4L\varepsilon \Big\} \leq 2(1 + 4e^{-2}\overline{\alpha}) \exp\Big\{ \frac{-(2L\varepsilon)^2 n^{(\alpha)}}{2(\sigma^2 + 2L\varepsilon M/3)} \Big\}.$$

By the fact that the probability of a union of events is bounded by the sum of the probabilities of these events, we can complete the proof of Theorem 2 by replacing ε by $\frac{\varepsilon}{4L}$.

In particular, if $0 < \varepsilon \leq \sigma^2$, then the exponent of inequality (5) in Theorem 2 becomes

$$\frac{-\varepsilon^2 n^{(\alpha)}}{8(\sigma^2 + \varepsilon M/6)} \leq \frac{-3n^{(\alpha)}\varepsilon^2}{4\sigma^2(6 + M)}.$$

By the same argument conducted as that in [2], we have that for some $c_p > 0$, and $0 < p < 2$, the covering number satisfies

$$\mathcal{N}\Big(\mathcal{F}, \frac{\varepsilon}{4L}\Big) \leq \exp\Big\{ c_p \Big(\frac{\varepsilon}{4L}\Big)^{-p} \Big\}.$$

Then by Theorem 2, we have that for any ε, $\sigma^2 \geq \varepsilon > 0$,

$$\mathrm{P}\Big\{ \sup_{f\in\mathcal{F}} |\mathcal{E}(f) - \mathcal{E}_n(f)| > \varepsilon \Big\} \leq 2(1 + 4e^{-2}\overline{\alpha}) \exp\Big\{ c_p \Big(\frac{\varepsilon}{4L}\Big)^{-p} - \frac{3n^{(\alpha)}\varepsilon^2}{4\sigma^2(6 + M)} \Big\}.$$

The critical value of ε occurs when

$$c_p \Big(\frac{\varepsilon}{4L}\Big)^{-p} = \frac{3n^{(\alpha)}\varepsilon^2}{4\sigma^2(6 + M)}$$

i.e., for $\varepsilon \doteq \varepsilon_n = \Big[\frac{c_p(4L)^p 4\sigma^2(6+M)}{3n^{(\alpha)}}\Big]^{\frac{1}{p+2}}$, by the same argument conducted as that in [9] we have that if $2\varepsilon_n \leq \varepsilon \leq \sigma^2$, there exists a constant C_1 such that

$$\mathrm{P}\Big\{ \sup_{f\in\mathcal{F}} |\mathcal{E}(f) - \mathcal{E}_n(f)| > \varepsilon \Big\} \leq 2(1 + 4e^{-2}\overline{\alpha}) C_1 \exp\Big\{ \frac{-3n^{(\alpha)}\varepsilon^2}{4\sigma^2(6 + M)} \Big\}, \qquad (6)$$

where C_1 is a positive constant. If $\varepsilon \leq 2\varepsilon_n$, the left-hand side of inequality (6) trivially holds, and if $n^{(\alpha)} > \frac{4c_p(2L)^p(6+M)}{3\sigma^{2(p+1)}}$ we have $\sigma^2 \geq \varepsilon$. Then we obtain the following proposition.

Proposition 1. *With all notations as in Theorem 1, and if for some $c_p > 0$, and $0 < p < 2$*

$$\mathcal{N}\Big(\mathcal{F}, \frac{\varepsilon}{4L}\Big) \leq \exp\Big\{ c_p \Big(\frac{\varepsilon}{4L}\Big)^{-p} \Big\}, \quad \forall \varepsilon > 0.$$

Then for any $u > 0$, and any $n^{(\alpha)}$, $n^{(\alpha)} > \frac{4c_p(2L)^p(6+M)}{3\sigma^{2(p+1)}}$, there exists a constant c'_p such that with probability at least $1 - e^{-u}$ for any $f \in \mathcal{F}$,

$$\mathcal{E}(f) \le \mathcal{E}_n(f) + \sqrt{\frac{4\sigma^2(6+M)[c'_p + B]}{3n^{(\alpha)}}}.$$

Proof. Let us rewrite inequality (6) in an equivalent form. Let

$$(1 + 4e^{-2}\overline{\alpha})C_1 \exp\left\{\frac{-3n^{(\alpha)}\varepsilon^2}{4\sigma^2(6+M)}\right\} = e^{-u}$$

which we solve with respect to ε. We obtain $\varepsilon \doteq \varepsilon(n) = \sqrt{\frac{4\sigma^2(6+M)[c'_p+B]}{3n^{(\alpha)}}}$, is used to solve inequality $\sup_{f\in\mathcal{F}}[\mathcal{E}(f) - \mathcal{E}_n(f)] \le \varepsilon(n)$, where $c'_p = \ln(C_1)$. As a result we obtain that with probability at least $1 - e^{-u}$ simultaneously for all functions in the set $\mathcal{F}$, the inequality $\mathcal{E}(f) \le \mathcal{E}_n(f) + \sqrt{\frac{4\sigma^2(6+M)[c'_p+B]}{3n^{(\alpha)}}}$ is valid. Then we complete the proof of Proposition 1.

4 Error Analysis

To bound the sample error of Tikhonov regularization with strongly mixing observations, our ideas are as follows: First, we estimate the quantities $\mathcal{E}(f) - \mathcal{E}_n(f)$ and $\mathcal{E}(f_{\mathbf{z},\lambda}) - \mathcal{E}_n(f_{\mathbf{z},\lambda})$ by Corollary 1 and Proposition 1 respectively. Second, we need to estimate the capacity of the function set that contains $f_{\mathbf{z},\lambda}$ and, to get better bounds we hope it is as small as possible. This is equivalent to find a small R such that $f_{\mathbf{z},\lambda} \in B_\Omega(R)$ with high probability. In the end, we adopt the iteration technique was used in [10] to obtain the following theorem.

Theorem 3. *Let $\underline{Z}$ be a stationary α-mixing sequence with the mixing coefficient satisfying Assumption 1, and assume the variance $\mathrm{D}[\ell(f, \mathbf{z})] \le \sigma^2$ for all $\mathbf{z} \in \underline{Z}$, and for all $f \in \mathcal{F}$. Let $D : \mathbb{R}_+ \to \mathbb{R}_+$ be a function satisfying $r(\lambda, f) \le D(\lambda)$ for all functions in $\mathcal{F}$. Then for any $\varepsilon > 0$, and any δ, $0 < \delta < 1$, there exists a constant $c''_p > 0$ independent of the observations' size n such that*

$$\mathcal{E}(f_{\mathbf{z},\lambda}) - \mathcal{E}(\tilde{f}) \le D(\lambda)\left\{4B + c''_p\left[\frac{M(0)}{D(\lambda)}\right]^{\frac{2p\varepsilon}{p+2}}\right\}$$

with probability $1 - \delta$ provided that $n^{(\alpha)} \ge \max\{n_1, n_2\}$, where

$$n_1 = \max\left\{\frac{17M}{3D(\lambda)}, \frac{2M^2(t)}{3D(\lambda)}\right\}, \qquad c''_p = c'_p(A_1 + A_2)^{\frac{2p}{p+2}}$$

$$n_2 = \max\left\{\frac{2M(6+M)}{3D(\lambda)}\left[\frac{(D(\lambda))^{\frac{1}{\theta}}}{R\lambda^{\frac{1}{\theta}}}\right]^{\frac{2p}{p+2}}, \frac{4c_p(4L)^p(6+M)}{3\sigma^{2(p+1)}}\right\},$$

and $A_1 = (c'_p)^{\frac{p+2}{(p+2)\theta-2p}}$, $A_2 = \left((c'_p)^{\frac{(p+2)\theta}{(p+2)\theta-2p}} + \frac{4(p+2)\theta}{(p+2)\theta-2p}\right)^{\frac{1}{\theta}}$.

Proof. Take $u \geq 1$, which will be determined later. We decompose our proof into the following three steps.

Step 1. By Corollary 1, we have that there exists a subset V_1 of $\underline{Z}^n$ with probability at least $1 - e^{-u}$ such that for any $S \in V_1$,

$$\mathcal{E}_n(f) - \mathcal{E}(f) \leq \frac{2BM}{3n^{(\alpha)}} + \sqrt{\frac{2B\sigma^2}{n^{(\alpha)}}}.$$

Since $\sigma^2 \leq \mathrm{E}\{[\ell(f,\mathbf{z})]^2\} \leq M\mathrm{E}[\ell(f,\mathbf{z})] = M\mathcal{E}(f)$, we have

$$\mathcal{E}_n(f) - \mathcal{E}(f) \leq \frac{2BM}{3n^{(\alpha)}} + \sqrt{\frac{2BM\mathcal{E}(f)}{n^{(\alpha)}}} \leq \frac{5BM}{3n^{(\alpha)}} + \frac{1}{2}\mathcal{E}(f). \tag{7}$$

Step 2. Let $\mathcal{H}_R = \{f : \ f \in B_\Omega(R)\}$, by assumption (2) we have

$$\mathcal{N}\Big(\mathcal{H}_R, \frac{\varepsilon}{4L}\Big) \leq \mathcal{N}\Big(B_\Omega(R), \frac{\varepsilon}{2L}\Big) \leq \exp\Big\{c_p(4RL)^p\varepsilon^{-p}\Big\}.$$

We apply Proposition 1 to $\mathcal{H}_R$ and find a subset $V(R)$ of $\underline{Z}^n$ with probability at least $1 - e^{-u}$ such that for any $S \in V(R)$,

$$\mathcal{E}(f) - \mathcal{E}_n(f) \leq \sqrt{\frac{4\sigma^2(6+M)[c'_p+u]}{3n^{(\alpha)}}}. \tag{8}$$

For the sake of simplicity, we denote the term on the right-hand side of inequality (8) by I. By the same argument as inequality (7), we have

$$I = \sqrt{\frac{4\sigma^2(6+M)[c'_p+B]}{3n^{(\alpha)}}} \leq \frac{2M[c'_p+B](6+M)}{3n^{(\alpha)}} + \frac{1}{2}\mathcal{E}(f). \tag{9}$$

Combine inequalities (8) and (9), we obtain

$$\mathcal{E}(f) - \mathcal{E}_n(f) \leq \frac{2M[c'_p+B](6+M)}{3n^{(\alpha)}} + \frac{1}{2}\mathcal{E}(f). \tag{10}$$

Let $W(R)$ be the subset of $\underline{Z}^n$ defined by $W(R) = \{S \in V_1 : f_{\mathbf{z},\lambda} \in B_\Omega(R)\}$. Let $S \in W(R) \cap V(R)$, and then inequality (10) holds for $f_{\mathbf{z},\lambda}$, together with inequality (7), we have

$$\mathcal{E}_n(f) - \mathcal{E}(f) + \mathcal{E}(f_{\mathbf{z},\lambda}) - \mathcal{E}_n(f_{\mathbf{z},\lambda}) \leq \frac{5BM}{3n^{(\alpha)}} + \frac{2M[c'_p+B](6+M)}{3n^{(\alpha)}} + \mathcal{E}(f).$$

Then we get

$$\begin{aligned}\Delta &\doteq \mathcal{E}(f_{\mathbf{z},\lambda}) - \mathcal{E}(\tilde{f}) + \lambda\Omega(f_{\mathbf{z},\lambda}) \\ &\leq \frac{17BM}{3n^{(\alpha)}} + \frac{2BM^2(t)}{3n^{(\alpha)}} + \frac{2M(6+M)}{3n^{(\alpha)}}c'_p + 2D(\lambda).\end{aligned}$$

If $n^{(\alpha)} > n_1$, we have $\frac{17BM}{3n^{(\alpha)}} + \frac{2BM^2(t)}{3n^{(\alpha)}} \le 2D(\lambda)B$. If $n^{(\alpha)} > n_2$, we also have

$$\frac{2M(6+M)}{3n^{(\alpha)}}c'_p \le D(\lambda)c'_p\Big[\frac{R\lambda^{\frac{1}{\theta}}}{(D(\lambda))^{\frac{1}{\theta}}}\Big]^{\frac{2p}{p+2}}.$$

Therefore we have

$$\Delta \le 2D(\lambda)(B+1) + D(\lambda)c'_p\Big[\frac{R\lambda^{\frac{1}{\theta}}}{(D(\lambda))^{\frac{1}{\theta}}}\Big]^{\frac{2p}{p+2}} = D(\lambda)\Big[4B + c'_p\Big[\frac{R\lambda^{\frac{1}{\theta}}}{(D(\lambda))^{\frac{1}{\theta}}}\Big]^{\frac{2p}{p+2}}\Big].$$

For $S \in W(R) \cap V(R)$, this implies $f_{\mathbf{z},\lambda} \in B_\Omega(g(R))$, that is $\Omega(f_{\mathbf{z},\lambda}) \le (g(R))^\theta$, where $g: \mathbb{R}_+ \to \mathbb{R}_+$ is a function defined by

$$g(R) = \Big[\frac{D(\lambda)}{\lambda}\Big]^{\frac{1}{\theta}}\Big\{4B + c'_p\Big[\frac{R\lambda^{\frac{1}{\theta}}}{D(\lambda)^{\frac{1}{\theta}}}\Big]^{\frac{2p}{p+2}}\Big\}^{\frac{1}{\theta}}.$$

It follows that

$$W(R) \cap V(R) \subseteq W(g(R)). \tag{11}$$

Step 3. We find a small ball $B_\Omega(R)$ that contains $f_{\mathbf{z},\lambda}$ by iteration. Since $\mathcal{E}_n(f_{\mathbf{z},\lambda}) + \lambda\Omega(f_{\mathbf{z},\lambda}) \le \mathcal{E}_n(0) + \lambda\Omega(0) \le M(0)$, we have $\Omega(f_{\mathbf{z},\lambda}) \le \frac{M(0)}{\lambda}$. Take $R_0 = \Big(\frac{M(0)}{\lambda}\Big)^{\frac{1}{\theta}}$, then $f_{\mathbf{z},\lambda} \in B_\Omega(R_0)$ and $W(R_0) = V_1$.

Denote $R_j = g(R_{j-1})$ for $j \in \mathbb{N}$. According to (11), we have

$$W(R_0) \cap \Big(\cap_{i=0}^{j-1} V(R_i)\Big) \subseteq W(R_j).$$

Define $r_j = \frac{R_j\lambda^{\frac{1}{\theta}}}{(D(\lambda))^{\frac{1}{\theta}}}$, $j \in \{0\} \cup \mathbb{N}$. Then $r_{j+1} = \Big(c'_p(r_j)^{\frac{2p}{p+2}} + 4B\Big)^{\frac{1}{\theta}}$. By Lemma 5.17 in [7], we have $r_j \le A_1 r_0^{(\frac{2p}{(p+2)\theta})^j} + A_2$, where

$$A_1 = (c'_p)^{\frac{p+2}{(p+2)\theta-2p}},\ A_2 = \Big((c'_p)^{\frac{(p+2)\theta}{(p+2)\theta-2p}} + \frac{4B(p+2)\theta}{(p+2)\theta-2p}\Big)^{\frac{1}{\theta}}.$$

This implies

$$R_{j_0} \le \Big[\frac{D(\lambda)}{\lambda}\Big]^{\frac{1}{\theta}}\Big\{A_1\Big(\frac{M(0)}{D(\lambda)}\Big)^{\frac{1}{\theta}(\frac{2p}{(p+2)\theta})^j} + A_2\Big\}.$$

For any $\varepsilon > 0$, choose $j_0 \in \mathbb{N}$ such that $j = \Big\lfloor \frac{\ln\frac{1}{\theta\varepsilon}}{\ln\frac{(p+2)\theta}{2p}} \Big\rfloor + 1$, it follows

$$R_{j_0} \le \Big[\frac{D(\lambda)}{\lambda}\Big]^{\frac{1}{\theta}}\Big\{A_1\Big[\frac{M(0)}{D(\lambda)}\Big]^{\varepsilon} + A_2\Big\}.$$

Set $R_\varepsilon = (A_1 + A_2)\Big[\frac{D(\lambda)}{\lambda}\Big]^{\frac{1}{\theta}}\Big[\frac{M(0)}{D(\lambda)}\Big]^{\varepsilon}$, then $W(R_{j_0}) \subseteq W(R_\varepsilon)$, and hence $W(R_\varepsilon)$ has probability at least $1 - (j_0+2)e^{-u}$. Taking $u = ln\frac{j_0+3}{\delta}$, the probability of the set $W(R_\varepsilon) \cap V(R_\varepsilon)$ is at least $1 - (j_0+3)e^{-u} = 1 - \delta$.

Then for each $S \in W(R_\varepsilon) \cap V(R_\varepsilon)$, with probability at least $1-\delta$, there holds

$$\mathcal{E}(f_{\mathbf{z},\lambda}) - \mathcal{E}(\tilde{f}) \leq D(\lambda)\Big\{4B + c_p''\Big[\frac{M(0)}{D(\lambda)}\Big]^{\frac{2p\varepsilon}{p+2}}\Big\},$$

where $c_p'' = c_p'(A_1 + A_2)^{\frac{2p}{p+2}}$. Then we complete the proof of Theorem 3.

Remark 4. *In Theorem 3, we use a technical condition, that is the upper bound $D(\lambda)$ of $r(\lambda, f)$. This is because it is usually hard to compute $r(\lambda, f)$ explicitly, the upper bound may be obtained by knowledge of the function $\tilde{f}$, and it is natural for $D(\lambda) \to 0$ since $\lambda \to 0$ in most situation.*

Remark 5. *Since $\lambda \to 0$ as $n \to 0$, and $D(\lambda) \to 0$ as $\lambda \to 0$, we have*

$$\mathcal{E}(f_{\mathbf{z},\lambda}) - \mathcal{E}(\tilde{f}) \to 0 \quad as \quad n \to 0.$$

This implies that Tikhonov regularization algorithm with exponentially strongly mixing sequence is consistent. In addition, Steinwart, Hush and Scovel [3] proved that the SVMs algorithm with α-mixing processes for both classification and regression are also consistent if the samples of processes satisfying the law of large numbers. In this paper we get the same result with [3] by making use of different method with [3], that is we obtain firstly the bound on the generalization performance of Tikhonov regularization algorithm, and then prove that Tikhonov regularization algorithm with strongly mixing sequence is consistent.

5 Conclusions

The fundamental problems in Statistical Learning Theory (SLT) are evaluation of generalization performance and consistency of learning algorithms. In this paper we have studied the extension problem of SLT from the classical i.i.d. sequences to the exponentially strongly mixing sequences. We have extended the classical generalization bound estimations of Tikhonov regularization algorithm through establishing a new bound on the samples error based on exponentially strongly mixing sequences for general loss function. From the established generalization bound estimations, we can conclude that Tikhonov regularization algorithm with strongly mixing samples is consistent. The obtained results perfectly extended the well-known statistical learning theory for Tikhonov regularization algorithm justified previously for i.i.d. observations. To our knowledge, the result here is the first explicit bound on the learning performance on this topic.

Along the line of the present work, several open problems deserves further research, e.g., how to control the generalization ability of Tikhonov regularization algorithm with exponentially strongly mixing samples? What is the essential difference of generalization ability of Tikhonov regularization algorithm with i.i.d. samples and dependent samples? how to develop the bounds on generalization ability of Tikhonov regularization algorithm for other dependent samples ? All these problems are under our current investigation.

References

1. Vapnik, V.: Statistical Learning Theory. John Wiley, New York (1998)
2. Cucker, F., Smale, S.: On the Mathematical Foundations of Learning. Bulletin of the American Mathematical Society 39, 1–49 (2002)
3. Steinwart, I., Hush, D., Scovel, C.: Learning from Dependent Observations. Los Alamos National Laboratory Technical Report LA-UR-06-3507 (submitted for publication) (2006)
4. Modha, S., Masry, E.: Minimum Complexity Regression Estimation with Weakly Dependent Observations. IEEE Trans. Inform. Theory 42, 2133–2145 (1996)
5. Vidyasagar, M.: Learning and Generalization with Applications to Neural Networks, 2nd edn. Springer, London (2002)
6. Zou, B., Li, L.Q., Xu, Z.B.: The Generalization Performance of ERM Algorithm with Strongly Mixing Observations. Machine Learning (accepted, 2008)
7. Wu, Q.: Classification and Regularization in Learning Theory. Thesis of doctor of philiosophy. Hong Kong: City University of Hong Kong (2005)
8. Tikhonov, A.N.: On Solving Ill-posed Problem and Method of Regularization. Dokl. Akad. Nauk, 501–504 (1963)
9. DeVore, R., Kerkyacharian, G., Picard, D., Temlyakov, V.: Approximation Methods for Supervised Learning. Foundations of Computational Mathematics 6(1), 3–58 (2006)
10. Wu, Q., Zhou, D.X.: Support Vector Machines: Linear Programming Virsus Quadratic Programming. Neural Computation 17, 1160–1187 (2005)

Efficient Learning from Few Labeled Examples

Jiao Wang, Siwei Luo, and Jingjing Zhong

School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China

Abstract. Active learning and semi-supervised learning are two approaches to alleviate the burden of labeling large amounts of data. In active learning, user is asked to label the most informative examples in the domain. In semi-supervised learning, labeled data is used together with unlabeled data to boost the performance of learning algorithms. We focus here to combine them together. We first introduce a new active learning strategy, then we propose an algorithm to take the advantage of both active learning and semi-supervised learning. We discuss several advantages of our method. Experimental results show that it is efficient and robust to noise.

Keywords: Active learning, Semi-supervised learning, Learning from examples, Selective sampling, Machine learning.

1 Introduction

Labeling the training data for a machine learning algorithm is time consuming, tedious, and sometimes expensive. Active learning and semi-supervised learning are two approaches to alleviate the burden of labeling large amounts of data.

Active learning algorithms cope with this problem by detecting and asking the user to label the most informative examples in the domain. For examples, Muslea et al. [1] and Freund et al. [2] generate a committee of hypotheses, and query the unlabeled data on which the disagreement within the committee is the greatest. Cohn et al. [3] queries the unlabeled data that minimizes the error rate of the classifier. Lewis et al. [4] queries the unlabeled data on which this classifier makes the least confident prediction. By these ways, active learning methods reduce the user's involvement in the data labeling process.

Semi-supervised learning algorithms use few labeled data together with a lot of unlabeled data to boost the performance of learning algorithms, thus reducing the need of labeled data. For examples, Blum et al. [5] generate two different hypotheses for co-training, and let the two hypotheses label the unlabeled data for each other. Nigam et al. [6] use the EM algorithm to estimate the labels of unlabeled data. Blum et al. [7] and Belkin et al. [8] construct a graph to encode the similarity between data, and use the spectral graph partition methods to get the final classification.

A natural idea is to combine active learning and semi-supervised learning together, in order to get the advantages of both. That is, we could allow the learning algorithm to pick some unlabeled data to be labeled by a domain expert, which will then be used as the labeled data set for learning a better hypothesis. But few works have been done

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 728–734, 2009.

except for [9-11]. Zhu et al [9] combine active learning and semi-supervised learning under a Gaussian random field model, they choose the samples actively by greedily selecting queries from the unlabeled data to minimize the estimated expected classification error, but estimating the expected classification error is computation expensive. Muslea et al [10] give a multi-view algorithm which interleaves active and semi-supervised learning, but their method can only be used when the problem has multi views. Wang et al [11] give a theoretical analysis on the sample complexityof combining multi-view active learning an semi-supervise learning, which showing that the combination can reduce the error rate quickly.

We introduce a new method, Efficient-Learning (E-Learning for short), which combines active learning and semi-supervised learning together. In E-Learning, a strategy of active learning is proposed firstly. Based on the observation that data near to the dividing hyperplane providing more information, the data nearest to the dividing hyperplane is chosen and its label is asked from user. Then E-Learning runs a semi-supervised algorithm with all the data (including labeled and unlabeled data) to get a new dividing hyperplane. This process is repeated until the results do not change. In others words, E-Learning uses the data proximate to the dividing hyperplane to optimize the dividing hyperplane with the help of unlabeled data.

Compared with previous work, E-Learning has several advantages:

1. Existing methods are computation expensive since they either need to estimate the expected classification error [9], or need to construct several learners [10]. In contrast, E-Learning is simple to compute by defining the data nearest to the dividing hyperplane as the most informative data.
2. Existing methods are typically designed to exploit a particular base learner, while E-Learning is a general purpose method that can be used together with any semi-supervised learning algorithms.
3. E-Learning is geometry intuitive, and it makes no assumption about the distribution of data. In experiment, it converges quickly to the target concept after labeling few data proximate to the dividing hyperplane.

The remainder of the paper is organized as follows. First, we introduce a new strategy of active learning, and then we propose the E-Learning algorithm. Finally, we present our empirical evaluation on some data sets.

2 A New Strategy of Active Learning

Consider the situation in which the learner has access to a large number of unlabeled data, which is often referred as selective sampling. The question is which samples, if labeled, would give more information.

A natural observation is that data near to the dividing hyperplane providing more information, since the classifier makes less confident prediction of its class label. If we ask its label from user, we can get more information to optimize the hyperplane, as illustrated in Fig.1:

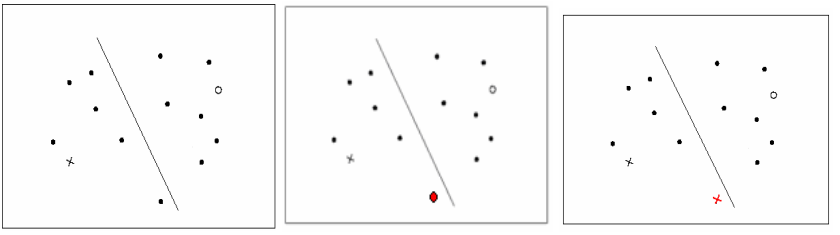


Fig. 1. Data near to the dividing hyperplane provide more information. (a) In two class classification problem, "×"and "○"represent the two class, the solid lines represent current dividing hyperplane. (b) The red point is the one nearest to the current hyperplane. (c) When the data is labeled as "×", the hyperplane is adjusted to the dotted. line.

Let X denote the set of all possible data, $x_i \in X$. Let $h(\theta)$ denote the dividing hyperplane, where $\theta \in \Theta$ is the parameter. Let $d(x_i, h(\theta))$ represent the distance of x_i to $h(\theta)$. We define a quantity $I(x_i)$ to represent the information contained by x_i:

$$I(x_i) = \frac{1}{d(x_i, h(\theta))^2} \tag{1}$$

which implies that the nearer a data to the hyperplane, the more informative the data is.

The strategy is to find

$$k = \arg\max_i \{I(x_i)\} \tag{2}$$

which is equal to find

$$k = \arg\min_i \{d(x_i, h(\theta))\} \tag{3}$$

To get $d(x_i, h(\theta))$, we only need to compute the dot product of $x_i \cdot h(\theta))$. So this strategy is computationally efficient.

Unlike previous uncertainty reduction active learning[4], which require the base learner to be able to estimate the confidence of its prediction, our strategy can be applied to any base learner.

3 The E-Learning Algorithm

The proposed E-Learning algorithm combines active learning and semi-supervised learning together. It uses the strategy introduced above to detect the most informative data, asks its label from user, and then it uses a semi-supervised learning method with current labeled and unlabeled data to adjust the dividing hyperplane. E-Learning repeats this process until the hyperplane does not change. This algorithm is illustrated below.

The E-Learning Algorithm

E-Learning (L , U , $SSLearn$)
Input: L : Original labeled data set
U : Unlabeled data set
$SSLearn$: semi-supervised learning algorithm
$h(\theta) \leftarrow SSLearn(L,U)$
Repeat until $h(\theta)$ do not change
$k = \arg\min_i \{d(x_i, h(\theta))\}$
Ask the user for the label of x_k
$L \leftarrow L \cup \{x_k, y_k\}$
$h(\theta) \leftarrow SSLearn(L,U)$
End of repeat
Output: $h(\theta)$

In this paper, in order to explain our algorithm in detail, we use the proposed active learning strategy to work with the semi-supervised learning work of Belkin [8] to show the entire process. We can see how this active learning strategy boosting semi-supervised learning on manifold. We do not claim that this method can only use on manifold, and indeed we aim to illuminate that applying our method, to any semi-supervised method, would always yield satisfying results.

Given a sample set $x_1, \cdots, x_n \in X$, we construct its neighborhood graph $G = (V, E)$, whose vertices are sample points $V = \{x_1, \cdots, x_n\}$, and whose edge weights $\{w_{ij}\}_{i,j=1}^{n}$ represent appropriate pairwise similarity relationships between samples. For example, w_{ij} can be the radial basis function:

$$w_{ij} = \exp(-\frac{1}{\sigma^2}\sum_{d=1}^{m}(x_{id} - x_{jd})^2) \tag{4}$$

where σ is a scale parameter, m is the dimensionality of x_i . The radial basis function of w_{ij} ensure that nearby points are assigned large edge weights.

Belkin [8] chooses the labeled data manually, and in some other works the labeled data is chosen randomly. We exploit a more automatically and actively way. Based on the assumption that data near to the dividing hyperplane providing more information, we choose the data nearest to the dividing hyperplane as equation (3), ask its label from user, and add it to the labeled data set.

Every time we get a new labeled data, we run semi-supervised learning algorithm [8] to adjust the dividing hyperplane. More specifically, we minimize the following cost function

$$\min_{f \in H} H[f] = \frac{1}{l}\sum_{i=1}^{l}(f(x_i) - y_i)^2 + \gamma_A \|f\|_K^2 + \gamma_I \sum_{i,j=1}^{n}(f(x_i) - f(x_j))^2 W_{ij} \tag{5}$$

where l is the number of labeled samples, γ_A, γ_I are regularization parameters, $\|f\|_K^2$ is some form of constraint to ensure the smoothness of the learned manifold.

4 Experimental Results

First, we test the E-Learning algorithm using the noise case of the two moons dataset [12]. Figure 2 shows the process. There are six pictures in figure 2, which represent six iterations. In each picture, the data nearest to the dividing hyperplane is chosen, and it is represented in green color. Then, the label of this data is given, which is represented in red or blue color (for two different classes). After that, the semi-supervised manifold learning algorithm is run with current labeled data and unlabeled data to get a new dividing hyperplane, which is represented in yellow curve. At the beginning, we randomly choose one data for each class to get an initialization.

As we can see, the proposed active learning strategy can choose the most informative data to give labels, and it is robust to noise. The E-Learning algorithm adjusts the dividing hyperplane to be better and better along the iteration. After labeling six data, it converges to the target concept as the hyperplane does not change.

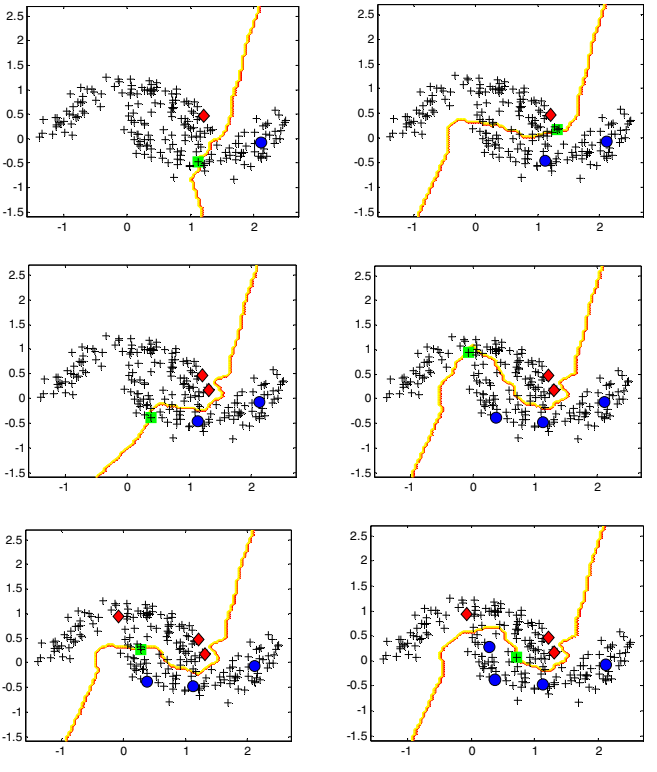


Fig. 2. The E-Learning process of the noise case of two moon dataset

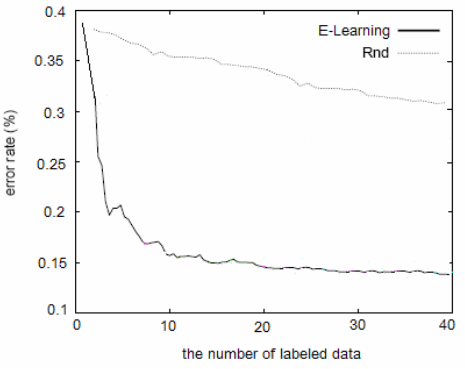


Fig. 3. The E-Learning algorithm reduce the error rate quickly in the "20 Newsgroups"data set

Then we apply the E-Learning algorithm to some real-world classification problem: the "20 Newsgroups"data set. In our experiments, each document was presented by a vector, and no pre-process was performed on the vectors. To verify the proposed algorithm, we randomly partition the data into two sets, traing-ing set and test set with equal size. We perform ten runs to average the results. Each experiment began with two randomly chosen initial examples for each class, and successive examples were added in batches of size 2. We compare the results of E-Learning with randomly chosen method. As we can see, E-Learning decrease the error rate quickly, and after labeling 10 data, the error rate converge (Fig. 3).

5 Conclusion

In order to label less data for a machine learning task, we combine active learning and semi-supervised learning together, and propose the E-Learning algorithm. Based on the assumption that data near to the dividing hyperplane providing more information, we choose the data nearest to the dividing hyperplane for active learning. This strategy can apply to any semi-supervised learning methods, and in this paper, we combine it with manifold regularization to show how it works.

Experiments results illustrate that the E-Learning algorithm converges quickly to target concept, and is robust to noise. It gives an indication of possible research direction in machine learning.

Acknowledgements. This work is supported by National High Technology Research and Development Program of China (2007AA01Z168), National Nature Science Foundation of China (60773016, 60805041, 60872082) and Doctoral Foundations of Ministry of Education of China (20050004001, 200800041049).

References

1. Muslea, I., Minton, S., Knoblock, C.A.: Active Learning with Multiple Views. Journal of Artificial Intelligence Research 27, 203–233 (2006)
2. Freund, Y., et al.: Selective Sampling Using the Query by Committee Algorithm. Machine Learning 28, 133–168 (1997)
3. Cohn, D.A., Ghahramani, Z., Jordan, M.I.: Active Learning with Statistical Models. In: Advances in Neural Information Processing Systems 7. MIT Press, Cambridge (1995)
4. Lewis, D.D., Catlett, J.: Heterogeneous Uncertainty Sampling for Supervised Learning. In: Proceedings of the 11th International Conference on Machine Learning (1994)
5. Blum, A., Mitchell, T.: Combining Labeled and Unlabeled Data with Co-Training. In: Proceedings of the 11th Annual Conference on Computational Learning Theory, Madison, WI (1998)
6. Nigam, K., et al.: Text Classification from Labeled and Unlabeled Documents Using EM. Machine Learning 39, 103–134 (1999)
7. Blum, A., Chawla, S.: Learning from Labeled and Unlabeled Data Using Graph Mincuts. In: Proceedings of the 18th International Conference on Machine Learning. Morgan Kaufmann, San Francisco (2001)
8. Belkin, M., Niyogi, P.: Semi-Supervised Learning on Riemannian Manifolds. Machine Learning 56, 209–239 (2004)
9. Zhu, X., Lafferty, J., Ghahramani, Z.: Combining Active Learning and Semi-supervised Learning Using Gaussian Fields and Harmonic Functions. In: ICML 2003 workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining (2003)
10. Muslea, I., Minton, S., Knoblock, C.A.: Active +Semi-supervised Learning Robust Multi-view Learning. In: Proceedings of the 19th International Conference on Machine Learning (2002)
11. Wang, W., Zhou, Z.: On Multi-view Active Learning and the Combination with Semi-Supervised Learning. In: Proceedings of the 25th nternational Conference on Machine Learning, Helsinki, Finland (2008)
12. Belkin, M., Niyogi, P., Sindhwani, V.: Manifold Regularization: A Geometric Framework for Learning from Examples. Department of Computer Science, University of Chicago, Technical Report (2004)

An Improved Quantum Evolutionary Algorithm with 2-Crossovers

Zhihui Xing[1], Haibin Duan[1,2], and Chunfang Xu[1]

[1] School of Automation Science and Electrical Engineering, Beihang University, Beijing, 100191, China
[2] Provincial Key Laboratory for Information Processing Technology, Suzhou University, Suzhou 215006, China
christianahui@126.com, hbduan@buaa.edu.cn, xiaofang88s@hotmail.com

Abstract. Quantum evolutionary algorithm (QEA) is proposed on the basis of the concept and principles of quantum computing, which is a classical meta-heuristic algorithm for the approximate solution of combinatorial optimization problems that has been inspired by the principles of evaluation of living organisms in nature. QEA has strong robustness and easy to combine with other methods in optimization, but it has the shortcomings of stagnation that limits the wide application to the various areas. In this paper, a hybrid QEA with 2-crossovers was proposed to overcome the above-mentioned limitations. Considering the importance of randomization, 2-crossovers were applied to improve the convergence quality in the basic QEA model. In this way, the new-born individual after each updating can to help the population jump out of premature convergence. The proposed algorithm is tested with the Benchmark optimization problem, and the experimental results demonstrate that the proposed QEA is a feasible and effective in solving complex optimization problems.

Keywords: Quantum evolutionary algorithm (QEA); Genetic algorithm (GA); Qubit chromosome; Crossover; premature.

1 Introduction

Since the concept of quantum was put forward, there was a revolution coming in the field of computing, and it was coming from quantum-the smallest of all places: the subatomic particles that form the basis of all matters.

Quantum computing has promised prodigious powers in the past years. Its basic currency, the qubit, exists in an ON or OFF verge, which you will never know until it's read out. Therefore, if you could operate on K qubits, a potentially vast space of 2^K values opens up for computation which means that we can solve many computing problems at the same time, which saves you a lot of time. The fundamental operation on qubits is a rotation. We have logic gates to combine the rotations. The algorithm is based on these logic gates. In principle, these algorithms can perform calculations far beyond classical computation's conceivable reach.

Genetic algorithm (GA) was firstly put forward by J. Holland in 1970s to study the self adaptation behavior of natural system [1]. It's a classical meta-heuristic algorithm

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 735–744, 2009.

for the approximate solution of combinatorial optimization problems that has been inspired by the principles of evaluation of living organisms in nature. The application of GA needs no initiating knowledge of the system, and it isn't limited by the form and property of the problem. Guided by fitness function and principle of probability, it can search in global according to self adaptation by using selection, crossover and mutation. Therefore, it's a comprehensive optimization method with extensive application in terms of processing complex non-linear problems.

GA has strong robustness and is easy to combine with other methods in optimization, but it has limited population size and the problems of premature convergence and stagnation that limit the wide application to the various area often exist. Qubit chromosomes enjoy a rapidly growing population and strong randomization.

To overcome the above-mentioned shortcomings of GA, quantum evolutionary algorithm (QEA) is proposed on the basis of the concept and principles of quantum computing. In QEA, qubit chromosomes, which can represent a linear superposition of solutions, are adopted to maintain solution diversity and overcome premature convergence. At the same time, quantum rotation gate, which make full use of the information of the current best individual, is used to update individual and avoid stagnation [2].

The common QEA uses qubit gate rotation in mutation and whole interference in crossover [3]. By using rotation operation, we can make full use of the information of the currently best individual to perform the next searching process, and the whole interference can avoid prematurity. In this way, the global search capacity can be greatly improved, while the convergence speed is slowed down. In order to further improve the whole performance of QEA, a new hybrid strategy was proposed in this paper.

The remainder of this paper is organized as follows. The next section introduces the main process of common QEA. Section 3 proposes a hybrid QEA model with 2-crossovers. Then, in Section 4, series of comparison experiments are conducted. Our concluding remarks and future work are contained in the final section.

2 Basic QEA

2.1 Qubit Chromosome

In QEA, a qubit chromosome as a string of n qubits can be defined as follows [4]:

$$q = \begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_m \\ \beta_1 & \beta_2 & \dots & \beta_m \end{bmatrix} \tag{1}$$

where $|\alpha_i|^2+|\beta_i|^2=1$, $i=1,\dots, m$, and m is the number of qubits and also the string length of the qubit individual. $|\alpha_i|^2$ gives the probability that the qubit will be found in the state of '0' and $|\beta_i|^2$ gives the probability that the qubit will be found in the '1' state. A qubit chromosome is able to represent a linear superposition of all possible solutions. It has a better characteristic of diversity than classical chromosome[5]. The process to get classical chromosome is: bring a random number between 0 and 1, if it's bigger than $|\alpha_i|^2$, this bit in classical chromosome is '1', else '0' is chosen.

2.2 Quantum Mutation

The standard mutation operation is totally random without any directions, so the speed of convergence is slowed down. But in QEA, the qubit representation can be used as a mutation operator. Directed by the current best individual, quantum mutation is completed through the quantum rotation gate U(θ),then the $[\alpha_i \ \beta_i]^T$ is updated as:

$$\begin{bmatrix} \alpha'_i \\ \beta'_i \end{bmatrix} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix} \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} \tag{2}$$

Look up the Table 1 to find out the right θ_i, which is determined by both quantum and classical chromosome.

Table 1. Rotation angle

x_i	$best_i$	$f(x)>f(\text{best})$	θ_i			
			$\alpha\beta_i>0$	$\alpha\beta_i>0$	$\alpha\beta_i>0$	$\alpha\beta_i>0$
0	0	*False*	0	0	0	0
0	0	*True*	0	0	0	0
0	1	*False*	0	0	0	0
0	1	*True*	-0.05π	0.05π	±0.05π	0
1	0	*False*	-0.05π	0.05π	±0.05π	0
1	0	*True*	0.05π	-0.05π	0	±0.05π
1	1	*False*	0.05π	-0.05π	0	±0.05π
1	1	*True*	0.05π	-0.05π	0	±0.05π

x_i is the *i*-th bit of the current classical chromosome, $best_i$ is the ith bit of the current best classical chromosome, $f(x)$ is the adaptation function [6].

The Figure 1 below describes the polar plot of the rotation operation on qubit. It tells us the reason why the rotation gate can increase the speed of convergence obviously [7].

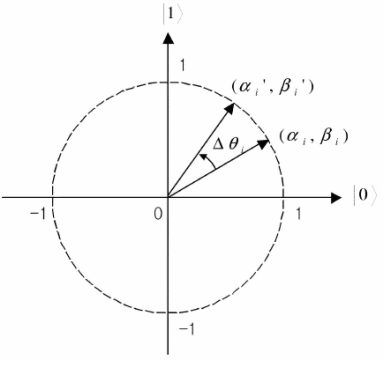


Fig. 1. Polar plot of the rotation gate for qubit chromosome

2.3 Quantum Whole Interference Crossover

This kind of crossover operation is constructed by the interference characteristic of qubit. All the quantum chromosomes are involved in. For example, when the population number is 5 and the length of chromosome is 6, the table 2 below introduces a kind of operation:

Table 2. The whole interference crossover operation

1	A(1)	E(2)	D(3)	C(4)	B(5)	A(6)	E(7)
2	B(1)	A(2)	E(3)	D(4)	C(5)	B(6)	A(7)
3	C(1)	B(2)	A(3)	E(4)	D(5)	C(6)	B(7)
4	D(1)	C(2)	B(3)	A(4)	E(5)	D(6)	C(7)
5	E(1)	D(2)	C(3)	B(4)	A(5)	E(6)	D(7)

The whole interference crossover operation can make full use of the information in the chromosome, improve the unilateralism of classical crossover and avoid premature convergence and stagnation problems.

3 The Proposed Hybrid QEA with 2-Crossovers

The QEA provide new ideas to improve the traditional GA. Firstly, the information in a quantum chromosome is more than that in a classical chromosome, the number of population is decreased and the diversity is improved. Secondly, the mutation operation is no longer totally random but directed by some rules to make the next generation better and increase the speed of convergence. Thirdly, whole interference crossover operation can avoid premature convergence and stagnation problems.

When the whole interference crossover operation try to produce new solution to avoid premature convergence, it's not good for the maintaining of current good solutions. Then, the convergence speed of QEA is decreased. In order to improve the convergence speed and avoid premature convergence, we proposed a hybrid QEA. The result is extraordinary both in theory and experiments.

In our proposed hybrid QEA, we use 2-crossover operations, and some improvements are also conducted in the quantum mutation.

The first crossover operation is the classical single point crossover. Roulette selection operation is used to choose two quantum chromosomes from the parent generations, then the child generation is produced by crossover. After this process, two better individuals can be chosen into the next generation by evaluating their fitness. This operation is mainly to improve the convergence speed and preserve the instructive information. Usually, we choose 0.6 to 0.9 as the crossover probabilities in this process.

Then, we evaluate the fitness of the whole population. We choose the best one as the mutation director, because the evaluation is just before mutation operation, the director individual is considered better than anyone else. Therefore, when we use the rules shown in Table 1, $f(x)$ is always smaller than $f(best)$. Here, we only need half of

the information presented in Table 1. The table is simplified, thus the evolution process is faster and easier. We choose 0.01 to 0.2 as the mutation probabilities. Although it's not traditional mutation and the individual can also converge by this operation, the mutation probability can't be very high. Because in every generation, we only choose one best individual, and we use quantum chromosome, the selection is full of randomicity, so we're not sure the "best solution" we choose is really the best result. Furthermore, too much probability isn't good for evolution.

Another crossover operation, the whole interference crossover operation, is adopted to prevent premature convergence. It can bring new individual to help the population jump out of premature convergence. We also choose 0.01 to 0.2 as the crossover probability. For it's just used to avoid premature convergence, we also can't use it in high frequency.

The process of our proposed hybrid QEA with 2-crossovers for solving complex optimization problems can be described as follows:

Step 1: Initialization of parameters: bring a random angle ω between 0 and 2*π, α=cos(ω),β=sin(ω), then a qubit is produced. Set other parameters: classical crossover probability-*Pcc*, mutation probability-*Pm*, whole interference crossover probability-*Pic*, the max circulation generation-*germax*, the number of population-*n*, and the length of chromosome-*L*.

Step 2: Produce a classical population by using this quantum chromosomes. It's the original classical population. Evaluate the fitness of each chromosome.

Step 3: Use roulette operation to select parents quantum chromosome, operate crossover in the classical crossover probability. Update the quantum population. Then produce a classical population, and find out the best solution which will be used in the mutation operation.

Step 4: Operate mutation in the mutation probability and update the quantum population. Then produce a new classical population, evaluate the fitness of each chromosome, compare with the old ones and update the classical population.

Step 5: Operate the whole interference crossover in it's probability and update the quantum population.

Step 6: If the stopping criterion is satisfied, the proposed QEA algorithm stops, and output the best solution, else return to Step 3.

The above-mentioned procedures of the proposed hybrid QEA process can also be described in the Figure 2.

Table 3. The concised rotation angle

x_i	best$_i$	$f(x)>f(best)$	θ_i			
			$\alpha\beta_i>0$	$\alpha\beta_i>0$	$\alpha\beta_i>0$	$\alpha\beta_i>0$
0	0	False	0	0	0	0
0	1	False	0	0	0	0
1	0	True	0.05π	-0.05π	0	±0.05π
1	1	False	0.05π	-0.05π	0	±0.05π

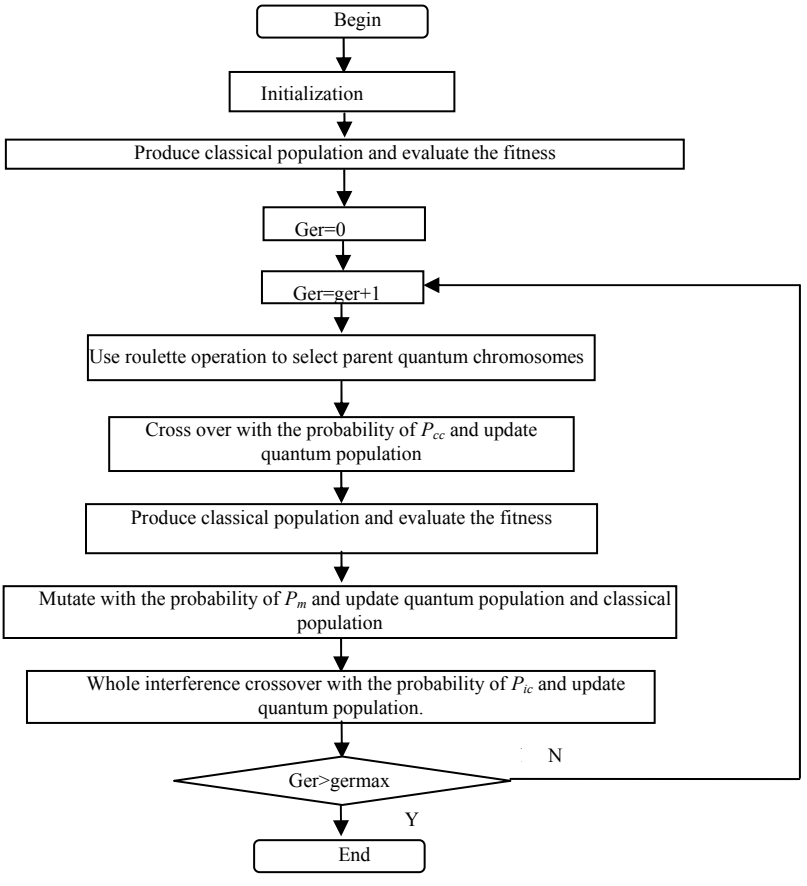


Fig. 2. The proposed hybrid QEA with 2-crossovers

4 Experimental Results

In order to investigate the feasibility and effectiveness of the proposed hybrid QEA with 2-crossovers, a series of experiments are conducted on the Benchmark problem: *Maxf*(*x*)='*x*+10*sin(*x*.*5)+7*cos(*x*.*4)' to find the maximum value. In the three conducted experiments, the differences lie in the adopted crossover operation. The first experiment adopts classical crossover, the second experiment uses whole interference crossover, while the third experiment combines both of them.

The three QEAs have been encoded in Matlab language and implemented on PC-compatible with 1024 Mb of RAM under the Windows XP. The parameters were set to the following values: *n*=50, *L*=22, *Pcc*=0.9, *Pic*=0.2, *Pm*=0.2, *germax* is different in the three experiments: 200, 1000and 100.

The original chromosomes is produced with strong randomization. Figure 3 shows the final position of the chromosomes in the first experiment. It's obvious that it's easy to get into premature convergence. The evolution curves presented in Figure 4 also reflects the slow convergence speed.

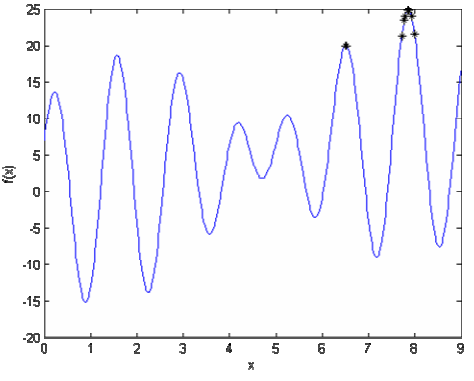


Fig. 3. Final position of the chromosomes in experiment 1

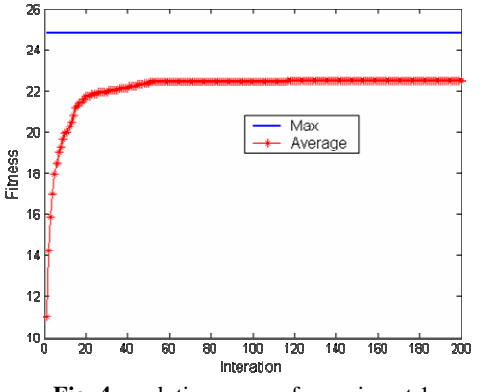


Fig. 4. evolution curve of experiment 1

Figure 5 – Figure 8 show the results in experiment 2 and 3.

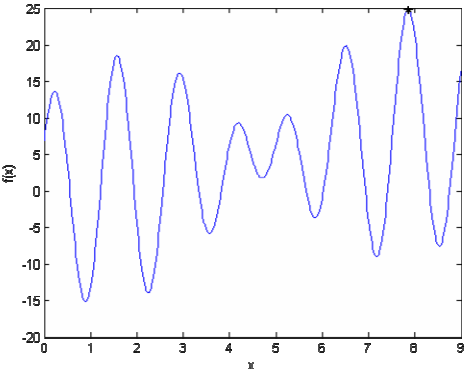


Fig. 5. Final position of the chromosomes in experiment 2

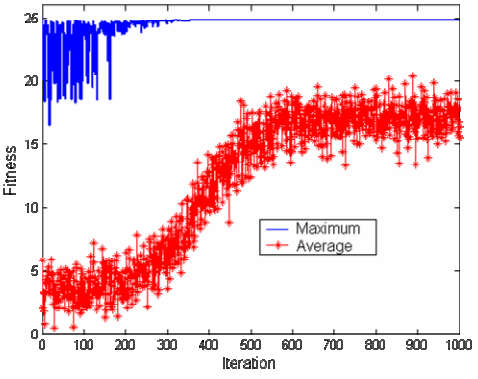


Fig. 6. Evolution curve of experiment 2

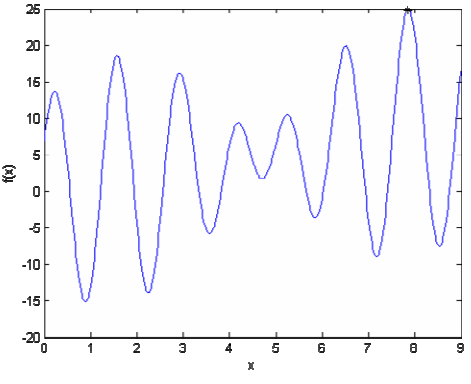


Fig. 7. Final position of the chromosomes in experiment 3

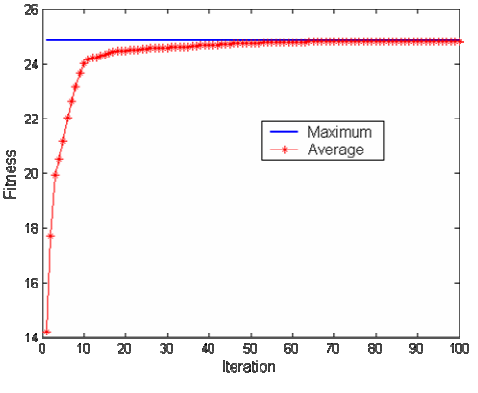


Fig. 8. Evolution curve of experiment 3

It is obvious that our proposed QEA model can find better solutions than the other basic QEAs in solving continuous optimization problems, and the improved QEA model can avoid premature convergence which happens in the first and the second experiments. Our proposed improved QEA with 2-crossovers has a more excellent performance with strong ability to find optimal solution and quick convergence speed.

5 Conclusions and Future Work

This paper has presented an improved QEA with 2-crossovers for solving the continuous optimization problems. The series experimental results verify that the proposed hybrid QEA model is a practical and effective algorithm in solving complex optimization problems, and also a feasible method for other complex real-world optimization problems.

Our future work will focus on applying the newly proposed QEA approach in this paper to other combinatorial optimization problems. Furthermore, we are also interested in the theoretical analysis on the proposed QEA model.

Acknowledgements. This work was supported by Natural Science Foundation of China under grant #60604009, Aeronautical Science Foundation of China under grant #2006ZC51039, "Beijng NOVA Program" Foundation of China under grant #2007A0017, "Science Research Training Program" Foundation of Ministry of Education of China, "New Star in Blue Sky" talent program of Beihang University, and Open Fund of the Provincial Key Laboratory for Information Processing Technology, Suzhou University under grant # KJS0821. The authors are grateful to Mr. Guanjun Ma and Mr. Xiangyin Zhang for their detailed guidance and great support in this research. The authors also want to show thanks to the anonymous referees for their valuable comments and suggestions, which lead to the better presentation of this work.

References

[1] Holland, J.: Adaptation in Natural and Artificial Systems. The University of Michigan Press, Ann Arbor (1975)

[2] Zhang, R., Gao, H.: Improved Quantum Evolutionary Algorithm for Combinatorial Optimization Problem. In: 6th International Conference on Machine Learning and Cybernetics, Hong Kong, pp. 3501–3505 (2007)

[3] Yang, S.Y., Liu, F., Jiao, L.C.: The Quantum Evolutionary Strategies. Acta electronica sinica 29(12A), 1873–1877 (2001)

[4] Zhang, W.F., Shi, Z.K., Luo, Z.Y.: Prediction of Urban Passenger Transport Based-on Wavelet SVM with Quantum-Inspired Evolutionary Algorithm. In: Proceedings of the 2008 International Joint Conference on Neural Networks, Hong Kong, pp. 1510–1514 (2008)

[5] Tayarani, M.H., Akbarzadeh. M. R.: A Cellular Structure and Diversity Preserving operator in Quantum Evolutionary Algorithms. In: Proceedings of the 2008 IEEE Congress on Evolutionary Computation, pp. 2670–2675, Hong Kong (2007)

[6] Xiao, J., Yan, Y.P., Lin, Y., Yuan, L., Zhang, J.: A Quantum-inspired Genetic Algorithm for Data Clustering. In: Proceedings of the IEEE Congress on Evolutionary Computation, Hong Kong, pp. 1513–1518 (2008)

[7] Wei, M., Li, Y.X., Jiang, D.Z., He, Y.F., Huang, X.Y., Xu, X.: A New Evolutionary Algorithm based on Quantum Statistical Mechanics. In: Proceedings of the IEEE Congress on Evolutionary Computation, Hong Kong, pp. 1722–1726 (2008)

A Maximum Power Point Tracking Method Based on Extension Neural Network for PV Systems

Kuei-Hsiang Chao, Ching-Ju Li , and Meng-Huei Wang

Department of Electrical Engineering, National Chin-Yi University of Technology, Taichung, Taiwan, R.O.C.
chaokh@ncut.edu.tw, s49612101@student.ncut.edu.tw, wangmh@ncut.edu.tw

Abstract. In this paper, a maximum power point tracking (MPPT) technique based on extension neural network (ENN) was proposed to make full utilization of photovoltaic (PV) array output power which depends on solar insolation and ambient temperature. The proposed ENN MPPT algorithm can automatically adjust the step size to track the PV array maximum power point (MPP). Compared with the conventional fixed step size perturbation and observation (P&O) and incremental conductance (INC) methods, the presented method is able to effectively improve the dynamic response and steady state performance of the PV systems simultaneously. A theoretical analysis and the designed principle of the proposed method are described in detail. And some simulation results are made to demonstrate the effectiveness of the proposed MPPT method.

Keywords: Maximum power point tracking (MPPT), Perturbation and observation (P&O) method, Incremental conductance (INC) method, Photovoltaic (PV) system, Extension neural network (ENN).

1 Introduction

Recently, there are many techniques have been proposed for tracking the maximum power point of PV arrays [1-6]. The voltage and current based photovoltaic generator methods [1] offer a simple and low-priced way to acquire the maximum power. Nevertheless, they require periodical disconnection or short-circuit of the PV modules to measure the open-circuit voltage or short-circuit current for reference, resulting in more power loss when scanning the entire control range. The perturbation and observation (P&O) method [2] is widely applied in the MPPT controller due to its simplicity and easy implementation, but its accuracy in steady-state is low because the perturbation process would make the operation point of the PV arrays to oscillate around the MPP. Therefore, the power loss may be increased. Furthermore, when the insolation changes rapidly, the P&O method probably fails to track MPP. The incremental conductance (INC) method [3], which is based on the fact that the slope of the PV array power versus voltage curve is zero at the MPP, has been proposed to improve the tracking accuracy and dynamic performance under rapidly varying conditions. The steady-state oscillations would be eliminated in theory since the derivative of the power with respect to the voltage vanishes at MPP. However, null value of the slope of the PV array power versus voltage curve seldom occurs due to the resolution of

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 745–755, 2009.

digital implementation. The INC MPPT algorithm usually uses a fixed iteration step size, which is determined by the accuracy and tracking speed requirement. Thus, the corresponding design should satisfactorily address the tradeoff between the dynamics and steady state oscillation. To solve these problems, a modified INC MPPT with variable step size and constant voltage tracking (CVT) at the start process [4] has been proposed to tune the step size automatically according to the inherent PV array characteristics. However, high complexity of the method requires high sampling accuracy and fast control speed, which might have resulted in a high cost system. And the system stability problem would be occurred once the control unit switches from CVT to INC MPPT mode or the starting point is not selected properly. The intelligent fuzzy, sliding mode and neural network methods [5-6] that focus on the nonlinear characteristics of PV array provide a good alternative for MPPT control. Since the output characteristics of the PV array should be well ascertained to create the MPPT control rules and the computation time of these algorithms are with high complexity, the versatility of these methods is limited.

In considering the aforementioned shortcomings and then to fulfill the requirements of dynamic response and steady-state performance of a MPPT control for PV arrays, an intelligent control algorithm based on P&O method with extension neural network [7] was proposed in this paper.

2 Extension Neural Network

2.1 The Structure of ENN

The schematic structure of the ENN is depicted in Fig. 1. The nodes in the input layer receive an input feature pattern and use a set of weighted parameters to generate an image of the input pattern.

In this network, there are two connection values (weights) between input nodes and output nodes, one connection represents the lower bound for this classical domain of the features, and the other connection represents the upper bound. The connection

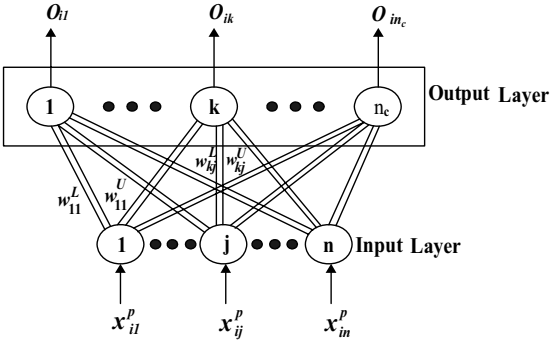


Fig. 1. The structure of extension neural network

weights between the *j*-th input node and the *k*-th output node are w_{kj}^{L} and w_{kj}^{U}. This image is further enhanced in the process characterized by the output layer. Only one output node in the output layer remains active to indicate a classification of the input pattern. The operation mode of the proposed ENN can be separated into the learning phase and the operation phase. The learning algorithm of the ENN will be discussed in the next section.

2.2 Learning Algorithm of the ENN

The learning of the ENN can be seen as supervised learning, and its purpose is to tune the weights of the ENN to achieve good clustering performance or to minimize the clustering error. Before the learning, several variables have to be defined. Let training pattern set be $X \equiv \left\{X_1, X_2, ..., X_{N_P}\right\}$, where N_P is the total number of training patterns. The *i*-th pattern is $X_i^p \equiv \left\{x_{i1}^p, x_{i2}^p, ..., x_{in}^p\right\}$, where n is the total number of the feature of patterns, and the category of the *i*-th pattern is p. To evaluate the clustering performance, the total error number is set as N_m, and the total error rate E_τ is defined below:

$$E_\tau = \frac{N_m}{N_p} \tag{1}$$

The detailed supervised learning algorithm can be described as follows:

Step 1: Set the connection weights between input nodes and output nodes. The range of classical domains can be either directly obtained from the previous requirement, or determined from training data as follows:

$$w_{kj}^{L} = \min_{i \in N}\left\{x_{kj}^{k}\right\} \tag{2}$$

$$w_{kj}^{U} = \max_{i \in N}\left\{x_{kj}^{k}\right\} \tag{3}$$

Step 2: Calculate the initial cluster center of every cluster.

$$Z_k = \left\{z_{k1}, z_{k2}, ..., z_{kn}\right\} \tag{4}$$

$$z_{kj} = \left(w_{kj}^{L} + w_{kj}^{U}\right)/2, \quad \text{for } k = 1, 2 \ldots n_c \text{ ; } j = 1, 2, \ldots n \tag{5}$$

Step 3: Read the *i*-th training pattern and its cluster number p.

$$X_i^p = \left\{x_{i1}^p, x_{i2}^p, ..., x_{in}^p\right\} \text{ , } p \in n_c \tag{6}$$

Step 4: Use the proposed extension distance (ED) to calculate the distance between the training pattern X_i^p and the *k*-th cluster as follows:

$$ED_{ik} = \sum_{j=1}^{n}\left[\frac{\left|x_{ij}^{p} - z_{kj}\right| - \left(w_{kj}^{U} - w_{kj}^{L}\right)/2}{\left|\left(w_{kj}^{U} - w_{kj}^{L}\right)/2\right|} + 1\right], \ k = 1, 2, ..., n_c \tag{7}$$

The proposed distance is a modification of extension distance [8], and it can be graphically presented as in Fig. 2. It can describe the distance between the x and a range $\left\langle w^{L}, w^{U}\right\rangle$. Figure 2 shows that different ranges of classical domains can arrive at different distances due to different sensitivities. This is a significant advantage in classification applications. Usually, if the feature covers a large range, the data should be fuzzy or less sensitive to distance. On the other hand, if the feature covers a small range, the data should be precise or highly sensitive to distance.

Step 5: Find the k^*, such that $ED_{ik^*}=\min\{ED_{ik}\}$, If $k^*=p$ then go to Step 7, otherwise Step 6.

Step 6: Update the weights of the p-th and the k^*-th clusters as follows:

(a) Update the centers of the p-th and the k^*-th clusters.

$$z_{pj}^{new} = z_{pj}^{old} + \eta\left(x_{ij}^{p} - z_{pj}^{old}\right) \tag{8}$$

$$z_{k^*j}^{new} = z_{k^*j}^{old} - \eta\left(x_{ij}^{p} - z_{k^*j}^{old}\right) \tag{9}$$

(b) Update the weights of the p-th and the k^*-th clusters.

$$\begin{cases} w_{pj}^{L(new)} = w_{pj}^{L(old)} + \eta(x_{ij}^{p} - z_{pj}^{old}) \\ w_{pj}^{U(new)} = w_{pj}^{U(old)} + \eta(x_{ij}^{p} - z_{pj}^{old}) \end{cases} \tag{10}$$

$$\begin{cases} w_{k^*j}^{L(new)} = w_{k^*j}^{L(old)} - \eta(x_{ij}^{p} - z_{k^*j}^{old}) \\ w_{k^*j}^{U(new)} = w_{k^*j}^{U(old)} - \eta(x_{ij}^{p} - z_{k^*j}^{old}) \end{cases} \tag{11}$$

where η is a learning rate. The result of tuning two clusters' weights shown in Fig. 3, which clearly indicates the change of ED_A and ED_B. The cluster of pattern x_{ij} is changed from cluster A to B because $ED_A > ED_B$. From this step, we can clearly see that the learning process is only to adjust the weights of the p-th and the k^*-th clusters. Therefore, the proposed method has a rapid speed advantage over other supervised learning algorithms and can quickly adapt to new and important information.

Step 7: Repeat Step 3 to Step 6, and if all patterns have been classified then a learning epoch is finished.

Step 8: Stop if the clustering process has converged or the total error rate E_τ has arrived at a preset value; otherwise, return to Step 3.

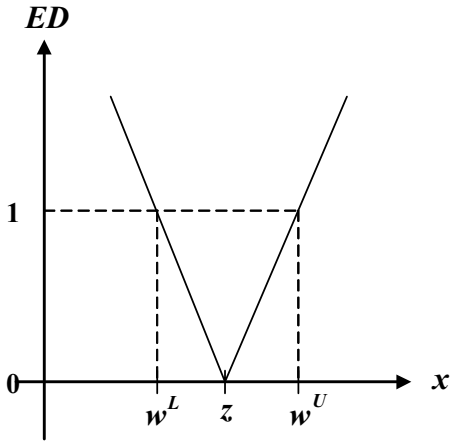


Fig. 2. The proposed extension distance

It should be noted that the proposed ENN can take input from human expertise before the learning, and it can also produce meaningful output after the learning, because the classified boundaries of the features are clearly determined.

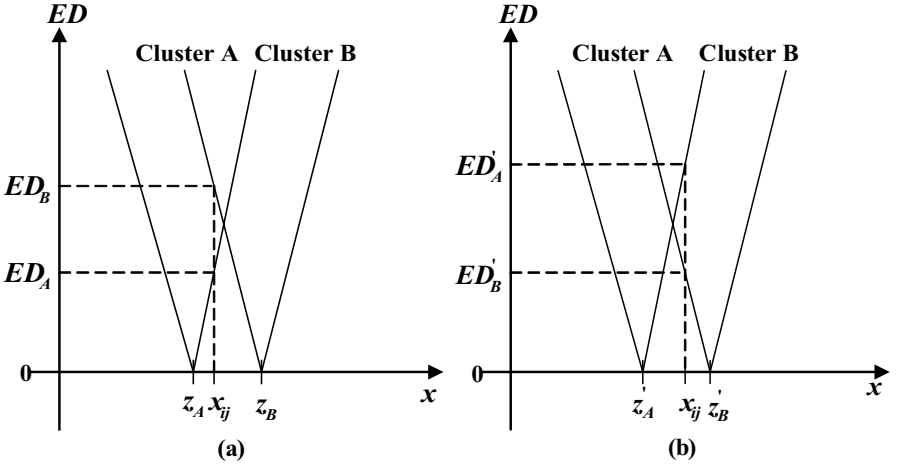


Fig. 3. The results of tuning cluster weights: (a) original condition; (b) after tuning

3 The Extension Neural Network MPPT Controller

3.1 The Proposed MPPT Scheme

A simple MPPT PV system shown in Fig. 4 is developed to test the effectiveness of the proposed method. A boost converter is used as the power interface between the PV array and the load to achieve maximum power. The output voltage V_o of the boost converter can be expressed as [9]

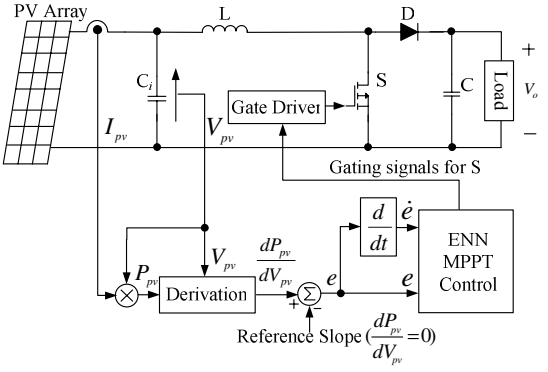


Fig. 4. The proposed extension MPPT scheme

$$V_o = \frac{V_{PV}}{1-D} \tag{12}$$

where D is the duty cycle of the converter. It can be seen that the input DC voltage V_{PV} can be shifted to a high level. This power converter is suitable for a lower PV output voltage and higher desirable DC link voltage case.

3.2 The Extension Neural Network MPPT Method

To let the MPPT method possess adaptive capability, it is proposed that the step size of P&O MPPT method of the PV arrays is adaptively tuned by the ENN error tuning scheme, which is driven by a slope error of the PV array power P_{pv} versus voltage V_{pv} and its error change which are defined as $e(k) \overset{\Delta}{=} \frac{P_{pv}(k)-P_{pv}(k-1)}{V_{pv}(k)-V_{pv}(k-1)} - 0 = \frac{dP_{pv}(k)}{dV_{pv}(k)}$ and $\dot{e}(k) \overset{\Delta}{=} e(k) - e(k-1)$ with P_{pv} and V_{pv} being the output power and voltage of the PV arrays at k-th sampling interval, respectively. The major purpose of this MPPT controller is to let the resulted dP_{pv}/dV_{pv} tracking response closely follow that of reference $dP_{pv}/dV_{pv}=0$ as shown in Fig. 5. Thus the general model error trajectory can be predicted and plotted in Fig. 6. Based on the experience about the P-V characteristic curve shown in Fig. 5 and incorporating with the extension matter-element, the numbers of quantization levels of the input variables $e(k)$ and $\dot{e}(k)$ are chosen to be 12 categories and listed in Table 1. Based on the experience about the MPP to be controlled and the properties of dynamic signal analyses made in Fig. 6, the linguistic rules of the ENN error tuning scheme are decided and listed in the Table 1. The slope error e is equal to zero at MPP of the PV array. When the operation point closes to MPP, the absolute value of slope error e becomes smaller; on the contrary, the value of e will becomes bigger. And, the polarity of tracking direction depends on the slope error and slope error change as those shown in Fig. 5. According to the range of e, $\dot{e}$ and tracking direction polarity, the ENN MPPT algorithm can discriminate the category and then determined the duty cycle step size of the boost converter.

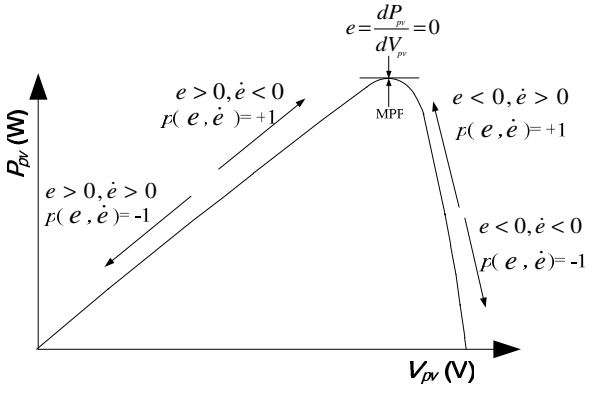


Fig. 5. The P-V curve slope error and error change of PV arrays

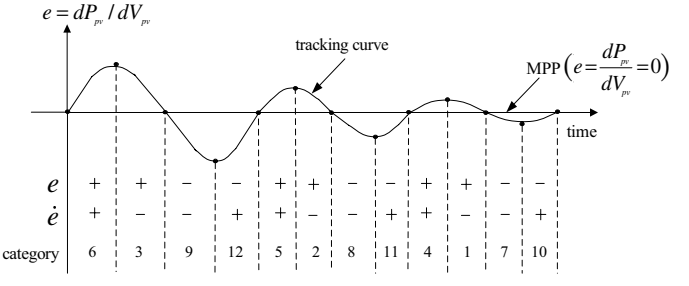


Fig. 6. General dP_{pv}/dV_{pv} reference tracking error dynamic behavior

Table 1. Quantized slope error, error change and decision duty cycle step size

Category number	Slope error region e	Slope error change region $\dot{e}$	Duty cycle step size ΔD	Tracking direction polarity $p(e,\dot{e})$
1	$0.01973 < e \le 13.0814$	$-30.33855 < \dot{e} \le -0.060754$	-0.01	+1
2	$13.1426 < e \le 18.9344$	$-32.80175 < \dot{e} \le -0.146825$	-0.012	+1
3	$19.2897 < e \le 48.3132$	$-77.2777 < \dot{e} \le -0.59005$	-0.025	+1
4	$0.02786 < e \le 13.3896$	$0.0602297 < \dot{e} \le 24.33907$	-0.01	-1
5	$13.4424 < e \le 18.9342$	$0.147061 < \dot{e} \le 32.80199$	-0.015	-1
6	$19.696 < e \le 48.3195$	$0.551126 < \dot{e} \le 77.2388$	-0.022	-1
7	$-20.7179 < e \le -0.34566$	$-140.5711 < \dot{e} \le -0.0238124$	0.01	-1
8	$-223.144 < e \le -21.9978$	$-142.668 < \dot{e} \le -0.87397$	0.012	-1
9	$-390.206 < e \le -224.154$	$-150.956 < \dot{e} \le -0.959167$	0.02	-1
10	$-20.7179 < e \le -0.34566$	$0.0238124 < \dot{e} \le 123.5711$	0.01	+1
11	$-251.144 < e \le -21.9978$	$0.87397 < \dot{e} \le 142.668$	0.025	+1
12	$-390.206 < e \le -252.154$	$0.959167 < \dot{e} \le 150.956$	0.03	+1

According to the *P-V* curve statistical records of PV arrays at insolation 200 W/m^2-1000 W/m^2, the lower and upper boundary of each classical region for slope error e and slope error change $\dot{e}$ are assigned as the connection weights w_{kj}^U and w_{kj}^L between input nodes and output nodes of the extension neural network.

The actual measured 892 data of slope error e and error change $\dot{e}$ at different regions of P-V curve are used to train the ENN proposed in the previous section. To obtain higher and precise convergence rate, the learning rate η and total error rate E_τ are set to be 0.2 and 0.1%, respectively. After the training procedure, one can find that the total error rate is 4.38% and only learning times 50 is needed.

The proposed ENN method can calculate the distance with respect to each region category, and accordingly the region category and duty cycle step size ΔD in next period can be determined. The operation procedure of ENN is summarized as follows:

Step 1: Read the weighting matrix of ENN.

Step 2: Calculate the initial cluster centers of every cluster by using equation (4) and equation (5).

Step 3: Read the test pattern.

$$X_t = \{x_{t1}, x_{t2}, ..., x_{tn}\} \tag{13}$$

Step 4: Use the proposed extension distance (*ED*) to calculate the distance between the inputted pattern and every existing cluster by equation (7).

Step 5: Find the k^*, such that $ED_{ik^*} = \min\{ED_{ik}\}$, and set the $O_{ik^*} = 1$ to indicate the cluster of the input pattern and recognize the category of the input slope error and error change information and determine the duty cycle step size ΔD of boost converter and the tracking direction polarity $p(e, \dot{e})$. To increase the sensitivity and adaptive capability, the new duty cycle D_{new} in next time period is determined as follows:

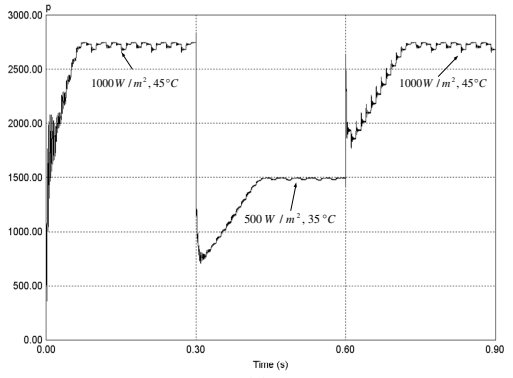
$$D_{new} = D_{old} + \Delta D + \left[(\Delta D_n - \Delta D_r) \times \frac{ED_r}{(ED_r + ED_n)} \right] \times p(e, \dot{e}) \times 0.5 \tag{14}$$

Where D_{old} is the duty cycle of boost converter in the previous sample period. The ΔD_r and ED_r are the nominal duty cycle step size and extended distance of the judged region cluster. Whereas ΔD_n and ED_n are the nominal duty cycle step size and extended distance next to the judged region cluster. And $p(e, \dot{e})$ is the tracking direction polarity as shown in Table 1.

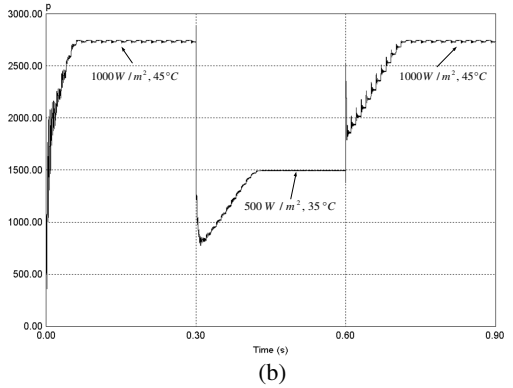
Step 6: Input a new slope error and error change, then go back to Step 2, or else end the process.

4 Simulation Results

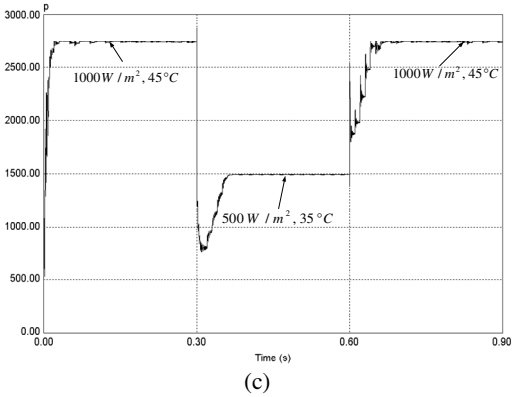
To verify the performance of the proposed ENN MPPT algorithm, a circuit-based PSIM model of a 3kW PV system with 4×10 series-parallel connection using



(a)



(b)



(c)

Fig. 7. Simulated dynamic response of PV array output power due to solar insolation step change from 1000*W/m*^{2} to 500 *W/m*^{2} and from 500*W/m*^{2} to 1000 *W/m*^{2} with (a) P&Q MPPT method; (b)INC MPPT method; (c)the proposed ENN MPPT method

SIEMENS SP75 Crystalline Silicon PV module is developed for simulation [10,11]. The specifications of the boost converter shown in Fig. 4 are chosen as follows:

DC capacitance : $C_i = 1000\mu F$, $C = 100\mu F$;

Filter inductance: $L = 200\mu H$;

Switching frequency: $f_s = 25kHz$.

To compare the performance of the proposed ENN MPPT algorithm which can automatically adjust the step size with the conventional fixed step size P&O and INC MPPT methods, the simulations are configured under exactly the same conditions. The sampling period used for P&O and INC MPPT algorithm is chosen as 0.01s. Therefore, the duty cycle of the boost converter is updated every 0.01s. The output power performance of P&O and INC MPPT methods with fixed step size 0.01 under insolation step change from 1000 W/m^2 (at temperature T=45℃) to 500 W/m^2 (at temperature T=35℃) at 0.3s are shown in Figs. 7(a) and 7(b), respectively. For the comparison, the corresponding PV output power response of the proposed ENN MPPT method with allowable maximum duty cycle step size ΔD_{max}=0.03 is also shown in Fig. 7(c). It is obvious that the oscillations occurred at stead-state in P&O and INC MPPT are almost eliminated by the proposed ENN MPPT algorithm. Moreover, the dynamic performance of the proposed method is obviously faster than that of P&O and INC MPPT with fixed step duty cycle step size 0.01.

To further verify the robustness of the proposed ENN MPPT method, the simulated power responses due to solar insolation step change from 500 W/m^2 (at temperature T=35℃) to 1000 W/m^2 (at temperature T=45℃) with P&O, INC and extension neural network MPPT methods are also shown in Figs. 7(a), 7(b), and 7(c) for comparison. The results indicate that the oscillations at steady-state are greatly reduced by utilizing the proposed ENN MPPT algorithm. Meanwhile, the dynamic response performance due to insolation change is improved, also.

5 Conclusions

An ENN MPPT method was presented in this paper. This method combined the extension theory and neural network with a boost converter to speed up responses for reaching the accurate MPP of PV arrays under solar insolation and ambient temperature changes. In addition, the proposed ENN MPPT method can also improve the steady-state performance and energy conversion efficiency. Both fixed step size MPPT methods (P&O and INC) and the proposed variable step size ENN MPPT methods are established with PSIM circuit-based model for simulation. The simulation results demonstrate the effectiveness and robustness of the proposed method. Furthermore, the proposed ENN MPPT algorithm needs less constructed data and simple learning procedure such that it can be easily implemented using microcontroller in the future.

Acknowledgments. The authors would like to thank the Photovoltaic Technology Center, Industrial Technology Research Institute, Taiwan for financially supporting this research.

References

1. Masoum, M.A.S., Dehbonei, H., Fuchs, E.F.: Theoretical and Experimental Analyses of Photovoltaic Systems with Voltage- and Current-Based Maximum Power-Point Tracking. IEEE Trans. Energy Conversion 17, 514–522 (2002)
2. Femia, N., Granozio, D., Petrone, G., Spagnuolo, G., Vitelli, M.: Predictive & Adaptive MPPT Perturb and Observe Method. IEEE Trans. Aerospace Electronics System 43, 934–950 (2007)
3. Wu, L., Zhao, Z., Liu, J.: A Single-Stage Three-Phase Grid-Connected Photovoltaic System with Modified MPPT Method and Reactive Power Compensation. IEEE Trans. Energy Conversion 22, 881–886 (2007)
4. Liu, F., Duan, S., Liu, B., Kang, Y.: A Variable Step Size INC MPPT Method for PV Systems. IEEE Trans. Industrial Electronics 55, 2622–2628 (2008)
5. Kottas, T.L., Boutalis, Y.S., Karlis, A.D.: New Maximum Power Point Tracker for PV Arrays Using Fuzzy Controller in Close Cooperation with Fuzzy Cognitive Networks. IEEE Trans. Energy Conversion 21, 793–803 (2006)
6. Veerachary, M., Senjyu, T., Uezato, K.: Neural-Network-Based Maximum-Power-Point Tracking of Coupled-Inductor Interleaved-Boost Converter-Supplied PV System Using Fuzzy Controller. IEEE Trans. Industrial Electronics 50, 749–758 (2003)
7. Wang, M.H., Hung, C.P.: Extension Neural Network. In: Proceedings of the International Joint Conference on Neural Network, pp. 399–403 (2003)
8. Cai, W.: The Extension Set and Incompatibility Problem. Journal of Scientific Exploration 1, 81–93 (1983)
9. Hart, D.W.: Introduction to Power Electronics. Prentice Hall, New Jersey (1997)
10. SIEMENS Solar Module SP75 Specifications, `http://www.abcsolar.com/pdf/sp75.pdf`
11. PSIM User's Guide, Powersim Inc. (2001-2003)

A Versatile Hyper-Ellipsoidal Basis Function for Function Approximation in High Dimensional Space

Saichon Jaiyen, Chidchanok Lursinsap, and Suphakant Phimoltares

Advance Virtual and Intelligent Computing (AVIC) Center, Department of Mathematics, Chulalongkorn University, Bangkok 10330, Thailand

Abstract. This paper presents a versatile hyper-ellipsoidal basis function for function approximation in a given high dimensional space. This hyper-ellipsoidal basis function can be translated and rotated to cover the data based upon the distribution of data in a given high dimensional space. Based on this function, we propose a one-pass hyper-ellipsoidal learning algorithm for which any new incoming data can be fed for learning without involving the previously learned one. This learning algorithm is used to adjust the parameters of the versatile hyper-ellipsoidal basis function. In addition, we propose the hyper-ellipsoidal basis function (HEBF) neural network that uses the one-pass hyper-ellipsoidal neural learning algorithm. The structure of this neural network is similar to the radial basis function (RBF) neural networks. The hidden neurons in the HEBF neural network can be increased or decreased during learning process. The number of the hidden neurons in the network can be grown based on geometric growth criterion and can be reduced by merging the two hidden neurons into a new hidden neuron based on merging criterion during learning process. The merging process can be done independently without considering the learned data set.

1 Introduction

Radial basis function (RBF) neural network is extremely popular and widely used in function approximation problems [6], [7] because of their ability to approximate complex nonlinear mappings directly from the input-output data with a simple neural network structure. The disanvantage of RBF neural networks is that the structure of the RBF neural network is fixed before training process and the number of hidden neurons cannot be known in advance. To overcome this problem, Platt [1] proposed the sequential learning algorithm for RBF neural network in which hidden neurons are added sequentially based on the novelty of the new data. This neural network is called resource allocation network (RAN). Kadirkamanathan and Niranjan [2] enhance RAN using an extended Kalman filter (EKF) for updating the network parameters instead of least-mean square (LMS) algorithm, known as a RAN extended Kalman filter (RANEKF). The drawback of RAN and RANEKF is that the hidden neurons can grow up but the hidden neuron is never removed. Yingwei [3] proposed the improvement of

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 756–765, 2009.

RANEKF by introducing a pruning strategy called minimal resource allocating network (MRAN). Li Yan [5] proposed an improved version of the MRAN algorithm called Extended-MRAN (EMRAN) algorithm in which the parameters that are related to the selected winner neurons are updated by the EKF algorithm. The disanvantage of these proposed neural networks is that there are a lot of parameters choosen by trial and error.

The performance of these RBF neural networks depend on the selection of the RBF centers [8], the shape of radial basis function and their weights. Although, new RBF neural networks are proposed consecutively, the learning algorithm is still based on Gaussian function with diagonal covariance matrix as a basis function. The shape related to Gaussian function in high dimensional input space is either an hyper-spherical shape when the values in diagonal are the same or an unrotated hyper-ellipsoidal shape when the values in diagonal are different. Nevertheless, the shape related to Gaussian function in a high dimensional input space is able to rotate the hyper-ellipsoidal shape if the Gaussian function is used with full covariance matrix, the parameters of this function such as mean vector and covariance matrix are difficult to compute and require a lot of epochs to estimate these parameters because of the optimization technique [9]. In this paper, we propose the new basis function for function approximation problem in high dimensional space called a versatile hyper-ellipsoidal basis function based on hyper-ellipsoidal function that can be translated and rotated to cover the data in the high dimensional space. In addition, the new proposed neural network is based on the versatile hyper-ellipsoidal basis function called a versatile hyper-ellipsoidal basis function (HEBF) neural network that uses only one-pass hyper-ellipsoidal neural learning algorithm to adjust the parameters of the versatile hyper-ellipsoidal basis function.

2 HEBF Neural Network

The versatile hyper-ellipsoidal basis function (HEBF) neural network consists of three layers, an input layer, a hidden layer, and an output layer. Given n neurons in the input layer, K neurons in the hidden layer, and one neuron in the output layer. For a given input vector, $\mathbf{x} = [x_1, x_2, ..., x_n]^T$, in $\Re^n$, the network output, $f(\mathbf{x})$, is defined by

$$f(\mathbf{x}) = \sum_{k=1}^{K} \alpha_k \left| \psi_k(\mathbf{x}) \right| \tag{1}$$

where α_k is the weight connecting the k^{th} hidden neuron to the output neuron, $\psi_k(\mathbf{x})$ is the response of k^{th} hidden neuron for an input vector $\mathbf{x}$, and $|\cdot|$ denotes the absolute value. In this paper, we define the new basis function, namely, *Versatile Hyper-Ellipsoidal Basis Function (HEBF)* as shown in the equation below. The versatile hyper-ellipsoidal basis function, $\psi_k(\mathbf{x})$, at the k^{th} neuron in hidden layer is defined as follows.

$$\psi_k(\mathbf{x}) = \sum_{i=1}^{n} \frac{((\mathbf{x} - \mathbf{c}_k)^T \mathbf{u}_i)^2}{a_i^2} - 1 \tag{2}$$

where the vector $\mathbf{c}_k = [c_1, c_2, ..., c_n]^T$ is the center of the versatile hyper-ellipsoidal basis function of k^{th} hidden neuron, a_i is the width of the i^{th} axis of the versatile hyper-ellipsoidal basis function of k^{th} hidden neuron, and $\mathbf{u}_i = [u_1, u_2, ..., u_n]^T$ is the unit basis vector along the i^{th} axis of the versatile hyper-ellipsoidal basis function of k^{th} hidden neuron as explained in the next section.

3 Incremental Mean Vector and Incremental Covariance Matrix Computation

Since the traditional mean vector requires all data vectors to compute their mean vector, it is not suitable for one-pass incremental learning since all training data are ignored after being learned. In order to compute the present value of mean vector, the value of mean vector must be rewritten in the form of a recurrence relation between the previous value of mean vector and the present value of mean vector. The following Theorem states the recurrence relation.

Theorem 1. *Let $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}$ be a set of N data vectors in $\Re^n$ and μ_{old} be the mean vector of the data set X. If $\boldsymbol{x}_{N+1} \in \Re^n$ is the new data vector added into the data set X then*

$$\mu_{new} = \alpha\mu_{old} + \beta, \tag{3}$$

where μ_{new} is the new mean vector, $\alpha = \frac{N}{N+1}$ and $\beta = \frac{\boldsymbol{x}_{N+1}}{N+1}$.

Like the mean vector, the present covariance matrix must be rewritten in the form of recurrence relation similar to the mean vector. The theorem below states this recurrence relation.

Theorem 2. *Let $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}$ be a set of N data vectors in $\Re^n$, μ_{old} the mean vector of the data set X, and S_{old} the covariance matrix of the data set X. If a new data vector $\boldsymbol{x}_{N+1} \in \Re^n$ is added into the data set X then*

$$S_{new} = \alpha S_{old} + \kappa \tag{4}$$

where $\alpha = \frac{N}{N+1}$, $\kappa = \kappa_1 + \kappa_2$, $\kappa_1 = \frac{\boldsymbol{x}_{N+1}\boldsymbol{x}_{N+1}^T}{N+1} - \mu_{new}\mu_{new}^T + \mu_{old}\mu_{old}^T$, $\kappa_2 = -\frac{\mu_{old}\mu_{old}^T}{N+1}$, μ_{new} is the new mean and S_{new} is the new covariance matrix.

4 The Proposed Learning Algorithm

In this paper, we denote $\Omega = \{\Omega_k | 1 \leq k \leq K\}$ be a set of K hidden neurons. Each hidden neural Ω_k is demonstrated as a 4-tuple, $\left(\mathbf{a}^{(k)}, \mathbf{c}^{(k)}, S^{(k)}, N_k\right)$, where

$\mathbf{a}^{(k)} = [a_1^{(k)}, a_2^{(k)}, ..., a_n^{(k)}]^T$ is the width vector of k^{th} hidden neuron, $\mathbf{c}^{(k)} = [c_1, c_2, ..., c_n]^T$ is the center of k^{th} hidden neuron, $S^{(k)}$ is the covariance matrix of k^{th} hidden neuron, and N_k is the total number of data covered by k^{th} hidden neuron. The element, $a_j^{(k)}$, of the width vector is the width parameter which is the width of the j^{th} axis of the versatile hyper-ellipsoidal basis function of k^{th} hidden neuron. Let $a_j^{(0)}$ be an initial width of j^{th} axis chosen appropriately. Let N_0 be the constant chosen appropriately for making a decision to adjust the width parameter $a_j^{(k)}$. Let $X = \{(\mathbf{x}_i, y_i)|1 \leq i \leq N\}$ be a finite set of N training data, where $\mathbf{x}_i \in \Re^n$ is an *input vector* and $y_i \in \Re$ is a desired output.

4.1 Geometrical Growth Criterion

Initially, there is no hidden neuron in the HEBF neural networks. A new hidden neuron can be added into the network. When an input-output vector $(\mathbf{x}_i, y_i)$ is presented into the network, the hidden neuron that is closest to the input vector $\mathbf{x}_i$ is assigned. If there exists the closest hidden neuron in the network, this data vector is temporarily considered to be an element of the closest hidden neuron. The new temporary parameters of the closest hidden neuron including the center $\mathbf{c}_{new}^{(cs)}$, the covariance matrix $S_{new}^{(cs)}$, and the number of elements N_{cs} of the closest hidden neuron are computed but the width of axes of the versatile hyper-ellipsoidal basis function have not been computed. Then, the output of the closest hidden neuron is computed using the new temporary parameters along with equation (2) as follows.

$$\psi_{cs}(\mathbf{x}_i) = \sum_{i=1}^{n} \frac{((\mathbf{x}_i - \mathbf{c}_{new}^{(cs)})^T \mathbf{u}_i)^2}{(a_i^{(cs)})^2} - 1,$$

where the index cs represents the index of the closest hidden neuron. If the geometrical growth criterion $\psi_{cs}(\mathbf{x}_i) > 0$, then a new hidden neuron is allocated and added into the network.

4.2 Merging Strategy

Given $\Omega_x = \left(\mathbf{a}^{(x)}, \mathbf{c}^{(x)}, S^{(x)}, N_x\right)$ and $\Omega_y = \left(\mathbf{a}^{(y)}, \mathbf{c}^{(y)}, S^{(y)}, N_y\right)$ be any two hidden neurons in HEBF neural network. If these two hidden neurons are merged into a new hidden neuron $\Omega_{new} = \left(\mathbf{a}^{(new)}, \mathbf{c}^{(new)}, S^{(new)}, N_{new}\right)$, the new parameters of a new hidden neuron can be computed as follows.

$$a_j^{(new)} = \sqrt{2\pi|\lambda_j|} + \delta, \ j = 1, ..., n. \tag{5}$$

$$\mathbf{c}^{(new)} = \frac{1}{N_x + N_y}\left(N_x \mathbf{c}^{(x)} + N_y \mathbf{c}^{(y)}\right) \tag{6}$$

$$\begin{aligned} S^{(new)} &= \frac{N_x}{N_x + N_y} S^{(x)} + \frac{N_y}{N_x + N_y} S^{(y)} \\ &\quad + \frac{N_x N_y}{(N_x + N_y)^2}(\mathbf{c}^{(x)} - \mathbf{c}^{(y)})(\mathbf{c}^{(x)} - \mathbf{c}^{(y)})^T \end{aligned} \tag{7}$$

$$N_{new} = N_x + N_y \tag{8}$$

4.3 One-Pass Hyper-Ellipsoidal Neural Learning Algorithm

In this section, *one-pass* means that each input data vector is learned only one time. Let K be the number of hidden neurons in HEBF neural network and ϵ be a threshold for merging the hidden neurons in the network. If there is no hidden neuron in the network, we set $K = 0$. The learning algorithm for HEBF neural network can be summarized as follows:

Algorithm

1. Present a training data $(\mathbf{x}_i, y_i)$ to the HEBF neural network.
2. **If** $K \neq 0$ **then** find a hidden neuron $\Omega_k \in \Omega$ where
$$k = \arg\max_j(||\mathbf{x}_i - \mathbf{c}^{(j)}||),\ j = 1, ..., K.$$
   (a) Compute $\mathbf{c}_{old}^{(k)} = \mathbf{c}^{(k)}$ and $S_{old}^{(k)} = S^{(k)}$.
   (b) Compute the new center $\mathbf{c}_{new}^{(k)}$ based on Theorem 1.
$$\mathbf{c}_{new}^{(k)} = \alpha \mathbf{c}_{old}^{(k)} + \beta$$
   (c) Compute the new covariance matrix based on Theorem 2.
$$S_{new}^{(k)} = \alpha S_{old}^{(k)} + \kappa$$
   else set $K = K + 1$ and create a new hidden neuron Ω_K.
   (a) Set the center $\mathbf{c}^{(K)} = \mathbf{x}_i$.
   (b) Set the covariance matrix $S^{(K)} = 0$ (*zero* matrix).
   (c) Set the parameter $N_K = 1$.
   (d) Set the width parameter $a_j^{(K)} = a_j^{(0)},\ j = 1, ..., n$.
   (e) Set $\Omega = \Omega \cup \Omega_K$ and remove $(\mathbf{x}_i, y_i)$ from the training set X.
   (f) Go to step 7.
   end
3. Compute the orthonormal basis for Ω_k.
   (a) Compute the eigenvalues of covariance matrix $S_{new}^{(k)}$:
$$\lambda_1 > \lambda_2 > ... > \lambda_n.$$
   (b) Compute the eigenvectors of the covariance matrix $S_{new}^{(k)}$:
$$\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_n,$$
   where $\mathbf{u}_j$ is the eigenvector corresponding to λ_j.
   (c) Assign $\{\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_n\}$ be the orthonormal basis for Ω_k.
4. Compute $\psi_k(\mathbf{x}_i) = \sum_{i=1}^{n} \frac{((\mathbf{x}_i - \mathbf{c}_{new}^{(k)})^T \mathbf{u}_i)^2}{(a_i^{(k)})^2} - 1$
5. **If** $\psi_k(\mathbf{x}_i) \leq 0$ **then** update the parameters of Ω_k using these steps:
   (a) Update the center of Ω_k, $\mathbf{c}^{(k)} = \mathbf{c}_{new}^{(k)}$.
   (b) Update the covariance matrix of Ω_k, $S^{(k)} = S_{new}^{(k)}$.

(c) Update the parameter $N_k = N_k + 1$.
(d) **If** $N_k \geq N_0$ **then** update the width parameter of Ω_k by setting

$$a_j^{(k)} = \sqrt{2\pi|\lambda_j|} + \delta,\ j = 1, ..., n,$$

where δ is a small value.
end

else set $K = K + 1$ and create a new hidden neuron Ω_K.
(a) Set the center $\mathbf{c}^{(K)} = \mathbf{x}_i$.
(b) Set the covariance matrix $S^{(K)} = 0$ (*zero* matrix).
(c) Set the parameter $N_K = 1$.
(d) Set the width parameter $a_j^{(K)} = a_j^{(0)},\ j = 1, ..., n$.
(e) Set $\Omega = \Omega \cup \Omega_K$ and remove $(\mathbf{x}_i, y_i)$ from the training set X.
end

6. Find the hidden neuron $\Omega_{nr} \in \Omega$ such that

$$nr = \arg\min_{j \neq cr} ||\mathbf{c}^{(cr)} - \mathbf{c}^{(j)}||, j = 1, ..., K,$$

where cr is the index of current updated neuron or the new added neuron and nr is the index of the nearest neuron.

If $||\mathbf{c}^{(cr)} - \mathbf{c}^{(nr)}|| < \epsilon$ **then** do the following steps:

(a) Merge the hidden neuron Ω_{nr} and Ω_{cr} into a new hidden neuron and compute its parameters by equation (5), (6), (7), and (8).
(b) Add the new hidden neuron into the network and then remove Ω_{nr} and Ω_{cr} from the network.
(c) Set $K = K - 1$.
end

7. **If** the training set X is not empty **then** go to step 1.
else, stop training.

Let $\mathbf{w} = [w_1, w_2, ..., w_K]^T$ be a weight vector of HEBF neural network where $w_i, i = 1, ..., K$, is the weight between i^{th} hidden neuron in the hidden layer and the output neuron in the output layer. After training, the weights between hidden layer and output layer are computed by the following equation.

$$\mathbf{w} = \mathbf{G}^+\mathbf{y}$$

where $\mathbf{G}^+$ is the pseudo inverse of matrix $\mathbf{G}$ and $\mathbf{y} = [y_1, y_2, ..., y_N]^T$ is the desired output vector. The matrix $\mathbf{G}$ is defined by

$$\mathbf{G} = \begin{bmatrix} |\psi_1(\mathbf{x}_1)| & |\psi_1(\mathbf{x}_2)| & \cdots & |\psi_1(\mathbf{x}_N)| \\ |\psi_2(\mathbf{x}_1)| & |\psi_2(\mathbf{x}_2)| & \cdots & |\psi_2(\mathbf{x}_N)| \\ \vdots & \vdots & & \vdots \\ |\psi_K(\mathbf{x}_1)| & |\psi_K(\mathbf{x}_2)| & \cdots & |\psi_K(\mathbf{x}_N)| \end{bmatrix}$$

5 Experimental Results

In this paper, the performance of HEBF neural network is evaluated and compared to the radial basis function (RBF) neural network, the minimal resource allocating network (MRAN), and the Extended-MRAN (EMRAN). The benchmark problems are the real world problems in the area of the function approximation and regression collected from UCI Repository of machine learning database [10]. In the experiments, the input attributes and the output of the data set are normalized within the range $[0, 1]$. The 5-fold cross-validation is used for training and testing all models mention above. In the 5-fold cross-validation, each data set is divided into five disjoint subsets. Then four subsets are used as a training set and the rest is used as a testing set. After training, we test all models by the training set and testing set. Then, the errors from these models are computed by using the mean absolute error (MAE) and the mean square error (MSE). This process is repeated five times, each of which one subset is used exactly once as the testing set. Eventually, the results from each training and testing set are averaged. The proposed (HEBF) neural network as well as the MRAN and EMRAN are trained only one epoch. Suppose that $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, where $\mathbf{x}_i \in \Re^n$, is the training set and μ is the mean of this data set. The initial width of HEBF neural network and RBF neural network are computed from the following equation.

$$a_0 = \rho \cdot \max_i \|\mathbf{x}_i - \mu\|, \ i = 1, ..., N \tag{9}$$

where ρ is the threshold. For the MRAN and EMRAN algorithms, the values of $\kappa = 0.1$, $\gamma = 0.99$, $Q_0 = 0.00001$, and $P_0 = 1.0$ are fixed for all experiments. Next, the results for each benchmark problem are presented.

5.1 Boston Housing Data Set

Boston housing data set [10] consists of 13 input attributes and one continuous output (median value of owner-occupied homes). There are 506 instances in this data set. For the MRAN and EMRAN algorithms, the model parameters are chosen as follows: $e_{min} = 0.001$, $e'_{min} = 0.002$, $\epsilon_{min} = 0.2$, $\epsilon_{max} = 1.15$, and a size of sliding window, M, for growing and pruning is chosen at $M = 80$. For the RBF algorithm, the initial width of Gaussian basis function is computed from equation (9) with $\rho = 0.25$. For our algorithm, the model parameters are initialized as follows: $N_0 = 15$, $\epsilon = 0.39$, $\delta = 0.01$, and the initial axes are computed as $a_i^k = a_0$, where a_0 is computed from equation (9) with $\rho = 0.25$ as same as RBF. The average results of the mean absolute error and the mean square error from the 5-fold cross-validation method together with both training and testing set are illustrated in Table 1. It can be obviously noticed that the average accuracy of the training set of the proposed algorithm is higher than the average accuracy of the training set of MRAN and EMRAN algorithm but slightly less than that of RBF. Furthermore, the average accuracy of testing of the proposed algorithm is higher than those of the others. In addition, the

Table 1. The comparison results trained by Boston Housing Data Set

Algorithms	Training Error		Testing Error		No. of	No. of
	MAE	MSE	MAE	MSE	Neurons	epochs
HEBF	0.0653	0.0077	0.0910	0.0144	42.6	1
RBF	0.0475	0.0043	0.0910	0.0164	43	43
MRAN	0.1299	0.0340	0.1319	0.0358	43	1
EMRAN	0.1397	0.0392	0.1461	0.0429	43.6	1

Table 2. The comparison results trained by Concrete Compressive Strength Data Set

Algorithms	Training Error		Testing Error		No. of	No. of
	MAE	MSE	MAE	MSE	Neurons	epochs
HEBF	0.0637	0.0069	0.0852	0.0120	80	1
RBF	0.0675	0.0072	0.1035	0.0184	80	80
MRAN	0.1395	0.0327	0.1523	0.0359	81.8	1
EMRAN	0.1480	0.0366	0.1752	0.0471	84.6	1

average number of the hidden neurons of the proposed algorithm is slightly less than those of the others.

5.2 Concrete Compressive Strength Data Set

Concrete compressive strength data Set [10] consists of 8 input attributes and one continuous output (Concrete compressive strength). There are 1030 instances in this data set. For the RBF algorithm, the initial width of Gaussian basis function is computed from equation (9) with $\rho = 0.2$. For our algorithm, the model parameters are initialized as follows: $N_0 = 15$, $\epsilon = 0.285$, $\delta = 0.01$, and the initial axes are computed as $a_i^k = a_0$, where a_0 is computed from equation (9) with $\rho = 0.2$. For the MRAN and EMRAN algorithms, the model parameters are chosen as follows: $e_{min} = 0.001$, $e'_{min} = 0.002$, $\epsilon_{min} = 0.1$, $\epsilon_{max} = 1.15$, and the size of sliding window for growing and pruning is chosen as $M = 80$. The average results of the mean absolute error and the mean square error from the 5-fold cross-validation method are shown in Table 2. It shows that our algorithm performs better than the others in terms of training and testing errors and the number of neurons in the network.

5.3 Computer Hardware Data Set

Computer hardware data set [10] consists of 6 predictive attributes, 2 non-predictive attributes, 1 goal field, and the linear regression's guess. There are 209 instances in this data set. In this data set, the 6 predictive attributes are

Table 3. The comparison results trained by Computer Hardware Data Set

Algorithms	Training Error		Testing Error		No. of	No. of
	MAE	MSE	MAE	MSE	Neurons	epochs
HEBF	0.0246	0.0013	0.0629	0.0122	13.2	1
RBF	0.0166	0.0006	0.0862	0.0436	14	14
MRAN	0.0754	0.0180	0.0949	0.0448	14	1
EMRAN	0.0795	0.0187	0.0938	0.0433	13.8	1

used as the input attributes and the goal field attribute (published relative performance) is used as the output. For the MRAN and EMRAN algorithms, the model parameters are chosen as follows: $e_{min} = 0.002$, $e'_{min} = 0.001$, $\epsilon_{min} = 0.04$, $\epsilon_{max} = 1.15$, and a size of sliding window for growing and pruning is presented at 80. For the RBF algorithm, the initial width of Gaussian basis function is computed from equation 9 with $\rho = 0.2$. For our algorithm, the model parameters are initialized as follows: $N_0 = 15$, $\epsilon = 0.31$, $\delta = 0.01$, and the initial axes are computed as $a_i^k = a_0$, where a_0 is computed from equation (9) with $\rho = 0.2$. The average results of the mean absolute error and the mean square error from the 5-fold cross-validation method are shown in Table 3. According to Table 3, the average accuracy of the training of the proposed algorithm is higher than the average accuracy of the training of MRAN and EMRAN algorithm but slightly less than that of RBF. Furthermore, the average accuracy of the testing of the proposed algorithm is higher than those of the others and, also, the average number of hidden neurons is less than those of the others. Here, we define one *epoch* as a period of unit time measured during the consecutive training from the first input pattern to the last input pattern. From all simulation results, the number of epochs of the proposed algorithm is equal to the number of epochs of MRAN and EMRAN but less than that of RBF.

6 Conclusion

In this paper, a versatile hyper-ellipsoidal basis function (HEBF) for function approximation in high dimensional space is proposed. The basis function can be translated and rotated to cover the input data in high dimensional space depend upon the distribution of the data set in the high dimensional data space. A one-pass hyper-ellipsoidal learning algorithm for adjusting the parameters of a versatile hyper-ellipsoidal basis function is mainly proposed. This algorithm can adjust the parameters of the versatile hyper-ellipsoidal basis function only one epoch for one incoming data. This algorithm includes the geometrical growth criterion to handle the new hidden neurons and merging strategy to merge the two hidden neurons into a new hidden neuron. This merging strategy does not require any previous trained data. Consequently, the number of hidden neurons can be reduced. Furthermore, the versatile hyper-ellipsoidal basis function (HEBF)

neural network is presented for approximating the function in high dimensional space based on the versatile hyper-ellipsoidal basis function. The performance of proposed learning algorithm has been compared with other sequential well known learning algorithms such as RBF, MRAN, and EMRAN on three real world problems in the function approximation area. The results indicate that the average accuracy of the testing set of the proposed model is better than the other models.

Acknowledgment

The authors would like to thank the Thailand Research Fund (TRF) for financial support of this research under the Royal Golden Jubilee (RGJ) Scholarship.

References

1. Platt, J.: A Resource-allocating Network for Function Interpolation. Neural Computation 3, 213–225 (1991)
2. Kadirkamanathan, V., Niranjan, M.: A Function Estimation Approach to Sequential Learning with Neural Networks. Neural Computation 5, 954–975 (1993)
3. Yingwei, L., Sundararajan, N., Saratchandran, P.: A Sequential Learning Scheme for Function Approximation Using Minimal Radial Basis Function (RBF) Neural Networks. Neural Computatation 9, 461–478 (1997)
4. Yingwei, L., Sundararajan, N., Saratchandran, P.: Performance Evaluation of a Sequental Minimal Radial Basis Function (RBF) Neural Network Learning Algorithm. IEEE Trans. Neural Networks 9, 308–318 (1998)
5. Li, Y., Sundararajan, N., Saratchandran, P.: Analysis of Minimal Radial Basis Function Network Algorithm for Real-time Identification of Nonlinear Dynamic Systems. IEEE Proceedings-Control Theory and Applications 147, 476–484 (2000)
6. Yan, X.B., Wang, Z., Yu, S.H., Li, Y.J.: Time Series Forecasting With RBF Neural Network. In: Proc. IEEE on Machine Learning and Cybernetics (2005)
7. Leung, H., Lo, T., Wang, S.C.: Prediction of Noisy Chaotic Time Series Using an Optimal Radial Basis Function Neural Network. IEEE Trans. Neural Networks 12 (2001)
8. Mao, K.Z.: RBF Neural Network Center Selection Based on Fisher Ratio Class Separability Measure. IEEE Trans. Neural Networks 13 (2002)
9. Mak, M.W., Kung, S.Y.: Estimation of Elliptical Basis Function Parameters by the EM Algorithm with Application to Speaker Verification. IEEE Trans. Neural Networks 11 (2000)
10. Asuncion, A., Newman, D.: UCI Machine Learning Repository. School of Information and Computer Sciences. Univ. of California, Irvine (2007)

Application of Item Response Theory to Collaborative Filtering

Biyun Hu, Yiming Zhou, Jun Wang, Lin Li, and Lei Shen

School of Computer Science & Engineering, Beihang University
XueYuan Road No.37, HaiDian District, Beijing 100191, China
byhu8210@yahoo.com.cn, zhouyiming@buaa.edu.cn,
junwang8151@163.com, lilinbuaa@gmail.com,
lorashen@cse.buaa.edu.cn

Abstract. Although many approaches to collaborative filtering have been proposed, few have considered the data quality of the recommender systems. Measurement is imprecise and the rating data given by users is true preference distorted. This paper describes how item response theory, specifically the rating scale model, may be applied to correct the ratings. The theoretically true preferences were then used to substitute for the actual ratings to produce recommendation. This approach was applied to the Jester dataset and traditional k-Nearest Neighbors (k-NN) collaborative filtering algorithm. Experiments demonstrated that rating scale model can enhance the recommendation quality of k-NN algorithm. Analysis also showed that our approach can predict true preferences which k-NN cannot do. The results have important implications for improving the recommendation quality of other collaborative filtering algorithms by finding out the true user preference first.

Keywords: Collaborative filtering, Rating quality, Rating scale model, Item response theory, k-NN algorithm.

1 Introduction

Collaborative filtering (CF) is a popular technique used in recommender systems [1, 2, 3]. It works by collecting user ratings for items in a given domain. Although many CF approaches have been proposed [4, 5, 6], few have explored the effect of the rating error on the recommendation quality. From psychology point of view, when an item is presented to a user, preference for it must be first estimated and then a rating is typed according to the user's understanding of the rating scale. Each step of the process can influence the rating, so error comes about. What if it were possible to correct a user's rating to eliminate the error? Would the resulting rating improve the recommendation quality? This paper first analyses the actual ratings from Item Response Theory (IRT) point of view and then uses the rating scale model in IRT to correct the ratings. The theoretically true preferences are at last processed by k-NN algorithm [7] to produce recommendation.

The remainder of the paper is organized as follows. Section 2 introduces IRT, Rating Scale Model (RSM) and rating correcting; Section 3 provides the experiments investigating whether the theoretically true preference is a better representation of

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 766–773, 2009.

user preference and can improve recommendation quality; Section 4 provides analyses from our preliminary experiments; Section 5 summarizes our conclusions and presents future works.

2 Item Response Theory

IRT, which sees increasing applications in the fields of attitudes and interests measurement [8, 9, 10], is first developed in psychometrics as a complement of Classical Test Theory (CTT). It is about the application of mathematical models to data from questionnaires or tests for measuring latent traits, such as attitudes, interests, and etc. It is based on the idea that the probability of getting a test item *correct* is a function of person *ability* and item *difficulty*. As the rating data in recommend systems are attitude data for measuring interests, for ease of understanding, we use the terms provided in [8] with some modifications in this paper and explain the basic idea again, that is, the likelihood of a *like* response to an item is a function of user *interest* and item *agreeability*.

In the next section, we analyze the rating data with responses classified as *like* (value 1) or *dislike* (value 0), derive rating distribution function and explains the rationale of IRT simultaneously. More elaborate explanation can be found in [11].

2.1 Why Apply IRT?

Most of IRT models assume that person's responses to items attribute to only one latent trait θ (the unidimensionality hypothesis), that is, all items in a questionnaire or test are used to measure the same variable, whether it is interest or attitude or ability. They also assume that latent trait θ, theoretically can be any value in $(-\infty, +\infty)$, is a determinate value for a specific person. When showing a user *j* an item *i*, his latent response γ_i is an indirect measurement of his *interest* θ_j because measurement includes error. We formulate this as (1),

$$\gamma_i = \rho_i \theta_j + \varepsilon_i \quad . \tag{1}$$

where ρ_i is the linear correlation coefficient between γ_i and θ_j, ε_i the user's random error for item *i*. IRT hypothesizes that the user's response to item *i* is independent of his responses to other items (the local independence hypothesis), so whether the user will show a *like* response rests with if his response γ_i is bigger than a threshold γ_i^* pertaining to item *i*'s *agreeability*, namely, the probability of a *like* response is $p(\gamma_i > \gamma_i^*)$. IRT assumes that ε_i has a normal distribution with mean 0 and variance $1-\rho_i^2$, so γ_i also obeys normal distribution with mean $\mu_i = \rho_i \theta$ and standard deviation $\sigma_i = \sqrt{1-\rho_i^2}$. We then have the formulation (2), where $r_{j,i}$ is user *j*'s rating for item *i*.

$$p(r_{j,i}=1\mid\theta_j)=\int_{\gamma_i^*}^{+\infty}\frac{1}{\sigma_i\sqrt{2\pi}}e^{-(\gamma_i-\mu_i)^2/2\sigma_i^2}d_{\gamma_i}\ . \tag{2}$$

IRT defines item *agreeability* b_i and item *discrimination* a_i as shown in formula (3) and (4). Discrimination a_i represents the degree to which item *i* discriminates between users in different regions on the latent *interest* continuum. More explanation about a_i and b_i which is often called item difficulty can be found in [12].

$$b_i=\gamma_i^*/\rho_i\ . \tag{3}$$

$$a_i=\rho_i/\sqrt{1-\rho_i^2}\ . \tag{4}$$

We substitute (3) and (4) into (2) and then (2) is transformed to formulation (5).

$$p(r_{j,i}=1\mid\theta_j)=\int_{-\infty}^{a_i(\theta_j-b_i)}\frac{1}{\sqrt{2\pi}}e^{-\frac{t^2}{2}}d_t\ . \tag{5}$$

While further hypothesizing that all items have same *discrimination a* (the equal discrimination hypothesis) and set $\theta_j^{'}=a\theta_j, b_i^{'}=ab_i$, the two parameter normal ogive model [13] (5) is changed to one parameter normal ogive model (6).

$$p(r_{j,i}=1\mid\theta_j)=\int_{-\infty}^{\theta_j^{'}-b_i^{'}}\frac{1}{\sqrt{2\pi}}e^{-\frac{t^2}{2}}d_t\xleftrightarrow[b_i=b_i^{'}]{\theta_j=\theta_j^{'}}\int_{-\infty}^{\theta_j-b_i}\frac{1}{\sqrt{2\pi}}e^{-\frac{t^2}{2}}d_t\ . \tag{6}$$

Through assuming that the error arising from item response process obeys normal distribution, we obtain user's rating distribution function (6); theoretically true preference can then be computed. That's why we consider using IRT model to correct ratings.

2.2 Rasch Model

IRT was first developed using normal ogive models, such as (5) and (6) in section 2.1, however, the complex integral computation had limited its practicality at the time. Researchers then found that logistic model with simpler calculation can be good approximation to normal ogive model [14], so logistic model has gradually substituted the latter. Corresponding logistic model has been proposed for each normal ogive model. The one parameter logistic model [15], also named Rasch model, approximating one parameter normal ogive model has the formulation (7), where the parameters have the same meaning as they are in formula (6).

$$p(r_{j,i}=1\mid\theta_j)=\frac{e^{\theta_j-b_i}}{1+e^{\theta_j-b_i}}\ . \tag{7}$$

Intuitively, when user *interest* θ_j is equal to item *agreeability* b_i , user *j* has a probability of 0.5 to give a *like* response. The more θ_j is larger than b_i, the more probable *j* will give a *like* response.

2.3 Rating Scale Model

Rasch model used for dichotomous response categories is extended to rating scale model (RSM) dealing with polychotomous ordered response categories [16]. A simple summary of the derivation can be found in [17]. RSM used in this paper is formula (8),

$$p(r_{j,i} = x \mid \theta_j, b_i, \kappa, m) = \frac{e^{\kappa_x + x(\theta_j - b_i)}}{1 + \sum_{k=1}^{m} e^{\kappa_k + k(\theta_j - b_i)}} \quad . \tag{8}$$

where $r_{j,i}$, θ_j and b_i have the same meaning as they are in formula (7), x a user rating taking from successive rating categories set $\{0,1,2,\ldots,m\}$, and $\kappa_0 = 0$, $\kappa_x = -\sum_{k=1}^{x} \tau_k$, $x = 1,2,..,m\text{-}1$, $\kappa_m = 0$ the category coefficients expressed in terms of the ordered thresholds $\tau_1, \tau_{2,\ldots,} \tau_m$ which assumed by RSM to separate successive categories. RSM also assumes that the discriminations at the thresholds are all equal [17].

2.4 Rating Correcting Using RSM

After having the rating distribution function (8), the theoretically true preference can be further computed as (9).

$$E(r_{j,i}) = x * p(r_{j,i} = x \mid \theta_j, b_i, \kappa, m) \quad . \tag{9}$$

We use $r'_{j,i}$ to stand for the actual rating given by user *j* for item *i*, we then define *residual* as (10) to represent the effect of the errors arising from the rating process.

$$residual_{j,i} = r'_{j,i} - E(r_{j,i}) \quad . \tag{10}$$

3 Experiments

In this section we first describe the dataset and the experiments exploring the effect of rating correcting on the recommendation quality and then provide the experimental results.

3.1 Experimental Datasets

We use the user-joke continuous ratings (-10.00 to +10.00) from the Jester dataset [18]. The subset we random chose has ratings of 100 users for 100 jokes with some ratings missing. Since RSM can only deal with discrete ordered rating categories, the ratings was first rounded. We then added 10 to the ratings to obtain the ordered rating categories set {0, 1,…, 20}. Although when using IRT models, violation of the hypotheses mentioned in section 2.1 will not cause big problem[11, 19] recommends deleting the items having loading less than 0.3 on the first factor after factorial

analysis. We at last obtained 6280 ratings of 100 users for 88 items. 80% of the ratings are random selected into a training set and the remaining ratings into a test set.

3.2 Methods

We first analyzed the training set using Facets 3.22 [20], which estimates the parameters in RSM while giving the theoretically true preferences for the analyzed ratings. The true preferences were then used as the training ratings. The result was compared with the recommendation quality obtained using the actual ratings as the training ratings. The whole process is repeated 6 times with different training set. For simplicity, we used k-NN recommendation algorithm [7]. Average experiments results are given in the next section.

3.3 Results

With the two kinds of training ratings mentioned in section 3.2, we report the Mean Absolute Error (MAE) [21] in Fig.1. MAE corresponds to the average absolute deviation of predictions to the actual ratings in the test set. A smaller MAE value indicates a better performance. In Fig.1, RSMk-NN stands for the recommendation method using the theoretically true preferences as the training ratings and k is the number of neighbors. The maximum of k is set to be 30 because of the suggestion in [7].

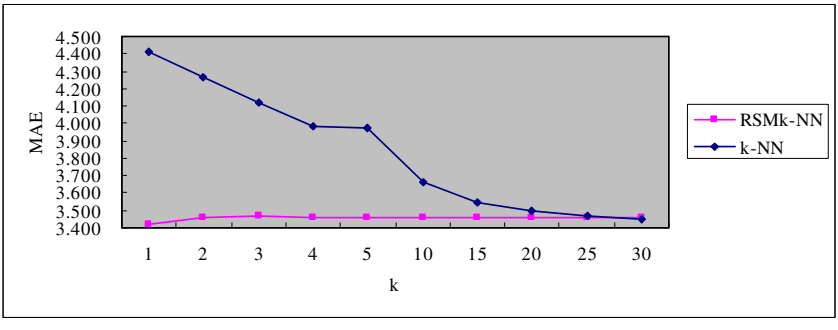


Fig. 1. Recommendation quality *MAE*s obtained using *RSMk-NN* and *k-NN*

From Figure 1, we see that RSM-kNN outperforms k-NN in all the experimental settings except k=30. We attribute the better performance of k-NN at k=30 to the effect of neighbors' number, as we can see clearly form Fig,1 that k influences k-NN more than RSMk-NN.

4 Discussion

In this section, we first discuss why RSMk-NN can outperform k-NN and then analyze the relationships between the two method's performance and user rating residual.

4.1 Why Can RSMk-NN Have Better Performance?

To explain the better performance of the RSMk-NN, we assume that *the neighbors found out by RSMk-NN are better.* We define the user-neighbors Distance as shown in formula (11) for evaluating neighbors' quality.

$$Distance = \frac{\sum_{j=1}^{u}\sum_{t=1}^{k} | \theta_j - \theta_{j_t} |}{uk} \quad . \tag{11}$$

In formula (11), u is the number of users in the test set, k the number of neighbors, θ_j user j's *interest* and θ_{j_k} user j's t-nearest neighbor's *interest*.

Upon the better neighbors assumption, we could infer that the *Distance* obtained using RSMk-NN will be smaller in all the experimental settings except k=30, that is, the neighbors found by RSMk-NN will be closer to the user on the latent *interest* continuum. The results are given in Fig.2, which confirm our inference.

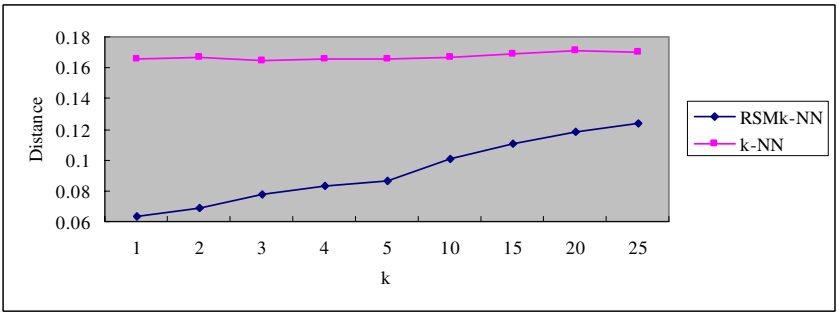


Fig. 2. User-neighbors *Distance*s obtained using *RSMk-NN* and *k-NN*

4.2 RSMk-NN and k-NN's Performance and Rating Residual

From psychology point of view, the ratings in the test set are also true user preferences with errors. Although we can get better prediction quality which we ascribe to better neighbors, we consider that *RSM-kNN should have even better performance for the actual ratings with less residual* (ref. formula (10)).

We then used the parameters given by Facets 3.22 [19] to compute the theoretically true preference and *residual* (ref. formula (9) and (10)) for each rating in the test set. MAEs obtained using RSMk-NN and k-NN for test ratings with |residual|< a (a was set to be 0.2, 0.4,…, 0.8, 2, 4,…, 10 respectively) are given in Fig.3. To eliminate the influence of different neighbors' number, k is set to be 30. The results show that RSMk-NN indeed has even better performance for the actual ratings with less residual.

RSMk-NN gets a very good performance with MAE equal to 0.214 when predicting test ratings with |residual|<0.2. From this point of view, we conclude that *RSMk-NN can predict true preferences*, which k-NN can not do due to lack of consideration of rating error.

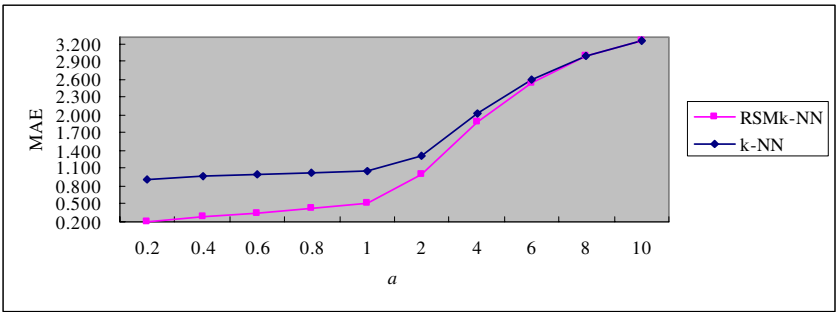


Fig. 3. *MAE*s obtained by *RSMk-NN* and *k-NN* using test ratings with |residual|<*a*

5 Conclusion and Future Work

Users' true preferences eliminated the effect of errors arising from the rating process can better predict their preferences in the future. In this paper, we proposed a recommendation method named RSMk-NN which used the rating scale model to obtain system users' theoretically true preferences. Experiments show that our RSMk-NN can outperform k-NN collaborative filtering algorithm. Experiments also demonstrate that RSMk-NN obtains better performance by finding out better neighbors. Further analysis indicates that RSMk-NN can predict true preferences. Our method has important implications for other collaborative filtering algorithms by considering the ratings quality first before giving recommendations.

In this paper, items were assumed with equal discrimination and unidimensional after deleting some of them. In future work, we would like to take different discrimination and multidimensionality into consideration, expecting that even better recommendation quality can be achieved.

Acknowledgments. We acknowledge the support of the National Basic Research Program of China under Grant No.2007CB310803.

References

1. Kim, J.H., Kang, U.G., Lee, J.H.: Content-Based Filtering for Music Recommendation Based on Ubiquitous Computing. In: Intelligent Information Processing III 2007. LNCS, vol. 228, pp. 463–472. Springer, Heidelberg (2007)
2. Golovin, N., Rahm, E.: Automatic Optimization of Web Recommendations Using Feedback and Ontology Graphs. In: Lowe, D.G., Gaedke, M. (eds.) ICWE 2005. LNCS, vol. 3579, pp. 375–386. Springer, Heidelberg (2005)
3. Amento, B., Terveen, L., Hill, W., Hix, D., Schulman, R.: Experiments in Social Data Mining: The Topicshop System. ACM Trans. Comput.-Hum. Interact. 10, 54–85 (2003)
4. Ko, S.-J., Lee, J.-H.: User Preference Mining through Collaborative Filtering and Content Based Filtering in Recommender System. In: Bauknecht, K., Tjoa, A.M., Quirchmayr, G. (eds.) EC-Web 2002. LNCS, vol. 2455, pp. 244–314. Springer, Heidelberg (2002)

5. Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A.: Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach. ACM Trans. Inf. Syst. 23(1), 103–145 (2005)
6. Lops, P., Degemmis, M., Semeraro, G.: Improving Social Filtering Techniques Through WordNet-Based User Profiles. In: Conati, C., McCoy, K., Paliouras, G. (eds.) UM 2007. LNCS, vol. 4511, pp. 268–277. Springer, Heidelberg (2007)
7. Herlocker, K.J., Borchers, A., Riedl, J.: An Algorithmic Framework for Performing Collaborative Filtering. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 230–237. ACM, New York (1999)
8. Kathy, E.: Applications of the Rasch Model to Evaluation of Survey Data Quality. New Directions for Program Evaluation 1996(70), 81–92 (1996)
9. Bradley, K., Sampson, S., Royal, K.: Applying the Rasch Rating Scale Model to Gain Insights into Students' Conceptualisation of Quality Mathematics Instruction. Mathematics Education Research Journal 18(2), 11–26 (2006) (Mathematics Education Research Group of Australasia, Australasia)
10. Meij, A., Kelderman, H., Flier, H.: Fitting a Mixture Item Response Theory Model to Personality Questionnaire Data: Characterizing Latent Classes and Investigating Possibilities for Improving Prediction. Applied Psychological Measurement 32, 611–631 (2008)
11. Yu, J.: Item Response Theory and its Applications. Jiang Su Educational Press, China (1992)
12. Item Response Theory, http://en.wikipedia.org/wiki/Item_response_theory
13. Lord, F.: A theory of test scores. Psychometric Society, New York (1952)
14. Birnbaum, A.: Some Latent Trait Models and Their Use in Inferring an Examinee's Ability. In: Lord, F.M., Novick, M.R. (eds.) Statistical Theories of Mental Test Scores. Addison-Wesley, Reading (1968)
15. Rasch, G.: On General Laws and the Meaning of Measurement in Psychology. In: Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, pp. 321–340. University of California Press, Berkeley (1961)
16. Andrich, D.: A Rating Formulation for Ordered Response Categories. Psychometrikia 43, 561–573 (1978)
17. Andrich, D.: An Extension of the Rasch Model for Ratings Providing Both Location and Dispersion Parameters. Psychometrikia 47, 105–113 (1982)
18. Goldberg, K., Roeder, K., Gupta, D., Perkins, C.: Eigentaste: A Constant Time Collaborative Filtering Algorithm. Information Retrieval 4, 133–151 (2001)
19. Armor, D.J.: Theta Reliability and Factor Scaling. In: Costner, H.L. (ed.) Sociological methodology, Jossey-Bass, San Francisco (1974)
20. Linacre, J.M.: A user's guide to FACETS: Rasch Measurement Computer Program. MESA Press, Chicago (2001)
21. Xue, G., Lin, C., Yang, Q., Xi, W., Zeng, H., Yu, Y., Chen, Z.: Scalable Collaborative Filtering Using Cluster-Based Smoothing. In: Proc. SIGIR, 2005, pp. 114–121. ACM, New York (2005)

Interactive Learning Neural Networks for Predicting Game Behavior

Qiaomei Sun, Guang Ren, and Xiaowei Qi

Marine Engineering College, Dalian Maritime University, Dalian 116026, China
xuanyuanhome@126.com, reng@dlmu.edu.cn, xiaowei0735@163.com

Abstract. Game theory is an interdisciplinary approach to the study of human behavior. Games describe a widely accepted framework for representing interactive decision-making. Artificial Neural Networks (ANNs) are universal approximators and have the ability of learning. Combining ANNs with game representation, we introduced a new architecture by which the learning abilities of ANNs are utilized to predict game behavior. Based on previous work, we investigated further the potential value of neural networks for modeling and predicting human interactive learning in repeated games. We conducted simulation studies based on the new model using experiments data which are provided by authors other than this paper. Through computer simulations and comparing with other models, we demonstrated that our model is superior in many respects to other models on ten experiments.

Keywords: Neural networks, Learning, Game, Prediction.

1 Introduction

Artificial Neural Networks (ANNs) contain a large number of simple processing units (artificial neurons) which are interconnecting widely to generate a self-learning, adaptive and self-organization of dynamic distributed parallel processor. Also the neural Network is a pattern of classified information processing system from which features can be extracted automatically. Due to high level of intelligence, high speed to identify, high rate of correct identification, ANNs have raised lots of concerns.

Hu X. and Wang J. presented a recurrent neural-network model for solving a special class of general variational inequalities (GVIs), which includes classical VIs as special cases [1]. It was proved that the proposed neural network for solving this class of GVIs could be globally convergent, globally asymptotically stable, and globally exponentially stable under different conditions. In 2008, a one-layer recurrent neural network with a discontinuous hard-limiting activation function was proposed for quadratic programming [2]. These contributions demonstrate that ANNs can be widely used in many fields.

Since the 1970s, economics has experienced a revolution of game theory. Game theory provides a valuable tool for many problems, but a variety of methods has limitations. In the 1970s and 1980s, researchers in economic psychology, such as Herbert Simon, challenged the strong assumption made by economic theorists that individual decision-making is purely rational. Economists responded with vigorous new lines of

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 774–783, 2009.

work that addressed many of these concerns. Since then, experimental and behavioral economics have flourished and have been honored with the Nobel Prizes awarded to Vernon Simith and Daniel Kahneman [3].

Recently, researchers used several modeling strategies to fit and predict how humans learn in repeated games in a laboratory setting, and great achievements were made. One modeling strategy extends a classical paradigm of learning theory (i.e., reinforcement learning) [4, 5] to games. The second strategy builds hybrid models that blend reinforcement learning with modeling the evolution of a player's beliefs about other players' moves: the relative weight of both learning processes depends on parameters that can be tuned, in turn, by experience [6, 7]. More recently, a model which emphasizes post-decision regret as the driver of learning has also been proposed [8]. As interest in neuroeconomics rising, a different modeling strategy adopting neural networks as models of human interactive behavior might be conceived. Recent advances in psychology are raising new challenges to economic assumptions. Marchiori and Warglien open a fresh avenue by which we can use models of neural networks to understand how humans learn as they make economic decisions [9].

Following the work made by Marchiori and Warglien [9], we establish a new structure of combining an artificial neural network with game behavior diagram. A conceived concept of 'regret' is used to establish learning strategies for neural networks. The payment function of the games is utilized to establish a 'regret' neural network model. We conducted simulation studies using Matlab on PC, and the classical experimental data are used for the simulation. We also investigate the human interactive learning behavior represented by mixed strategy of repeated games. The simulation results are analyzed and compared with that got by other methods. The results show that neural network technology can be used for simulation of game behavior.

This paper is organized as follows. The second part presents our learning model structure. The third part introduces the criteria we considered. The fourth part describes other established models for prediction in repeated games. The fifth part shows our computer simulation results and analysis. The sixth part concludes this paper.

2 The Learning Model

Modeling human interactive decision-making using neural networks is a new research direction. To explore the possibility of human interactive learning behavior, we conceived the network architecture and selected input information to the network that had to be both economically and neurophysiologically motivated.

The work presented here treats interactive learning which differs from individual learning in that, given n agents, each agent adapts to behaviors that are modified by the concurrent learning of the other n--1 agents. Interactive learning is our model's significant characteristic. Experimental game theory has provided a large set of laboratory data on human interactive learning in repeated games [10], often contradicting the predictions of standard game theory. The need for models of interactive learning in games arises from the difficulties of ordinary game-solution concepts to explain both the trajectories and the long-run stationary state of experimentally observed human behavior in repeated games. Because Nash equilibrium [11] not only fails to

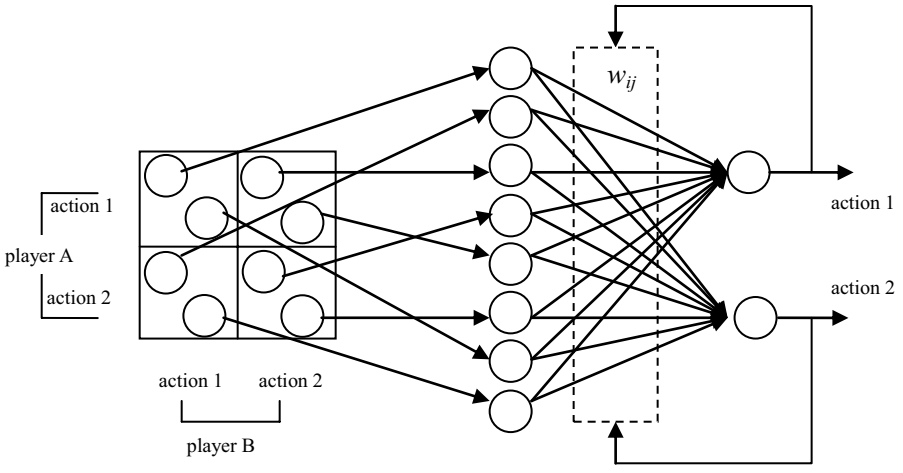


Fig. 1. The new architecture of regret model

approximate behavior in early rounds but also is often a poor predictor of the stable behavior emerging in the long run. In this paper, we focus on games with unique equilibria in mixed strategies.

We'd like to formulate a concise model using one of the most elementary learning neural network architectures: the simple (one-layered) analog perceptron [12, 13]. Our model maps the structure of a strategic game onto a neural network in a direct way. The model has an input node x_j corresponding to each payoff in the game utility matrix and the opponent's payoffs. The output node y_i presents for each action available to a player k. The learning model structure is plotted in Fig.1. The feedback process is modified according to some essential economic considerations (in accordance with both theoretical insights and empirical evidence). All of our research is based on a premise that learning is driven by a sort of 'ex-post' rationalizing process [14]: once individuals know what the other individual's move was, they modify their behavior by looking backward to what might have been their best move. They make adjustment in the direction of such an ex-post best response.

Recent neuroscience research on individual decision-making shows that regret affects learning and that both neurophysiological and behavioral responses to the experience of regret are correlated to its amplitude [15, 16].Thence we assume that the intensity of the adjustment is proportional to a measure of regret: how much they have missed by not playing such move. The input information is coded taking the value of the corresponding payoff in the current game's utility matrix; the output node activation is computed by summing up inputs to each output node weighted by the value of the incoming connections w_{ij} and transforming the summation via the hyperbolic log-*sig* activation function

$$y = \log sig\left(\sum_j w_{ij} x_j\right) \tag{1}$$

The activate function values of the output nodes can be viewed as propensities of actions. And after normalization, they are transformed into actual probabilities of play. Thus far, the learning model is a very conventional, simple analog perceptron, where learning is modeled, as usual, as adaptive updating of the connections' weights.

Connection weight adjustment we adopted is driven by a series of factors that can be summarized as adjustment is equal to the product of learning rate, distance from ex-post best response, regret and input saliency. As compared with Hopfield's perceptron rule, the main difference of this variant is that the error feedback is multiplied by the regret size. The update rule [12, 13] is shown below:

$$w_{ij}^{t} = w_{ij}^{t-1} + \Delta w_{ij} \tag{2}$$

given the action m chosen by player k, a_m^k

$$\Delta w_{ij} = \lambda \times \left[t_i\left(a^{-k}\right) - y_i \right] \times R^k\left(a_m^k, a^{-k}\right) \times x_j \tag{3}$$

where λ is the learning rate; $t_i\left(a^{-k}\right)$ is the ex-post best response of player k to the other players actions a^{-k}; y_i is its propensity to play action i; $R^k(\square)$ is the regret given the action a_m^k and other players' actions a^{-k} ; and x_j is presented by payoff saliency. Regret is computed as the difference between the actual payoff received by a player k and the maximum payoff obtainable, given other players' actions.

3 Criteria Consideration

In order to better compare the performances of different models with different degrees of complexity, we considered four measures: the Mean Square Deviation (MSD), Sum of Squared Residuals (SSR), the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC).The last two criteria take into account the number of free parameters which suggests that a measure of model error is penalized proportionally to the number of free parameters in the model [17].

MSD is one of the most accepted ways to measure the distance between the estimated and the observed vectors of subjects' choice frequencies [18]. Labeling with y the vector of the observed choice frequencies and with y' the vector of the estimated ones, MSD is defined as follows:

$$MSD = \frac{\sum_{i}^{N}\left(y_i^{'} - y_i\right)^2}{N} \tag{4}$$

As a series of empirically observed average frequencies of play, and estimated average frequencies of play are concerned, then we define:

$$SSR = \sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2 \tag{5}$$

as the Sum of Squared Residuals. In the case in which the residuals can be assumed normally distributed with a constant variance, then AIC and BIC can be computed as:

$$AIC = n \cdot \log\left(\frac{SSR}{n}\right) + 2 \cdot k \tag{6}$$

and

$$BIC = n \cdot \log\left(\frac{SSR}{n}\right) + k \cdot \log(n) \tag{7}$$

where k is the number of free parameters in the model.

For our analysis we use AIC and BIC defined above, since it is reasonable to assume that residuals are normally distributed. Indeed, residuals are defined as the differences between observed and estimated average frequencies of play, which are asymptotically normally distributed (Central Limit Theorem).

4 Competing Models

Established learning models in economics have two main interactional component processes: behavior is generated by some stochastic choice rule that takes as input raw 'propensities' to play actions, which can be transformed into probabilities to play the action; learning use feedback from the outcomes to modify propensities, which in turn affect subsequent choice.

In a typical economic learning model, choice is only a function of propensities. But here it is a combining function of propensities and the payoffs in the game utility matrix. We compared our model with several established ones, such as Basic Reinforcement Learning (BRL) Model(Erev et al.1998)[5], the one-layer analog perceptron model using ordinary error feedback measure(NNET), Erev and Roth's Reinforcement Learning (REL) Model (Erev et al. 2002)[19], Self Tuning Experience Weighted Attraction (stEWA) Model (Camerer, Ho and Chong, 2007) [8], Normalized Fictitious Play (NFP) Model (Ert and Erev 2007, Erev et al. 2007) [6, 9].Because we focus on the first two models, in the following, we just take(BR).and NNET for example.

In BRL Model, the reinforcement deriving from receiving payoff x is given by

$$RF(x) = x - x_{\min} \tag{8}$$

where $x_{\min}$is the minimum payoff.

Given that in period t the i-th agent has played her k-th pure strategy receiving a payoff x:

$$\text{if} \quad j=k \qquad a_{ij}(t+1) = a_{ij}(t) + RF(x) \tag{9}$$

$$\text{otherwise} \qquad a_{ij}(t+1) = a_{ij}(t) \tag{10}$$

The probability $p_{ik}(t)$ for player *i* to play her *k*-th pure strategy at time *t*, is defined as:

$$p_{ik}(t) = \frac{a_{ik}(t)}{\sum a_{ij}(t)} \tag{11}$$

In the first period, agent *i*'s initial attractions for her *j*-th and *k*-th pure strategies satisfy the condition: $a_{ik}(1) = a_{ij}(1)$, for all the possible combinations of *k*-th and *j*-th pure strategies.

Denoted with X_i the average absolute payoff for player *i*, the initial strength parameter is defined as follows:

$$s_i(1) = \frac{\sum a_{ij}(1)}{X_i} \tag{12}$$

This parameter is assumed to be equal for all players. Hence, player *i*'s initial propensities (attractions) are defined as:

$$a_{ij}(1) = p_{ij}(1) \cdot s(1) \cdot X_i \tag{13}$$

where $p_{ij}(1)$, the initial probability of choice, is given by $p_{ij}(1) = 1/M_i$, with M_i the number of all pure strategies of player *i*.

The model NNET consists in a traditional one-layer analog perceptron, where output units are fed back as usually by an error (target-output) measure:

$$\Delta w_{ij} = \lambda \cdot \beta \cdot \left[t_i\left(a^{-k}\right) - y_i \right] \cdot x_j \tag{14}$$

It is a two-parameter model with independent β and λ.

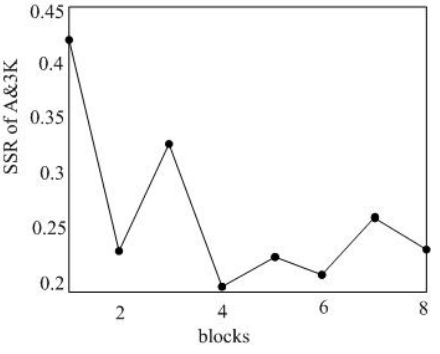
5 Result Analysis

To test the descriptive and predictive accuracy of our model, we made the program on Matlab running on computer and conducted simulation studies over previous data. We considered experiments on 10 different games with unique equilibria in mixed strategies [6, 20-22]. The games have at least two actions available to each player. For each one of the experimental conditions, the remaining 9 conditions have been used to estimate the free parameters values that minimize a Mean Square Deviation measure over such 9 conditions. The ensuing parameter estimates have subsequently been used to generate predictions over the left-out condition [5].

We discuss a class of games which have unique equilibria. Game theory lends a unique prediction of agents' behavior, providing a nonequivocal benchmark. Besides this, all the ten games that we discuss have nondegenerate solutions: In

Player 2 / Player 1	A2	B2
A1	(3,7)	(8,2)
B1	(4,6)	(1,9)

(a)



(b)

Fig. 2. S&A3k Game, payoff matrix(a) and prediction SSR Scores(b)

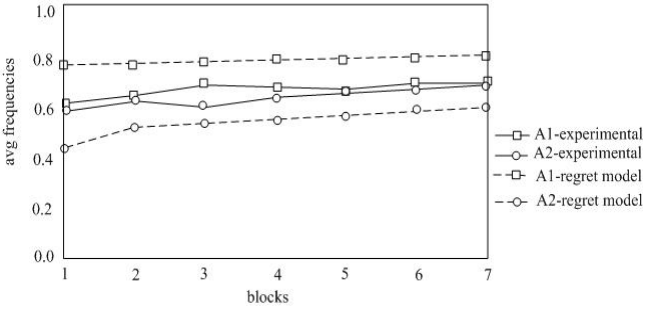


Fig. 3. Experimental and estimated frequencies of S&A3k Game

equilibrium, subjects have to randomize their behavior. Due to space constraints, we choose one out of the 10 games to present in this paper: Suppes and Atkison (1960) [6].

Experimental settings: this game was played by 20 pairs of subjects for 210 times.The payoff matrix was known to the subjects. The authors presented the data they gathered in 7 blocks of 30 repetitions of the stage game.

Payoff matrix of S&A3k Game and our model prediction SSR results are plotted in fig.2. Experimentally observed and estimated frequencies of choices are presented in fig.3.

We compared our model with different breeds of models: in particular, we took the Basic Reinforcement Learning (BRL) model [5] as competitors. To single out the value added by introducing a regret term in the perceptron feedback, we further compared our model with the corresponding one-layer analog perceptron (NNET) that uses the ordinary error feedback measure and has independent λ and β free parameters. The results are shown below:

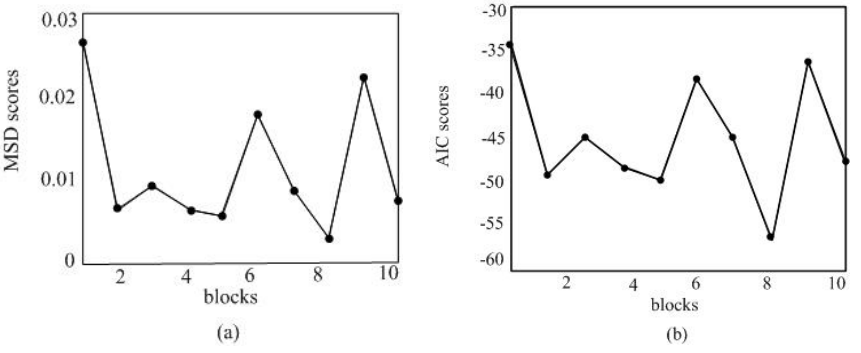


Fig. 4. MSD scores of regret-model (a) and AIC scores of regret-model (b)

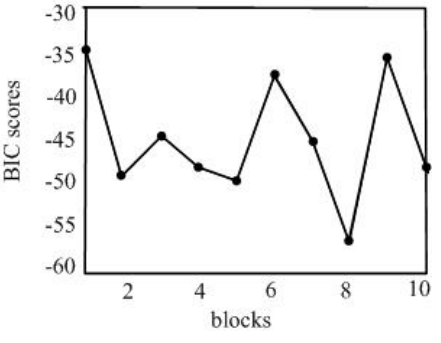


Fig. 5. BIC scores of regret model

Fig. 4 and fig.5 show prediction results of our model. To differentiate among all the models, we present our outcome in the following tables.

Table 1. MSD Scores for Predictions

	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	Avg MSD
BRL	0.420	0.131	0.490	0.342	0.240	0.188	0.500	0.415	0.405	0.288	0.3419
NNET	0.123	0.107	0.162	0.139	0.171	0.179	0.074	0.126	0.202	0.105	0.1388
Regret model	0.027	0.006	0.009	0.012	0.006	0.018	0.009	0.003	0.022	0.007	0.0118

Table 2. AIC Scores for Predictions

	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	Avg AIC
BRL	-5.803	-16.28	-4.422	-8.435	-10.85	-13.06	-4.238	-5.908	-6.135	-9.206	-8.434
NNET	-16.84	-18.10	-14.36	-16.17	-13.90	-13.50	-21.37	-16.67	-12.39	-18.29	-16.161
Regret model	-30.44	-44.19	-40.1	-40.01	-44.66	-34.15	-40.49	-50.58	-32.27	-43.19	-40.011

Table 3. BIC Scores for Predictions

	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	Avg-BIC
BRL	-5.606	-16.08	-4.225	-8.237	-10.65	-12.87	-4.041	-5.712	-5.938	-9.009	-8.237
NNET	-16.64	-17.90	-14.16	-15.96	-13.70	-13.30	-21.17	-16.47	-12.19	-18.09	-15.96
Regret model	-30.24	-43.99	-39.90	-39.81	-44.46	-33.96	-40.29	-50.39	-32.07	-42.99	-39.81

We number the 10 games as E1-E10. From table 1 to 3, the simulation results show that our model has the best MSD scores on the prediction tasks, and has better scores in most predicting tasks than that other two models have. Once the number of free parameters is taken into account, however, our model has the lowest AIC and BIC scores and is more favorable to other models in most of the games. Thus, no matter which measure performance is taken into account, our model is superior to the BRL model and NNET model.

Obviously, our model outperformed the traditional NNET analog perceptron. And this demonstrates the definitive role played by introducing regret as a source of feedback for learning. In other words, introducing regret in the feedback dramatically improved the performance of the neural network.

6 Conclusion

In this paper, we first formulated a framework of combining artificial neural network with game paradigm. We utilized the conceived concept of 'regret' to establish learning strategies for neural networks. The payment function of the games was used to establish a 'regret' neural network model. Using Matlab we made the program running on PC and conducted simulation studies according to the classical experimental data. Next, we investigated the human interactive learning behavior through the representation of mixed strategy of repeated games. Then, the results are analyzed and compared with that got by other methods. The experimental results show that neural network technology can be used for simulation of game behavior. Our method may provide a new avenue to both the artificial neural network application and the human interactive learning behavior research via game theory.

Acknowledgments. The work is supported by the Ministry of Communication of P.R. China (Grant#200332922505).

References

1. Hu, X., Wang, J.: A Recurrent Neural Network for Solving a Class of General Variational Inequalities. IEEE Transactions on Systems, Man and Cybernetics 37(3), 528–539 (2007)
2. Liu, Q., Wang, J.: A One-Layer Recurrent Neural Network With a Discontinuous Hard-Limiting Activation Function for Quadratic Programming. IEEE Transactions on Neural Networks 19(3), 558–570 (2008)
3. Cohen, M.D.: Learning with Regret. Science 319, 1052–1053 (2008)

4. Erev, I., Roth, A.: Predicting How People Play Games: Reinforcement Learning in Experimental Games with Unique, Mixed-strategy Equilibria. Am. Econ. Rev. 88, 848–881 (1998)
5. Erev, I., Roth, A., Slonim, L., Barron, G.: Learning and Equilibrium as Useful Approximations: Accuracy of Prediction on Randomly Selected Constant Sum Games. Econ. Theory 33, 29–51 (2007)
6. Camerer, C., Ho, T.H.: Experience-weighted Attraction Learning in Formal Games. Econometrica 67, 827–874 (1999)
7. Ho, T.-H., Camerer, C., Chong, J.-K.: Self-tuning Experience-weighted Attraction Learning in Games. J. of Econ. Theory 133, 177–198 (2007)
8. Ert, E., Erev, I.: Replicated Alternatives and the Role of Confusion, Chasing and Regret in Decisions from Experience. J. Behav. Decision Making 20, 305–322 (2007)
9. Marchiori, D., Warglien, M.: Predicting Human Interactive Learing by Regret-Driven Neural Networks. Science 319, 1111–1113 (2008)
10. Camerer, C.F.: Behavioral Game Theory: Experiments on Strategic Interaction. Princeton University Press, Princeton (2003)
11. Nash, J.F.: Equilibrium Points in N-Person Games. Mathematics 36, 48–49 (1950)
12. Hertz, J., Krogh, A., Palmer, R.G.: Introduction to the Theory of Neural Computation. Addison-Wesley, Redwood City (1991)
13. Hopfield, J.J.: Learning Algorithms and Probability Distributions in Feed-forward and Feed-back Networks. Proc. Natl. Acad. Sci. U.S.A. 84, 8429–8433 (1987)
14. Selten, R., Stöcker, R.: End behavior in Sequences of Finite Prisoner's Dilemma Supergames a learning Theory Approach. J. Econ. Behav. Organ. 7, 47–70 (1986)
15. Camille, N., et al.: The Involvement of the Orbitofrontal Cortex in the Experience of Regret. Science 304, 1167–1170 (2004)
16. Coricelli, G., et al.: Regret and Its Avoidance: a Neuroimaging Study of Choice Behavior. Nat. Neurosci. 8, 1255–1262 (2005)
17. Lee, D.: Neuroeconomics Best to Go with What You Know. Nature 441, 822–823 (2006)
18. Selten, R.: Axiomatic Characterization of the Quadratic Scoring Rule. Exp. Econ. 1, 43–61 (1998)
19. Erev, I., Roth, A., Slonim, L., Barron, G.: Predictive Value and the Usefulness of Game Theoretic Models. Int. J. of Forecasting 18, 359–368 (2002)
20. Lieberman, B., Malcom, D.: The Behavior of Responsive Individuals Playing a two- person, Zero Sum Game Requiring the Use of Mixed Strategies. Psychonomic Science 12, 373–374 (1965)
21. O'Neill, B.: Nunmetric Test of the Minimax Theory of Two Person Zerosum Games. Proc. Natl. Acad. Sciences U.S.A. 84, 2106–2109 (1987)
22. Avrahami, J., Guth, W., Kareev, Y.: Games of Competition in a Stochastic Environment. Theory and Decision 59, 255–294 (2005)

Asymmetric Learning for Pedestrian Detection Based on Joint Local Orientation Histograms

Junfeng Ge and Yupin Luo

Tsinghua National Laboratory for Information Science and Technology(TNList), Department of Automation,Tsinghua University, Beijing 100084, P.R. China
gejf03@mails.tsinghua.edu.cn, luo@tsinghua.edu.cn

Abstract. We present a cost-sensitive learning framework for pedestrian detection in still images based on the novel Joint Local Orientation Histograms (JLOH) features and the Asymmetric Gentle AdaBoost. The JLOH features capture the co-occurrence of local histograms and make it possible to classify the difficult examples. The proposed Asymmetric Gentle AdaBoost takes account of the situation that the rare positive targets have to be distinguished from enormous negative patterns in practical applications. The quantitative evaluation on the well-defined INRIA data set demonstrates the effectiveness of our methods.

Keywords: Joint local orientation histograms, Cost-sensitive learning, Pedestrian detection.

1 Introduction

Detecting pedestrians in still images or videos has received significant attention due to its crucial value in visual applications including surveillance, robotics, intelligent transportation, and human-computer interaction. However, large within-class variations caused by the wide range of human poses, different clothing and variable appearance, as well as varying backgrounds and imaging conditions, make this problem particularly challenging.

Generally, the computational approaches to object detection tasks can be divided into two components: feature extraction and learning. In feature extraction, good features which are informative, invariant to noise or some transformations will be extracted from the training samples. In learning process, effective classifiers or models are constructed from the samples. Thus, the pedestrian detection algorithm with high performance depends on both the discriminative features and the efficient learning algorithm.

In this paper, we present a cost-sensitive learning framework for pedestrian detection based on the novel feature called Joint Local Orientation Histograms (JLOH) and AdaBoost. The new feature takes advantage of the co-occurrence of the extended Histograms of Oriented Gradients(HOG) [1] features and captures more information than a single histogram, which makes it possible to construct an effective classifier. We also propose a Look-Up-Table(LUT)-type weak learner based on Weighted Fisher Linear Discriminant (WFLD) to adapt

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 784–793, 2009.

the boosting framework to vector-valued features. Moreover, the Asymmetric Gentle AdaBoost algorithm which takes account of the situation that the rare positive targets have to be distinguished from enormous negative patterns in most practical applications is applied to learn the cascaded pedestrian detector.

The rest of the paper is organized as follows: After reviewing the related work in Section 2, we introduce the JLOH features and cost-sensitive learning algorithm in Section 3 and 4. The results of the experimental study on INRIA database are presented in Section 5 and we conclude in Section 6.

2 Related Work

Many pedestrian detection approaches with different features and learning methods have been proposed in the literature. These features including edges, filter responses, wavelet coefficients, Haar-like features, edge orientation histograms, region covariance, edgelet, shapelet and affine invariant features can be distinguished into global features, local features, and key-points depending on how the features are measured.

Gavrila [2] proposes a hierarchy of global pedestrian silhouettes using Chamfer matching and the distance transform to compare the silhouettes with the image content. Zhao and Thorpe[3] apply a fully connected feed-forward neural network to high-pass filtered images for detecting pedestrians.

Regarding the local interest points and regions, Seemann et al. [4] present a detailed evaluation about the performance of various interest point detectors and different shape descriptors for pedestrian detection. Leibe et al. [5] start with a local feature detection to generate a set of pedestrian hypothesis. The detection results come from a top-down verification step using the segmentation masks and Chamfer matching. The improved version is presented in [6].

Local features based approaches become more popular after Viola and Jones [7] proposed their successful detection framework for face detection using AdaBoost and Haar-like features. However, the Haar-like features only using intensity information encounter difficulties in detecting people in cluttered scene.

Recently, features based on edge orientations have been shown effective for pedestrian detection. Dalal and Triggs [1] design a descriptor using a dense grid of normalized HOG features computed over blocks of predefined size and obtain promising performance with the help of SVM. Zhu et al. [8] further speed up this approach by integrating a boosted cascade with the HOG features of variable-size blocks, while maintaining a similar accuracy level. Similar framework is proposed by Ivan Laptev [9] with more patterns of subregions and a different weak learner, which achieves better performance on PASCAL VOC 2005 data set. Another local feature named shapelet [10] adopts multi-layer AdaBoost to explore the distinctive features from local gradients. And region covariance [11] which combines pixel coordinates, intensity, higher order derivatives and orientation etc. into a covariance matrix can be used in detection, tracking, matching and texture classification tasks. The experimental evaluation of several local features can be found in [12].

Besides selecting more distinctive features, an alternative improvement can be done on the learning methods, especially the efficient boosting algorithm. There are Real AdaBoost [13], KLBoost [14], FloatBoost [15] and Vector Boosting [16] for object detection. Moreover, many cost-sensitive extensions of discrete AdaBoost [17,18] have been proposed to deal with the asymmetric problem in detection task that needs to distinguish the rare positive objects from a large number of negative patterns.

3 Joint Local Orientation Histograms

Feature co-occurrence, which captures the characteristics of objects, makes it possible to construct a more powerful classifier [16,19]. In this section, we focus our work on the feature level to learn discriminative features based on the combinations of the basic local orientation histograms (LOH).

3.1 LOH Features

We define the LOH feature F as an ensemble of subregion based statistics about gradient magnitude in several orientations like HOG, but with more patterns of local regions, which are called templates.

$$F(x_t, y_t, w_t, h_t, K, T_i(n, R_+, R_-)) = (f_1^1, \ldots, f_1^k, \ldots, f_2^1, \ldots, f_n^K)^T \ . \quad (1)$$

where (x_t, y_t, w_t, h_t) denotes the template position, K is the orientation bin number. $T_i(n, R_+, R_-)$ represents the i-th template with n histograms and R_+ indicates the *white* region with positive magnitude, R_- indicates the *gray* region with minus magnitude.

$$f_j^k = \sum \psi_k(x, y), (x, y) \in R_j \ . \quad (2)$$

where R_j is the labeled region in the template for calculating the j-th histogram.

$$\psi_k(x, y) = \begin{cases} g(x, y) & \text{if } \theta(x, y) \in bin_k \text{ and } (x, y) \in R_+ \\ -g(x, y) & \text{if } \theta(x, y) \in bin_k \text{ and } (x, y) \in R_- \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where $g(x, y)$ and $\theta(x, y)$ are the strength and orientation of the gradient at the point (x, y).

There are eleven different templates presented in Figure 1, which contain the HOG features proposed by [8] and [9]. Moreover, the ratio between w_t and h_t is not limited. In our implementation, $w_t = 12, 14, \ldots, 32$, $h_t = 12, 16, 20, \ldots, 84$, and the size of the reference detection window is 32×84. Thus, if the scanning steps are 2 pixels in width and 4 pixels in height, the whole basic feature pool contains 137940 local histograms, which are over-completed for the 2688-dimensional space. This redundant basic feature set allows us to build up compact and powerful JLOH.

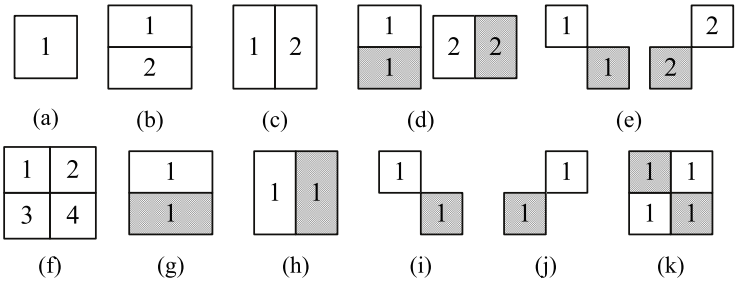


Fig. 1. The 11 templates for extracting local orientation histograms. The number indicate the region R_j. The white region denotes R_+, and the gray region refers to R_-.

3.2 The Metrics for Feature Selection

Having defined the feature set for further combination, the second work is to select proper distinctive features from the pool, because the conventional brute-force method that searches all possible combinations exhaustively can not be accomplished in limited training time.

Regarding the feature selection methods [20], we can utilize the class separability of weighted Fisher discriminant analysis (WFLD) and the Kullback-Leibler (KL) divergence to measure the discrimination of features. The class separability of WFLD is defined as follows [9]:

$$WFLD(\mathbf{w}) = \frac{(\mathbf{w}^t\mathbf{m}_+ - \mathbf{w}^t\mathbf{m}_-)^2}{\mathbf{w}^t(S_+ + S_-)\mathbf{w}}, \text{with } \mathbf{w} = (S_+ + S_-)^{-1}(\mathbf{m}_+ - \mathbf{m}_-) \ . \quad (4)$$

where $\mathbf{m}_+, \mathbf{m}_-$ are the weighted class means and S_+, S_- are the weighted class covariance matrices. The KL divergence is based on the projected distribution of the positive and negative samples. Here, we use $\mathbf{w}$ from WFLD and histograms to compute the symmetric KL divergence as in [14]:

$$KL(\mathbf{w}) = \int [h_k^+(\mathbf{w}^T x) - h_k^-(\mathbf{w}^T x)] \log \frac{h_k^+(\mathbf{w}^T x)}{h_k^-(\mathbf{w}^T x)} d\mathbf{w}^T x \ . \quad (5)$$

Considering the training process of AdaBoost, the normalization factor of sample weights $Z_t = 2\sqrt{\epsilon_t(1-\epsilon_t)}$ is a natural measure of the informativeness of a single feature. So, we can defined the third criterion based on Z_t and $\mathbf{w}$ during the AdaBoost learning process with a single feature.

$$B(\mathbf{w}) = 1 - Z_t \ . \quad (6)$$

Finally, we select a subset containing 100 best local histograms to lean the JLOH features. And the performance of the above three criterions will be compared in the experiments.

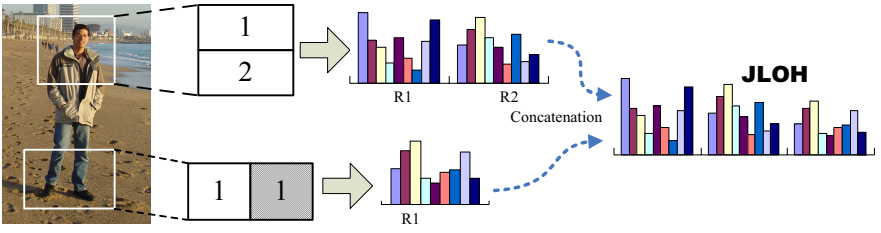


Fig. 2. The mtehod to combine the two local orientation histograms

3.3 The Combination of Local Features

After selecting a subset of the basic feature pool, the final step for building up JLOH is to combine the single local orientation histograms into more discriminative features. For the sake of computational efficiency in feature extraction, we only take account of the combination of TWO different histograms.

The way to form the JLOH features that capture the co-occurrence of spatially separated histograms is shown in Figure 2, which directly concatenates the two histograms as the new feature vector.

However, measuring how informative a combined JLOH feature is not as straightforward, we adopt brute-force search mehtod for selecting the best JLOH features by minimizing the weighted error in each round of the boosting process. Moreover,the normalization of JLOH is retained to improve its robustness.

4 Cost-Sensitive Learning

The cost-sensitive learning has attracted more and more attention in recent years because the asymmetric probability for observing objects and non-objects is common in many practical application [17,18]. In this section, we will derive the Asymmetric Gentle AdaBoost algorithm from minimizing the expectation of cost-sensitive exponential loss function [21] and construct the cascaded pedestrian detector following the way described in [22] .

4.1 Asymmetric Gentle AdaBoost

From a statistical perspective, AdaBoost acts as a method for fitting an additive model $F(x) = \sum_j f_j(x)$ in a forward stage-wise manner. Both discrete and real-valued AdaBoost can be derived in the form of additive logistic regression by minimizing the expectation of exponential loss of training samples (7), which is proved to be an upper bound on misclassification error and equal to the product of every normalization factor Z_t [23].

$$R(F(\mathbf{x})) = E\left[e^{-yF(\mathbf{x})}\right] = \sum_{i=1}^{n} w_i^0 \exp(-y_i F(\mathbf{x}_i)) \geq \frac{1}{n}\sum_{i=1}^{n} [\![y_i \neq H(\mathbf{x}_i)]\!] \,. \quad (7)$$

where $H(\mathbf{x}) = \text{sign}(F_T(\mathbf{x})) = \text{sign}\left(\sum_{j=1}^{T} f_j(\mathbf{x})\right)$ is the final strong classifier, and $\sum_{i=1}^{n} w_i^0 \exp(-y_i F(\mathbf{x}_i)) = \prod_{t=1}^{T} Z_t$.

Consequently, different loss functions lead to different boosting algorithms. In order to obtain the cost-sensitive extension of Gentle AdaBoost, we consider an asymmetric loss,with a cost of C_1 for false rejected samples and C_2 for false accepted samples.

$$Loss_{asym}(x_i, y_i) = \begin{cases} C_1 \text{ if } y_i = 1 \text{ and } H(x_i) = -1 \\ C_2 \text{ if } y_i = -1 \text{ and } H(x_i) = 1 \\ 0 \quad \text{otherwise} \end{cases} . \tag{8}$$

Obviously, the upper bound of the above asymmetric loss is held by the exponential loss defined in (9), where $C_1, C_2 \in [0 \quad 1]$.

$$ALoss(F(\mathbf{x})) = \begin{cases} e^{-C_1 y_i F(\mathbf{x}_i)} \text{ if } y_i = 1 \\ e^{-C_2 y_i F(\mathbf{x}_i)} \text{ if } y_i = -1 \end{cases} . \tag{9}$$

Therefore, the Asymmetric Gentle AdaBoost can be derived from minimizing the cost function in (10).

$$R_{asym}(F(\mathbf{x})) = E\left[I(y=1)e^{-yC_1F(\mathbf{x})} + I(y=-1)e^{-yC_2F(\mathbf{x})}\right] . \tag{10}$$

where $I(\cdot)$ is the indicator function.

Suppose a current hypothesis $F((x))$ has been obtained in the additive model, the next step is to learn an optimal weak classifier $f(\mathbf{x})$ to add in. Thus, the overall training risk turns into

$$\begin{aligned} R_{asym}(F(\mathbf{x}) + f(\mathbf{x})) &= E\left[I(y=1)w(\mathbf{x},y)e^{-C_1 f(\mathbf{x})} + I(y=-1)w(\mathbf{x},y)e^{C_2 f(\mathbf{x})}\right] \\ &= E_w\left[I(y=1)e^{-C_1 f(\mathbf{x})} + I(y=-1)e^{C_2 f(\mathbf{x})}\right] \\ &= P_w(y=1|\mathbf{x})e^{-C_1 f(\mathbf{x})} + P_w(y=-1|\mathbf{x})e^{C_2 f(\mathbf{x})} \end{aligned} \tag{11}$$

where $w = w(\mathbf{x}, y) = e^{-yCF(\mathbf{x})}$, $C = C_1$ if $y = 1$, $C = C_2$ if $y = -1$. $E_w[\cdot]$ is the weighted expectation defined by

$$E_w\left[g(\mathbf{x}, y)\right] = \frac{E\left[w(\mathbf{x}, y)g(\mathbf{x}, y)\right]}{E\left[w(\mathbf{x}, y)\right]} \tag{12}$$

By minimizing (11), the optimal weal hypothesis of Asymmetric Gentle AdaBoost can be obtained using Newton steps as is done by Gentle AdaBoost and described as follows:

$$f(\mathbf{x}) = \frac{C_1 P_w(y=1|\mathbf{x}) - C_2 P_w(y=-1|\mathbf{x})}{C_1^2 P_w(y=1|\mathbf{x}) + C_2^2 P_w(y=-1|\mathbf{x})} \tag{13}$$

And the algorithm use $w_i^{(t+1)} = w_i^{(t)} e^{-Cyf(\mathbf{x})}$ for updating weights.

4.2 Domain-Partition-Based Weak Learner

The piece-wise function [13,16] is a straightforward realization of the domain partition based weak classifiers that are required for Asymmetric Gentle AdaBoost learning, which can be implemented by look up table (LUT) easily.

As the JLOH features are not one-dimensional, the WFLD described in Section 3.2 is used for dimension reduction. Therefore, the weak hypothesis in (13) can be realized as follows.

$$f(\mathbf{x}) = \sum_{k=1}^{n} \frac{C_1 \overline{W}_{+1}^k - C_2 \overline{W}_{-1}^k}{C_1^2 \overline{W}_{+1}^k + C_2^2 \overline{W}_{-1}^k} I_n^k(\mathbf{w}^t \mathbf{x}) \ . \tag{14}$$

where $\mathbf{w}$ is the direction for WFLD, $\overline{W}_{+1}^k$ and $\overline{W}_{-1}^k$ are the estimated likelihood in bin_k. $bin_k = [(k-1)/n, k/n), k = 1, \ldots, n$. $D_t(i)$ is the weight of each sample.

$$\overline{W}_l^k = P(\mathbf{w}^t \mathbf{x} \in bin_k, y = l) = \sum_{i: \mathbf{w}^t \mathbf{x}_i \in bin_k \bigwedge y_i = l} D_t(i), \quad l = \pm 1 \ . \tag{15}$$

$I_n^k(u)$ is a indicator function. And in our experiments, the LUT size $n = 32$.

5 Experimental Results

We evaluate the proposed algorithms on the well-defined INRIA data set [1], in which, the training set consists of 2416 cropped pedestrian images (with left-right reflections), 1237 labeled pedestrians and 1218 person-free images. Meanwhile, there are 288 images containing 589 labeled pedestrians ($566 \times 2 = 1132$ cropped images) and 453 images without pedestrians in the testing set.

5.1 Evaluation Criteria

The Receiver-Operator-Characteristic (ROC) curve is a common way to measure the discriminability of a given classifier, however, it requires a set of well-cropped or resized object images. So does the Detection Error Tradeoff (DET) curve proposed by [1]. This requirement prevents them from evaluating the detector's localization ability. Therefore, we use Recall-Precision (RP) curve to quantifies the different detectors's performance and adopt the evaluation criteria and software realized in PASCAL VOC challenge Development Kit. A detection is considered correct, if the area of overlap between the predicted bounding box B_p and ground truth bounding box B_{gt} exceeds 50% by the formula $\frac{\text{area}(B_p \cap B_{gt})}{\text{area}(B_p \cup B_{gt})} > 0.5$. We use the same post-processing method as [9] and the VOC development kit to generate the RP curves in the following experiments.

5.2 Evaluation and Comparison

For convenience, we randomly select 1000 cropped positive samples and 1000 person-free images from the labeled training set of INRIA data set as the smaller set for learning the proposed algorithms with different parameters.

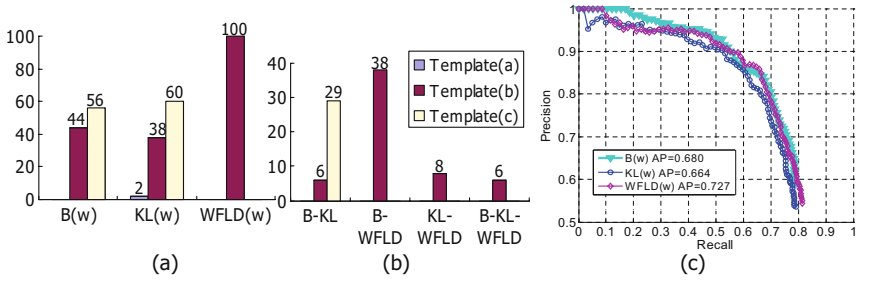


Fig. 3. The type distribution of the selected 100 local orientation features and the corresponding performance. The features from template(f) are excluded, and only features from template(a)(b)and(c) are selected on the first round.

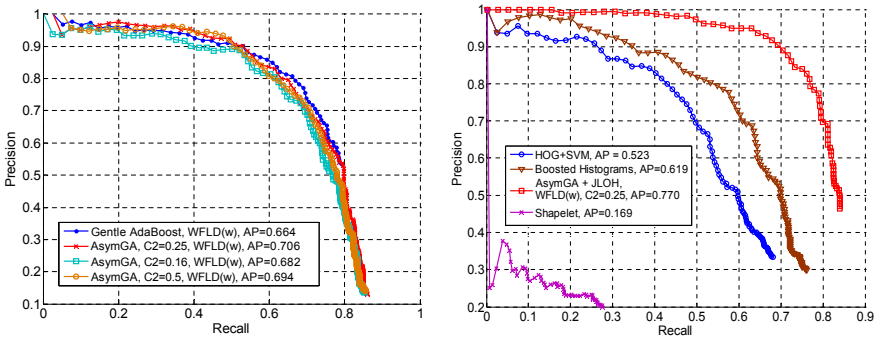


Fig. 4. The performance of Asymmetric Gentle AdaBoost. (a)Trained on the small set. (b)Performance comparison with the results from [1], [9] and [10].

The distribution of the 100 best local orientation histograms' types, which are selected by three different rules on the first round learning, are shown in Figure 3(a). The local features from template(f) are excluded in the test because they are more informative than others. The type distribution of the same features selected by two or three rules simultaneously is presented in Figure 3(b). It shows that the features selected by (4) and (5) are very different. Features selected by (6) share most with others. The performance of the detectors trained on the three different subsets is given in Figure 3(c), where the AP means average precision. The worst result comes from the subset selected by (5).

In order to evaluate the performance of the Asymmetric Gentle AdaBoost, we set the parameters $C_1 = 1, C_2 = 0.5, 0.25, 0.16$, and train the different detectors on the JLOH features selected by (4) from the smaller training set. The results in Figure 4(a) confirm that the proposed cost-sensitive algorithm can improve the performance, reaching higher recall rate, and the best parameters are $C_1 = 1, C_2 = 0.25$. All these detectors and above ones consist of 14 stages, trained with the same parameters for cascade, but contain different number of weak learners.

For comparing our detectors with others, like HOG+SVM [1],Shapelet [10] and Boosted Histograms [9] detectors, we train our detectors with different parameters on the whole training set of INRIA database. And the results of other detectors are obtained by their provided binaries or source code. In addition, we adjust the output of HOG+SVM and Shapelet detectors to make them achieve the best statistical results. The results of our best detector based on JLOH, and others are shown in Figure 4(b). The proposed algorithm greatly outperforms the state-of-the-art methods. And the poor performance of shapelet may be caused by the serious error in its implementation [24].

6 Conclusion

We present a cost-sensitive learning framework for pedestrian detection that combines Asymmetric Gentle AdaBoost and JLOH features to improve the generalization performance. The Asymmetric Gentle AdaBoost algorithm takes account of the asymmetric situation that the rare positive targets have to be distinguished from enormous negative patterns in practical applications. The JLOH features take advantage of the co-occurrence of the extended HOG features, and make it possible to classify the difficult examples that are misclassified by weak classifiers using a single feature. We also propose a LUT-type weak learner based on Weighted Fisher Linear Discriminant to adapt the boosting framework to vector-valued features. The experimental results evaluated on the INRIA data set demonstrate that the proposed approaches have superior performance compared to the state-of-the-art methods on the pedestrian detection task.

References

1. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 886–893. IEEE Press, New York (2005)
2. Gavrila, D.M.: Pedestrian Detection from a Moving Vehicle. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1843, pp. 37–49. Springer, Heidelberg (2000)
3. Zhao, L., Thorpe, C.: Stereo and Neural Network-Based Pedestrian Detection. IEEE Trans. on Intelligent Transportation Systems 1, 148–154 (2000)
4. Seemann, E., Leibe, B., Mikolajczyk, K., Schiele, B.: An Evaluation of Local Shape-Based Features for Pedestrian Detection. In: Proc. British Machine Vision Conference, pp. 949–958. BMVA Press, Worcs (2005)
5. Leibe, B., Seemann, E., Schiele, B.: Pedestrian Detection in Crowded Scenes. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 878–885. IEEE Press, NewYork (2005)
6. Mikolajczyk, K., Leibe, B., Schiele, B.: Multiple Object Class Detection with a Generative Model. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 26–36. IEEE Press, New York (2006)
7. Viola, P.A., Jones, M.J.: Rapid Object Detection Using a Boosted Cascade of Simple Features. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 511–518. IEEE Press, New York (2001)

8. Zhu, Q., Yeh, M.C., Cheng, K.T., Avidan, S.: Fast Human Detection Using a Cascade of Histograms of Oriented Gradients. In: Proc. IEEE Conf. Computer Vision And Pattern Recognition, pp. 1491–1498. IEEE Press, New York (2006)
9. Laptev, I.: Improvements of Object Detection Using Booste Histograms. In: Proc. British Machine Vision Conference, pp. 949–958. BMVA Press, Worcs (2006)
10. Sabzmeydani, P., Mori, G.: Detecting Pedestrians by Learning Shapelet Features. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 1–8. IEEE Press, New York (2007)
11. Tuzel, O., Porikli, F., Meer, P.: Region Covariance: A Fast Descriptor for Detection and Classification. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 589–600. Springer, Heidelberg (2006)
12. Munder, S., Gavrila, D.M.: An Experimental Study on Pedestrian Classification. IEEE Trans. on Pattern Analysis and Machine Intelligence 28, 1863–1868 (2006)
13. Wu, B., Ai, H., Huang, C., Lao, S.: Fast Rotation Invariant Multi-View Face Detection Based on Real Adaboost. In: Proc. IEEE Int. Conf. Automatic Face and Gesture Recognition, pp. 79–84. IEEE Press, New York (2004)
14. Liu, C., Shum, H.: Kullback-Leibler Boosting. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 587–594. IEEE Press, New York (2003)
15. Li, S., Zhang, Z.: Floatboost Learning and Statistical Face Detection. IEEE Trans. on Pattern Analysis and Machine Intelligence 26, 1112–1123 (2004)
16. Huang, C., Ai, H., Li, Y., Lao, S.: High-Performance Rotation Invariant Multiview Face Detection. IEEE Trans. on Pattern Analysis and Machine Intelligence 29, 671–686 (2007)
17. Viola, P.A., Jones, M.J.: Fast and Robust Classification Using Asymmetric Adaboost and a Detector Cascade. In: 14th Conf. on Advances in Neural Information Processing Systems, pp. 1311–1318. MIT Press, Cambridge (2002)
18. Hou, X., Liu, C.L., Tan, T.: Learning Boosted Asymmetric Classifiers for Object Detection. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 330–338. IEEE Press, New York (2006)
19. Mita, T., Kaneko, T., Hori, O.: Joint Haar-Like Features for Face Detection. In: Proc. IEEE Int. Conf. on Computer Vision, pp. 1619–1626. IEEE Press, New York (2005)
20. Dollar, P., Tu, Z., Tao, H., Belongie, S.: Feature Mining for Image Classification. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1–8. IEEE Press, New York (2007)
21. Masnadi-Shirazi, H., Vasconcelos, N.: Asymmetric Boosting. In: Proc. 24th Int. Conf. on Machine Learning, pp. 609–619. ACM Press, New York (2007)
22. Xiao, R., Zhu, L., Zhang, H.: Boosting Chain Learning for Object Detection. In: Proc. IEEE Int. Conf. on Computer Vision, pp. 709–715. IEEE Press, New York (2003)
23. Friedman, J., Hastie, T., Tibshirani, R.: Additive Logistic Regression: A Statistical View of Boosting. The Annals of Statistics 28, 337–374 (2000)
24. Sabzmeydani, P., Mori, G.: http://www.cs.sfu.ca/~mori/research/papers/sabzmeydani_shapelet_cvpr07.html

Using Strongly Connected Components as a Basis for Autonomous Skill Acquisition in Reinforcement Learning

Seyed Jalal Kazemitabar and Hamid Beigy

Intelligent Systems Lab. Computer Eng. Department
Sharif Univeristy of Technology
{jkazemitabar,beigy}@ce.sharif.edu

Abstract. Hierarchical reinforcement learning (HRL) has had a vast range of applications in recent years. Preparing mechanisms for autonomous acquisition of skills has been a main topic of research in this area. While different methods have been proposed to achieve this goal, few methods have been shown to be successful both in performance and also efficiency in terms of time complexity of the algorithm. In this paper, a linear time algorithm is proposed to find subgoal states of the environment in early episodes of learning. Having subgoals available in early phases of a learning task, results in building skills that dramatically increase the convergence rate of the learning process.

Keywords: skill acquisition, hierarchical reinforcement learning, strongly connected components.

1 Introduction

During the last decade, Hierarchical Reinforcement Learning (HRL) has shown quite successful in solving a wide range of problems in machine learning and beyond. This variety incorporates applications such as robotics, spoken dialogue management, decision making and control engineering. There are three fundamental frameworks to this area: Parr's HAM method [9], Sutton's Options formalism [14], and Dietterich's MAXQ framework [4].

Although different in formulation, all the hierarchical frameworks have two major contributions in common. Firstly, they reduce the curse of dimensionality in problems. As an example [15] has used options to solve a mission planning task including a state-action space of nearly 25 billion elements. By injecting domain-specific knowledge through the use of temporal abstraction, they reduced the flat RL problem to a hierarchical one with less than a million elements. As described in [4] state abstraction -or the abstraction resulted by ignoring some aspects of the state of the environment- can also greatly ease problem resolution. The second affecting involvement of hierarchical methods is that they significantly increase the agent's efficiency in solving problems. For an illustration consider the four-room gridworld problem described in [15] where the agent achieves its

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 794–803, 2009.

goals much faster when he knows how to get to the doorway. Room to room navigation is made easy with the aid of hand-made options.

A temporally-extended action, or a skill, is a closed-loop policy over one-step actions [11]. The primary question that rises is how an agent can find useful skills autonomously. In other words, how can an agent use HRL without being supported with predefined macro-actions? The skills formed from the agent's current experience in an environment can be applied to domains with familiar characteristics even with different rewarding functions. Moreover, they allow more efficient exploration of the state space by providing more direct access to those regions that the agent does not tend to go to easily [10].

In recent years a number of methods have been proposed to find proper skills for an RL agent. Most of these methods have focused on the idea of finding subgoals. Arriving at goal incorporates going through some milestones. A skill in this regard is a hierarchical structure that leads the agent to a useful subgoal. Thus, to acquire skills one should discover subgoals of interest. McGovern in [7] characterizes subgoals as states frequently visited in successful trajectories but never visited in unsuccessful ones (for a proper definition of success). Some other methods look for subgoals by discussing the frequency of visitations of each state [12,2]. Asadi in [1] uses Monte Carlo sampling to analyze a learned policy to discover subgoals for similar tasks. Others search for states that are situated between densely-connected areas of the state space. [10] represents the problem of finding subgoals as a pattern classification issue. A state in this respect is a target (subgoal) or otherwise a non-target object. Furthermore, there are methods that use the observations to form a graphical view of the discovered environment. Most of such approaches use graph partitioning techniques and select the connecting states as subgoals [11,6,8].

Graph-based approaches, although successful in finding useful subgoals, mostly suffer from the intrinsic NP-hardness of the partitioning problem. This has imposed the use of approximation techniques which result in either loosing precision of the initially suggested concept or incurring a computational cost to the online acting agent.

In this paper, we present a new subgoal-based method for discovering skills in RL. Similar to [8,6,10], our subgoals of achievement are states that connect densely-connected regions of the observed state space. A directed graph is called *strongly connected* if there is a path from each vertex to every other one. The strongly connected components (SCCs) of a directed graph are its maximal strongly connected subgraphs.We search for SCCs in the local transition graphs collected during the previous episodes to find the cutting edges of regions. However, our subgoals are encountered with an exact linear algorithm. The subgoals are then used to form skills which are used by the agent to attain the goal. Meanwhile, our algorithm needs fewer episodes than many others to find the subgoals. Additionally, since our algorithm is defined independently of the reward function, it does well in solving many other tasks defined in the same environment. In the discussion section we present a comparative study of these subgoal-based algorithms.

The paper begins (in section 2) with an introduction to options framework as the used hierarchical formalism. Section 3 then introduces the strongly connected components concept and describes our algorithm in detail. In section 4 we review the empirical results and finally conclude with a comparison of different algorithms and future directions.

2 The Options Framework

We used the options framework to represent our skills. According to [14] a (Markov) option $\mathcal{O}$ is a triplet $\langle \mathcal{I}, \pi, \beta \rangle$ in which $\mathcal{I} \subseteq \mathcal{S}$ is a subset of states in which an option can start. The second component $\pi : \mathcal{S} \times \mathcal{A} \mapsto [0, 1]$ is the policy that constructs the behavior of the agent. This function which is defined over the states in which the option can execute, identifies the action to be chosen in each state in case of running the option. As the finishing condition on an option, a function $\beta : \mathcal{S} \mapsto [0, 1]$ is used to characterize the probability of the option termination in each state. Traditionally, this function is defined to be 1 for states in which the macro-action should finish and 0 in other regions of the state space. The options framework provides methods for learning and planning using options as temporally extended actions in the standard reinforcement learning framework [13].

A policy μ over options selects option o in state s with probability $\mu(s, o)$. The option-value function for μ is defined as follows:

$$Q^{\mu}(s, o) \stackrel{\text{def}}{=} \left\{ r_{t+1} + \gamma r_{t+2} + \cdots \,\middle|\, \mathcal{E}(o\mu, s, t) \right\},$$

In the above formula $\mathcal{E}(o\mu, s, t)$ is the event that option o is initiated at time t in state s and terminates stochastically τ time steps later. The policy then continues according to μ.

By the aid of dynamic programming and RL algorithms, the options framework results in an SMDP Q-learning equation much like the conventional Q-learning update:

$$Q(s, o) \leftarrow Q(s, o) + \alpha \Big[r_s^o + \gamma^k \max_{o' \in \mathcal{O}_{s'}} Q(s', o') - Q(s, o) \Big].$$

This relation is applied when o finishes in s' after running for k time steps. In the above equation r_s^o is the accumulated reward during o's execution. Under conditions similar to those for Q-learning, $Q(s, o)$ converges to the optimal value function over options.

3 Description of the Algorithm

In this section, we introduce an algorithm for finding skills. Algorithm 1 shows the proposed method in four steps.

Primarily, the agent utilizes Q-learning (or any other flat RL method) in early episodes of learning. Then a graph is made by collecting trajectories in previous

Algorithm 1 SCC-based Skill Acquisition.

1: Run Q-learning for k episodes
2: Construct the transition graph from trajectories
3: Find strongly connected components of the graph by calling *SCC-Inspector(G)*
4: Generate Skills using subgoals identified in step 3.

episodes. In the third step the environment is partitioned into some regions based on the constructed graph. Finally skills are extracted from movements of the agent from large regions to others. Autonomously gained skills are then used as heuristics to search the environment for goal states. In fact skills guide the agent for a goal oriented exploration and thus accelerate the learning process.

In this section, steps 2 to 4 of the algorithm are described separately.

3.1 Constructing the Transition Graph

In order to construct a skill we need to have a proper description of a subgoal. Our explanation of subgoal states is a mixture of two previous definitions presented in [8,6,10] and [11]. According to the first view, subgoals are states that lie between densely-connected regions of the state space. In the early phases of a learning task or an exploration process, the agent acts much like a random walk. It hangs around moving in one region for a long time until it finds a way through a neighboring region accidentally. Considering the trajectories of an agent in primary consecutive episodes, we can intuitively understand that there should be a path from each state s_{G_1} of a region G_1 to the other state (say s'_{G_1}) of the same region and vice versa. On the other hand, since the probability of transition from region G_1 to the other one (say G_2) is low, it is unlikely that the agent has experienced a significant number of paths to and from the two outer states of s_{G_1} and s'_{G_2}.

Bearing in mind the above explanation, we construct the transition graph $G = (V, E)$ in the early episodes. In this graph V corresponds to the set of observed states in the first k trajectories and each directed weighted edge $(u, v) \in E$ corresponds to a transition from state u to state v. Weights show the number of times the transition has taken place. This transition graph is the union of k local transition graphs collected in elementary episodes of the learning task.

In order to take into account the low probability of transition between outer states of two regions we used a threshold t_t as a minimum number of transitions between two states needed to stand for an edge in graph G. For detecting dense regions we used the strongly connected components concept, which will be described in the next part.

3.2 Finding Strongly Connected Components

A strongly connected component (SCC) of a directed graph $G = (V, E)$ is a maximal set of vertices $C \subseteq V$ such that for every pair of vertices u and v in C, we have a path from node u to node v and vice versa; that is, vertices u and v

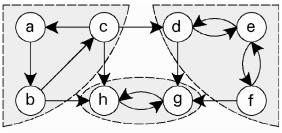


Fig. 1. a graph with three SCCs

are reachable from each other [3]. Figure 1 illustrates the SCCs of a DG with eight vertices.

In what follows, we describe an algorithm for finding SCCs of a directed graph (DG). The method, as shown in algorithm 2, is based on depth first search (DFS) and consists of four steps.

First in step 1 the graph is traversed using DFS. This traversal method also assigns two timestamps to every vertex v. The first timestamp, $d(v)$, records when v is first *discovered* and the second timestamp, $f(v)$, records when the search *finishes* examining the adjacent nodes of v. In step 2 the transpose graph, G^T, is constructed, which has the same nodes as G and the direction of all edges are reversed. After that in step 3 the transpose graph is searched with DFS starting from an unmarked node which has the largest finishing time. This is repeated until all nodes have been marked. Finally in step 4 the vertices of each tree in the depth first forest generated in step 3 are output as a separate strongly connected component. See [3] for a detailed proof on correctness of the algorithm.

Algorithm 2 SCC-Inspector(G)

1: Call DFS(G) to compute finishing times $f(u)$ for each vertex u
2: Compute G^T
3: Call DFS(G^T), but in the main loop of DFS, consider the vertices in order of decreasing $f(u)$ (as computed in line 1)
4: Output the vertices of each tree in the depth-first forest formed in line 3 as a separate strongly connected component

Using the adjacency list representation, the DFS algorithm creates the depth first forest of a DG in a running time of $\Theta(V + E)$ and creation of G^T takes $O(V + E)$ time. As a result SCC-Inspector(G) is a linear-time algorithm.

Unlike the adjacency matrix which incurs a cost of $\theta(V^2)$, in an adjacency list representation, for each vertex we only keep those vertices to which it has an edge. Using such a data structure is quite fitting to our requirements. Almost all the domains of interest in HRL field are those with large state spaces and only a few admissible actions are available in each decision point. Thus, the resulting transition graph would be a sparse one. This is exactly the dominance point of adjacency list representation to the matrix one. This is because it does not stand for an edge that does not exist in the graph which results in an order of magnitude efficiency in taking space and processing time in sparse matrices.

In our SCC-based subgoal discovery method, the agent constructs the transition graph in early episodes of learning. After that, edges with low weight are filtered. The resulting graph is passed to the SCC inspector algorithm to obtain the components which form the state space. The states that connect two SCCs are subgoal states. Subgoals of interest are those that are the windows of *larger* components to the environment. As a heuristic for selecting such components, we took into consideration the fact that in a Gaussian distribution with μ as the mean number of states among all components and σ as the standard deviation, $P(X < \mu + 2\sigma) = 0.97$. So to find useful subgoals, one way is to choose states that are the edge points of components with more than $\mu + 2\sigma$ states.

3.3 Generating Skills

After constructing the SCCs of the graph and finding subgoals, skills will be generated using the options framework. In order to generate skills, the algorithm finds a local policy π using experience replay and a pseudo-reward function [5,4]. After that, β is set to be 1 for the noticed states and states not included in the initiation set. The probability of option termination is set to 0 for states in the initiation set. We included all nodes of the strongly connected component in the initiation set. This selection seems to be of help in the learning phase since the primary goal of an option is to lead the agent to the outer states of a (large) component, where it can easily transit to other components for further exploration. As a result, instead of using hand-craft options made available by the system designer, the agent is capable of building its own skills. Evidently, this results in acceleration of learning process and also great ease in knowledge transfer among agents.

4 Experimental Results

In this section we present some experiments to empirically assess our SCC-based skill acquisition algorithm. Similar to [7,8,10,11,2] we use the two-room gridworld as the test bench. A domain we understand well and can locate its subgoals intuitively. As shown in figure 2, this domain contains two rooms that are connected through a hallway. The agent starts randomly from a square in the left room and tries to get to the bottom right corner of the right room. There are four primitive actions available to the agent in all states: up, right, down, left. If the direction of movement is blocked, the agent remains in the same location. Getting in the goal square results in a reward value of 1 and a small punishment of 10^{-6} is received for all other states. To take into account the stochastic nature of the environment, by choosing an action, the agent moves in the intended direction with probability 0.9 and in a uniform random direction with probability 0.1.

In our simulations the agent used Q-learning with ϵ-greedy exploration with $\epsilon = 0.1$. All through the experiment the learning rate α was kept unchanged at 0.05 and the Q-values were set to 0 in the beginning. We collected the trajectories

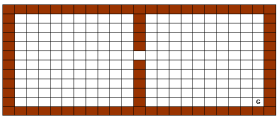
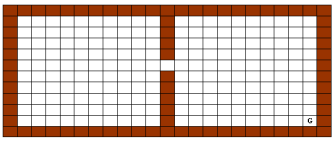
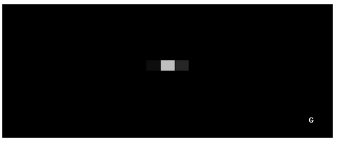


Fig. 2. Two-room gridworld with a subgoal in hallway

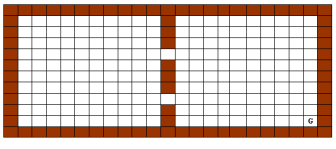


(a) Hallway position



(b) Identified subgoals

Fig. 3. Two-room gridworld with one hallway



(a) Hallway positions

(b) Identified subgoals

Fig. 4. Two-room gridworld with two hallways

in the $k = 10$ first episodes of learning and included edges with a weight equal to or bigger than $t_t = 14$ in the SCC-Inspector algorithm.

Figure 3 shows a gridworld with only one hallway. We ran the algorithm for 50 times. In 37 runs, the hallway was the outer state of an SCC containing all the states in the left room. In 7 other runs the right square neighboring the hallway was identified as subgoal. Figure 3b shows the obtained results. In %96 of the runs (=48 runs) either the hallway or one of its three neighboring states were identified as subgoal (as bleached in figure 3b).

Figure 4a depicts another gridworld with two hallways as subgoal. The obtained results after 50 executions are shown in 4b . For both subgoals, in %94 of times the subgoal state was the hallway or one of its three neighboring squares. we used $k = 10$ and $t_t = 12$ to obtain these results. According to the results, the identified subgoals are quite satisfactory. Meanwhile, the heuristic from Gaussian distribution suits this domain well. In all experiments only one component (containing the squares in the left room) was distinguished to be large.

Other values for k and t_t are also applicable. As an example, for the gridworld in figure 3 the parameters $k = 6$ and $t_t = 10$ are quite suitable. Having found useful subgoals, the agent builds options and employs them in the learning task. Figure 5 compares the average number of steps to goal in a two-room gridworld with one hallway when learning using primitive actions and the case of learning skills using SCC-based subgoal discovery method. The figure also contains the

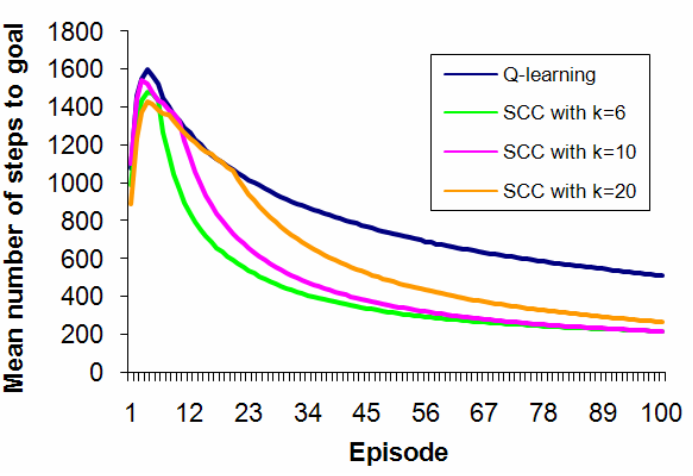


Fig. 5. Average steps to goal with and without skills in a two-room gridworld

learning curve for different values of k. It can be observed that making use of skills can considerably speed up the learning task.

5 Discussion

It can be seen that the suggested SCC-based algorithm works fine in gridworld environment. These results are quite satisfactory, but evaluating the performance of the algorithm in other domains is in progress.

Comparing our method with the one proposed in [7] we understand that although both methods find the subgoals properly, ours has an advantage in terms of utilizing the identified subgoals. SCC-based method finds subgoals after ten episodes. These subgoals are then used to build options that form suitable skills. The latter achieves this goal in twenty episodes. The sooner we make options, the more we stop the Q-learning agent from wandering in early episodes. Approved by our simulations and in agreement with diagrams in [7], Q-learning spends thousands of steps in this ten episode interval. All this is in the condition that in the examined two-room gridworld the maximum distance of the initial state to the goal is no more than thirty squares. This dominance is more vivid comparing some frequency based methods like [2] where subgoals in the same environment are recognized in 500^{th} episode. Such a method may be more applicable to other following learning tasks and not the current learning problem.

Unlike our method that has a threshold on the number of episodes for selecting subgoals, there are a number of methods that use other limits. For example in her LCut method, Şimşek dictates thresholds on the number of times a square has been identified as subgoal (or as *access state* in her terminology) [11] . Only squares that have been recognized as hit more than the threshold value are named subgoals. To be accepted as subgoal, the state should also be observed more than t_o times. She has also set such limits in her RN method. The core

concept of this algorithm is that states that have more novelty relative to the previously observed states are more probable to lead the agent to new regions in the environment and thus can be identified as subgoals. With the aid of bayesian decision theory she classifies the states. Coincidentally, in %96 of the time RN resulted in subgoals which were within two steps of the doorway [10]. Subgoals are discovered in each episode and there's no limit on the number of subgoals. A good point about RN is that deciding whether a state is a subgoal or not, only takes $O(1)$ time which is unique among all methods. But this immediate decision making is the outcome of some estimations and off-line learning done before the main learning task. Other than making the agents behavior domain specific, these approximations blur the rich theoretical basis of the algorithm.

Approximating a value or solution to a problem has been the case of some other methods. The graph partitioning problem has been the bottleneck of some prominent methods. According to [8] Menache's Q-Cut algorithm uses *min-cut/max flow* to partition the transition graphs. An algorithm which takes $O(N^3)$ time where N is the number of states in environment. In [11] Şimşek used an approximation of *NCut* metric plus a clustering method that works on an undirected version of the local transition graph. Considering the MDP nature of RL, using an algorithm for directed graphs seems more suitable.

SCC-based subgoal discovery on the other hand is a good mixture of theory, exactness and low computational complexity. Using the SCC concept, no approximation algorithm is needed to be applied on transition graphs. It also has a linear complexity in terms of the number of states. It is important to note that LCut is more efficient in one respect: Its complexity is $O(h^3)$ where h is the number of states in the most recent part of the transition history [11]. A weakness of the algorithm shared by most other methods is the heuristic setting of its parameters (t_t, k). An important direction for future research is to set up a mechanism for setting suitable parameters. This parameter setting is more important when building hierarchical collections of skills automatically.

References

1. Asadi, M., Huber, M.: Accelerating action dependent hierarchical reinforcement learning through autonomous subgoal discovery. In: Proceedings of the ICML 2005 Workshop on Rich Representations for Reinforcement Learning (2005)
2. Chen, F., Gao, Y., Chen, S., Ma, Z.: Connect-based subgoal discovery for options in hierarchical reinforcement learning. In: International Conference on Natural Computation, vol. 3 (2007)
3. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 2nd edn. MIT Press, Cambridge (2001)
4. Dietterich, T.G.: Hierarchical reinforcement learning with the MAXQ value function decomposition. Journal of Artificial Intelligence Research 13, 227–303 (2000)
5. Lin, L.-J.: Self-improving reactive agents based on reinforcement learning, planning and teaching. Mach. Learn. 8(3-4), 293–321 (1992)
6. Mannor, S., Menache, I., Hoze, A., Klein, U.: Dynamic abstraction in reinforcement learning via clustering. In: Proceedings of the Twenty-First International Conference on Machine Learning, pp. 560–567. ACM Press, New York (2004)

7. McGovern, A., Barto, A.G.: Automatic discovery of subgoals in reinforcement learning using diverse density. In: International Conf. on Machine Learning, vol. 18, pp. 361–368. Morgan Kaufmann, San Francisco (2001)
8. Menache, I., Mannor, S., Shimkin, N.: Q-cut - dynamic discovery of sub-goals in reinforcement learning. In: Elomaa, T., Mannila, H., Toivonen, H. (eds.) ECML 2002. LNCS, vol. 2430, pp. 295–306. Springer, Heidelberg (2002)
9. Parr, R., Russell, S.: Reinforcement learning with hierarchies of machines. In: Jordan, M.I., Kearns, M.J., Solla, S.A. (eds.) Advances in Neural Information Processing Systems, vol. 10, pp. 167–173. MIT Press, Cambridge (1997)
10. Şimşek, Ö., Barto, A.G.: Using relative novelty to identify useful temporal abstractions in reinforcement learning. In: International Conference on Machine Learning, vol. 21, pp. 751–758. ACM Press, New York (2004)
11. Şimşek, Ö., Wolfe, A.P., Barto, A.G.: Identifying useful subgoals in reinforcement learning by local graph partitioning. In: International Conference on Machine Learning, Bonn, Germany, vol. 22 (2005)
12. Stolle, M., Precup, D.: Learning options in reinforcement learning. In: Koenig, S., Holte, R.C. (eds.) SARA 2002. LNCS, vol. 2371, pp. 212–223. Springer, Heidelberg (2002)
13. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning). MIT Press, Cambridge (1998)
14. Sutton, R.S., Precup, D., Singh, S.P.: Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. Artificial Intelligence 112(1-2), 181–211 (1999)
15. Sutton, R.S., Singh, S.P., Precup, D., Ravindran, B.: Improved switching among temporally abstract actions. In: International Conference on Machine Learning, vol. 15. Morgan Kaufmann, San Francisco (1998)

The Research of Negative Correlation Learning Based on Artificial Neural Network

Yi Ding[1,2], Xufu Peng[1], and Xian Fu[1]

[1] Department of Computer Science and Technology,
Hubei Normal University, Huangshi, 435002, China
[2] Department of Computer Science and Technology,
Huazhong University of Science and Technology, Wuhan, 430074, China
teacher.dingyi@gmail.com

Abstract. The integrated technology of the artificial neural network is a research focus of the neural computing technology, which possesses ripe applications in a lot of fields. The neural network ensemble studies the same question with limited neural networks. The output of the ensemble under some input example is determined by all the output of the neural network forming the ensemble under the same input example. The negative correlation learning, which encourages different individual network to study and train different parts of the ensemble in order to make the whole ensemble study the whole training data better, is a training method for the neural network ensemble in this paper. Using a BP algorithm with impulse in the error function is an improvement of the method of negative correlation learning in the paper. The method is an algorithm in batches with more powerful generalization ability and studying of speed, because it combines primitive correlation learning with BP algorithm of impulse.

Keywords: Artificial neural networks, Neural Network Ensemble, Negative Correlation Learning.

1 Introduction

In recent years, the neural network Ensemble (NNE) has already become a focus of the field of neural network. Lacking of the instruction of tightly theoretical system, so the result of application of the nerve calculation totally depends on user's experience. Although some people like Hornik[1] have demonstrated that only one single latent feedforward network of layer can approach arbitrary function of any complexity, it is not confirmable that how to find the suitable network configuration. In 1990, a initiative method of Hansen and Salamon[2], named the neural network ensemble, offered a simple and easy and feasible scheme for settlement of the problem described above. In this method, the generalization performance of the learning system can be improved prominently by briefly training a lot of neural networks and formatting its result. Thanks to its easily using and obvious results, an ordinary engineering technician without the experience of nerve to calculating could benefit from it. Consequently the study

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 804–812, 2009.

on the neural network ensemble will not only promote neural calculation and even theoretical research of all statistics learning methods, but also promote the process of the neural calculation into engineering application greatly.

Some commonly used neural network models are easy to fall into partial minima in the course of studying, which are usually considered to be one of the main shortcomings of the neural network. However, some believe this characteristic has played important role of the improvement of the integrated generalization ability of the neural network. The reason is that the neural network will possibly fall into different partial minima if each neural network is independent when they study, thus the variance of the neural network ensemble can be great, and generalization error can be reduced. In other words, the negative correlation of every partial minima cancels out one another. In this respect, the learning method of negative correlation provided by Liu and Yao in 1990 is a very potential method. In this paper, the author will make some improvement based on the primitive learning method of negative correlation and provide corresponding algorithm and procedure.

2 Artificial Neural Networks

Artificial neural network (ANN) research is enlightened by biology to a certain extent, because the learning system of organism is made from an extremely complicated network of mutual joint neuron. And the artificial neural network is similar on the whole with it. It is formed by the intensively joint of a series of simple units. Every unit has certain amounts of real number values input. (It may be an output of other units), and produce single real number values output. (The output can become input of a lot of other units). At the beginning of the eighties, Scholars such as Rumelhart and Lecun[3] proposed the backpropagation algorithm of the multi-layer perceiving devices, which makes the research of the neural network become the focus of the study. From then on, the development of the neural network research is speedy.

2.1 Perceptron

By the end of the 50s, the esthesia device that Rosenblatt had put forward is one of the main units forming the neural network. Esthesia device regards a real number value vector quantity as input, calculates the linear association of the input, then if the result is greater than a certain threshold value, export 1, otherwise export-1. More precisely, if the input is x_1 to x_n, the output calculating of the esthesia device is:

$$o(x_1, x_2, \cdots, x_n) = \begin{cases} 1, & if \quad w_0 + w_1 x_1 + \cdots + w_n x_n > 0 \\ -1, & otherwise \end{cases} \tag{1}$$

Every w_i is a real constant, or known as weight, which decides the contribution rate of output of every x_i.Among them, $-w_0$ is a threshold value, which is to

make a sensor output, input and $w_1x_1 + w_2x_2 + \cdots + w_nx_n$ must be weighted more than the threshold value.

To simplify the procedures, attached to a constant input $x_0 = 1$, the formula above can be written in the style:

$$o(\overrightarrow{x}) = sgn(\overrightarrow{w}, \overrightarrow{x}) \tag{2}$$

Among it:

$$sgn(y) = \begin{cases} 1, \ if \quad y > 0 \\ -1, otherwise \end{cases} \tag{3}$$

Studying an esthesia device, which means selecting the value of the weigh. So what the esthesia device should consider is that the supposed candidate space is equal to the weigh vectorial gathering of all possible real number value. Formula (2)is called this unit to activate the function. In fact, it also can be the linear function, sigmoid function and so on, to activate the function.

2.2 Feed-Forward Neural Network

The neural network of feedforward is formed by several joint unit. A neuron of the feed-forward neural network accepts the input from the front, and outputs to the behind, without feedback. It can be described by a direction figure without cycling.The nodes are divided into two groups in the picture, which are input nodes and calculation units. The input of every calculation unit is unconditional, but the output is only one. And the output can couple the inputs of other unconditional nodes. There are usually different layers in the feedforward network. The input of layer i is associative with $i - 1$ only, considering the input node is the first layer. The nodes of input and output are called seen-layers for they can be linked up to the external world and influenced by the environment directly, and the other intermediate layers are called latent layers, as shown in Fig. 1.

Kolmogorov proves that any continuous shining-upon function from input to output can be realized by a latent network of 3 layers when the activating function of a neuron is micro. The premise is that enough latent units, proper

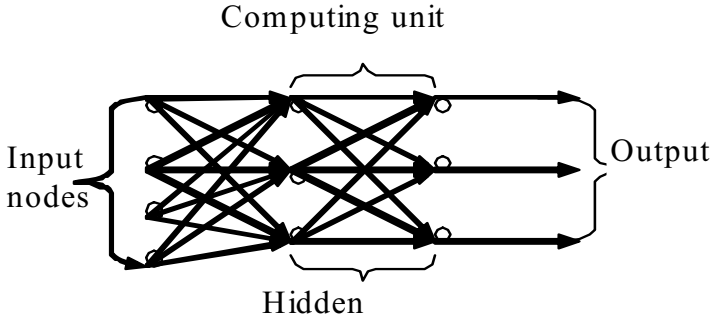


Fig. 1. Feed-forward neural network diagram

activating functions of nonlinearity and weighs values. Normally function to activate the neurons chooses Sigmoid function $f(x) = \dfrac{1}{1+e^{-x}}$, because the sigmoid function is non-linear, monotone, an unlimited number of micro, approximately regarding as threshold function with great weigh and close to an linear function with small weigh.

The common neuron activation functions are also including hyperbolic function $f(x) = tanh(x)$, and so on.

3 Negative Correlation Learning

1996, Solich and Krogh [4] provided the definition for the neural network ensemble, which was the neural network ensemble studied a same question with limited neural networks, The output of the ensemble under some input example is determined by all the output of the neural network forming the ensemble under the same input example.

In the respect of the individual network of producing ensemble, the process of two stage designing is adopted by most methods, that is, firstly the mission is to train individual networks independently and in turn, then make them form neural network ensemble. In this way, it not only loses the interaction between the individual networks, but also is without feedbacked between the stages of training and forming. So it could lead some individual networks having no contribution to the whole ensemble. In addition, the document points out, when the individual networks are different greatly in the neural network ensemble, the effect of the ensemble is better. However, it's still a focus of research that how to get the individual networks with great difference and how to appraise difference degree between several networks at present. In this respect, the learning method of negative correlation provided by Liu and Yao in 1990 is a very potential method.

3.1 The Basic Concepts of Negative Correlation Learning

The order of the negative correlation learning is to encourage differently individual networks of the different individual networks ensemble study learn to train different parts in the data, in order to study the whole training gathering better for the neural network ensemble. The difference between the negative correlation learning and the other methods of neural network ensemble is that, the way to train individual networks in other method is an independent or in- order way; while the way in the negative correlation learning is to induce a relevant punishment item for training these individual network simultaneously through the error function of the individual network of the neural network ensemble. That is to say, the purpose of every individual network's training is to make the result of the whole neural network ensemble best.

Supposing the gathering of training is $D = \{(\overrightarrow{x}_1, d(1)), \cdots, (\overrightarrow{x}_n, d(N))\}$

Among it:

N stands for the sample size, $\overrightarrow{x}_i \in R^p$ with generalization yet, $d(i)$ stands for the output amount of the goal;simply regarding the average value of the output sum of all the individual networks in the neural network ensemble as the output of the ensemble.

$$F(n) = \frac{1}{M}\sum_{i=1}^{M} F_i(n) \tag{4}$$

M stands for the number of the individual networks in the ensemble. $F_i(n)$ stands for the output of individual network j from training sample n. $F(n)$ stands for the output of the ensemble from training sample n.

In the method of negative correlation learning, the error function of every individual network induces a relevant punishment item, and makes all the individual networks trained in the same ensemble at the same time. The error function of individual network i can be defined as:

$$E_i = \frac{1}{N}\sum_{n=1}^{N} E_i(n) = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}(F_i(n) - d(n))^2 + \frac{1}{N}\sum_{n=1}^{N} p_i(n) \tag{5}$$

$E_i(n)$ is the value of the error function of individual networks i in sample n. In the right of the formula, the first item is the experience error function for individual networks i. The second is the item of relevance punishment. p_i stands for the function of the relevance punishment item of individual networks i. By minimizing p_i, every individual network Error is negative correlated with all the rest of the networks of individual errors, adjusting the punishment power through the reference $0 \leq \lambda \leq 1$. The form of the punishment function of $p_i(n)$ on the sample n is:

$$p_i(n) = (F_i(n) - F(n))\sum_{j\neq i}(F_j(n) - F(n)) \tag{6}$$

$$\begin{aligned}\frac{\partial E_i(n)}{\partial F_i(n)} &= F_i(n) - d(n) + \lambda\frac{\partial p_i(n)}{\partial F_i(n)} \\ &= F_i(n) - d(n) + \lambda\sum_{j\neq i}(F_j(n) - F(n)) \\ &= F_i(n) - d(n) - \lambda(F_i(n) - F(n)) \\ &= (1-\lambda)(F_i(n) - d(n)) + \lambda(F(n) - d(n))\end{aligned} \tag{7}$$

Observing formula (4), (5), (6), and (7), we can get:

(1)While using the method of negative correlation learning, all the individual networks interact by the relevant punishment items of the error function in the neural network ensemble. The output of F_i of every individual network have to minimize not only the difference of $F_i(n)$ and $d(n)$, but also the difference of $F(n)$ and $d(n)$. That is to say, it should be considered the error of other trained sub networks before train a certain sub network in the neural network ensemble.

(2)The independent and in-order training is a special form of the negative correlation learning.

(3)When $\lambda = 1$:

$$\frac{\partial E_i(n)}{\partial F_i(n)} = F_i(n) - d(n) \tag{8}$$

The error function of the neural network ensemble on sample n can be defined as

$$E_{nes}(n) = \frac{1}{2}(\frac{1}{M}\sum_{i=1}^{M} F_i(n) - d(n))^2 \tag{9}$$

$$\frac{\partial E_{nes}(n)}{\partial F_i(n)} = \frac{1}{M}(\frac{1}{M}\sum_{i=1}^{M} F_i(n) - d(n)) = \frac{1}{M}(F(n) - d(n)) \tag{10}$$

From (8) and (10), we can get:

$$\frac{\partial E_i(n)}{\partial F_i(n)} \propto \frac{\partial E_{nes}(n)}{\partial F_i(n)} \tag{11}$$

In (11), it is obvious that we can minimize the error function of the whole neural network ensemble by minimizing the error function of individual networks. Therefore, a new way offered by the negative correlation learning is that the whole neural network ensemble learning can be disintegrated into different submissions of individual networks.

3.2 Deviation-Variance-a Compromise between the Covariance

Mean-Squared Error (is abbreviated as MSE) can be used for weighing the generalization performance of neural network and neural network ensemble.

The mean-squared error formula of the neural network that is widely known to all:

$$\begin{aligned} E_{mse} &= E_D[(E_D[d \mid x] - F(x, D))^2] \\ &= (E_D[F(x, D] - E_D[d \mid x])^2 + (E_D[F(x, D] - E_D[F(x, D])^2 \end{aligned} \tag{12}$$

The formula (12) indicates, the mean-square error can be shown as the sum of deviation item and the variance item. The best way to obtain smaller deviation and smaller variance is to try hard to understand the priori information of the goal equation. The observation of deviation and variance is useful to explain the following practices: go to look for the accurate priori knowledge of the form of solution; make use of training samples as much as possible; it is essential to study the match situation of algorithms and designated question. The mean-squared error formula of neural network ensemble is similar to the mean-squared error formula of the neural network:

$$E_{mse} = E_D[(E_D[d \mid x] - F(x, D))^2]$$
$$= (E_D[F(x, D] - E_D[d \mid x])^2 + E_D[\frac{1}{M^2}\sum_{i=1}^{M}(F_i(x, D) - E_D[F_i(x, D)])^2]$$
$$+ E_D[\frac{1}{M^2}\sum_{i=1}^{M}\sum_{i\neq j}(F_i(x, D) - E_D[F_i(x, D)]) \times (F_j(x, D) - E_D[F_j(x, D)])] \tag{13}$$

In the right of (13), the first item is reflected the deviation of the neural network ensemble, the second is reflected the variance, and the third is reflected the covariance of the neural network ensemble. The negative correlation learning can make the generalization performance of the neural network ensemble better through adjusting to keep the balance of deviation, variance and covariance.

3.3 Improved Negative Correlation Learning

In the primitive negative correlation learning, the training algorithm of every individual network in the neural network ensemble is the standard BP algorithm. Because of the inherent characteristic of standard BP algorithm, the shortcoming is slow restraint for individual networks. The algorithm with impulse BP makes the renewing weigh in No.n time partly depend on the renewing weigh in No.n-1 time in order to search for the result with long step and accelerate the restraint. In this paper, combining the algorithm with impulse BP, the author provides an improved negative correlation learning for the study in batches of the neural network ensemble.

Step 1. Begin.

Step 2. Create neural networks of number M a single-unit output of BP as individual networks in neural network ensemble of number M, BPN_i stands for individual network j;

Step 3. Initialize the weigh value of all individual networks into small value at random;

Step 4. Before meeting the condition of stopping:

(I) D-training for every training sample $(\overrightarrow{x}, t)$, do:

(i) For every individual Network BPN_i do:

Enter BPN_i into samples $\overrightarrow{x}$ and calculate the output O_u of every unit in BPN_i;

(ii) Calculate $O_{total} = \frac{1}{M}\sum_{k=1}^{M} O_k$ stands for the BPN_i output of individual network of No. k;

(iii) For every BPN_i of individual networks, do:

a. For the output of every unit k in BPN_i, calculate the error item $\delta_k \longleftarrow (1-\lambda)f_k(d_k - O_k) + \lambda(d_k - O_{total})$, f_k stands for the derivation of the activation function of unit k, λ stands for the punishment factor;

b. For every hidden unit h in BPN_i, calculate the error item

$$\delta_h \longleftarrow f_h \sum_{j \in Downstream(h)} \delta_j W_{jh}.$$

Downstream (h) stands for the output units gathering containing unit h in the direct input. w_{jh} is the weigh value associated with the input of No.h in unit j.

(iv) About every individual network BPN_i, calculate the renewed value $\Delta_{W_{mn}}(i)$ of every hidden- layer node and output node according the current sample. For example, the current training sample is the first sample in this turn, so $\Delta_{W_{mn}}(i) = \eta\delta_m x_{mn}$, otherwise, $\Delta_{W_{mn}}(i) = \eta\delta_m x_{mn} + \Delta_{W_{mn}}(i)$. In the formula, $\Delta_{W_{mn}}(i)$ stands for the renewed part of weigh value between node n to m in the training turn of No.i, x_{mn} stands for the input from node n to unit m, η is the study rate;

(II) Renew every weigh value of the network $w_{mn} = w_{mn} + \Delta_{W_{mn}}(i) + \alpha\Delta_{W_{mn}}(i-1)$,in the formula, α is the impulse factor, Goto 4;

Step 5 End.

4 Conclusion

In this paper, the author introduces the basic conception of the negative correlation learning of the neural network ensemble, and analyzes deviation, variance, a compromise between the covariance. Besides, the author provides an improved negative correlation learning, combined by the negative correlation learning and the study algorithms with impulse. Although there are some improvement in the study of the negative correlation learning of the neural network ensemble in the paper, much work are still waiting for complete in the study. For example, when selecting the proper punishment factor, the neural network ensemble can obtain very good generalization performance. But the selection of punishment factor often depends on experience parameters. It is the coming mission that how to select the proper punishment factor quickly.

Acknowledgments. This work was supported by Natural Science Foundation of Hubei Province under Grant 2007ABA183.

References

1. Hornik, K.M., Stinchcombe, M., White, H.: Multiayer Feedforward Networks are Universal Approximators. Neural Networks 2, 359–366 (2002)
2. Hansen, L.K., Salamon, P.: Neural Network Ensembles. IEEE Transactions on Pattern Analysis and Machine Intellience 12, 993–1001 (1999)
3. Rumelhar, D.E., Hinton, G.E., Williams, R.J.: Learning Internal Representations by Error Propagation. In: Rumlhart, D.E., Mcclell, J.L. (eds.) MIT Press, pp. 318–362. MIT Press, Cambridge (2001)

4. Sollich, P., Krogh, A.: Learning with Ensembles: How Overfitting can be Useful. In: Touretzky, D., Mozer, M. (eds.) Hasselmo Meds Advance in Neural Information processing Systems, pp. 190–196. MIT Press, Cambridge (1996)
5. Cooper, L.N.: Hybrid Neural Network Architectures: Equilibrium Systems that Pay Attention. In: Mamone, R.J., Zeevi, Y.Y. (eds.) Neural Network: Theory and Applications, pp. 81–96. Acaddem ic Press, New York (1991)
6. Liu, Y., Yao, X.: Simultaneous Training of Negatively Correlated Neural Networks in An Ensemble. IEEE Transactions on Systems 29, 297–310 (1999)

Algorithm of Neural Network Ensembles and Robust Learning

Hai Qian and Youping Fan

Faculty of Electrical Engineering, Wuhan University, Hubei Prov., Wuhan 430072, China
Fyoupingnxinrong@yahoo.com.cn

Abstract. Neural networks ensemble (NNE) has recently attracted great interests because of their advantages over single neural networks (SNN) as the ability of universal approximate and generalization. However, the design of neural network ensembles is a complex task. In this paper, we propose a general framework for designing neural network ensembles by means of cooperative co-evolution. The proposed model has two main objectives: first, the improvement of the combination of the trained individual networks; second, the cooperative evolution of such networks, encouraging collaboration among them, instead of a separate training of each network. In order to favor the cooperation of the networks, each network is evaluated throughout the evolutionary process using a PSO algorithm based on bootstrap technology (BPSO). A simulation example of the 3-D Mexican Hat is given to validate the method. The result proved its effectiveness.

1 Introduction

Neural network ensembles [1] are receiving increasing attention in recent NN research, due to their interesting features. Neural networks ensemble is a set of neural networks whose decisions are combined to improve the performance of the overall system [2]. It originates from Hansen and Salamon's work [3], which shows that the generalized ability can be significantly improved through training many neural networks and then combining their predictions. Each network within the ensemble has a potentially different weight in the output of the ensemble. Several works have shown [4,5] that the network ensemble has a generalization error generally smaller than that obtained with a single network and also that the variance of the ensemble is lesser than the variance of a single network. The output $\boldsymbol{y}$ of a typical ensemble [6] with $\boldsymbol{k}$ constituent networks when an input pattern $\boldsymbol{x}$ is presented is

$$y(x) = \sum_{i=1}^{k} w_i y_i(x) \tag{1}$$

where $\boldsymbol{y_i}$ is the output of network $\boldsymbol{i}$ and $\boldsymbol{w_i}$ is the weight associated to that network. If the networks have more than one output, a different weight is usually assigned to each output. The ensembles of neural networks have some of the advantages of large networks without their problems of long training time and risk of overfitting. For more detailed descriptions of ensembles the reader is referred to [3,7].

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 813–818, 2009.

The rest of the paper is organized as follows. Section 2 describes the proposed *model of cooperative ensembles*, *Cooperative Ensemble of Neural Networks and Robust Learning Algorithm of Neural Networks Ensemble*. Section 3 describes *Simulations and Experimental Results*. Finally, Section 4 states the conclusions of our work and the most important lines for future research.

2 Neural Network Ensembles and Robust Learning

A. Architecture of RNNE

Our basic network is a generalized multilayer perceptron (GMLP), as defined in [8]. It consists of an input layer, an output layer, and a number of hidden nodes interconnected among them.

Given a GMLP with $\boldsymbol{m}$ inputs, $\boldsymbol{N}$ hidden nodes, and $\boldsymbol{n}$ outputs, and $\boldsymbol{X}$ and $\boldsymbol{Y}$ being the input and output vectors, respectively, it is defined by the equations [8]

$$\begin{aligned} x_i &= X_i, && 1 \le i \le m \\ h_i &= \sum_{j=1}^{i-1} w_{ij} x_j, && m < i \le m + N + n \\ x_j &= f(h_j), && m < j \le m + N + n \\ Y_i &= x_{i+m+N}, && 1 \le j \le n \end{aligned} \tag{2}$$

where $\boldsymbol{w_{ij}}$ is the weight of the connection from node $\boldsymbol{j}$ to node $\boldsymbol{i}$ .The representation of a GMLP can be seen in [8]. We see that the $\boldsymbol{i}$-th node, provided it is not an input node, has connections from every $\boldsymbol{j}$-th node $\boldsymbol{j < i}$.

The main advantage of using a GMLP is the parsimony of the evolved networks. Its structure allows the definition of very complex surfaces with fewer nodes than in a standard multilayer perceptron with one or two hidden layers.

The architecture of a specific Robust Learning of Neural Networks Ensemble (RNNE) is defined as follows: RNNE = <X, SN, IU, Y>.

Where, $x \in R^n$ is the input vector; SN represents a set of subnets; IU represents the integrating unit which performs selective ensemble of modules; $Y \in R^n$, is the output vector.

B. Cooperative Ensemble of Neural Networks

Particle Swarm Optimization (PSO) algorithm is an evolution computation technology based on swarm intelligent methodology. PSO is initialized as a swarm of arbitrary particles (arbitrary solution), and then the optimal solution is discovered by iteration. But in course of iterative process for basic PSO methodology, pre-convergence problem always appears with the decreasing of particles' diversity, which un-benefits for acquiring suitable combination weights. So modification is needed.

BPSO algorithm is a PSO algorithm based on bootstrap technology [9]. Bootstrap technology is to arbitrarily select data set from original data set D to form new data set with N sampling points. This technology can produce different data set so as to get different target function, and evolve multi-particle seed swarm with certain differentia degree. After limiting optimal particle searching range and synthesizing optimal

particle searching result in every seed swarm, corresponding combination weights can be adjusted flexibly in a certain range and over-imitation degree of the noise in the original training data caused by integration can be reduced, which will strengthen the extensive ability for nerve net integration. If $\boldsymbol{M}$ well-trained nerve net $f_1, f_2, \cdots, f_M$ is given, BPSO algorithm optimal combination weight $\vec{a}(a_1, a_2, \cdots, a_M)$ is as follows:

① Input data $D(x_j, f(x_j))$, $j = 1, 2, \cdots, N$ used for optimization.

② While $t \le T$ (initial t = 1, T is the most sampling time).

i) Data D is repeatedly sampled for the time of $\boldsymbol{t}$, Number t-group of data $D_t(x_j^t, f(x_j^t))$, $j = 1, 2, \cdots, N$ can be acquired.

ii) The most iterative time $\boldsymbol{L}$ for optimal particle searching is given, PSO algorithm is transferred to correspond number t particle seed swarm of D_t, where there are $\boldsymbol{q}$ particles and every particle is coded as $1 \times M$ vector $\vec{a}^T(a_1^t, a_2^t, \cdots, a_M^t)$; Target function for optimal combination weight is

$$MSE^t = \frac{1}{N}\sum_{j=1}^{N}\left(\sum_{I=1}^{M} a_i^t f_i(x_j^t) - f(x_j^t)\right)^2 + PE \tag{3}$$

Supposing n = 40, $PE = N\left(\sum_{i=1}^{M} a_i^t - 1\right)^2$ is penalizing function, which can restrain weight sum to be 1. Adapting function is $F^t = 1/MSE^t$

Grain the particles depend on as follows formula to adjust an own speed:

$$\begin{aligned} v[\ ] = {} & w * v[\ ] + c1 * rand() * (pBest[\ ] - present[\ ]) \\ & + c2 * rand() * (gBest[\ ] - prent[\ ]) \end{aligned} \tag{4}$$

in which $w = (w2 - w1) * (iw - 1)/(J - 1) + w1$, $w1 = 0.9$, $w2 = 0.2$, $J = 100$. $\boldsymbol{iw}$ is iterative times for particles.

The most value for velocity v is $v_m = 0.05$; optimal particle searching range is $[-am, 1]$, $am = 0.05r$, $\boldsymbol{r}$ is arbitrary data adapting function between **(0 1)**. The formulas (4) and $present[\ ] = present[\ ] + v[\ ]$ are used to adjust velocity and position of particles. Until iterating to $\boldsymbol{L}$, the optimal particle $\boldsymbol{P_t}$ is acquired corresponding optimal combination weight $\vec{a}_b^t$.

iii) $t = t + 1$.

③ Then acquire $\boldsymbol{T}$ optimal particles $\boldsymbol{P_T}$ and its corresponding combination weight value $\vec{a}_b^t$ $(t = 1, 2, \cdots, T)$; The final weight value $\vec{a} = \frac{1}{T}\sum_{t=1}^{T} \vec{a}_b^t$ is gained by averaging T combination weight value, and corresponding nerve net integration output is $f(x) = \sum_{i=1}^{M} a_i f_i(x)$.

C. Robust Learning Algorithm of Neural Networks Ensemble

Least squares method (LSE) is a kind of popular and important methods in many fields including ANN's learning. But some deficiencies lie in this method. One of these deficiencies is it strengthens the destructive impacts of outliers for the square error function [10]. In other words, it has poor robustness. Generally, an algorithm is more robust if the outliers have less impacts on it [10].

Least absolute distance (LAD) method presented by R.J. Boscovitch (1755) is such a more robust method [10]. In 1964 and 1975, P. J. Huber generalized the above idea and he practiced it into linear statistic model [10,11,12]. In statistics, this method is named as M-estimation.

In this paper, synthesis cost function [13,14] is adopted as the target of component neural networks:

$$\phi_1(x) = \beta In(\frac{\cosh x}{\beta}) \tag{5}$$

$$\phi_2(x) = \frac{1}{2}x^2 \tag{6}$$

$$G(\lambda) = \lambda \sum_{k=1}^{m}\sum_{i=1}^{n_0}\phi_2(e_{i,k}) + (1-\lambda)\sum_{k=1}^{m}\sum_{i=1}^{n_0}\phi_1(e_{i,k}), 0 \le \lambda \le 1 \tag{7}$$

Here $\phi_2(x)$ represent the square of error. And $\phi_1(x)$ is an adjustable criterion, it trends to absolute value criterion when β augmenting, otherwise it trends to square criterion. $G(\lambda)$ is the cost function, λ adjust the weight of $\phi_1(x)$ and $\phi_2(x)$.

The reason we adopt the above method is that robustness is in contradiction with efficiency. General speaking, the more robust an algorithm is, the less efficiency it has. The robust learning algorithm composed of (5)-(7) is just a reasonable compromise for the mentioned contradiction. This method consists of two steps:

Step1. train component neural networks (***network***$_i$) with the cost function $G(\lambda)$.

Step2. combine the component predictions as (1).

3 Simulations and Experimental Results

The aforementioned method is applied in various modeling problems, and the results are satisfactory. Limited by the length of this paper, only the prediction for 3-D Mexican Hat is reported.

3-D Mexican Hat is defined as follows:

$$y = \frac{\sin\sqrt{x_1^2+x_2^2}}{\sqrt{x_1^2+x_2^2}}, x_i \sim U[-4\pi, 4\pi] \tag{8}$$

The data set comprise 2000 instances with the x_i randomly generated in terms of uniform distribution over the interval $[-4\pi, 4\pi]$, which are equally divided into two sets: one is the training set; the other is the test set. The train set includes 1000 patterns, and 10% of the patterns are randomly corrupted by impulsive noise .The train set is illustrated in Fig.1, the "*"represent the train set.

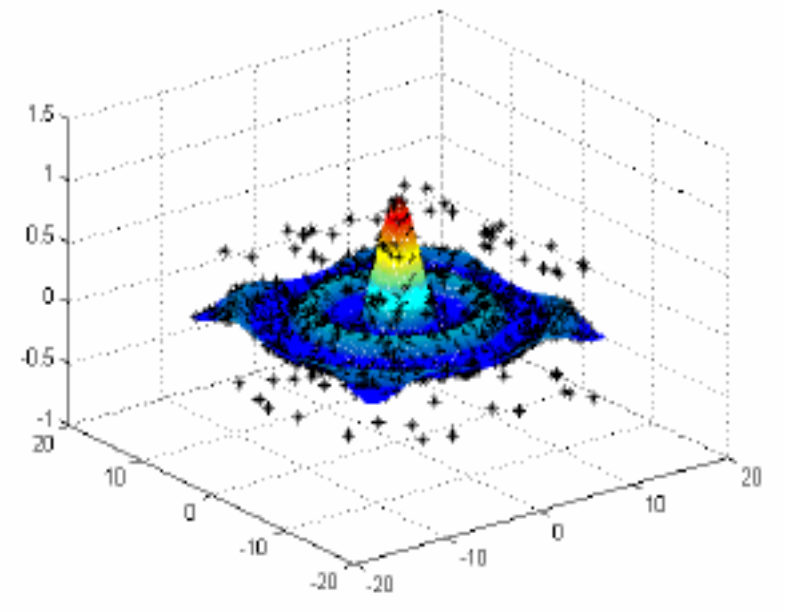


Fig. 1. 3-D Mexican Hat model and train set

Table 1. MSE OF TEST SET

	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5	Mean
NNE	1.18 e-2	9.06 e-3	1.12 e-2	1.33 e-2	1.64 e-2	1.235 e-2
RSNN	9.05 e-3	1.12 e-2	8.50 e-3	1.30 e-2	8.26 e-3	9.988 e-3
RNNE	6.70 e-3	5.62 e-3	4.97 e-3	5.81 e-2	4.25 e-3	5.474 e-4

In order to reduce the incidental factors, we perform five runs for the same experiment. To validate the precision and generalization of RNNE, this paper adopt the mean square error (MSE) of test set as the criterion and use the NNE and robust single neural networks (RSNN) paradigm of reference. The simulation result shows in TABLE II, and the criterion use the mean square error (MSE) as above.

4 Conclusion

The results in **Table 2** show that: learning under the contaminated data, RNNE gets the best performance, single neural networks (SNN) with robust algorithm get a tolerable result, while NNE is deteriorated by the outliers. And the result of this test shows that the RNNE fits quite well. During the empirical study, we discover that the more complex the real-world model is the greater difference between the RNNE and RSNN perform. RNNE is equal to those complex system contaminated by unknown distribution noise.

Acknowledgements. Thank you for Supported by the National Natural Science Foundation of China (50477018) and the Post-doctoral Science Foundation of China (2004036135).

References

1. Perrone, M.P., Cooper, L.N.: When Networks Disagree: Ensemble Methods for Hybrid Neural Networks. In: Neural Networks for Speech and Image Processing, pp. 126–142. Chapman& Hall, London (1993)
2. Zhou, Z.H., Wu, J.X., Tang, W.: Ensembling Neural Networks: Many Could be Better than All. Artificial Intelligence 137, 239–263 (2002)
3. Hansen, L.K., Salamon, P.: Neural Network Ensembles. IEEE Trans. Pattern Analysis and Machine Intelligence 12, 993–1001 (1990)
4. Fern, A., Givan, R.: Online Ensemble Learning: An Empirical Study. Mach. Learn. 53, 71–109 (2003)
5. Bakker, B., Heskes, T.: Clustering Ensembles of Neural Network Models. Neural Netw. 16, 261–269 (2003)
6. Haykin, S.: Neural Networks – A Comprehensive Foundation, 2nd edn. Prentice–Hall, New Jersey (1999)
7. Dzeroski, S., Zenko, B.: Is Combining Classifiers with Stacking Better than Selecting the Best One? Mach. Learn. 54, 255–273 (2004)
8. Chen, D.S., Jain, R.C.: A Robust Back Propagation Learning Algorithm for Function Approximation. IEEE Trans. Neural Networks 5, 993–1001 (1994)
9. Riget, J., Vesterstro, M.J.S.: A Diversity Guided Particle Swarm Optimizer-the ARPSO. EVA Life Technical Report No.2002-02[EB/ OL], http://www.evalife.dk/publications.php
10. Chen, X.: Historical Backgrounds and Present State of Least Squares Method. Journal of Graduate School, Academ ia Sinica 15, 4–11 (1998)
11. Hube, P.J.: Robust Estimation of a Location Parameter. Ann. Math. Statist. 35, 73–101 (1964)
12. Huber, P.J.: Robust Statistics. Wiley, New York (1981)
13. Wang, Y.J., Tu, J.: Neural Networks Control. Chinese Machinery Industry Press, Peking (1998)
14. Wang, L.X.: A Course in Fuzzy Systems and Control. Prentice-Hall, New Jersey (1994)

The Heuristic Algorithm Based on Learning-Competing Model and Its Application to Task Assignment Problem

Zhe Wang, Hongwei Wang, Xi Chen, and Yi Jiang

Institute of System Engineering, Huazhong University of Science and Technology, 430074 Wuhan, China
philo_wang@163.com

Abstract. A universal model-learning-competing one is established for solving task assignment problem in National Economy Mobilization(NEM), in which local search was focused in learning model and global search in competing. Thereafter the strong points of the two models were amalgamated in the algorithm. A team of parameters was used to coordinating the correlation of the two models. Applying learning-competing model to an actual case, using greed algorithm in learning model, genetic algorithm in competing model, the results obtained were in coincidence with the analysis.

Keywords: Heuristic algorithm, Combinatorial optimization, Learning-competing mode, Task assignment.

1 Introduction

Since a long time ago, an effective algorithm in polynomial time to solve combinatorial optimization problem, such as task assignment problem[1], is difficult to seek. Pertinent literature[2] shows, in case of P≠NP(nondeterministic polynomial), there is not an algorithm that ensure to gained optimal solution in polynomial time. In this background, a kind of heuristic algorithm based on biology, physics and artificial intelligence; with global optimize capability, strong robust, high universality and suitability for parallel processing has introduced to solve such complex combinatorial optimization problem.[3,4] Because of its strongpoint such as effect optimize capability and needless of special information about problem, the algorithm is rapidly applied in computer science, optimal scheduling, traffic problem, engineering optimization et al. Now such heuristic algorithm is including genetic algorithm(GA), A* algorithm, ant colony algorithm(ACA), simulated annealing algorithm (SA), evolutionary algorithm (EA), tabu search (TS) and artificial neural networks(ANN). But because of their different research mechanisms and emphases, the research works are quite fragmented.

For the sake of effectual unification for those works, we attempt a new universal model, learning-competing one[5], to solve such complex problem. Applying the model to task assignment problem in National Economy Mobilization(NEM), we get a good application. Task assignment problem in NEM, as a complex combinatorial optimization problem, mainly means that concentrating lots of resources to a certain

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 819–826, 2009.

place in a short period of time for the needs of emergency or war. Because the emergencies happen in a short time, have a destructive power, and change rapidly; many requests must be met, such as time, cost, product quality, the reliability of the task, the effect on economic system and so on. For such reasons, the heuristic algorithms, such as genetic algorithm(GA)[3] and A* algorithm[6], are widely used in solving the problem. Based on previous scholars' works, we advanced the research by learning-competing model.

2 The Basic Idea

Learning-competing model simulates in real world people must learn in school for well education, in order to enter the competition in social to realize their value. The model divides algorithm into two phases-learning model and competing model. In learning model, multi individuals search different local optimal solution, and then those individuals are amalgamated to global more-optimal solution though competing in competing model. Overlapping the method, until the global optimal solution or satisfactory solution can be gained. The model centralizes strongpoint from kinds of algorithms, so it fits for the large-scale practice problem, such as task assignment problem.

2.1 Learning Model Phase

The task for learning model is to seek good individuals for competing model. The graduate qualification is designed for measuring whether those individuals are eligibility. The graduate qualification is formed by three evaluations:

1. Viability target

   For combinatorial optimization problem, viability target is a pass line that the individuals' objective function value must achieve. Avoiding weak individuals enter into competing model and seize resources; individuals must get enough viability by leaning in learning model phase.

2. Individuation target

   Individuation target is used to measure the individuals' merits in different regions. High individuation means that individual is very good in some respects. For combinatorial optimization problem, high individuation means each individual is different local optimal solution. The greater the difference among the individuals, the better the overall operation effect in competing model. It is just a case in point to illustrate the heterosis in biology. Therefore ensuring individuals' individuation can embody the advantage of learning-competing model. Specially, some individuals that have high individuation evaluation must be precedence reserved.

3. Learning time

   Learning time is arithmetic complexity for engendering an individual. In order to ensure the overall optimal effectiveness, learning model can't be designed too complicated. Grouping can control learning time.

The purpose of setting the three indicators is to harmony the relationship of learning model and competing model. The stricter graduate qualifications demanded, the

more task that learning model shares. If the problem has not any assist info, we can adopt two disposal methods:

1. Record best individual we have, and recording the individual's evaluations, viability target is m_{maxw}, individuation target is w_{max}. Enact relevant constant m and w. The graduate qualification is set to: viability target is larger than m_{max}-m, individuation target is larger than w_{max}-w.
2. Recording best individual we have, and recording the individual's evaluations, viability target is m_{maxw}, individuation target is w_{max}. Enact relevant constant n_m and n_w. The graduate qualification is set to: viability target is larger than $m_{max}*n_m\%$, individuation target is larger than $w_{max}*n_w\%$.

Aim to the three evaluations that learning model wants to achieve, we can use the algorithm that it well solves the small-scale problem and has faster convergence to realize. In this paper, we choose greed algorithm to realize learning model.

2.2 Competing Model Phase

The aim for competing model is to amalgamate individuals' choiceness information from learning model. Therefore competing model must be realized by global optimization algorithm. In this paper, we choose genetic algorithm to realize learning model.

Genetic algorithm (GA)[7] is first advanced by USA scholar, Holland in 1975. The algorithm is a calculate model simulating Darwin's biological theory of evolution. Its basic idea is: the solution to the problem is expressed as a chromosome in GA, and the chromosome is a binary-system code or ten-system code. Besides, a group of chromosomes, the hypothetical solutions, is given before executing GA. Then put the hypothetical solutions into the question's environment, choose more adaptable chromosomes by the rule of survival of the fittest, copy them, afterward produce a new generation of chromosomes that more adapt the environment by chiasma, aberrance. In this way, after generations of evolution, the chromosome that is best adapting the environment will be convergences finally.

It is obvious that, "Survival of the fittest" in genetic algorithm and "competition" in competing model are anastomosed.

2.3 The Whole Computing Process

The whole computing process includes those steps next:

Step1: confirm coding scheme and randomly generated individuals.

Step2: grouping and then set the graduate qualification.

A success grouping can make individuals both get better individuation target in learning model and faster constringency in competing model. It is just corresponding that a good training scheme can make pupils both get more grades in school and express more compete ability in society in real world. But actually prefect grouping schemes are always too complication and hard to operating.

Step3: the implementation computing in the learning model.

Step4: judging individuals that satisfy the graduate qualification whether enough or not? If yes, turn to step5; no, means grouping inappropriate or the graduate qualification unreasonable, turn to step2.

Step5: the implementation computing in the competing model.

Step6 judging the best individual whether satisfies the terminating qualification or not? Yes, turn to step7; no, turn to step3.

Step7 output the option solution.

3 Applying to Task Assignment Problem

Task assignment problem in NEM[6] is: concentrating a certain amount Q of a product to place A in T day. Through surveying the potential data, the enterprises that can participate the task assignment is get; then get the candidate enterprises that can assume the task by FAHP (the fuzzy analytic hierarchy process). In order to facilitate the task assignment model, this paper set the following variables: the number of the candidate enterprises related to task assignment is n (n is a natural number that bigger than one; if n=1, the task assignment meaningless). Assuming the number of task assigned to enterprise i is $x_i (0 \le x_i \le Q)$, then $Q = x_1 + x_2 + ... + x_n$. If $x_i = 0$, means no task for the enterprise. Assuming a_i is enterprise i's production capacity, c_i is enterprise i's cost.

Because the task is shared by all enterprises, and various enterprises are parallel producing; the task completion time is the longest task time among all enterprises, that is $\max(t_1(x_1), t_2(x_2), ..., t_n(x_n))$, and $t_i(x_i)$ is enterprise i's task completion time. Then assuming enterprise i's original production task is producing q_i production; set the impact factor of mobilization task μ ($0 \le \mu \le 1$), means the influence that the mobilization task to the enterprise's operation. If $\mu = 1$, means no affect on the enterprise's operation, that is, using idle production capacity to carry out the mobilization task, applicable to the case of non-emergency situation. If $\mu = 0$, means not to consider the enterprise's original production, that is, all production capacity will be used to complete the mobilization task; it is applicable to the case of emergency situation, but it is the most effect on the normal economic order.

According to the above analysis, the mathematical expression of task assignment model as follows:

$$obj.\ \min(\sum_{i=1}^{n} c_i * x_i)$$

$$\min(\max(t_1(x_1), t_2(x_2), ..., t_n(x_n)))$$

$$s.t. \begin{cases} x_i + \mu * q_i \le T * a_i, \\ t_i(x_i) \le T, \\ \sum_{i}^{n} x_i = Q, \\ x_i \ge 0, i = 1, 2, ..., N \end{cases}$$

This is a multi-objective decision-making problem. Through the introduction of the time-cost[6], time will be converted into quantifiable monetary cost by parameter λ. Then a simplified task assignment model as follows:

$$obj.\ \min\left(\sum_{i=1}^{n}\sum_{t=1}^{t_i(x_i)}(c_i * P_i(t) * (1+\lambda_i)^t)\right)$$

$$s.t.\begin{cases} x_i + \mu * q_i \le T * a_i, \\ t_i(x_i) \le T, \\ \sum_i^n x_i = Q, \\ x_i \ge 0, i = 1,2,...,N \end{cases}$$

In which $P_i(t)$ is the number that No.i enterprise products in No.t day, $x_i = \sum_{t=1}^{t_i(x_i)} P_i(t)$.

3.1 Realization of Learning Model

We adopt heuristic greed algorithm to realize learning model. The basic idea is: the task to the enterprise that has lower cost $\mathbf{c}_i$ is as far as possible.

By way of predigesting algorithm, we adopt such coding scheme: firstly, assuming l is the greatest common denominator among Q,a_1,……., a_n ,then the task is average divided into J=Q/l sub-task, and each sub-task products l products. Sub-task will be allocated to each candidate enterprise one by one. Secondly, the candidate enterprises are arranged in non-descending order by the cost $\mathbf{c}_i$, that is, the smaller the enterprise's cost is, the smaller the number that stands for the enterprise is. Thirdly, the solution to the question is expressed by a string x[j],j=1,2,…,n.x[j]=l means No.j sub-task is allocated to No.l enterprise. The viable solution that randomly generated is implementing computing in the learning model as follows:

Step1. set k=0 , y=1.

Step2. get the number z that is counting if x[j] = y. Determine whether a_j *T is greater than z*l or not? If yes, turn to Step4; no, turn to Step 3.

Step3. y++, turn to Step 2.

Step4. Determine whether x[k] is greater than y or not? If yes, turn to Step 6; no, turn to Step 5.

Step5. k++, turn to Step 4.

Step6. x[k]-1. Then determine x [j], j = 1,2, ..., whether or not to meet the constraints? If yes, turn to Step 5; no, x[k]+1, then turn to Step 7.

Step7. Determine whether x[k] is satisfied the graduate qualification or not? If satisfied, then the individuals can enter the next model.

3.2 Realization of Competing Model

We adopt genetic algorithm to realize the competing model. Above all, convert x[j],j=1,2,…,n , to $P_j(t)$. And its adapting function is constructed as follows:

$$f(P_i'(t)) = \sum_{i=1}^{n} \sum_{t=1}^{t_i(x_i)} (c_i * P_i'(t) * (1+\lambda_i)^t)$$

hereinto $P_j'(t)$ is a correct to $P_j(t)$.

Adopting correct algorithm as next: if No.j enterprise can't finish its task, then the extra tasks are priority allocated to the enterprise that has greater parameter $a_i' = a_i - \mu * \frac{q_i}{T}$. So the incapable solution will become capable solution. Then, considering the replacement rate that the corrected chromosomes displace the former chromosomes. The replacement rate can change from 0% to 100%. We accept 5%.[8]

In this way, after appointed generations, we can determine whether the solutions are satisfied or not. If we don't satisfactory the solutions, put the individuals implementing computing in the learning model again, until the satisfactory solution is get.

3.3 Analyses

It is worth noting that: if the graduate qualification is m_{max} and w_{max}, the algorithm will become the greed algorithm; if we don't set the graduate qualification, the algorithm will become the hybrid genetic algorithm. In practice, we often have the following experience: when the global is too concerned, the local will be not clear; but when the local is too concerned, the global will be not clear. With a view to better results, the algorithm is coordinating the correlation of global and local by the graduate qualification. At the same time, the computing process is become an open process. When switch from the learning model to the competing model, we can control the algorithm complexity by mediating the graduate qualification. The practical significance is: we can solve the tasks that have different degree of emergency with the same set of algorithms.

1. Compared with the greedy algorithm: in fact, the greedy algorithm can only get the second-best solution under normal circumstances. But in this algorithm, it can out of local optimal solution by the competing model.
2. Compared with the genetic algorithm: the GA's main flaw is precocious problem and the slow convergence's problem.

The cause of precocious problem is: the individuals that maybe the optimal solutions are eliminated early, because of their bad early adaptability. But in this algorithm, those individuals can get high early adaptability by "studying" in the learning model.

Furthermore, the slow convergence's problem is settled by putting the individuals from the competing model into the learning model again. Sometimes, the optimal solution appears very small probability. For the strong randomness of GA, the optimal solution is difficult to search out in the whole GA running. In this algorithm, the

possibility that optimal solution appears is greatly enhanced by repeated local search in the learning model; and once the optimal solution appears, it will be reserved for its high viability.

4 Result

To consider the production task: product 2500 tents in 10 days. Assuming there are 6 candidate enterprises, their parameters shown in the table below.

Table 1. Enterprises' parameter table

	cost	**production capacity**	**ë**
Enterprise A	8	200	0.07
Enterprise B	6	120	0.06
Enterprise C	10	160	0.08
Enterprise D	9	220	0.06
Enterprise E	8	210	0.06
Enterprise F	7.5	130	0.06

In our experiment, flock size is 100, chiasma probability is 0.65, and aberrance probability is 0.05. General descendible generation is 60. Recording best individual we have, and recording the individual's evaluations: viability evaluation is m_{maxw}, and individuation evaluation is w_{max}. The graduate qualification is set to: viability target is larger than m_{max}*0.75%, individuation target is larger than w_{max}*0.75%. The results will be as follows.

Table 2. Algorithms' result

	learning-competing model	**greed algorithm**	**standard genetic algorithm**
Total cost	18284.26	18310.22	18303.39
TIME/s	24	1	24

By the table above, we can find the algorithm is obviously better than both standard genetic algorithm and greed algorithm. The algorithm is not slower than the two others, while its total cost is less than the two others'.

References

1. Ronald, G.A., Chen, J.Q.: Dynamic Task Assignment for Throughput Maxization with Worksharing. European Journal of Operational Research 168, 853–869 (2006)
2. Gutin, G., Punnen, A.P.: The Traveling Sales-man Problem and Its Variations. Kluwer Academic Publishers, Boston (2002)
3. Tal, S., Steven, J.R., Andrew, G., Sparks, Passino, K.M.: Multiple task assignments for Cooperating Uninhabited Aerial Vehicles Using Genetic Algorithms. Computers & Operations Research 33, 3252–3269 (2006)

4. Cormen, T.H., Leiserson, C.E.: Introduction to Algorithms. MIT Press, Cambridge (2002)
5. Chen, T.Y., Wang, Z.: The Implementation of Heuristic Algorithm Based on Learning-competing Mode. J. Huazhong Univ. of Sci. & Tech. 35, 38–40 (2007)
6. Xiong, J.: The Research on the Task Assignment in National Economy Mobilization Simulation and Training System. Master's degree thesis, Huazhong University of Science and Technology, Wuhan (2006)
7. Holland, J.H.: Adaptation in Natural and Artificial Systems. The University of Michigan Press, ANN Arbor (1975)
8. Michalewicz, Z.: Genetic Algorithms + data structures = evolution programs. Springer, Berlin (1994)

Comparisons of Machine Learning Methods for Electricity Regional Reference Price Forecasting

Ke Meng[1], Zhaoyang Dong[2], Honggang Wang[3], and Youyi Wang[4]

[1] School of Information Technology & Electrical Engineering, the University of Queensland, St. Lucia, QLD 4072, Australia
{kemeng@itee.uq.edu.au}
[2] Department of Electrical Engineering, The Hong Kong Polytechnic University, Hong Kong
{zydong@ieee.org}
[3] State Key Laboratory of Chemical Engineering, East China University of Science & Technology, Shanghai 200237, China
{hgwang@mail.ecust.edu.cn}
[4] School of Electrical & Electronic Engineering, Nanyang Technological University, Singapore
{eyywang@ntu.edu.sg}

Abstract. Effective and reliable electricity price forecast is essential for market participants in setting up appropriate risk management plans in an electricity market. In this paper, we investigate two state-of-the-art statistical learning based machine learning techniques for electricity regional reference price forecasting, namely support vector machine (SVM) and relevance vector machine (RVM). The study results achieved show that, the RVM outperforms the SVM in both forecasting accuracy and computational cost.

Keywords: Electricity reference price forecasting, Support vector machine, Relevance vector machine.

1 Introduction

With the deregulation of power industry, price forecasting has been increasingly important for market participants, especially generators and the system operator. Price forecasting provides key input toward proper risk management plans and helps determine the optimal bidding strategy into the market. In addition to the economic price modeling methods such as mean reversion and jump diffusion, recent advances in price forecasting uses regression models [1], neural networks [2,3], and data mining [4,5] based techniques. Statistic time series models including ARIMA [6] and GARCH models [7] have been proven to be effective with satisfactory prediction performance. According to the study in [8], the machine learning based methods outperform the time series approaches. In this paper, we investigate two state-of-the-art statistical learning based machine learning techniques for electricity regional reference price forecasting. The two approaches we consider here are both kernel-based, namely support vector machine (SVM) [9] and relevance vector machine (RVM) [10].

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 827–835, 2009.

The paper is organized as follows, after the introduction section; the machine learning methods used in the study, SVM and RVM, are reviewed for completeness, followed by benchmark dataset test. Then an evaluation study of the machine learning methods is carried out with electricity reference price from the Australian National Electricity Market (NEM). Conclusions are drawn in the last section.

2 Overview of Machine Learning Methods

This section provides a brief introduction to the two kernel-based machine learning methods that are considered in this paper. To begin, sample dataset is given $\{\mathbf{x}_i, t_i\}_{i=1}^{l}$, $\mathbf{x}_i \in R^d$ is input vector, and $t_i \in R$ is the associated desired output value.

2.1 Support Vector Machine (SVM)

SVM, proposed in the middle of 1990s, is considered as an innovation of machine learning method, which is based on the statistical learning theory. It approximates the relation curve by using only a small amount of training data, which are known as the support vectors (SVs). Furthermore, SVM can effectively avoid the over-fitting problem by reaching a proper trade-off between empirical accuracy and model complexity [9]. Therefore, SVM usually show better performance than many traditional methods. Here we describe the regression function of SVM for time series forecasting.

For given sample dataset, the SVM model used for function approximation is:

$$t_i = y(\mathbf{x}_i) = \omega_i \Phi(\mathbf{x}_i) + b_i, \tag{1}$$

where, b_i is output bias; ω_i is model weights; Φ is nonlinear mapping from input space to high dimensional feature space.

The ω_i and b_i can be determined by given training samples. This is accomplished through optimization of the following structural risk function, where C is a pre-specified value:

$$Minimize \quad E = \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{l}\left|t_i - y(\mathbf{x}_i)\right|_\varepsilon . \tag{2}$$

In the function, we adopted the Vapnik's linear loss function with ε-intensive zone as a measure for empirical error:

$$\left|t_i - y(\mathbf{x}_i)\right|_\varepsilon = \begin{cases} 0, \ if \left|t_i - y(\mathbf{x}_i)\right| < \varepsilon \\ \left|t_i - y(\mathbf{x}_i)\right| - \varepsilon, otherwise \end{cases}. \tag{3}$$

After introducing two positive slack variables μ_i and μ_i^*, the structural risk function can be converted to:

$$Minimize \quad E = \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{l}\left(\mu_i + \mu_i^*\right) \tag{4}$$

$$Subject\ to \quad \begin{cases} \left[\omega \cdot \Phi\left(\mathbf{x}_i\right) + b\right] - t_i \le \varepsilon + \mu_i^* \\ t_i - \left[\omega \cdot \Phi\left(\mathbf{x}_i\right) + b\right] \le \varepsilon + \mu_i \\ \mu_i, \mu_i^* \ge 0, i = 1, 2, \cdots, l \end{cases} \tag{5}$$

Introducing Lagrange multipliers, the decision function can be expressed as:

$$y\left(x, \alpha_i, \alpha_i^*\right) = \sum_{i=1}^{l}\left(\alpha_i - \alpha_i^*\right)K\left(\mathbf{x}_i, \mathbf{x}\right) + b\,, \tag{6}$$

where α_i, α_i^* are the Lagrange multipliers.

Using Mercer's theorem, the regression is obtained by solving a finite dimensional QP problem in the dual space avoiding explicit knowledge of the high dimensional mapping and using only the related kernel function [9]. In this paper, we select RBF kernel for all approaches, and the Lagrange multiplier can be obtained by maximizing the following form:

$$Maximize$$
$$-\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\left(\alpha_i - \alpha_i^*\right)\left(\alpha_j - \alpha_j^*\right)K\left(\mathbf{x}_i, \mathbf{x}_j\right) + \sum_{i=1}^{l}\left(\alpha_i - \alpha_i^*\right)t_i - \varepsilon\sum_{i=1}^{l}\left(\alpha_i + \alpha_i^*\right) \tag{7}$$

$$Subject\ to \quad \begin{cases} \sum_{i=1}^{l}\left(\alpha_i^* - \alpha_i\right) = 0 \\ 0 \le \alpha_i^*, \alpha_i \le C \end{cases}, i = 1, 2, \cdots, l \tag{8}$$

Through adjusting the parameters C and ε, the generalized performance can be controlled in high-dimension space.

2.2 Relevance Vector Machine (RVM)

RVM, a new statistical learning technique based on Bayesian estimation theory, is developed for regression and classification problems. The key feature is that it can yield a solution function that depends on only a very small number of training samples; relevance vectors (RVs) [10]. It shows better performance than many other methods with higher accuracy and faster speed.

For given dataset, the output can be expressed as:

$$\begin{cases} y\left(\mathbf{x}, \boldsymbol{\omega}\right) = \sum_{i=1}^{l}\omega_i K\left(\mathbf{x}, \mathbf{x}_i\right) + \omega_0 \\ t_i = y\left(\mathbf{x}_i, \boldsymbol{\omega}\right) + \varepsilon_i \end{cases}, \tag{9}$$

where $K(\cdot)$ is kernel function; ω_i is model weights; and ε_i is output noise.

Assume $p(t \mid \mathbf{x})$ is Gaussian $N\left(t \mid y(\mathbf{x}), \sigma^2\right)$. The likelihood function of the dataset can then be written as:

$$p\left(\mathbf{t} \mid \boldsymbol{\omega}, \sigma^2\right) = \left(2\pi\sigma^2\right)^{-l/2} \exp\left\{-\frac{\left\|\mathbf{t} - \boldsymbol{\Phi}\boldsymbol{\omega}\right\|^2}{2\sigma^2}\right\}, \tag{10}$$

where $\mathbf{t} = (t_1, \cdots, t_l)$; $\boldsymbol{\omega} = (\omega_0, \cdots, \omega_l)$; $\boldsymbol{\Phi}_{l\times(l+1)} = \left[\boldsymbol{\varphi}(\mathbf{x}_1), \cdots, \boldsymbol{\varphi}(\mathbf{x}_l)\right]$; $\boldsymbol{\varphi}(\mathbf{x}_i) = \boldsymbol{\varphi}_i = \left[1, K(\mathbf{x}_i, \mathbf{x}_1), \cdots, K(\mathbf{x}_i, \mathbf{x}_l)\right]$.

From the structural risk minimization theory of statistics learning, maximum-likelihood estimation of value $\boldsymbol{\omega}$ and σ^2 without constraints will generally lead to severe over-fitting. In order to improve model generalization ability, RVM defines Gaussian prior probability distribution over the weights [11], which is the key feature of RVM and is ultimately responsible for its sparsity properties [10].

$$p(\boldsymbol{\omega} \mid \boldsymbol{\alpha}) = \prod_{i=0}^{l} N\left(\omega_i \mid 0, \alpha_i^{-1}\right), \tag{11}$$

where $\boldsymbol{\alpha}$ are vector of hyperparameters.

For the given prior probability distribution and likelihood distribution, posterior probability distribution for calculating the weights by Bayesian inference can be expressed as:

$$\begin{cases} p\left(\boldsymbol{\omega} \mid \mathbf{t}, \boldsymbol{\alpha}, \sigma^2\right) = \dfrac{p\left(\mathbf{t} \mid \boldsymbol{\omega}, \sigma^2\right) p(\boldsymbol{\omega} \mid \boldsymbol{\alpha})}{p\left(\mathbf{t} \mid \boldsymbol{\alpha}, \sigma^2\right)} \\ p\left(\boldsymbol{\omega} \mid \mathbf{t}, \boldsymbol{\alpha}, \sigma^2\right) = N(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \end{cases}, \tag{12}$$

where $\boldsymbol{\mu} = \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{\Phi}^T\mathbf{t}$; $\boldsymbol{\Sigma} = \left(\sigma^{-2}\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \mathbf{A}\right)^{-1}$; $\mathbf{A} = diag\left(\alpha_0, \alpha_1, \cdots, \alpha_l\right)$, $\alpha_i \to \infty$, $\mu_i = 0$.

The weights estimation can be achieved by the mean value of posterior probability distribution $\boldsymbol{\mu}$, and uncertainty of best weights values $\boldsymbol{\Sigma}$ can be used to represent the uncertainty of model prediction. In order to estimate the model weights, we need to estimate the best values of hyperparameters whose likelihood distribution can be calculated according to Bayesian framework which is the marginal likelihood [12]:

$$p\left(\mathbf{t} \mid \boldsymbol{\alpha}, \sigma^2\right) = \int p\left(\mathbf{t} \mid \boldsymbol{\omega}, \sigma^2\right) p(\boldsymbol{\omega} \mid \boldsymbol{\alpha})\, d\boldsymbol{\omega} = N(0, \mathbf{C}), \tag{13}$$

where $\mathbf{C} = \sigma^2\mathbf{I} + \boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^T$.

We can find the best possible hyperparameters $\boldsymbol{\alpha}_{MP}, \sigma^2_{MP}$ by the type II maximum hyperparameters likelihood method [12]. Here we adopt an iterative re-estimation approach, using the direct differentiation and rearranging:

$$\alpha_i^{new} = \frac{\gamma_i}{\mu_i^2}, \tag{14}$$

where $\gamma_i = 1 - \alpha_i \Sigma_{ii}$.

For the noise variance, the re-estimate can be calculated by:

$$\left(\sigma^2\right)^{new} = \frac{\left\|\mathbf{t} - \mathbf{\Phi}\boldsymbol{\mu}\right\|}{l - \sum_i \gamma_i}, \tag{15}$$

In practice, many of the α_i approach infinity and according to (11), $p\left(\boldsymbol{\omega} \mid \mathbf{t}, \boldsymbol{\alpha}, \sigma^2\right)$ becomes in finitely peaked at zero. The ω_i corresponding to these values can be regarded equal to 0. And the dataset corresponding to the non-zero ω_i are the RVs, like the SVs in SVM.

If the above hyperparameters estimation converges, we can predict the new dataset x_* according to weights posterior and best hyperparameters $\boldsymbol{\alpha}_{MP}, \sigma_{MP}^2$. The prediction distribution can be calculated by:

$$p\left(t_* \mid \mathbf{t}, \boldsymbol{\alpha}_{MP}, \sigma_{MP}^2\right) = \int p\left(t_* \mid \boldsymbol{\omega}, \sigma_{MP}^2\right) p\left(\boldsymbol{\omega} \mid \mathbf{t}, \boldsymbol{\alpha}_{MP}, \sigma_{MP}^2\right) d\boldsymbol{\omega}. \tag{16}$$

Because the two integral parts are all Gauss distribution, so

$$p\left(t_* \mid \mathbf{t}, \boldsymbol{\alpha}_{MP}, \sigma_{MP}^2\right) = N\left(\mu_*, \sigma_*^2\right), \tag{17}$$

where $\mu_* = \boldsymbol{\mu}^T \boldsymbol{\varphi}\left(\mathbf{x}_*\right)$, $\sigma_*^2 = \sigma_{MP}^2 + \boldsymbol{\varphi}\left(\mathbf{x}_*\right)^T \boldsymbol{\Sigma} \boldsymbol{\varphi}\left(\mathbf{x}_*\right)$. The prediction values are $y\left(\mathbf{x}_*; \boldsymbol{\mu}\right)$.

In the following sections, the proposed method will be tested with chaos time series generated by mathematic function before being applied to analyze some realistic highly volatile electricity price date series to show its effectiveness under different conditions.

3 Benchmark Dataset

In order to mathematically compare the performance of these two techniques, a benchmark function is used here.

$$\operatorname{sinc}\left(x\right) = \frac{\sin\left(x\right)}{x}, x \in \left[-10, 10\right]. \tag{18}$$

Total 100 data samples are collected including random noise with 0.1 deviations. For comparison purpose, the RBF kernel is selected and associated width is chosen as 3. Here a common criterion is used to evaluate the accuracy of price forecasting: Root Mean Square Error (RMSE).

$$RMSE = \sqrt{\frac{1}{l}\sum_{i=1}^{l}(t_i - \overline{t}_i)^2} \ . \tag{19}$$

Note that both of them are associated with a few parameters that need to be fine-tuned for best fitting performance. Furthermore, the RVM outperforms the SVM in fitting accuracy, and it needs less number of RVs than that of SVs for SVM.

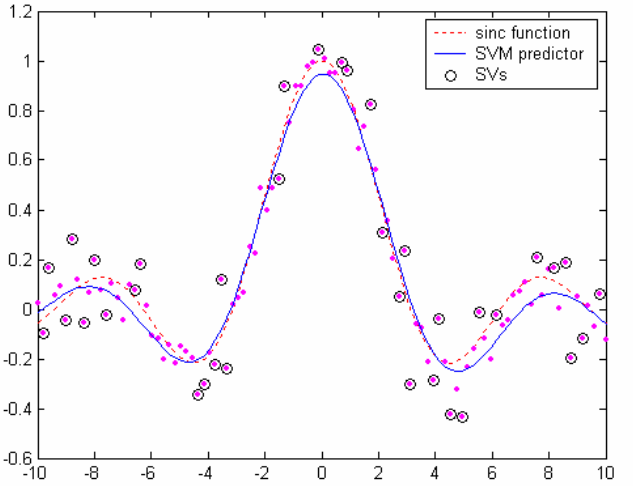


Fig. 1. Sinc Function (with noise) Fitting with SVM

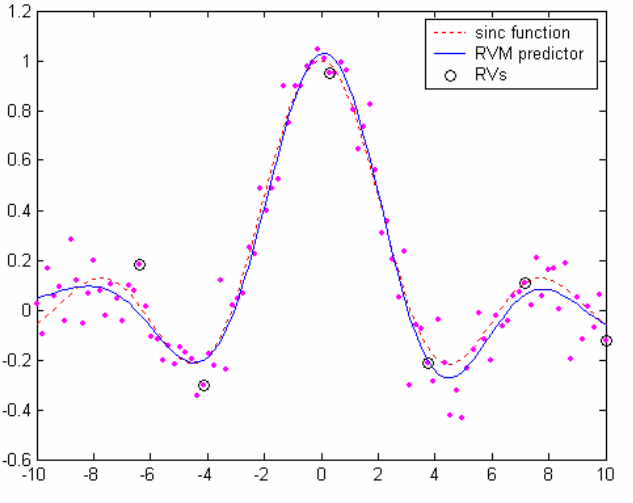


Fig. 2. Sinc Function (with noise) Fitting with RVM

Table 1. Comparisons of Sinc Function (with noise) Fitting with SVM and RVM

Methods	RMSE	Support Vectors / Relevance Vectors
SVM	0.04791	36
RVM	0.03818	6

4 Electricity Market Price Forecasting Case Study

In this section, the two machine-learning methods are tested with the Queensland (QLD) electricity regional reference price. The QLD market is part of the Australian National Electricity Market (NEM) which is composed of the states of New South Wales, Victoria, South Australia, Queensland, and Tasmania. The market data are taken from National Electricity Market Management Company Limited (NEMMCO) website (http:\\www.nemmco.com.au\.). It is widely accepted that electricity price is highly volatile and difficult to predict. Here two methods will be used on such dataset to compare the capability in handling electricity price data series.

4.1 Data Description and Pre-processing

The simulation experiment is performed based on history data of the Australian NEM. The NEM is a day-ahead energy-only market. Trading in the NEM is based on a 30 minute trading interval. In NEM, generators submit offers every five minutes every day. A dispatch price is determined every 5 minutes: 288 prices in a day, and six dispatch prices are averaged every half-hour to determine the spot price for each of the regional reference price (RRP). NEMMCO uses the spot price as the basis for the settlement of financial transaction for all energy traded in the NEM. In this case, 48 market prices should be predicted in day-ahead price forecasting.

The collected data cover three months in 2007, namely the August, September, and October. For the 4,416 collected price data, the pre-processing is carried out in order to reduce the influence of spike prices. The price data are normalized after being transferred to the logarithm scale. Here we take the four input values as arguments and return one output value. The four inputs are the one-day-ahead, two-days-ahead, three-days-ahead, and one-week-ahead electricity price at same time as the predicted value every day. Finally, 96 data points (two days) are predicted for each model from 0:00 to 24:00 next day and totally 96 electricity price forecast models are generated for each method.

$$p(x) = y\left[p(x-1), p(x-2), p(x-3), p(x-7) \right]. \tag{20}$$

4.2 Simulation Results

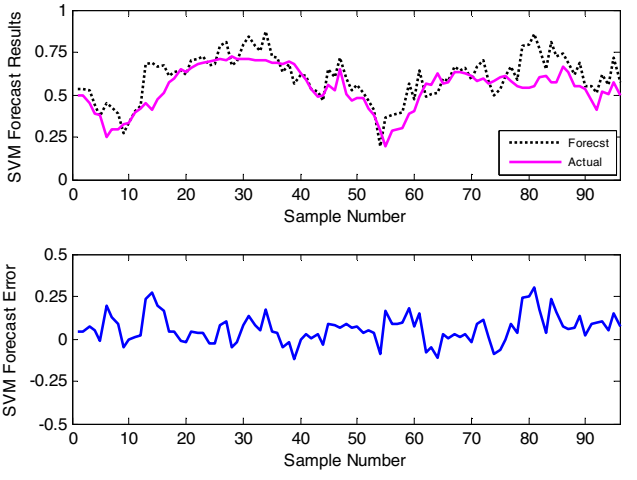


Fig. 3. QLD Electricity Reference Price Forecast with SVM

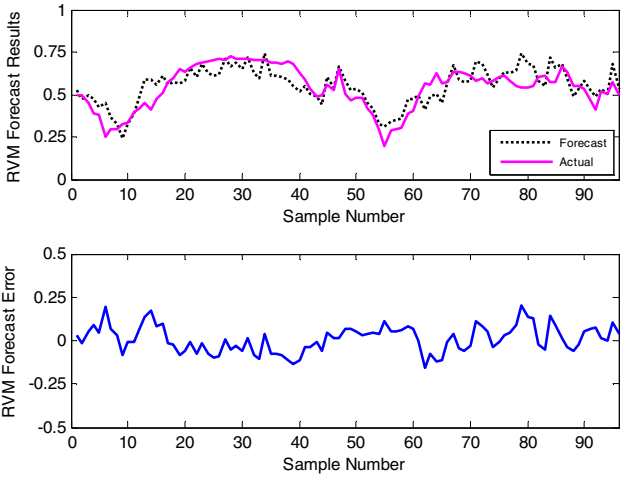


Fig. 4. QLD Electricity Reference Price Forecast with RVM

Table 2. Comparisons of Electricity Price Forecast with SVM and RVM

Methods	RMSE	Time (s)	Support Vectors / Relevance Vectors
SVM	0.10391	75.2268	55
RVM	0.07634	3.7724	8

5 Conclusion

As clearly shown in Table 2, it can be concluded that, the RVM outperforms the SVM in both forecast accuracy and computational cost. And the RVM need far fewer RVs than the SVs for SVM. The promising prediction performance on the price data illustrated the efficiency of the two methods. However, note that each approach is typically associated with a few model parameters that need to be fine-tuned.

References

1. Hu, Z., Yu, Y., Wang, Z., Sun, W., Gan, D., Han, Z.: Price Forecasting using an Integrated Approach. In: Proc. Electric Utility Deregulation, Restructuring Power Technologies (2004)
2. Meng, K., Dong, Z., Wong, K.P.: Self-adaptive RBF Neural Network for Short-term Electricity Price Forecasting, accepted by IET Generation. Transmission & Distribution 51, 120–127 (2008)
3. Zhang, B., Dong, Z.: An Adaptive Neural-wavelet Model for Short Term Load Forecasting. International Journal of Electric Power Systems Research 59, 121–129 (2001)
4. Lu, X., Dong, Z., Li, X.: Electricity Market Price Spike Forecast with Data Mining Techniques. International Journal of Electric Power Systems Research 73, 19–29 (2005)
5. Zhao, J., Dong, Z., Li, X., Wong, K.P.: A Framework for Electricity Price Spike Analysis with Advanced Data Mining Methods. IEEE Transactions on Power Systems 22, 376–385 (2007)
6. Contreras, J., Espinola, R., Nogales, F.J., Conejo, A.J.: ARIMA Models to Predict Next-day Electricity Prices. IEEE Transactions on Power Systems 18, 1014–1020 (2003)
7. Garcia, R.C., Contreras, J., Akkeren, M.V., Garcia, J.: A GARCH Forecasting Model to Predict Day-ahead Electricity Prices. IEEE Transactions on Power Systems 20, 867–874 (2005)
8. Li, G., Liu, C., Mattson, C., Lawarree, J.: Day-ahead Electricity Price Forecasting in a Grid Environment. IEEE Transactions on Power Systems 22, 266–274 (2007)
9. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
10. Tipping, M.E.: Sparse Bayesian Learning and Relevance Vector Machine. Journal of Machine Learning Research 1, 211–244 (2001)
11. Neal, R.M.: Bayesian Learning for Neural Networks. Springer, New York (1996)
12. Berger, J.O.: Statistical Decision Theory and Bayesian Analysis. Springer, New York (1985)

Prediction of Chaotic Time Series Based on Neural Network with Legendre Polynomials

Hongwei Wang and Hong Gu

School of Electronic and Information Engineering, Dalian University of Technology
Dalian, Liaoning, China
wanghw@dlut.edu.cn

Abstract. In this paper, a modeling method based on the orthogonal function neural network is proposed. Legendre orthogonal polynomials are selected as the basic functions of the neural network. Kalman filtering algorithm with singular value decomposition is used to confirm the parameters of orthogonal function neural network in order to avoid error delivery and error accumulation. To demonstrate the performance of this modeling method, the simulation on Mackey-Glass chaotic time series is performed. The results show that this method provides effective and accurate prediction.

Keywords: Neural network, Legendre orthogonal polynomials, Kalman filtering, Singular value decomposition, Chaotic time series.

1 Introduction

The prediction methods of time series have been widely applied to many fields. The examples can be found in weather forecast, speech coding and noise cancellation, etc. When a time series is chaotic, it implies that the laws underlying the time series can be expressed as a deterministic dynamical system. However, these deterministic equations are not usually given explicitly. Predictions rely on the empirical regularities derived from the experimental observations of the real system.

Different methods have been used in the prediction of chaotic time series, such as radical basis function neural network[1], recurrent neural network[2] and wavelet neural network[3-5], which possess the abilities to approximate nonlinear systems. However, the specified solution based on the training of these networks is satisfactory to the given signal but unsatisfactory to the new input signal. These networks can only approximate the given trajectory based on the training of the sampling points. In recent years, the chaotic nonlinear systems have been discussed in many ways. Genetic programming modeling algorithm[6]was proposed to forecast chaotic time series by optimizing the model structure and the parameters of the prediction model. A new technique based on the least squares support vector machines (LS-SVM)[7] was proposed to make one-step or multi-step prediction of chaotic time series. In addition, different filtering methods have been employed in the prediction of chaotic time series such as neural Volterra filter[8] and Kalman filter [9]. In this research area, there have been also many attempts to create prediction models of chaotic time series based on statistical methods [10,11].In the prediction of chaotic time series, there are two

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 836–843, 2009.

factors to achieve a certain performance goal, including satisfactory prediction precision and better robustness.

In the paper, Legendre orthogonal polynomial functions are selected as the basic functions of the orthogonal function neural network. The parameters of the orthogonal function neural network are confirmed by using Kalman filtering algorithm with singular value decomposition, which is used to avoid the error delivery and the error accumulation. To demonstrate the performance of the forecasting method, the simulation on Mackey-Glass chaotic time series is performed. The results show that this method provides effective and accurate prediction. The remainder of this paper is organized as follows. In section 2, the structure of the orthogonal function neural network is introduced. The parameter estimation of neural network is presented in Section 3. To demonstrate the performance, the prediction simulation on Mackey-Glass chaotic time series is performed in Section 4. Some concluding remarks are made in Section 5.

2 The Description of the Orthogonal Neural Network

The orthogonal function neural network can approximate to any nonlinear function on the tight set, which has simple structure, fast convergence with the comparison of the common BP neural network. The structure of the orthogonal function neural network is shown as Figure 1. In the paper, Legendre orthogonal polynomials are selected as the basic functions of the orthogonal function neural network.

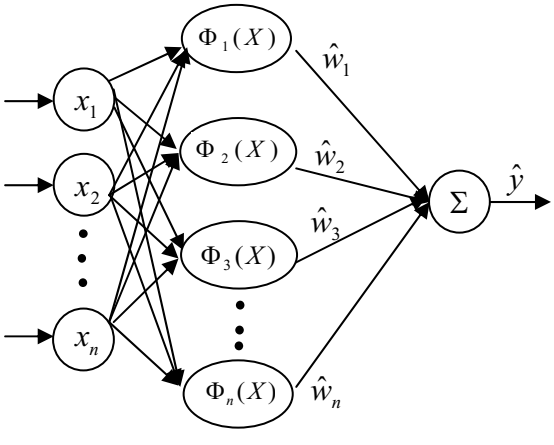


Fig. 1. The structure diagram of the orthogonal function neural network

The Legendre polynomials are defined as the following form.

$$\begin{cases} P_0(x) = 1 \\ P_i(x) = \dfrac{1}{2^i i!}\dfrac{d^i}{dx^i}\left[\left(x^2-1\right)^i\right] \end{cases} \quad i = 1,2,....c\ ,\ x \in [-1,1] \tag{1}$$

For the Legendred polynomial, its recursive property is

$$P_i(x) = a_{i-1}xP_{i-1}(x) - b_{i-1}P_{i-2}(x),\ i \geq 2 \tag{2}$$

where $P_0(1) = 1,\ P_2(1) = x,\ a_i = 2 - \frac{1}{i}$, and $b_i = 1 - \frac{1}{i}$.

Equation (2) is arranged as the following form.

$$P_i(x) = \left[(2i-1)xP_{i-1}(x) - (i-1)P_{i-2}(x)\right]/i \tag{3}$$

According to the definition of ONN, the global output of the orthogonal function neural network is defined as Equation (4).

$$y = \sum_{i=1}^{c} W_i^T \Phi_i(x) \tag{4}$$

where $\Phi_i(x) = P_{1i}(x_1) \times P_{2i}(x_2) \times \cdots \times P_{ni}(x_n) = \prod_{j=1}^{n} P_{ji}(x_j)$, $P_{ji}(x_j)$ is Legendre polynomial, $P_{j1}(x_j) = 1, P_{j2}(x_j) = x_j,\ P_{j3}(x_j) = (3x_j^2 - 1)/2, P_{j(i+1)}(x_j) = [(2i-1)x_j P_{ji}(x_j) - (i-1)P_{j(i-1)}(x_j)]/i$ $i = 1, 2, ..., c$, $j = 1, 2, ..., n$.

The global output of the neural network is also written as the following form.

$$\overline{y} = A \cdot \theta \tag{5}$$

where A is the input matrix, satisfying $A = \begin{bmatrix} \Phi_1(1) & \Phi_2(1) & \cdots & \Phi_c(1) \\ \Phi_1(2) & \Phi_2(2) & \cdots & \Phi_c(2) \\ \vdots & \vdots & \cdots & \vdots \\ \Phi_1(N) & \Phi_2(N) & \cdots & \Phi_c(N) \end{bmatrix}$; θ is the parameter vector, $\theta = (W_1, W_2, \cdots, W_c)^T$; N is the number of the sampling data.

3 The Recursive Estimation of Parameters of Orthogonal Function Neural Network

The parameters of neural network model are determined on basis of the minimum of mean squared error between the output of the system and the output of neural network. The parameter vector θ is estimated by the static Kalman filtering algorithm.

$$\theta_k = \theta_{k-1} + K(k-1)[y_k - A^T(k)\theta_{k-1}] \tag{6}$$

$$P(k) = P(k-1) - P(k-1)A(k)S^{-1}(k)A^T(k)P(k-1) \tag{7}$$

$$S(k) = A^T(k)P(k-1)A(k) + 1 \tag{8}$$

$$K(k-1) = P(k)A(k) \quad k = 1, 2, ..., n \tag{9}$$

where θ is the parameter vector, $\theta = (W_1, W_2, \cdots, W_c)^T$; y_k is the real output of the kth sampling point; P is the covariance matrix, $P \in R^{c \times c}$; $P(0) = \alpha I$ (I is an identity matrix and α is a large positive number); $A^T(k) = [\Phi_1(k), \Phi_2(k), \cdots, \Phi_c(k)]^T$.

The key problem of the Kalman filtering algorithm is how to calculate the covariance matrix $P(k)$. Equation (7) includes two positive definite and symmetrical matrixes, namely $P(k)$ and $P(k-1)$. Because of the limit of character length of the computer, and the delivery and the accumulation of cut-off errors, $P(k)$ may lose positive definite capability, and then the stability of the Kalman filtering algorithm will be unsatisfactory. In order to avoid error delivery and error accumulation, the covariance matrix $P(k)$ is calculated to improve the quality of the Kalman filtering algorithm by means of singular value decomposition.

Singular value decomposition has been widely applied to many calculation fields such as linear least square algorithm, ill-condition equations and so on. The basic concepts and properties are shown as follows.

If A is a real matrix, $A \in R^{N\times M}$ ($N \le M$), then A is decomposed into the following form.

$$A = U\Lambda V^T, \quad \Lambda = \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix} \tag{10}$$

where U and V are the column orthogonal matrixes, $U=[u_1,u_2,..,u_N]\in R^{N\times N}$, $V=[v_1,v_2,...,v_M]\in R^{M\times M}$, satisfying $U^TU=I$, $V^TV=I$; S is the diagonal matrix, $S=diag(\delta_1,\delta_2,...,\delta_N)$, where $\delta_i(i=1,2,...,N)$ is the singular value of matrix A, satisfying $\delta_1 \ge \delta_2 \ge ... \ge \delta_N$.

The column vectors of matrix U and matrix V are defined as left singular vectors and right singular vectors, respectively. If A is a positive definite and symmetrical matrix, then A can be decomposed into the following equation.

$$A = U\Lambda U^T = UD^2U^T \tag{11}$$

where the left singular vectors are equivalent to the right singular vectors.

Applying singular value decomposition to $P(k)$ in (7), it yields the following equation.

$$P(k) = U(k)D(k)U^T(k) \tag{12}$$

where $U(k)$ is an orthogonal matrix, $U(k)=[U_1,U_2,\cdots,U_c]\in R^{c\times c}$, satisfying $U^TU=UU^T=I$. The matrix D is a diagonal matrix, $D=diag(\sigma_1^2,\sigma_2^2,\cdots,\sigma_c^2)$, in which $\sigma_i^2(i=1,2,...,c)$ are singular values of $P(k)$, satisfying $\sigma_1^2 \ge \sigma_2^2 \ge \cdots \ge \sigma_c^2 \ge 0$.

Equation (12) is written as the following form

$$\begin{aligned} U(k)D(k)U^T(k) &= U(k-1)D(k-1)U^T(k-1) - \frac{U(k-1)g(k)g^T(k)U^T(k-1)}{S(k)} \\ &= U(k-1)[D(k-1) - \frac{g(k)g^T(k)}{S(k)}]U^T(k-1) \end{aligned} \tag{13}$$

where $g(k)$ and $f(k)$ satisfy

$$\begin{cases} g(k) = [g_1(k), g_2(k), \cdots, g_c(k)]^T = D(k-1) f(k) \\ = [\sigma_1^2(k-1) f_1(k-1), \sigma_2^2(k-1) f_2(k-1), \cdots, \sigma_c^2(k-1) f_c(k-1)]^T \\ f(k) = [f_1(k), f_2(k), \cdots, f_c(k)]^T = U^T(k-1) A(k) \end{cases} \tag{14}$$

The right part of (13) can be rewritten as the following equation by using singular value decomposition.

$$D(k-1) - \frac{g(k) g^T(k)}{S(k)} = \overline{U}(k) \overline{D}(k) \overline{U}^T(k) \tag{15}$$

where $\overline{U}(k)$ is an orthogonal matrix, $\overline{U}(k) = [\overline{U}_1, \overline{U}_2, \cdots, \overline{U}_c] \in R^{c \times c}$, satisfying $\overline{U}^T \overline{U} = \overline{U} \overline{U}^T = I$. The matrix $\overline{D}$ is a diagonal matrix, $\overline{D} = diag(\overline{\sigma}_1^2, \overline{\sigma}_2^2, \cdots, \overline{\sigma}_c^2)$, where $\overline{\sigma}_i^2 (i = 1, 2, \ldots c)$ are singular values, satisfying $\overline{\sigma}_1^2 \geq \overline{\sigma}_2^2 \geq \cdots \geq \overline{\sigma}_c^2 \geq 0$.

Equation (13) is transformed into the following equation.

$$U(k) D(k) U^T(k) = U(k-1) \overline{U}(k) \overline{D}(k) \overline{U}^T(k) U^T(k-1) \tag{16}$$

$$\begin{cases} D(k) = \overline{D}(k) \\ U(k) = U(k-1) \overline{U}(k) \end{cases} \tag{17}$$

The matrix $K(k)$ is calculated as the following equation.

$$K(k) = U(k) D(k) U^T(k) X(k) \tag{18}$$

4 Simulation

Mackey-Glass time series which are derived from the differential equation are shown as the following equation.

$$\dot{x} = \frac{\alpha x(t-\tau)}{1 + x^{\gamma}(t-\tau)} - \beta x(t) \tag{20}$$

The problem is how to use the past values of x to forecast the future value of x. The same example has been compared with the published results in the paper [13]. To obtain the time series value at each integer point, the fourth Runge-Kutta method is applied to get the numerical solution of Equation (16). The initial values of the system (20) are $x(0) = 1.2$, $\alpha = 0.2$, $\beta = 0.1$, $\gamma = 10$. τ is the time-delay parameter. If $\tau \geq 17$, Equation (16) shows the chaotic phenomenon.

Four variables $x(t-18), x(t-12), x(t-6), x(t)$ are selected as the input variables of the orthogonal neural network, and the variable $x(t+6)$ as the output variable of the orthogonal neural network. We use 1000 simulation data points to build the orthogonal neural network of Mackey-Glass time series.

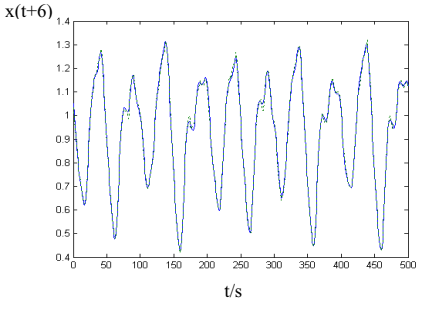


Fig. 2. The output of the fuzzy model and the training points

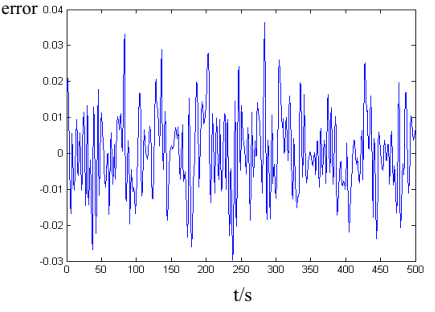


Fig. 3. The error diagram

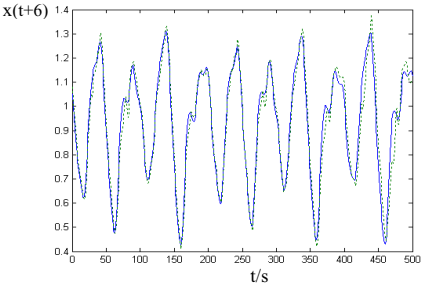


Fig. 4. The prediction output of the fuzzy model and the testing points

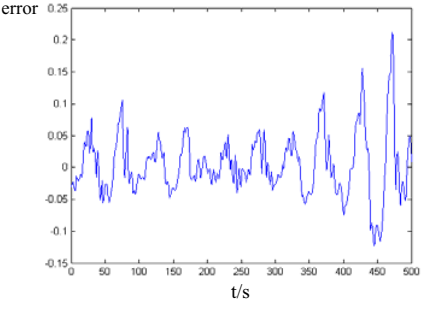


Fig. 5. The error diagram

In the 1000 simulation data points, the former 500 points are selected as the training data points to build the orthogonal neural network, and the rest 500 points as the testing data to test the validity of the orthogonal neural network. There are 12 neurons to describe the orthogonal neural network. After the orthogonal neural network model having been constructed on the basis of the training data, the following point will be forecasted by the orthogonal neural network, and the estimated value will be used to adjust the orthogonal neural network. The prediction model of Mackey-Glass time series is built via the above-mentioned process. The simulation results are given from Figure 2 to Figure 5.

Figure 2 is the comparison between the output of the orthogonal neural network and the training points. The real line represents the training points and the dot line represents the output of the orthogonal neural network in Figure 2. Figure 3 is the error diagram between the output of the orthogonal neural network and the training points. Figure 4 is the comparison between the prediction output of the orthogonal neural network and the testing points. The real line represents the testing points and the dot line represents the prediction output of the orthogonal neural network in Figure 4. Figure 5 is the error diagram between the output of the orthogonal neural network and the testing points.

In table 1, we compare our prediction method with other prediction methods from same data. It can be seen that the performance of our method is superior to that of other prediction methods, expect the proposed method in the paper [14]. Though our method cannot outperform ANFIS [14] in the prediction of Mackey-Glass time series, it still has three superiorities:1) The proposed method has stronger vitality than ANFIS when there are many uncertainties in the complex system; 2) The proposed method can obtain a simpler model structure (12 neurons) than that of ANFIS (16 rules), which will contribute to satisfying the high efficiency requirement for the on-line prediction; 3) The proposed method is more efficient than ANFIS. In the prediction of the time series, the modeling time of our method was only about 6 minutes on PC (Pentium), while the ANFIS simulation took about 1.1 hours on PC (Pentium).

Table 1. Generalization Result Comparisons

Methods	Training Cases	Mean Square Errors
ANFIS [13]	500	0.007
Cascade-Correlation [12]	500	0.06
Back-Prop NN [12]	500	0.02
6^{th} –Order Polynomial [12]	500	0.04
Our method	500	0.016

5 Conclusion

This paper presents an orthogonal function neural network to the prediction of complex systems. The recursive Kalman filtering algorithm based on SVD method is used to confirm the parameters of the orthogonal function neural network in order to avoid error delivery and error accumulation. The simulation results have shown that Mackey-Glass time series can be accurately predicted. In a word, the proposed method has shown the advantages of simplicity, flexibility and high accuracy.

References

1. Zeng, Z.C., Duan, Y.R., Shao, G.: Chaotic Time Series Analysis Based on Radial Basis Function Network. Chinese Journal Of Chongqing University 22, 113–120 (1999)
2. Luo, J.F., Zheng, J.L., Sun, S.Y.: Prediction Algorithm Based on the Combination of Chaos and Neural Network and its Application. Chinese Journal of Radio Science 14, 172–177 (1999)
3. Jiang, Y.D., Shen, J.T., Yang, B.R.: A Method of Chaotic Time Series Prediction Based on Wavelet Neural Network. Microcomputer Development 12, 37–39 (2001)
4. Zhang, Q.H., Benvensite, A.: Wavelet Network. IEEE Trans. on Neural Network. 37, 433–448 (2000)
5. Zhang, J., Gilbert, G.: Wavelet Neural Network for Function Learning. IEEE Trans. Signal Processing. 43, 1485–1497 (1995)
6. Zhang, W., Wu, Z.M., Yang, G.K.: Genetic Programming Modeling on Chaotic Time Series. Acta Electronica Sinica 33, 748–751 (2005)

7. Ye, M.Y., Wang, X.D.: Chaotic Time Series Using Least Squares Support Vector Machine. Chinese Physics 13, 454–458 (2004)
8. Li, H.C., Zhang, J.S., Xiao, X.C.: Neural Volterra Filter for Chaotic Series Prediction. Chinese Physics 14, 2181–2188 (2005)
9. Pisarenko, V.F., Sornette, D.: Statistical Methods of Parameter Estimation of Deterministically Chaotic Time Series. Physics Review E 69, 1–12 (2004)
10. Stefania, T., Massimiliano, G., Roberto, B.: Reconstruction of Chaotic Time Series by Neural Models: A Case Study. Neuralcomputing 55, 581–591 (2003)
11. Ding, T., Zhou, H.C.: Local Prediction for Chaotic Time Series and Its Application. The Journal of Dalian University of Technology 44, 445–448 (2004)
12. Crowder, R.S.: Prediction the Mackey-Glass Time Series with Cascade-Correlation Learning. In: Proc. Of the 1990 Connectionist Models Summer School, Carngie Mellon University, pp. 117–123 (1990)
13. Roger, J.: ANFIS: Adaptive-Network –Based Fuzzy Inference Systems. IEEE Transactions on Systems, Man, Cybernetics 23, 665–685 (1993)

Estimation for Speed and Leakage Power of Dual Threshold Domino OR Based on Wavelet Neural Networks

Jinhui Wang[1], Lei Zuo[1], Na Gong[2], Daming Gao[1], Shuqin Geng[1], Wang Zhang[1], Ligang Hou[1], Xiaohong Peng[1], and Wuchen Wu[1]

[1] VLSI and System Lab, Beijing University of Technology, Beijing 100124, China
wangjinhui888@yahoo.com.cn
[2] College of Electronic and Informational Engineering, Hebei University, Baoding 071002, China

Abstract. An approach for estimating the leakage power and the speed of the dual threshold domino OR gates based on Wavelet Neural Networks (WNN) in 45 nm technology is proposed. The estimating system has fast convergence and high precision. By studying the impact of the dual threshold voltage technique (DTV) on leakage reduction and delay increase, it successfully estimates the nonlinear changing of the leakage power and delay of the different inputs domino OR gates. At last, the reason for the estimating error and the trend of the estimating curve are explained, respectively.

Keywords: Wavelet Neural Networks, Dual threshold domino OR, Speed, Leakage power.

1 Introduction

Wide domino OR gates or like structures are commonly employed in register and cache array bit line design [1]. As technology scales down, the threshold voltage (Vt) and gate oxide thickness (tox) of the transistors must be reduced to accompany the supply voltage scaling to meet the performance requirements. However, the leakage power increase exponentially with the scaling of Vt and tox [2][3]. The 2005 International Technology Roadmap for Semiconductors (ITRS) [4] predicted that by the sub-65nm generation, leakage may constitute as much as 50 percent of the total power consumption.

Therefore, there exists the need to find a solution that can suppress leakage power. The dual threshold voltage technique (DTV) [5] is one of the most popular techniques to achieve this goal. But, because of the low speed characteristics of high threshold transistors, the dual threshold voltage technique increases the delay of the domino gates. Hence, before applying DTV, the leakage power reduction and delay increase should be forecasted and then traded off, which can help judge if the application of DTV in domino gates meets the design constrains

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 844–851, 2009.

of the power and the delay. Especially in EDA design flow, this estimation will save designers a huge amount of time and reduce iteration.

Leakage power and delay estimating for domino gates is challenging because of nonlinear effects. Neural networks have emerged to provide a very appealing approach to estimate nonlinear changing of leakage power and delay. An important advantage of neural networks based estimation is no explicit models are required [6]. Due to the quick convergence, effective classification and highly accuracy, the wavelet neural networks (WNN) has been widely known and successfully used in most control systems and information processing systems [7][8]. In this paper, a novel approach for estimating leakage power and delay of the dual threshold domino OR gate based on WNN in 45 nm technology is proposed. And its accuracy is validated with simulation test.

2 Estimating System Based on Wavelet Neural Networks

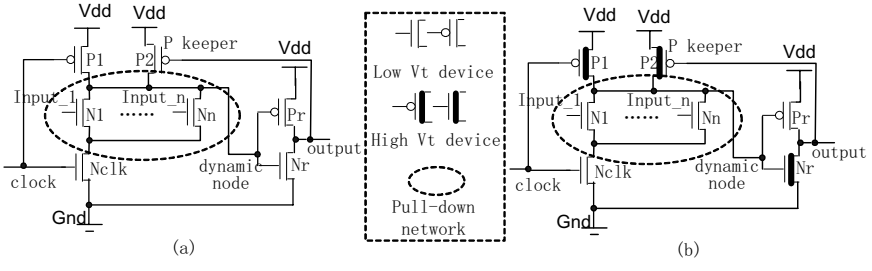


Fig. 1. Domino OR gates (a) Low Vt domino OR gate (b) High Vt domino OR gate

2.1 Dual Threshold Domino OR Gate

High leakage power consumption has become an important issue affecting domino OR gates performance in 45nm technology. The dual Vt technique is proposed to efficiently lower the leakage power [9][10]. As can be seen from Fig.1 (b), the critical signal transitions determining the domino circuit delay occur along the evaluation path. In a dual Vt domino circuit, therefore, all of the transistors, activated during the evaluation phase (Nclk, N1...Nn, Pr), have a low Vt. Alternatively, the precharge/predischarge phase transitions (P1, P2, Nr) are not critical for the performance of a domino circuit and they all have a high Vt [11]. The leakage current decreases with the increasing of Vt and it can be expressed as followed [12]

$$I_{sub} = \frac{W_{eff}}{L_{eff}} u \sqrt{\frac{q\varepsilon_{si} N_{ch}}{2\Phi_s}} V_T^2 exp(\frac{V_{gs} - V_t}{nV_T})(1 - exp(-\frac{V_{ds}}{V_T})), \quad (1)$$

where I_{sub}, L_{eff} and W_{eff} are the sub-threshold leakage current, the effective channel length and width respectively, and other parameters have their usual meanings. However, high Vt transistors in circuits will degrade the speed of the domino gates. It can be expressed as followed [12]

$$v \propto \frac{V_{dd}^{0.3}(1-\frac{V_t}{V_{dd}})^{1.3}}{t_{ox}^{0.5}}, \tag{2}$$

where v is the speed of the transistor, and other parameters have their usual meanings. Thus, in a dual threshold domino OR Gate, the number of the transistors in the pull-down, that is, the number of the inputs, has a crucial impact on its leakage power and speed characteristics. On the one hand, most of its leakage current is produced by the transistors in the pull-down network and the number of these transistors greatly influences the DTV effect on lowering leakage power. On the other hand, with the increasing of fan-in, the more transistors in the pull-down network, the more current path are constructed from dynamic node to ground. These increased current paths also decrease the delay of the evaluation stage and compensate the speed loss that DTV induces.

Therefore, constructing a precise estimating system that details the relation between the number of the inputs and the leakage reduction and delay increase can help determine whether to apply DTV in domino gates or not. In the next section, we present a model based on Wavelet Neural Networks.

2.2 Implement Wavelet Neural Networks

WNN is constructed based on wavelet analysis, which has similar structure of feed-forward neural networks. Three-layer WNN is embedded with wavelet functions as hidden layer neurons, which take wavelet space as feature space of pattern recognition. This is a multi-layer feedback architecture with wavelet, allowing the minimum time to converge to its global maximum. The WNN employs a wavelet base rather than a sigmoid function, which discriminates it from general back propagation neural networks.

The function of mapping can be expressed as:

$$f(x) = \sum_{o=1}^{h}\sum_{j=1}^{m}\omega_o \frac{1}{\sqrt{|a_o|}}\psi_{jo}\left(\frac{\sum_{i=1}^{n} x_{ij} - b_o}{a_o}\right), \tag{3}$$

where 0 (0=1, 2..., h) and j_0 are output of hidden layer neurons and the wavelet bases, respectively. Networks have three parameters to be trained: output weight translation factors a and dilation factors b.

In this paper, a set of training samples with labels D= {(yi, xi), i=1,2...,N}. And morlet wavelet is used as stimulation function of hidden layer

$$\psi(x) = \cos(1.75x)e^{\frac{-x^2}{2}}. \tag{4}$$

The error performance function is given by:

$$J = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{m}(y_{j,i}^{d} - y_{j,i})^2, \tag{5}$$

where N is the total number of training patterns, $y_{j,i}^{d}$ and $y_{j,i}$ are the desired and real outputs, respectively.

The training process of WNN is performed as following:

1. Create the initial population of individuals according to the initiation strategy–output weight, translation factors a and dilation factors b are in (0,1).

2. Calculate the fitness function by (5).

3. To minimize the fitness function in (5), the weights and coefficients a and b can be updated using the following formulas:

$$\omega_j(k) = \omega_j(k-1) + \eta(y^d(k) - y(k))h_j + \alpha(\omega_j(k-1) - \omega_j(k-2)), \quad (6)$$

$$\Delta a_j = a_j(y^d(k) - y(k))\omega_j h_j \frac{(\sum_{i=1}^{n} x_{ij} - b_j)^2}{a_j^3}, \quad (7)$$

$$a_j(k) = a_j(k-1) + \eta\Delta a_j + \alpha(a_j(k-1) - a_j(k-2)), \quad (8)$$

$$\Delta b_j(k) = (y^d(k) - y(k))\omega_j \frac{\sum_{i=1}^{n} x_{ij} - b_j}{a_j^2}, \quad (9)$$

$$b_j(k) = b_j(k-1) + \eta\Delta b_j + \alpha(b_j(k-1) - b_j(k-2)), \quad (10)$$

$$h_j = \frac{1}{\sqrt{|a_j|}}\psi_j(\frac{\sum_{i=1}^{n} x_{ij} - b_j}{a_j})(j = 1, 2..., m), \quad (11)$$

where η=0.01, α=0.05. j is the number of hidden layer neurons.

4. Repeat step 2) to step 3) until some constraint condition is satisfied, then stop and the desired individuals are obtained.

3 Simulation Result and Analysis

As described in section 2.1, the input of the WNN is the number of inputs of dual threshold domino OR gates and the outputs are leakage reduction percentage and delay increase percentage as compared to the low threshold domino OR gates. The training data and testing data are collected from the simulation results with HSPICE tools based on 45nm BSIM4 model, as can be seen in Table1, 2, 3. When simulating the leakage power, all of the domino OR gates are set in CHIH (clock=1, IN_1=IN_2=...INn=1) state, which can ensure every gates in the lowest

Table 1. Parameter of devices

Technology Node	45nm	
	Low V_t	High V_t
NMOS Threshold Voltage	0.22V	0.35V
PMOS Threshold Voltage	-0.22V	-0.35V
Supply Voltage	0.8V	0.8V

Table 2. Training Data

In	1	3	5	10	15	20	25	30	35	40	45	50	55	60	65	70
LR/%	82.9	75.6	69.6	56.2	49.0	43.3	41.8	38.1	35.0	34.9	32.6	30.6	28.9	27.3	25.9	24.7
DI/%	21.8	27.1	27.5	32.4	33.2	34.3	23.9	22.6	24.6	18.4	20.1	20.2	21.5	22.2	20.6	21.8
In	75	80	85	90	95	100	105	110	115	120	125	130	135	140	145	150
LR/%	23.6	22.5	21.6	20.8	20.0	19.2	18.6	18.0	17.4	16.9	16.4	16.0	15.5	15.1	14.7	14.3
DI/%	24.2	23.9	24.5	25.6	21.3	21.5	24.3	24.6	21.9	20.4	20.1	23.0	22.9	22.8	22.6	19.2

In: the number of the transistors in the pull-down network. LR: the percentage of the leakage reduction. DI: the percentage of the delay increase.

Table 3. Testing Data

In	2	4	8	16	32	48	64	96	128
LR/%	79.0	72.5	62.2	47.7	36.8	31.4	26.2	19.9	16.1
DI/%	21.9	28.3	28.4	33.8	23.6	20.2	20.7	21.3	23.0

In: the number of the transistors in the pull-down network. LR: the percentage of the leakage reduction. DI: the percentage of the delay increase.

Table 4. Testing Error

In	2	4	8	16	32	48	64	96	128
LR error/%	+3.80	+1.99	-1.26	-2.11	-1.77	-2.81	-2.96	-1.87	-1.81
DI error/%	+0.01	-1.51	+2.30	+0.28	-3.92	+1.39	+2.96	+2.46	-3.77

In: the number of the transistors in the pull-down network. LR: the percentage of the leakage reduction. DI: the percentage of the delay increase.

leakage state [13]. In order to test the availability of the estimating model, the testing data are selected from 2, 4, 8, 16, 32, 48, 64, 96, 128 inputs domino OR gates, because these typical gates are usually utilized in practice. Based on the previous discussion, the architecture was used for the wavelet network is 1-1-1 (one input layer, one hidden units and one output unit). The network was trained with 2500 learning iterations. The largest error E or given precision is 0.00001. The Matlab languages make up the software programs including the leakage power and delay sampling procedure, leakage power and delay analysis, WNN training and so on. The WNN training is ended when the value of objective function is less or equal to 0.00001 and it converge very fast, as can be seen from fig.2.

The testing errors are shown in table 4. And all of the errors are less 5% including leakage power reduction errors and delay increase errors. Obviously, the estimating system possesses the high estimation accuracy. Therefore this estimating system based on wavelet neural network has strong stability, and it can be embedded in EDA tools as power and delay estimation programming.

Fig.3 is the estimating curve based on the wavelet neural networks and testing curve. As can be seen from it, when the number of the inputs is less than eight, the testing leakage curve is over the estimating curve. Alternatively, the testing curve from the domino OR gates with eight or more inputs positions under the estimating curve.

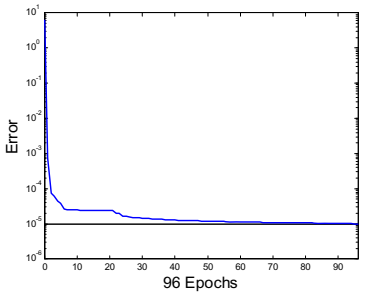


Fig. 2. The training error curve of the wavelet neural networks

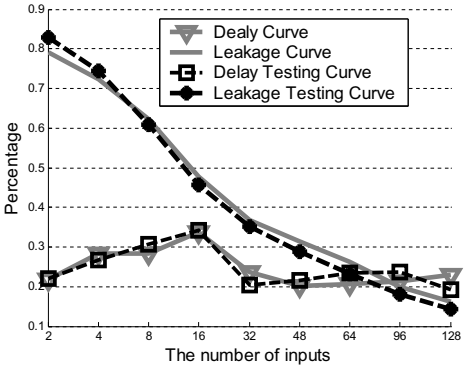


Fig. 3. The estimating curve and testing curve based on the wavelet neural networks

This can be explained as follows. In the nanometer CMOS technologies, the leakage power consists of the sub-threshold leakage power and the gate leakage power (produced by the gate leakage current: Igate) [14]. With the scaling of the oxide thickness (tox), Igate increases exponentially and is becoming a significant contributor to the total leakage current as CMOS process advances to sub-65nm regime, according to ITRS2005 [4][15]. What's more, in a dual threshold domino OR Gate, Igate increases greatly with the increasing of the fan-in because the transistors in the pull-down leads to the greatest forward Igate [16]. However, the DTV aims at suppressing the sub-threshold leakage power, and has little impact on the gate leakage power. Therefore, when the number of the inputs is less than eight, the gate leakage power is much smaller than the sub-threshold leakage power and then it can be ignored and the effect of DTV achieves maximum. However, with the increasing of the number of the inputs, the gate leakage power rises and comes up with the sub-threshold leakage power and thus the effect of DVT decreases. So as compared to the testing data, the training data values in the training curve are bigger and, therefore, the testing curve is under the estimating curve, as can be seen in fig.3.

As described in section 2.1, the number of inputs has significant influence on the delay of the domino OR gates. The delay is determined by the relative contribution of the capacitance of P1+P2 (in fig.1) and the capacitance in

pull-down network. When the number of the inputs is approximate sixteen, these two capacitances match well with each other, and then the gate delay is smallest. DVT application makes these two capacitances mismatch, which induces the most amount of delay increase as compared to other inputs domino OR gates, as can be seen in fig.3.

4 Conclusion

Based on WNN, a system for estimating the leakage power and the delay of the dual threshold domino OR gates in 45 nm technology is proposed. The estimating system has fast convergence and high precision. By studying the impact of the dual threshold voltage technique (DTV) on leakage reduction and delay increase, it successfully forecasts the nonlinear changing of the leakage power and delay of the different inputs domino OR gates. The simulation results for verification shows that the accuracy ratio of estimating system is more than 95%. Therefore, it can be well applied in VLSI design flow. At last, the reason for the estimating error and trend of the estimating curve are explained, respectively. The leakage power estimating error is produced because of ignoring gate leakage power which is becoming an important contributor with the increasing number of the transistors in pull-down network. And the delay penalty of sixteen inputs domino OR gate achieves maximum. In addition, through the extension of this estimating system, we can construct other estimating system to estimate the impact of different optimization techniques on different logical gates, which can significantly reduce the design period.

References

1. Chatterjee, B., Sachdev, M., Krishnnamurthy, R.: Designing Leakage Tolerant, Low Power Wide-OR Dominos for Sub-130 nm CMOS Technologies. Microelectronics Journal 36, 801–809 (2005)
2. Liu, Z., Kursun, V.: Leakage Power Characteristics of Dynamic Crcuits in Nanometer CMOS Technologies. IEEE Transactions on Circuits and Systems II 53, 692–696 (2006)
3. Guo, B.Z., Gong, N., Wang, J.H.: Designing Leakage-Tolerant and Noise-Immune Enhanced Low Power Wide OR Dominos in Sub-70nm CMOS Technologies. Chinese Journal of semiconductors 27, 804–811 (2006)
4. International Technology Roadmap for Semiconductors, http://public.itrs.net/
5. Kuroda, T., Fujita, T., Mita, S., et al.: A 0.9 V 150 MHz 10 mW 4 mm2 2-D Discrete Cosine Transform Core Processor With Variable-Threshold-Voltage Scheme. In: 43rd IEEE International Solids-State Circuits Conference, pp. 1770–1779. IEEE Press, New York (1996)
6. Luo, Z.Y., Shi, Z.K.: Wavelet Neural Network Method for Fault Diagnosis of Push-Pull Circuits. In: 2005 International Conference on Machine Learning and Cybernetics, pp. 3327–3332. IEEE Press, New York (2005)

7. Aminian, F., Aminian, M., Collins Jr., H.W.: Analog Fault Diagnosis of Actual Circuits Using Neural Networks. IEEE Transaction on Instrumentation and Measurement 51, 544–550 (2002)
8. Oonsivilai, A., El-Hawary, M.E.: Wavelet Neural Network Based Short Term Load Forecasting of Electric Power System Commercial Load. In: 1999 IEEE Canadian Conference on Electrical and Computer Engineering, pp. 122–128. IEEE Press, New York (1999)
9. Phillip, C., Charles, A.Z., George, G., Stephen, V.K.: Characterization of Logic Circuit Techniques and Optimization for High-Leakage CMOS Technologies. The VLSI journal 38, 491–504 (2005)
10. Wang, J.H., Gong, N., Feng, S.B., et al.: A Novel P-type Domino AND Gate Design in Sub-65nm CMOS Technologies. Chinese Journal of Semiconductors 28, 1818–1823 (2007)
11. Kursun, V., Friedman, E.G.: Sleep Switch Dual Threshold Voltage Domino Logic with Reduced Standby Leakage Current. IEEE Transaction on VLSI Systems 12, 485–496 (2004)
12. Taur, Y., Ning, T.H.: Fundamentals of Modern VLSI Devices. Cambridge University, Cambridge (1998)
13. Kao, J.T., Chandrakasan, A.P.: Dual-Threshold Voltage Techniques for Low Power Digital Circuits. IEEE Journal of Solid-State Circuits 35, 1009–1018 (2000)
14. Wang, J.H., Gong, N., Hou, L.G., et al.: Charge Self-compensation Technology Research for Low Power and High Performance Domino circuits. Chinese Journal of Semiconductors 29, 1412–1416 (2008)
15. Wang, J.H., Gong, N., Hou, L.G., et al.: Low Power Wide Dominos Design in sub-65nm CMOS Technologies. In: 8th International Conference on Solid-State and Integrated Circuit Technology, pp. 1864–1866. IEEE Press, New York (2006)
16. Gong, N., Guo, B.Z., Lou, J.Z., Wang, J.H.: Analysis and Optimization of Leakage Current Characteristics in Sub-65nm Dual Vt Footed Domino Circuits. Microelectronics Journal 39, 1149–1155 (2008)

Using Wavelet Based Neural Networks for Feedback Signals Estimation of a Vector Controlled Induction Motor Drive

Hassan Moghbelli[1], Akbar Rahideh[2], and Ali A Safavi[3]

[1] Science and Mathematics, Texas A&M University at Qatar, Doha, Qatar
hassan.moghbelli@qatar.tamu.edu
[2] School of Electrical and Electronic Engineering,
Shiraz University of Technology, Shiraz, Iran
[3] School of Engineering, Shiraz University, Shiraz, Iran

Abstract. In this investigation, a vector controlled induction motor drive is simulated and the feedback signals of this vector controlled drive are estimated using neural networks. The neural networks receive the machine terminal signals as inputs and estimate the rotor flux and unit vectors $\cos\theta_e$ and $\sin\theta_e$ as outputs. These outputs are used in the vector controlled drive system. The calculated feedback signals by the neural networks are not sensitive to the motor parameter variations. In this paper, three types of neural networks (i.e. multilayer perceptron (MLP), radial basis function (RBF) and wavenet) are used and the obtained results are compared. Finally, on the basis of the advantages of wavenets, the results prove the accuracy and effectiveness of the wavenet based estimator.

Keywords: Neural Networks, wavenet, vector control of induction motors, feedback signals.

1 Introduction

Among various types of electric motors, induction motors and in particular squirrel cage rotor type motors have a very wide spread applications due to their several advantages such as: low maintenance, high reliability, low weight and inertia, low cost and high efficiency. Despite these advantages, until a few years ago induction motors were only used in fixed speed applications. But with the development in power electronics and using modern control techniques like vector control and also using the methods based on artificial intelligence, suitable control for induction motors have been suggested and allowed the use of these motors for variable speed applications.

The control methods for induction motors are divided in two distinct categories: scalar control and vector control. In the scalar control method, voltage and frequency are the main control variables for induction motors. On the other hand, torque and flux are the functions of both voltage and frequency. Such a coupling causes a low response speed in induction motors. This limitation can be removed by employing a vector control approach. In vector control techniques, an ac motor can be controlled like a separated excited dc motor [1].

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 852–862, 2009.

In the last two decades, neural networks have been used in many applications. For instance, a stator-flux-oriented vector-controlled induction motor drive has been described where the space-vector pulsewidth modulation (SVM) and stator-flux-vector estimation are implemented by artificial neural networks [2]. A neural-network-based implementation of space-vector modulation (SVM) of a three-level voltage-fed inverter has been proposed in [3]. Karanayil *et al.* [4] have presented a new observer for the rotor resistance of an indirect vector controlled induction motor drive using artificial neural networks supplemented by a fuzzy logic based stator resistance observer. The estimation of the feedback signals of an induction motor using an MLP neural network can be found in [5]. Identification and adaptive control of induction motors have been suggested in [6]. Robustifying a vector controlled induction motor using feedback and feedforward control simultaneously can be found in [7].

Despite the potential of neural networks, several problems remain to be solved before their more wide spread applications become possible. For instance, most often no physical interpretation can easily be attributed to a model generated by a trained network. Some of the activation functions being used are *global* functions which do not allow local training. The set of processing functions, which also determines the structure of the network, may need to be determined empirically by trial and error. The convergence of training algorithms is not normally guaranteed. Attempts have been made to solve these and other problems by considering rigorous mathematical frameworks for neural networks. The recently developed and fast growing theory of wavelets and multiresolution analysis (MRA) may provide a powerful mathematical framework for wavelet networks. Wavelet based neural networks with a multiresolution learning, called wavenets, have overcome some of the weaknesses of the conventional neural networks. Wavenets have been used for modeling and optimization of distillation column [8]. More detailed descriptions about the so-called multiresolution analysis can be found in [9].

In this paper, three types of neural networks, i.e. multilayer perceptron (MLP) networks, radial basis function (RBF) networks and wavenet, have been used. The advantages of the wavenet are presented via a comparative study.

2 Feedback Signals

In the vector control method, three phase stator currents are converted to the direct and quadrature currents (i_{ds}, i_{qs}) where the flux and torque can be independently controlled using these two currents respectively. In the vector controlled induction motors, the feedback signals (i.e. rotor flux and unit vectors) are very sensitive to the motor parameter variations that is clear from the following equations. These feedback signals can be calculated from the d-axis stator current (i^s_{ds}), the q-axis stator current (i^s_{ds}), the q-axis stator flux (ψ^s_{qs}), the d-axis stator flux (ψ^s_{qs}) and motor parameters.

In a vector control drive system, the feedback signals are calculated from the current and voltage terminal of the machine using the following equations:

$$\psi_{qs}^{s} = \int (v_{qs}^{s} - R_s i_{qs}^{s})dt \tag{1}$$

$$\psi_{ds}^{s} = \int (v_{ds}^{s} - R_s i_{ds}^{s})dt \tag{2}$$

$$\psi_{qm}^{s} = \psi_{qs}^{s} - L_{ls} i_{qs}^{s} \tag{3}$$

$$\psi_{dm}^{s} = \psi_{ds}^{s} - L_{ls} i_{ds}^{s} \tag{4}$$

$$\psi_{qr}^{s} = \frac{L_r}{L_m}\psi_{qm}^{s} - L_{lr} i_{qs}^{s} \tag{5}$$

$$\psi_{dr}^{s} = \frac{L_r}{L_m}\psi_{dm}^{s} - L_{lr} i_{ds}^{s} \tag{6}$$

$$\psi_r = \sqrt{(\psi_{qr}^{s})^2 + (\psi_{dr}^{s})^2} \tag{7}$$

$$\cos\theta_e = \frac{\psi_{dr}^{s}}{\psi_r} \tag{8}$$

$$\sin\theta_e = \frac{\psi_{qr}^{s}}{\psi_r} \tag{9}$$

where,

R_s	Stator resistance
L_{ls}	Stator leakage inductance
L_{lr}	Rotor leakage inductance
L_m	Magnetizing inductance
L_r	Rotor inductance
P	Number of poles
$v^s_{ds}\left(v^s_{qs}\right)$	d-axis (q-axis) stator voltage
$i^s_{ds}\left(i^s_{qs}\right)$	d-axis (q-axis) stator current
$\psi^s_{ds}(\psi^s_{qs})$	d-axis (q-axis) stator flux linkage
$\psi^s_{dm}(\psi^s_{qm})$	d-axis (q-axis) air-gap flux linkage
$\psi^s_{dr}(\psi^s_{qr})$	d-axis (q-axis) rotor flux linkage.

3 Neural Networks and Estimation

In this study, the d-axis stator current (i^s_{ds}), the q-axis stator current (i^s_{ds}), the q-axis stator flux (ψ^s_{qs}) and the d-axis stator flux (ψ^s_{qs}), that all are in the stationary reference frame, are the inputs of the neural network and the rotor flux (ψ_r), and the unit vectors ($\cos\theta_e$ and $\sin\theta_e$) are considered to be the outputs of the neural network, where θ_e is the rotor flux angle. The block diagram of a vector controlled induction motor drive with a neural network based feedback signals estimator is illustrated in Fig. 1.

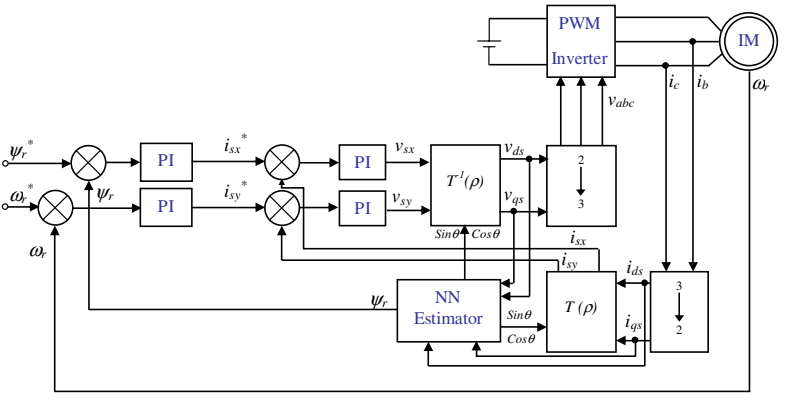


Fig. 1. Block diagram of a vector controlled induction motor drive with a neural network based feedback signals estimator

The estimation of the rotor flux and unit vectors using three classes of neural networks have been performed, and then applied in a vector controlled induction motor drive in a simulation environment. A comparative study of the results is also presented.

3.1 Multilayer Perceptron (MLP) Network

This network consists of three layers: input, hidden and output layers. The activation functions that can be used for this network are generally global (for instance *tanh*). The input signals are normalized using a set of normalizing factors and after computation, the output signals are converted to the actual value using their corresponding denormalizing factors. The most suitable learning algorithm that used for this network is found to be the Levenberg Marquardt due to the fact that the number of parameters (weights and biases) is not too high. Initial weights and biases are usually chosen randomly. The learning of an MLP network is very slow and time consuming and also the use of global functions is one of the other disadvantages of these networks. For more information about MLP networks see [10].

3.2 Radial Basis Function (RBF) Network

Fig. 2 illustrates a radial basis function network. In this network, the input signals directly enter the hidden layer and the basis functions are normally local. In order to train this network, in addition to weights adjustment, finding the centers of the basis

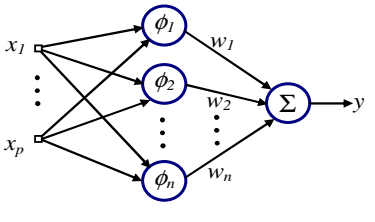


Fig. 2. A radial basis function network construction

functions is also necessary. The least mean squared error is selected for the training purpose. The number of neurons of the hidden layer is chosen by a trial and error process in the case of both RBF and MLP networks.

3.3 Wavenet

Fig. 3 illustrates a wavenet construction with p inputs and a single output. This construction is very similar to the radial basis function network. The wavenet construction includes two functions: scaling functions and wavelets. Wavelets are a new family of localized basis functions and they are functions with a combination of powerful features, such as orthonormality, locality in the time and frequency domains, different degrees of smoothness, fast implementations, and in some cases compact support. Some typical scaling functions and their wavelets are shown in Fig. 4.

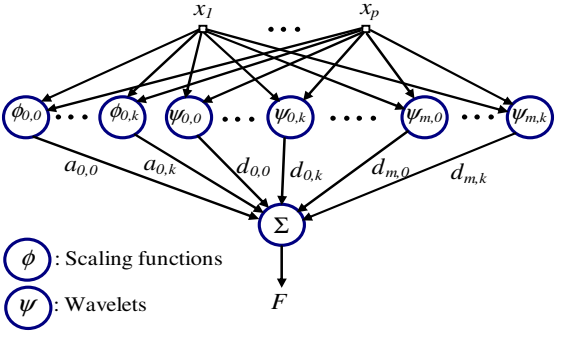


Fig. 3. A wavenet construction with p inputs and a single output

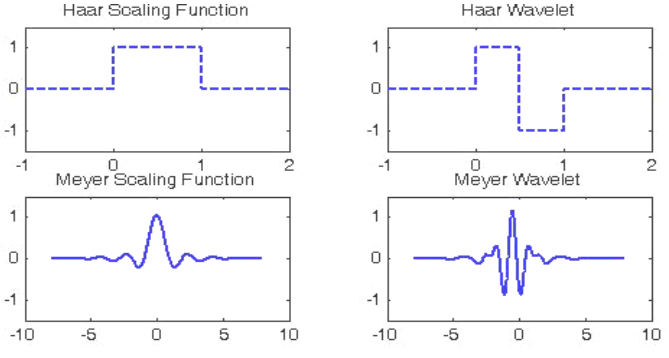


Fig. 4. Some typical scaling functions and their wavelets

Wavelets are usually introduced in a multiresolution framework developed by Mallat [9]. The function ϕ is called a scaling function of the multiresolution analysis, and a family of scaling functions of the MRA can be expressed as,

$$\varphi_{m,k}(x) = 2^{-m/2}\varphi(2^{-m}x - k) \tag{10}$$

where 2^{-m} and k correspond, respectively, to the dilation and translation factors of the scaling functions, while $2^{-m/2}$ is an energy normalization factor. The orthonormal basis functions corresponding to detailed spaces of multiresolution analysis are named wavelets and denoted by $y_{m,k}$'s. These can easily be obtained from $\phi_{m,k}$'s. A family of wavelets may be represented as,

$$\psi_{m,k}(x) = 2^{-m/2}\psi(2^{-m}x - k) \qquad m,k \in z\,. \tag{11}$$

In general any physically measureable function can be expressed as

$$f(x) = f_0(x) + \sum_{m=-\infty}^{0}\sum_{k=-\infty}^{\infty} d_{m,k}\psi_{m,k}(x) \tag{12}$$

where

$$f_0(x) = \sum_k a_{0,k}\varphi_{0,k}(x)\,. \tag{13}$$

To construct a feedforward neural network, one needs to describe the location of the nodes, the type of basis functions employed in the nodes and the learning algorithm. The basis functions in a wavenet are wavelets (and scaling functions). Therefore, it only remains to define the locations of these basis functions and a learning algorithm for the network. The locations of the basis functions for different resolutions are shown in Fig. 5.

Equation (12) describes the basic framework of a wavenet, in that it explains how each wavelet cooperates in the whole approximation scheme. It also shows that the scaling functions are only used at the earliest stage of the approximation to produce f_0, after which the approximation scheme uses only wavelets. Once the first approximation to a function f is obtained, that is f_0, one can get a better approximation, namely f_{-1}, by including wavelets of the same dilation factor as the scaling function, here $m = 0$. Adding wavelets of the next highest resolution, here $m = -1$, leads to a finer approximation f_{-2}, than the previous one f_{-1}. This process is continued until the original function is reconstructed, or an arbitrary degree of accuracy for the approximation is obtained. Three learning algorithms for wavenets (i.e. finding the network coefficients $a_{m,k}$ and $d_{m,k}$) have been presented in [8]. Since some of these learning methods are not iterative, they are very fast methods for learning.

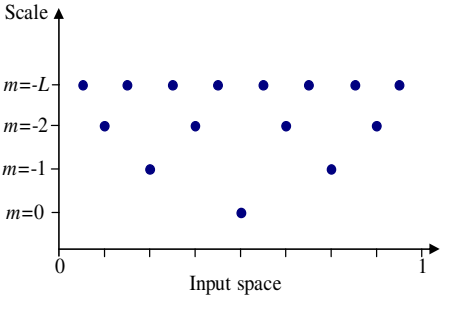


Fig. 5. The locations of basis functions for different resolutions

4 Results

As mentioned before, in this paper three types of neural network-based estimators have been developed and the results have been presented. The estimation problem can be expressed as below to find *g:*

$$[\cos\theta_e \quad \sin\theta_e \quad \psi_r] = g(i_{ds}^s, i_{qs}^s, \psi_{ds}^s, \psi_{qs}^s) \,. \tag{14}$$

In the case of the MLP network, the input/output patterns for training include a set of 2000 data points and the network has 4 inputs, 4 neurons at the first hidden layer, 10 neurons at the second hidden layer and 3 neurons at the output layer that have been chosen via trial and error. The activation functions for all three layers are *tanh.* The learning algorithm used for this network is the Levenberg Marquardt.

After the MLP based estimator, the RBF based estimator has been constructed. The number of neurons at the hidden layer has been selected by trial and error and is set to be 40 neurons and the input/output data for training is the same as MLP. The basis functions used are Gaussian functions.

Finally, the wavenet based estimator has been developed with 15 wavelets and 1 scaling function in the hidden layer for each of the outputs and the basis functions are selected to be the Meyer type. At the first level of the resolution in the training stage, the coefficient of the scaling function, $a_{0,0}$, is found and at the next resolution, the coefficients of the wavelets, $d_{0,0}$ to $d_{0,14}$,are obtained.

The main advantages of the wavenet are the multiresolution construction and also guaranteed convergence. The number of nodes in the hidden layer can be obtained using $(2^n)^{|m|}$ where n and m are the number of inputs of the network and the resolution respectively. On the other hand, the number of cells in each node can be calculated using 2^n-1. For instance, for a four-input network at the first resolution, m=0, there are only one scaling function and 15 wavelets. At the next resolution, m=-1, there are 240 wavelets.

Table 1 shows a comparison of the results of these three neural network-based estimators. The neural networks have the same number of inputs (four inputs). The three types of neural networks for signal estimation have shown reasonable performance; however, the wavenet based estimator outperforms its rivals in terms of sum squared error. Furthermore, the training time for this network is better. Therefore because of these advantages and also for the sake of brevity, only the results related to the vector controlled drive with the wavenet based estimator have been presented. However, the training and test results of all these three neural networks have been presented.

All of the neural networks have been trained with the same set of data, called training data, and then tested with another set of data, called test data, that has not been used in the training stage. Figs. 6(a) to 6(c) are the rotor flux and unit vectors ($\cos\theta_e$ and $\sin\theta_e$) estimated by the MLP network respectively. Figs. 7(a) to 7(c) show the same outputs related to the RBF based estimator. The outputs of the wavenet based estimator are shown in Figs. 8(a) to 8(c). The left hand sides of all these figures are related to the training stage and the right hand sides represent the test results.

Table 1. A comparative study of the results of the three classes of neural network based estimators

	MLP	RBF	Wavenet
Sum squared error	8.71	2.83	0.92
Network construction	4×4×10×3	4×40×3	4×48×3
Training time	9t	1.8t	t

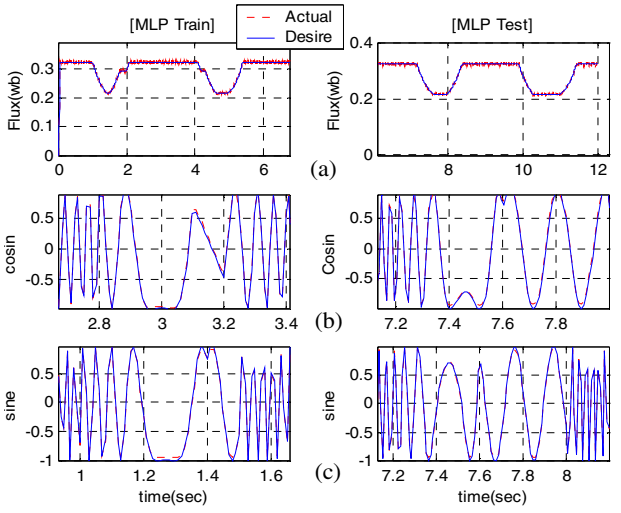


Fig. 6. The outputs of the MLP based estimator

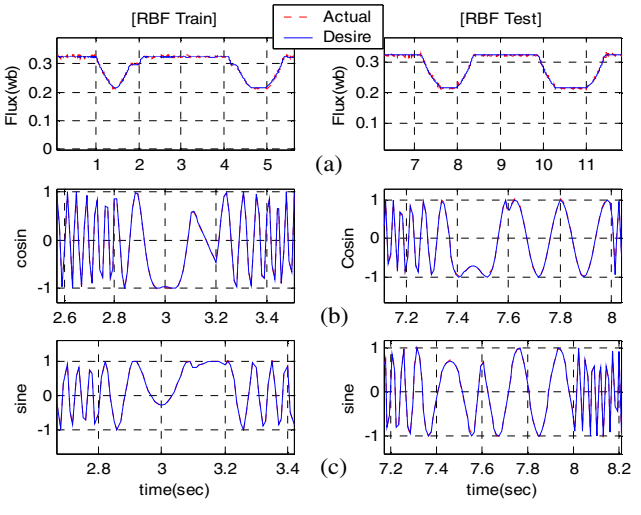


Fig. 7. The outputs of the RBF based estimator

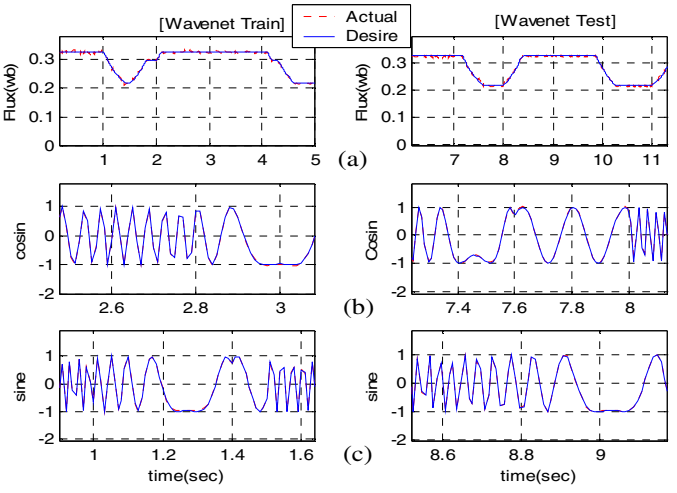


Fig. 8. The outputs of the wavenet based estimator

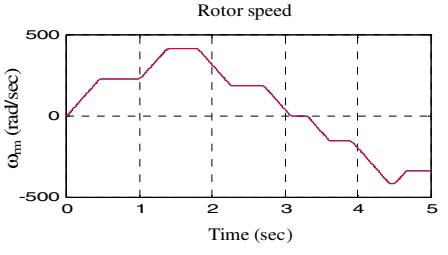


Fig. 9. The reference and actual rotor speeds

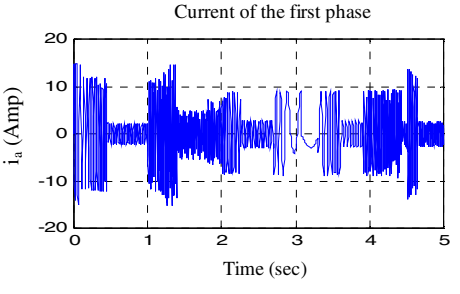


Fig. 10. The first phase of the stator current

Finally, these three neural network models have been used as the feedback signal estimators. Fig. 9 shows that the actual rotor speed closely follows the reference trajectory. Figs. 10 to 12 show the first phase of the stator current, d-axis and q-axis stator currents respectively. All of these figures are related to the vector controlled induction motor with the wavenet based estimator.

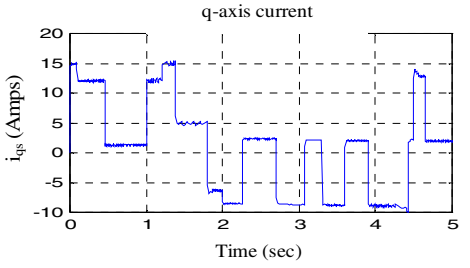


Fig. 11. The d-axis stator current

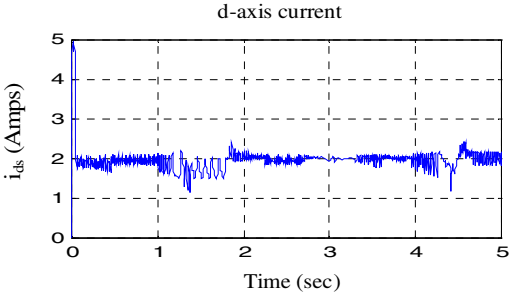


Fig. 12. The q-axis stator current

5 Conclusion

In this paper, the estimation of feedback signals for a vector controlled induction motor drive using three types of neural networks (i.e. MLP, RBF and wavenet) has been investigated and a comparative study has been presented. The neural networks receive the machine terminal signals as inputs and estimate the rotor flux and unit vectors $\cos\theta_e$ and $\sin\theta_e$ as outputs. These outputs have been used in the vector controlled drive system. The calculated feedback signals by the proposed neural networks are not sensitive to the motor parameter variations. Based on the simulation results, the wavenet based estimator is found to be the most suitable estimator due to its simple structure, less error, fast training, guaranteed convergence and the multiresolution learning. It can be seen from the simulation results that the actual rotor speed exactly tracks the reference speed and the wavenet based estimator estimates the feedback signals accurately.

References

1. Wade, S., Dunnigan, M.W.: Modeling and Simulation of Induction Machine Vector Control with Rotor Resistance Identification. IEEE Transaction on Power Electronics 12, 495–506 (1997)

2. Pinto, J.O.P., Bose, B.K., da Silva, L.E.B.: A Stator-Flux-Oriented Vector-Controlled Induction Motor Drive With Space-Vector PWM and Flux Vector Synthesis by Neural Networks. IEEE Transaction on Industry Application 37, 1308–1318 (2001)
3. Mondal, S.K., Pinto, J.O.P., Bose, B.K.: A Neural-Network-Based Space Space-Vector PWM Controller for a Three-Level Voltage-Fed Inverter Induction Motor Drive. IEEE Transaction on Industry Application 38, 660–669 (2002)
4. Karanayil, B., Rahman, M.F., Grantham, C.: Stator and Rotor Resistance Observers for Induction Motor Drive Using Fuzzy Logic and Artificial Neural Networks. IEEE Transaction on Energy Conversion 20, 771–780 (2005)
5. Simoes, M.G., Bose, B.K.: Neural Network Based Estimation of Feedback Signals for a Vector Controlled Induction Motor Drive. IEEE Transaction of Industry Application 31, 620–629 (1995)
6. Wishart, M.T., Harley, R.G.: Identification and Control of Induction Machines Using Artificial Neural Networks. IEEE Transaction on Industry Application 31, 612–619 (1995)
7. Tadakuma, S., Tanaka, S., Naitoh, H., Shimane, K.: Improvement of Robustness of Vector-controlled Induction Motor Using Feed Forward and Feedback Control. IEEE Transaction on Power Electronics 12, 221–227 (1997)
8. Safavi, A.A., Romagnoli, J.A.: Application of Wavelet-based Neural Network to the Modeling and Optimization of an Experimental Distillation. Engineering Application of Artificial Intelligence 10, 301–313 (1997)
9. Mallat, S.G.: A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 11, 674–693 (1989)
10. Haykin, S.: Neural Networks. Macmillan College Publishing Company (1999)

Study on Method of Identifying Dissolved Gases in Transformer Oil Based on Improved Artificial Neural Network Algorithm

Xingang Chen[1], Weigen Chen[2], Yi Yang[1], and Liangling Gu[1]

[1] Chongqing Institute of Technology, Chongqing 400050, China
[2] Chongqing University, Chongqing 400044, China

Abstract. Dissolved Gas-in-oil Analysis (DGA) plays an important role in fault diagnosis of power transformers. BP (Back Propagation) algorithm is used to diagnosis for dissolves gases in the oil of transformer in this paper. But typical BP algorithm has some defects, such as converging slowly, searching space possessing local minima and oscillation. The algorithm using additional momentum method and L-M (Lerenberg-Marquardt) to train BP Neural Network has been proved to have good performance in avoiding the local trap and converging slowly. So this paper adopts BP artificial neural network with algorithm of additional momentum method and L-M in diagnosis of dissolves gases in the oil. A mass of gases samples are analyzed in the algorithm and the results are compared with the swatches forecasted. The comparison result indicates that the improved algorithm has better classify capability for single-gases swatch as well as high diagnosis precision.

Keywords: Gas sensor array, Cross sensitivity, Artificial neural network, Gases identification.

1 Introductions

The large-scale power transformer is an important device in the power system, whether it works in a fit state concerns the safety and stabilization of the power system. Therefore, it is very valuable to examine transformer's early latent fault. There is a new settle approach had been offered to diagnose early fault when artificial neural network appears. [1-4] Construct smart sensor array system according components and contents differ of dissolves gases in the oil when transformer happen fault, neural network are trained with enough known samples through BP network's improved algorithm, and mapping relation from sensor array signal to gas concentration is saved on the linking weight of network. It can obtain satisfying ratio of diagnoses and can effectively conquer disadvantage of BP artificial neural network when use this technique to diagnosed dissolves gases in the oil.

2 Principle of Smart Sensor Array System

When semiconductor gas sensor (mostly the metal oxide) touches the surface of the gases, its physical characters changes, for example, the conductance .As a result, the

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 863–869, 2009.

sensor can be used to monitor gases. It's rapid and many different gases can be monitored, but has mistake selectivity for gas. The non-single selectively is decided by its sensitive mechanism. Although its selectivity can be improved by some methods such as adding an amount of precious metal Pt or Pd. it still has certain sensitivity to other gases. This is gas sensors "cross sensitivity" [5].

Identifies and tests gases by the array which is made of different selective gas sensor had been put forward by American Zaromb and Stetter. The theory of this method constitutes an array with different selective gas sensor and then building up the corresponding connections between the gas and the responses of the array. When the responses are gained, the gases can be identified and tested by inverse solution [6].

The principle of gas's identification and measurement by gas sensor array is shown in figure 1. Gas sensor array is used to detect mixed gas and form a high dimensional response model corresponding to measured media. Suppose that there are n components in the mixed gases and their concentration are c_1 , c_2,…, c_n etc. , gas sensor array's dimension is m, and then its response model is shown as follows:

$$\begin{cases} s_1 = g_1(c_1, c_2, \cdots, c_m) \\ s_2 = g_2(c_1, c_2, \cdots, c_m) \\ \quad\vdots \\ s_n = g_n(c_1, c_2, \cdots, c_m) \end{cases} \tag{1}$$

Formula 1 can be simplified as $S = G(c)$. In the simplified formula, input vector is $C = (c_1 , c_2 , \ldots , c_n)$, output vector is $S = [s_1 , s_2 , \ldots , s_m]$, where s_i is the ith sensor's output.

In addition, mapping relationship $G(\cdot) = [g_1(\cdot) , g_2(\cdot) , \ldots , g_m(\cdot)]$ reflect the complex cross-sensitivity of the gas sensor array [7-8]

The formation of mapping relationship H depends on the learning of network itself. When neural network are trained with enough known samples and network's output error approaches zero, mapping relation from sensor array signal to gas concentration is saved on the linking weight of network which is approximate to sensors inverse mapping and can be used to unknown pattern's identification of sensor array.

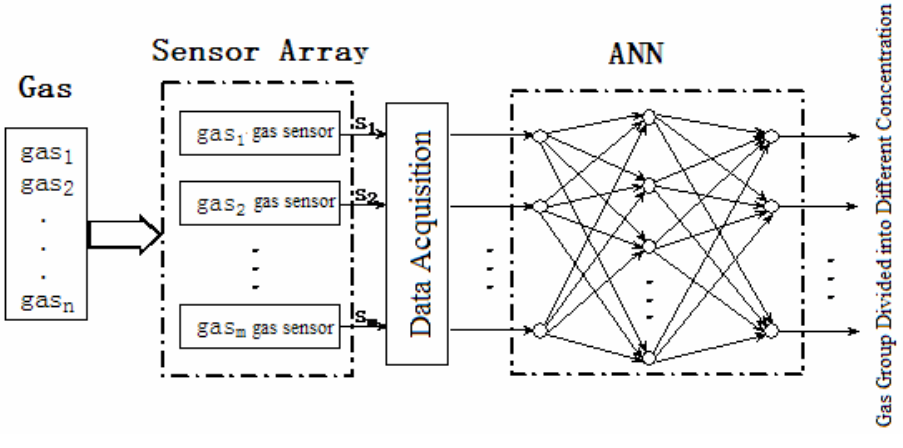


Fig. 1. Principle of gas's identification and measurement by gas sensor array

3 Feedforward Neural Network and Its Error Back Propagation Algorithm (BP Algorithm)

Artificial neural network (ANN) pattern identification method has several advantages. Firstly, it has simulated neural signal's transmission and integration mechanism which make it most close to olfactory system. Secondly, it has good fault-tolerance and strong adaptability. Moreover, it has strong self-learning ability and fast operation performance after network trained. Therefore, ANN has good application prospect in smart sensor technology research.

3.1 BP Neural Network and Its Algorithm

BP neural network's structure is shown in figure 2. It is made up of one input layer, one output layer, and several hidden layers. Each layer has several neurons. BP algorithm's learning process is made up of forward propagation and backward propagation [6-7].

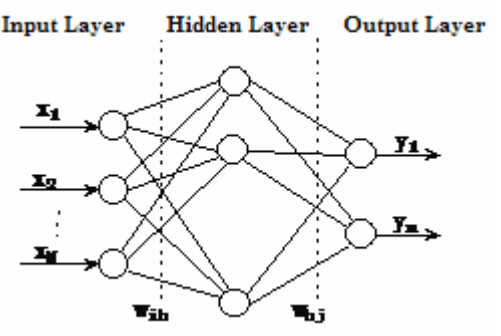


Fig. 2. BP network structure

Forward propagation is that input signals transmit from input layer to hidden layers and then to output layer. It is described as formula 2 and formula 3.

$$y_h = f_1(\sum_i w_{ih} x_i) \tag{2}$$

$$y_f = f_2(\sum_h w_{hj} f_1(\sum_i w_{ih} x_i)) \tag{3}$$

Here x_i is input signal, y_h is output of hidden layer and f_1 is its transfer function, y_f is output of output layer and f_2 is its transfer function, w_{ih} is linking weight from input layer to hidden layer, w_{hj} is linking weight from hidden layer to output layer.

If target value of output layer cannot be obtained, let difference between target value and output value transmit backward, modify linking weight of each neuron layer by layer which make output error reduce to allowable range. The process is called backward propagation. Its weight value is modified as follows:

$$w_{hj}(n+1) = w_{hj}(n) + \eta \delta_j y_h \tag{4}$$

where $\delta_j = f_2^{'}(\sum_h (w_{hj}(n) y_h)) * (T_j - y_j)$

$$w_{ih}(n+1) = w_{ih}(n) + \eta \delta_h x_i \tag{5}$$

where $\delta_h = f_1^{'}(\sum_i (w_{ih}(n) x_i)) * \sum_j (w_{hj}(n) \delta_j)$

In formula 4 and formula 5, $w_{hj}(n+1)$ and $w_{ih}(n+1)$ are the *nth* weight coefficients, t_j is the *jth* node target value, η is the training speed coefficient which is between 0 and 1.

3.2 BP Network's Improved Algorithm

Above traditional BP network is easy to converge to local minimum point [9]. Moreover its convergent speed is slow. To avoid this defect, additional momentum method was put forward to modify weight value. When modifying weight value, not only error's effect on grads but also error's trends on curved surface would be considered, a value directly proportional to previous weight variable value would be added to each variable weight in forward propagation. Moreover a new variable weight would be obtained according to backward propagation. This is so called "additional momentum method". Its regulating formula is:

$$w_{ij}(n+1) = w_{ij}(n) + (1-mc)\eta \delta_j y_i + [mc](w_{ij}(n) - w_{ij}(n-1)) \tag{6}$$

where $[mc]$ is momentum factor which is close to 0.95(0.9 in this paper), w_{ij} is linking weight from the *ith* layer to *jth* layer.

Additional momentum's essence is to transmit the last weight value's influence through a momentum factor $[mc]$. When $[mc]=0$, it is traditional BP algorithm. When $[mc]=1$, new weight increment is set to be the last weight increment. After increasing momentum, weight is urged to change to the average way of error curved surface's bottom. When network weight entering the flat region of curved surface's bottom, δ_i would be very small and $\Delta w_{ij}(n+1) \approx \Delta w_{ij}(n)$. It is helpful for the weight to get away from the minimum value of the error curved surface. According to additional momentum theory, if modifying lead to error larger growth, modifying should be cancelled that make momentum influence stop and network not to enter larger error curved surface. When new error exceeds a fixed maximal error change rate, new weight would be cancelled. Practice shows that additional momentum method can effectively avoid the network to fall into local minimum point and its operation speed can be improved greatly.

Another effective method is L-M (Lerenberg-Marquardt) algorithm to speed up feedforward network's training speed. L-M algorithm is a training algorithm which based on an optimization method. Traditional BP algorithm usually need longer time for training. However, L-M algorithm is much faster than BP algorithm but need more memory. Generally speaking, Lerenberg-Marquardt algorithm need memory which is traditional BP algorithm's S*Q times. Where S is the number of output neurons and Q is input-output vector number of training network. L-M algorithm's rule for updating parameters is shown as follows:

$$\Delta\omega = (J^T J + \mu I)^{-1} \bullet J^T e \tag{7}$$

where J is Jacobian matrix of differentiation for error to weight, e is error vector; μ is a scalar which is adjusted adaptively in the method.

Variable μ determine whether network's learning algorithm according to Newton algorithm or gradient algorithm. With the increasing of μ, the item $J^T J$ can be ignored in L-M rule which make formula 7 close to gradient algorithm. When variable μ is very small, formula 7 changed to be Guass-Newton method. So learning process descends mainly according to gradient method especially item $\mu^{-1} J^T e$. If error increase in iterative process, μ would increase accordingly until error stop increasing. But if μ is too large, learning would stop. Because if item $\mu^{-1} J^T e$ is very close to 0 and minimum error is found, this case would happened which make leaning stop when μ reach the maximal value.

4 Test Result Analysis

Six semiconductor gas sensors constitute an array in experiment. Each of the six sensors is sensitive with four different gases, but the degree is different .Deploy sensors in this way can form muti-dimensional response pattern for the mixed gases generally. At the same time, this method can give prominence to the information of one gas in one or more dimensions.

4.1 Sample Preprocessing

When neural network learning, there are abnormal samples in input training samples or data sample which added to input end is too large. As a result, neuron nodes arrives saturation state rapidly which lead to network's paralysis phenomenon. Moreover, logsig functions are mostly used in neural network. Its output range is between 0 and 1 but cannot reach 0 or 1. In this paper, formula 8 and formula 9 are adopted in data's normalization processing. After transformation, the value of input sample is between 0 and 1, and the value of output sample is between 0.05 and 0.95. Input data is between 0 and 1 sometimes, but its value is close to high precision that learning is difficult to reach. After normalization processing, data difference would get larger.

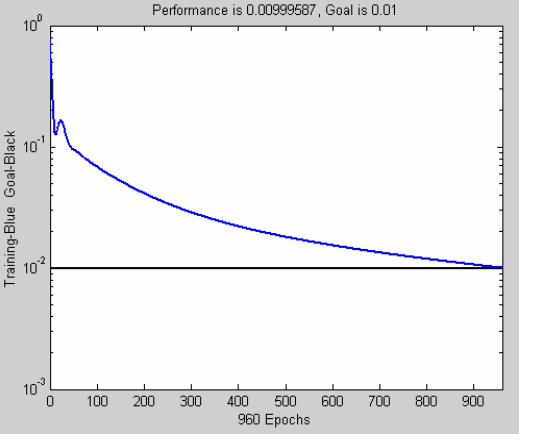
$$\overline{x_i} = \frac{(x_i - x_{\min})}{(x_{\max} - x_{\min})} \tag{8}$$

$$\overline{y_i} = \frac{0.9(y_i - y_{\min})}{(y_{\max} - y_{\min})} + 0.05 \tag{9}$$

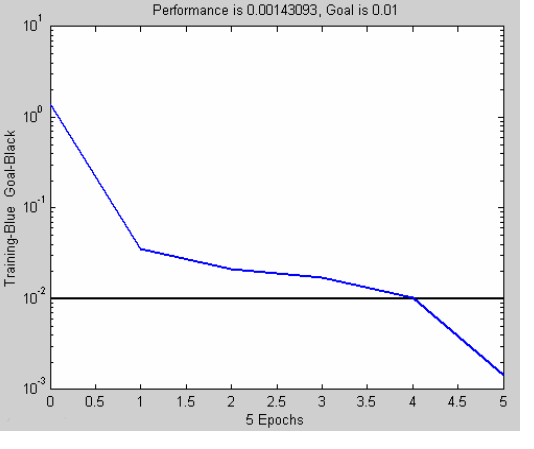
Here $\overline{x_i}$ is input data and $\overline{y_i}$ is output data after normalization processing, x_i and y_i are original data, $x_{max}, x_{min}, y_{max}, y_{min}$ are maximum and minimum value of input and output respectively.

Table 1. Neural network output result

Sample	Gas's Calibration Value (10^{-6})				Identification Result of Neural Network (10^{-6})			
	C_2H_6	C_2H_2	C_2H_4	CO	C_2H_6	C_2H_4	C_2H_2	CO
1	40	0	0	0	40.6765	0.7978	2.4444	0.6986
2	30	0	0	0	29.5880	0.4107	2.3616	2.1790
3	0	0	40	0	1.9578	0.1342	41.9649	1.8562
4	0	0	30	0	1.8239	0.2202	34.5201	1.6776
5	0	0	0	40	5.5188	0.0265	1.9064	37.6513
6	0	0	0	30	4.0514	0.0412	1.9593	29.3990
7	0	8	0	0	1.4485	7.4232	2.0728	2.0267
8	0	6	0	0	1.5349	5.7999	2.3844	2.1042



(a) Additional momentum algorithm



(b) L-M algorithm

Fig. 3. Error curves of Additional momentum algorithm and L-M algorithm

4.2 Test Result Analysis

In order to train and validate the Neural-Network designed, we can collect the swatches of C_2H_6,C_2H_4,C_2H_2 and CO. Thirty-two of the forty swatches in all are used to train the Neural-Network; the others are used to test.

Additional momentum Method and L-M Method are given in this report. They can be used to train network to the same error level respectively. Different methods use different training steps. The curves of the two methods are shown at Chart 3. After several iterative trainings, the error level can converge to 10^{-2}. And then fix up the weight value and threshold; validate it with the swatches forecasted. The results are as List1.It can be seen from the list that the results of the swatches forecasted by the network tally with the physical truth.

5 Conclusion

The sensor array system concerned in this test makes gas sensor array technology and BP man-made N-N band together. Moreover, it gives a method to analyzed one gas with the density in certain extend by experiment. The experiment results demonstrate that the intelligent sensor system can be used to analyses four characteristic gases in fault include C_2H_2 at 1×10^{-6}~10×10^{-6} grade and CO, C_2H_4, C_2H_6 at 10×10^{-6}~50×10^{-6} grade . There are such advantages as rapid learning speed and high disposal precision.

Smart sensor system which based on gas sensor array and ANN can complete single gas's qualitative identification and quantitative analysis. Research of this domain would be strengthened for its potential application value.

Reference

1. Jiao, L.C.: Theory of Neural Networks System. Xidian University Press, Xi'an (1990)
2. Hu, W.P., Yin, X.G., Zhang, Z.: Fault Diagnosis of Transformer Insulation Based on Compensated Fuzzy Neural Network. In: IEEE Annual Report Conference on Electrical Insulation and Dielectric Phenomena. pp. 273–276 (2003)
3. Vanegas, O., Mizuno, Y., et al.: Diagnosis of Oil-insulated Power Apparatus by Using Neural Network Simulation. IEEE Transactions on Dielectrics and Electrical Insulation 4, 290–299 (1997)
4. Xu, W., Wang, D., Zhou, Z., et al.: Fault Diagnosis of Power Transformers: Application of Fuzzy Set Theory, Expert and Artificial Neural Networks. IEE Proc. Sci. Meas. Technol. 1, 39–44 (1997)
5. Liu, J.H.: System of Intelligence Sensor. Xidian University Press, Xi'an (1999)
6. Zaromb, S., Stetter, J.: Theoretical Basis for Identification and Measurement of Air Contaminants Using an Array of Sensors Having Partly over Lapping Selectivities. Sensors and Actuators 6, 225–243 (1984)
7. Wu, H.Y., Chang, B.Y., et al.: Gas Sensor Array System for Analyzing Fault in Transformer. Journal of Xi'an Jiaotong University 34, 23–25 (2000)
8. Hanaki, S., Nakamoto, T., Moriizumi, T.: Artificial Odor-recognition System Using Neural Network for Estimating Sensory Quantities of Blended Fragrance. Sensors and Actuators A 57, 65–71 (1996)
9. Barker, B.S.: Vapour Recognition Using Organic Films and Artificial Neural Networks. Sensors and Actuators 17, 143–147 (1994)

Using Chaotic Neural Network to Forecast Stock Index

Bo Ning[1], Jiutao Wu[1], Hui Peng[2,*], and Jianye Zhao[1,*]

[1] School of Electronics Engineering and Computer Science,
Peking University, Beijing 100871, China
phdzjy@263.net
[2] School of Economics and Management,
Beijing University of Posts and Telecommunications,
Beijing 100876, China
gracepenghui@126.com

Abstract. In this paper, a new scheme based on chaotic neural network for stock index prediction is proposed. The data from a Chinese stock market, Shenzhen stock market, are applied as a case study. The chaotic neural network is used to learn the non-linear stochastic and chaotic patterns in the stock system and forecast a new index with former indexes. The validity of the scheme is analyzed theoretically, and the simulation results show that it has a good performance.

Keywords: Stock index, Chaotic neural network, Forecast.

1 Introduction

People tend to invest in stock because it has a high returns over time. Stock markets are affected by economic, social, political and even psychological factors, and these factors interact with each other in a very complicated manner. Therefore, it is very difficult to forecast stock index in general ways.

Recent studies have introduced neural network to analysis stock market systems [1-5]. Yao et al. [1] proposed a profit based adjusted weight factor for back propagation in which factors containing the profit, direction, and time information were added to the error function. In [2], the authors used both feed forward neural network and simple recurrent neural network, trained by time and profit based back propagation algorithm with early stopping to make the prediction. The network inputs, architecture, training strategies were decided based on experimental results. [3] introduces how the idea of Kalman prediction can be used to the study of the prediction of neural networks. Flexible neural trees are also introduced in [4]. However, some data coming from stock markets could be irregular and random. None of these schemes give due attention to reducing the influence of irregular and random data to the prediction.

In this paper, a new scheme of stock index forecast is proposed. Unlike schemes in [2] and [3], we bring chaotic neural network in to make the forecasting. Because of its complex dynamics, chaotic neural network has more memory capacity and error tolerance than other neural networks [7]. With this advantage, chaotic neural network

* Corresponding author.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 870–876, 2009.

can be used in detecting and data processing [9]. Therefore, we use chaotic neural network to reduce the influence of irregular and random data to the prediction. In the proposed scheme, the chaotic neural network is trained with historical data and then forecast new stock indexes with former ones. The results of simulation show that the proposed scheme has a good performance.

This paper is organized as follows: In Section 2, the detail design of the chaotic neural network is theoretically analyzed. Section 3 describes the system structure. Results of simulation are given in Section 4. Section 5 shows some concluding remarks.

2 Theory Analysis

The system structure is set based on Takens embedding theorem. According to the theorem, the geometrical structure of d-dimension dynamics system could be observed from the D-dimension vector

$$X_R(n) = [x(n), x(n-\tau), \ldots, x(n-(D-1)\tau)]^T, \tag{1}$$

where τ is an integer denoting the embedded delay. If D≥2d+1, reconstruction could be achieved from $X_R(n)$.

In the proposed scheme, the Aihara chaotic neuron is used as the basic element of the chaotic neural network module. The chaotic neuron model is expressed as follows [8]:

$$x(t+1) = f[A(t) - \alpha \sum_{d=0}^{t} k^d g\{x(t-d)\} - \Theta], \tag{2}$$

where x(t+1)is the output of the chaotic neural network, t is the discrete time steps, t=0,1,2,…n…, A(t) is the external stimulation at the time t, f is the output function, g is the refractory function, α , k and Θ are the refractory scaling parameter, the refractory decay parameter and the threshold, respectively.

Based on experimental results, we set f, g and Θ as follows: f adopts the function $\varphi_j(x_j(n)) = a\tanh(bx_j(n))$, (a, b)>0; g(y) = y; Θ=a. Because of the characteristic of the function y = tanh (x), $bx_j(n)$ should be among -1.0 and 1.0 to be in the linear area. The dynamics of the *i*th chaotic neuron is simplified as follows:

$$x_i(t+1) = f[\sum_{j=1}^{M} \varepsilon_{ij} \sum_{d=0}^{t} k_e A_j(t) + \sum_{j=1}^{N} \omega_{ij} \sum_{d=0}^{t} k_f x_j(t) - \alpha \sum_{d=0}^{t} k_r g\{x_j(t)\} - a], \tag{3}$$

where k_e, k_f and k_r are the parameters for the external inputs, the feedback inputs, and the refractoriness, respectively, ε_{ij} and ω_{ij} are synaptic weights to the *i*th neuron from the *j*th external input and from the *j*th neuron respectively. A chaotic neural of the model is illustrated in Figure 1.

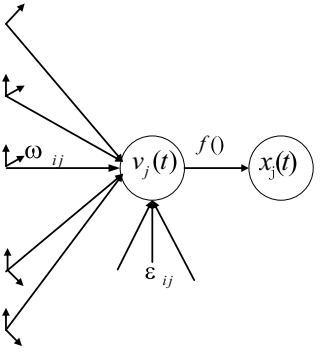


Fig. 1. A chaotic neural of the model

In the proposed scheme, the value of the weights is learning based on LMS algorithm. The learning processing is as follows:

$$\Delta\varepsilon_{ij}(n)=-\eta\frac{\partial E(n)}{\partial\varepsilon_{ij}(n)}=\eta e_j(n)\cdot\varphi_j'(v_j(n))\cdot x_j(n)=\eta\frac{\partial E(n)}{\partial v_j(n)}x_j(n), \tag{4}$$

$$\Delta\omega_{ij}(n)=-\eta\frac{\partial E(n)}{\partial\omega_{ij}(n)}=\eta e_j(n)\cdot\varphi_j'(v_j(n))\cdot x_j(n)=\eta\frac{\partial E(n)}{\partial v_j(n)}x_j(n), \tag{5}$$

where is $x_j(n)$ the output of the *j*th neuron, v_j is the inner state variable.

3 System Structure

The new scheme of stock index processing is illustrated in Figure 2. Its data processing flow is as follows:

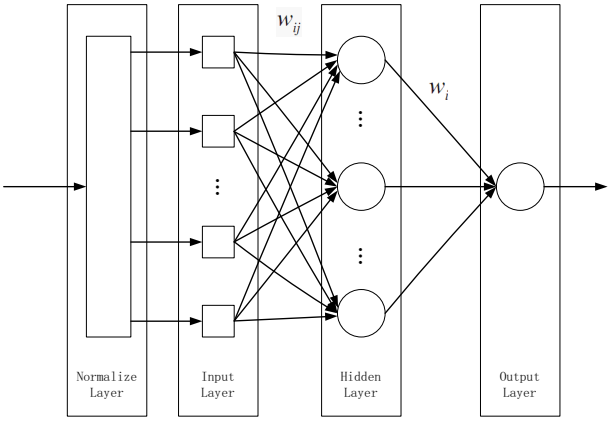


Fig. 2. The block diagram of stock index processing

Before forecasting, the stock index processing scheme is working in the "training period". During the training period, the stock indexes pass through normalize layer. The normalized data is among -1.0 and 1.0 to make the function f in (2) work in linear area. Then the data is input to the chaotic neural network. The chaotic neural network is trained to learn the non-linear stochastic and chaotic patterns in the stock system. The number of training and the number of chaotic neural of every layer is adjusted based on experimental results.

The $\frac{\partial \mathrm{E}(n)}{\partial v_j(n)}$ in expression (4) and (5) is different in output layer and hidden layer. For output layer,

$$\frac{\partial \mathrm{E}(n)}{\partial v_j(n)} = \eta e_j(n) \cdot \varphi_j'(v_j(n)) = \frac{b}{a}[d(n) - o(n)][a - o(n)][a + o(n)], \tag{6}$$

where o(n) is the predictive stock index, d(n) is the real stock index, value, a and b are the parameters of the output function $\varphi_j(x_j(n)) = a\tanh(bx_j(n))$, (a, b)>0. For hidden layer,

$$\frac{\partial \mathrm{E}(n)}{\partial v_j(n)} = \varphi_j'(v_j(n)) \cdot \sum_k \frac{\partial \mathrm{E}(n)}{\partial v_k(n)} \cdot \omega_{kj} = \frac{b}{a}[a - x_j(n)][a + x_j(n)] \sum_k \frac{\partial \mathrm{E}(n)}{\partial v_k(n)} \cdot \omega_{kj} \cdot \tag{7}$$

After the system has been trained successfully, the system begins to work in the "forecasting period". The stock indexes pass through every layer without changing the weights. The output of the chaotic neural network becomes the forecasted stock index by being anti-normalized.

4 Simulations

The stock indexes used for training and forecasting come from Shenzhen stock market. We use more than 33000 stock indexes to train the chaotic neural network. Then we inspect the performance. Our target is to develop efficient forecast models that could predict the index value of the following trade day based on former indexes. The simulation result is illustrated in Fig.3, Fig.4 and Fig.5. The real line corresponds to the real indexes, and the dashed corresponds to the predictive indexes.

For (a) and (b) in Fig.3, the chaotic neural network is working in the training period. With the number of training increasing, the predictive error goes down. This implies that the chaotic neural network is trained successfully.

For Fig.4, the chaotic neural network is working in the forecasting period. It can be seen that despite some predictive errors are high, most of the errors are below 100. This implies that in most situations the chaotic neural network has a good performance in forecasting.

In Fig.5, the relative errors of forecasted indexes are shown. In order to illustrate the advantage of the proposed scheme, we use back propagation neural network of similar complexity as a comparison. The same input data is used to train both networks. The number of training is also the same. (a) and (b) show the performance of chaotic neural network and back propagation neural network, respectively. It can be seen that the relative errors of chaotic neural network is smaller in general. The average errors of chaotic neural network and back propagation neural network are 0.77% and 1.07%, respectively. This implies that chaotic neural network has a better performance than back propagation neural network.

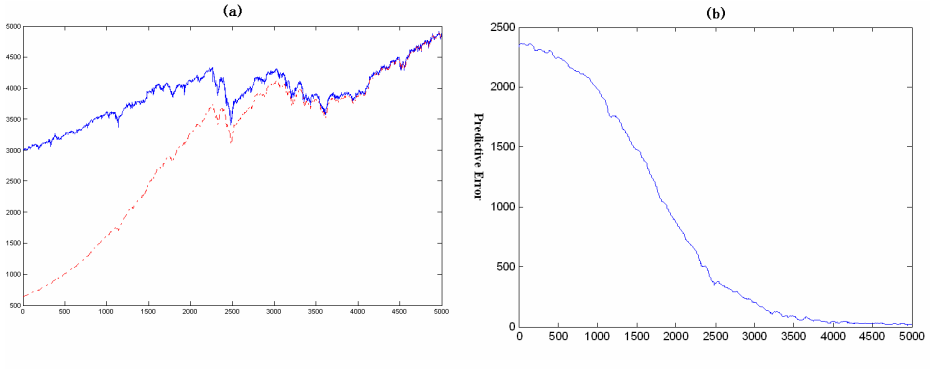


Fig. 3. The training of the chaotic neural network

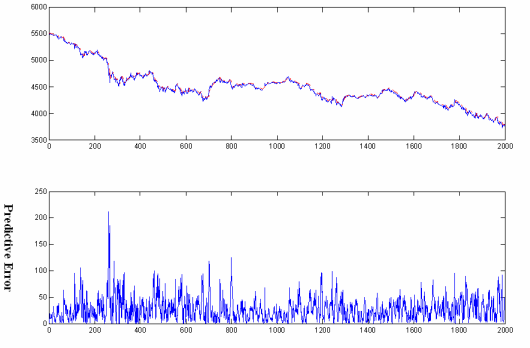


Fig. 4. Forecasting with the chaotic neural network

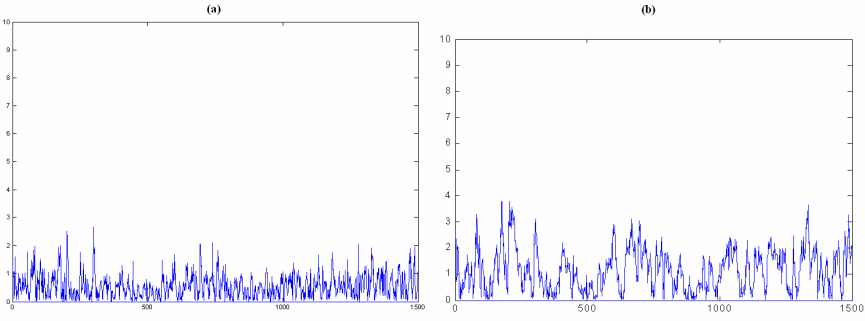


Fig. 5. The relative errors of forecasted indexes

5 Conclusion

In this paper we proposed a new scheme of stock index forecasting. The chaotic neural network is designed to approach the inherent function hid in the stock indexes and to learn the non-linear stochastic and chaotic patterns in the stock system. Because of its complex dynamics, chaotic neural network has more memory capacity and error tolerance than other neural networks. Therefore, we use chaotic neural network to reduce the influence of irregular and random data to the prediction. The validity of the scheme is analyzed theoretically. The simulation results show that it has good performance in forecasting new stock indexes, and has a better performance than back propagation neural network of similar complexity.

Acknowledgments. The letter is supported by NSF China, Grand No. 60704040.

References

1. Yao, J.T., Tan, C.L.: Time dependent directional profit model for financial time series forecasting. In: Proceeding of the IJCNN, Como, vol. 5, pp. 291–296 (2000)
2. Khoa, N.L.D., Sakakikara, K., Nishikawa, I.: Stock price forecasting using back propagation neural networks with time and profit based adjusted weight factors. In: SICE-ICASE International Joint Conference 2006, Busan, pp. 5484–5488 (2006)
3. Quah, T.S.: DJIA stock selection assisted by neural network. Expert Systems with Applications 35, 50–58 (2008)
4. Zhang, L., Zhong, C.Q., Zhang, L.Y., Ma, F., Zhang, L.: Application of innovations feedback neural networks in the prediction of ups and downs balue of stock market. In: Proceedings of the 6th World Congress on Intelligent Controland Automation, Dalian, vol. 1, pp. 4162–4166 (2006)
5. Tseng, C.H., Cheng, S.T., Wang, Y.H., Peng, J.T.: Artificial neural network model of the hybrid EGARCH volatility of the Taiwan stock index option prices. Physica A 387, 3192–3200 (2008)

6. Chen, Y.H., Yang, B., Abraham, A.: Flexible neural trees ensemble for stock index modeling. Neurocomputing 70, 697–703 (2007)
7. Adachi, M., Aihara, K.: Associative dynamics in a chaotic neural network. Neural Networks 10, 83–98 (1997)
8. Aihara, K., Takabe, T., Toyoda, M.: Chaotic neural networks. Physics Letters A 144, 333–340 (1990)
9. Ren, Q.S., Wang, J., Meng, H.L., Zhao, J.Y.: An adaptive radar target signal processing scheme based on AMTI filter and chaotic neural networks. In: Liu, D., Fei, S., Hou, Z., Zhang, H., Sun, C. (eds.) ISNN 2007. LNCS, vol. 4492, pp. 88–95. Springer, Heidelberg (2007)

A Fast Algorithm for 2-D ARMA Parameters Estimation

Zhipo Deng and Youshen Xia

College of Mathematics and Computer Science,
Fuzhou University, Fuzhou 350108, China
dicktank@gmail.com, ysxia2001@yahoo.com

Abstract. This paper proposed a fast algorithm for 2-D ARMA parameter estimation under noise environments. The proposed algorithm is based on a kind of inverse filter technique for the approximation of random excitation. The proposed algorithm only solves a low order 2-D AR system for the 2-D ARMA parameter. In contrast with Zhang and Cheng 's and Kizilkaya and Kayran's algorithms for 2-D ARMA parameter estimation, the proposed algorithm considers the noisy observation and is more suitable for practical applications. Moreover, the proposed algorithm has a fast speed due to a low computational cost. Simulation results show that the proposed algorithm can obtain more accurate estimates with a faster speed than the two algorithms.

Keywords: ARMA, Algorithm.

1 Introduction

The parameter estimation of 2-D autoregressive moving average(ARMA) process has found many applications in blind image restoration [1-3], texture analysis [4], image encoding [5-7], system identification and spectral estimation [8,9]. The key of modeling 2-D random field by 2-D ARMA process is to estimate the 2-D ARMA model parameters since it directly affects applications of the 2-D ARMA model. There are a lot of methods developed for 1-D autoregressive(AR) and ARMA parameter estimation, such as [10,11,12]. However, these methods for 1-D cases can't be easily extended and effectively applied to 2-D cases due to 2-D dimensional complexity.

There are several conventional algorithms for 2-D ARMA parameter estimation. Tekalp et al developed a maximum-Likelihood algorithm, which has a complexity problem for implementation [2,3]. Zhang and Cheng proposed a two-step algorithm [9,13]. In their algorithms, a form of 2-D MYW equation is introduced. AR parameters are obtained by solving this new form of 2-D modified Yule-Walker(MYW) equations in the first step. Relationship between MA spectrum coefficients and MA parameters are used in MA parameters estimation. MA parameters are then obtained by solving this set of nonlinear equations. Because of computational complexity and convergence problems, this algorithm could not got better estimate accuracy. Recently, Kizilkaya and Kayran proposed a more

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 877–886, 2009.

efficient algorithm [14]. In their algorithm, a 2-D EAR model parameters are first estimated by 2-D MYW equations. Unlike Zhang and Cheng's algorithm, Kizilkaya and Kayran's algorithm can get a better estimate accuracy due to no convergence problem. On the other hand, the accuracy of the estimated parameters are highly dependent of the accuracy of EAR parameters. Also, Kizilkaya and Kayran's algorithm need constructing many matrices with high dimension to obtain MA parameters and AR parameters. Thus there is a problem of computational cost. Moreover, both Zhang and Cheng's algorithm and Kizilkaya and Kayran's algorithm have no consideration on noisy cases.

In this paper, we proposed a fast algorithm for 2-D ARMA parameters estimation under noise environments. This algorithm is mainly based on the the relationship between ARMA parameters and covariance. Because the knowledge of random excitation is unknown, the inverse filter technique is first used to predict the random excitation from observation data. We then transform a 2-D ARMA(p_1, p_2, q_1, q_2) model into an approximate $AR(p_1 + q_1 + 1, p_2 + q_2 + 1)$ model. As a result, the proposed algorithm is simple and needs less computational cost. Compared with conventional algorithms [9,13] and [14], the proposed algorithm can reduce computational complexity and is more suitable for applications in noisy observation cases. Simulation results show that the proposed algorithm can obtain more accurate estimates with a fast speed than two standing algorithms.

2 2-D ARMA Model and Algorithms

2.1 Model and Assumption

Considering following quarter plane causal 2-D ARMA model of order (p_1,p_2,q_1,q_2)

$$\sum_{i=0}^{p_1}\sum_{j=0}^{p_2} a_{i,j}x(n_1-i,n_2-j) = \sum_{m=0}^{q_1}\sum_{n=0}^{q_2} d_{m,n}w(n_1-m,n_2-n) \tag{1}$$

where $x(n_1, n_2)$ is a stability random field, random excitation $w(n_1, n_2)$ is white with zero mean and variance σ_w^2. In practice, measurements of the random field is noisy:

$$y(n_1, n_2) = x(n_1, n_2) + u(n_1, n_2)$$

for $1 \le n_1 \le N_1,\ 1 \le n_2 \le N_2$, where $u(n_1, n_2)$ is observation noise of zero mean. It is assumed that $a_{0,0} = 1$ and $d_{0,0} = 1$. So our object is to estimate the AR parameters $d_{i,j}(0 \le i \le p_1, 0 \le j \le p_2, (i, j) \ne (0, 0))$ and MA parameters $a_{m,n}(0 \le m \le q_1, 0 \le n \le q_2, (i, j) \ne (0, 0))$.

2.2 Zhang and Cheng's Algorithm

Zhang and Cheng (1991) presented a three-step algorithm [9,13]. First, the following 2-D modified Yule-Walker(MYW) equation is solved for AR parameters $\{a_{i,j}\}$:

$$\sum_{i=0}^{p_1}\sum_{j=0}^{p_2} a_{i,j} r_{xx}(q_1+l-i, q_2+m-j) = d_{0,0} d_{q_1,q_2} \sigma_w^2 \delta(l,m) \tag{2}$$

where $\delta(l,m)$ is Kronecker delta function with $\delta(l,m)=1$ for $(l,m)=(0,0)$, otherwise $\delta(l,m)=0$. $d_{0,0}$ and d_{p_1,p_2} are MA parameters, $d_{0,0}d_{q_1,q_2}\sigma_w^2$ in (2) can be can be computed by the assumption that $a_{0,0}=1$. $r(k_1,k_2)$ denotes the covariance of the random field which is given by

$$\begin{aligned} r_{xx}(k_1,k_2) &= \frac{1}{(N_1-k_1)(N_2-k_2)} \sum_{n_1=1}^{N_1-k_1}\sum_{n_2=1}^{n_2-k_2} x(n_1,n_2)\cdot x(n_1+k_1,n_2+k_2) \\ &= r_{xx}(-k_1,-k_2) \qquad for(\quad k_1 \geq 0, \quad k_2 \geq 0) \\ r_{xx}(k_1,-k_2) &= \frac{1}{(N_1-k_1)(N_2-k_2)} \sum_{n_1=1}^{N_1-k_1}\sum_{n_2=1}^{n_2-k_2} x(n_1+k_1,n_2)\cdot x(n_1,n_2+k_2) \\ &= r_{xx}(-k_1,k_2) \qquad for(\quad k_1 \geq 1, \quad k_2 \geq 1) \end{aligned} \tag{3}$$

Next, MA spectrum parameters $c_{k,m}$ and $f_{k,m}$ are computed. Finally, MA parameters $d_{i,j}$ are obtained by solving the following nonlinear equations:

$$\sigma_w^2 \sum_{i=0}^{q_1}\sum_{j=0}^{q_2} d_{i,j}^2 = 2c_{0,0} \tag{4}$$

$$\sigma_w^2 \sum_{i=0}^{q_1-k}\sum_{j=0}^{q_2-m} d_{i,j} d_{i+k,j+m} = c_{k,m} \tag{5}$$

$$k=0,\cdots,q_1, m=0,\cdots,q_2, but(k,m)\neq q(0,0)$$

$$\sigma_w^2 \sum_{i=0}^{q_1-k}\sum_{j=m}^{q_2} d_{i,j} d_{i+k,j-m} = f_{k,m} \tag{6}$$

$$k=0,\cdots,q_1, m=0,\cdots,q_2$$

2.3 Kizilkaya and Kayran's Algorithm

Kizilkaya and Kayran developed an equivalent AR algorithm for ARMA parameters estimation [14]. Unlike Zhang and Cheng's algorithm, equivalent AR parameters $b_{i,j}$ are first solved by following 2-D MYW equation

$$\sum_{s=0}^{L_1}\sum_{t=0}^{L_2} b_{s,t} r_{xx}(l-s,m-t) = \sigma_w^2 \delta(l,m) \tag{7}$$

where $r_{xx}(i,j)$ and $\delta(l,m)$ are defined in (2). Then AR parameters are obtained by minimizing

$$\Phi = \Big(\sum_{k=0}^{q_1}\sum_{j=1}^{q_2} \boldsymbol{B}_{k,j} a_{k,j} + \sum_{h=1}^{q_1} \boldsymbol{B}_{h,0} a_{h,0} + \boldsymbol{B}_{0,0} - \boldsymbol{D}\Big)^2 \tag{8}$$

where $\boldsymbol{D}$ and $B_{k,j}$ are matrixes having dimension $(L_1+1)\times(L_2+1)$, defined as follows

$$\boldsymbol{D}=\begin{bmatrix} d_{0,0} & \cdots & d_{0,p_2} & 0\cdots 0 \\ \vdots & \cdots & \vdots & \vdots\cdots\vdots \\ d_{p_1,0} & \cdots & d_{p_1,p_2} & 0\cdots 0 \\ 0 & \cdots & 0 & 0\cdots 0 \\ \vdots & \cdots & \vdots & \vdots\cdots\vdots \\ 0 & \cdots & 0 & 0\cdots 0 \end{bmatrix}$$

$$B_{k,j}=\begin{bmatrix} 0\cdots 0 & 0 & 0 & \cdots & 0 \\ \vdots\cdots\vdots & \vdots & \vdots & \cdots & \vdots \\ 0\cdots 0 & 0 & 0 & \cdots & 0 \\ 0\cdots 0 & b_{0,0} & b_{0,1} & \cdots & b_{0,L_2-j} \\ 0\cdots 0 & b_{1,0} & b_{1,1} & \cdots & b_{1,L_2-j} \\ \vdots\cdots\vdots & \vdots & \vdots & \cdots & \vdots \\ 0\cdots 0 & b_{L_1-k,0} & b_{L_1-k,1} & \cdots & b_{L_1-k,L_2-j} \end{bmatrix}$$

The MA parameters are finally computed by

$$d_{m,n}=b_{m,n}+\sum_{k=0}^{q_1}\sum_{j=1}^{q_2}\boldsymbol{B}_{k,j}(m,n)a_{k,j}+\sum_{h=1}^{q_1}\mathbf{B}_{h,0}(m,n)a_{h,0} \tag{9}$$

Because Kizilkaya and Kayran's algorithm doesn't need solving nonlinear equations, they can obtain better estimate accuracy than Zhang and Cheng 's algorithm.

3 Our Algorithm for Parameter Estimation of 2-D ARMA Model

3.1 Our Estimation Method

Our estimation method is based on the obtained EAR parameters $\{b_{s,t}\}$ such that

$$w(n_1,n_2)\approx\sum_{s=0}^{L_1}\sum_{t=0}^{L_2}b_{s,t}x(n_1-s,n_2-t) \tag{10}$$

Substituting (10) into (1) we have

$$\begin{aligned} x(n_1,n_2)=&-\sum_{\substack{i=0\\(i,j)\neq(0,0)}}^{p_1}\sum_{j=0}^{p_2}a_{i,j}x(n_1-i,n_2-j) \\ &+\sum_{m=0}^{q_1}\sum_{n=0}^{q_2}\sum_{s=0}^{L_1}\sum_{t=0}^{L_2}d_{m,n}b_{s,t}x(n_1-m-s,n_2-n-t) \end{aligned} \tag{11}$$

Using $y(n_1, n_2) = x(n_1, n_2) + u(n_1, n_2)$ we have

$$y(n_1,n_2) = -\sum_{\substack{i=0\\(i,j)\neq(0,0)}}^{p_1}\sum_{j=0}^{p_2} a_{i,j}y(n_1-i,n_2-j) + \sum_{m=0}^{q_1}\sum_{n=0}^{q_2}\sum_{s=0}^{L_1}\sum_{t=0}^{L_2} d_{m,n}b_{s,t}y(n_1-m-s,n_2-n-t) + \overline{n}(n_1,n_2) \tag{12}$$

where $\overline{n}(n_1, n_2)$ is colored noise. Multiple both sides of (12) by $y(n_1 - k, n_2 - l)$ and taking expectation yields

$$r_{yy}(k,l) = -\sum_{\substack{i=0\\(i,j)\neq(0,0)}}^{p_1}\sum_{j=0}^{p_2} a_{i,j}r_{yy}(k-i,l-j) + \sum_{m=0}^{q_1}\sum_{n=0}^{q_2}\sum_{s=0}^{L_1}\sum_{t=0}^{L_2} d_{m,n}b_{s,t}r_{yy}(k-m-s,l-n-t) \tag{13}$$

where $r_{yy}(k, l)$ is the covariance of observation data, which is similar to (3).

Let $r'(k,l) = \sum_{s=0}^{L_1}\sum_{t=0}^{L_2} b_{s,t}r_{yy}(k-s,l-t)$, (13) can be rewrite as

$$r_{yy}(n_1,n_2) = -\sum_{\substack{i=0\\(i,j)\neq(0,0)}}^{p_1}\sum_{j=0}^{p_2} a_{i,j}r_{yy}(n_1-i,n_2-j) + \sum_{m=0}^{q_1}\sum_{n=0}^{q_2} d_{m,n}r'(n_1-m,n_2-n); \tag{14}$$

Note that the system is assumed to be causal, $w(n_1, n_2)$ is independent of $x(n_1 - k, n_2 - l)$ when $(k < 0, l < 0)$. Then

$$E[w(n_1,n_2)x(n_1-k,n_2-l)] = E[w(n_1,n_2)]\cdot E[x(n_1-k,n_2-l)] = 0 \quad for \quad (k<0,l<0) \tag{15}$$

It follows that

$$\begin{aligned} r'(k,l) &= \sum_{s=0}^{L_1}\sum_{t=0}^{L_2} b_{s,t}r_{yy}(k-s,k-t) \\ &= \sum_{s=0}^{L_1}\sum_{t=0}^{L_2} b_{s,t}E[y(n_1-k,n_2-l)y(n_1-s,n_2-t)] \\ &= \sum_{s=0}^{L_1}\sum_{t=0}^{L_2} b_{s,t}E[x(n_1-k,n_2-l)x(n_1-s,n_2-t)] \\ &\quad - \sum_{s=0}^{L_1}\sum_{t=0}^{L_2} b_{s,t}E[\overline{n}(n_1-k,n_2-l)\overline{n}(n_1-s,n_2-t)] \\ &= \sum_{s=0}^{L_1}\sum_{t=0}^{L_2} b_{s,t}E[x(n_1-k,n_2-l)x(n_1-s,n_2-t)] \end{aligned} \tag{16}$$

Using (10) we can obtain

$$\begin{aligned} r'(k,l) &= \sum_{s=0}^{L_1}\sum_{t=0}^{L_2} b_{s,t} E[x(n_1-k,n_2-l)x(n_1-s,n_2-t)] \\ &= E[\sum_{s=0}^{L_1}\sum_{t=0}^{L_2} b_{s,t} x(n_1-k,n_2-l)x(n_1-s,n_2-t)] \\ &\approx E[w(n_1,n_2)x(n_1-k,n_2-l)] = 0. \end{aligned} \tag{17}$$

So we only need computing $r'(k,l)$ for $(k \geq 0, l \geq 0)$.

Let values of k rang from 0 to p_1+q_1 and l rang from 0 to p_2+q_2 to (14). We denote column vector c which consists of both AR and MA parameters given by

$$\begin{aligned} \mathbf{c} = [&a(0,1),\cdots,a(0,p_2),a(1,0),\cdots,a(1,p_2),\cdots,a(p_1,0),\cdots,a(p_1,p_2), \\ &d(0,0),\cdots,d(0,q_2),d(1,0),\cdots,d(1,q_2),\cdots,d(q_1,0),\cdots,d(q_1,q_2)]^T \end{aligned} \tag{18}$$

Then (15) can be written as a system of $(p_1+q_1+1)\times(p_2+q_2+1)$ linear equations with respect to the unknown AR and MA parameters:

$$\mathbf{Rc} = \mathbf{g} \tag{19}$$

where

$$\begin{aligned} \mathbf{g} = [&r_{yy}(0,0),\cdots,r_{yy}(0,p_2+q_2),r_{yy}(1,0),\cdots,r_{yy}(0,p_2+q_2), \\ &\cdots,r_{yy}(p_1+q_2,0),\cdots,r_{yy}(p_1+q_1,p_2+q_2)]^T \end{aligned} \tag{20}$$

$$\begin{aligned} R = [&\mathbf{r}_{0,0},\cdots,\mathbf{r}_{0,(p_2+q_2)},\mathbf{r}_{1,0},\cdots,\mathbf{r}_{1,(p_2+q_2)} \\ &,\cdots,\mathbf{r}_{(p_1+q_1),0},\cdots,\mathbf{r}_{(p_1+q_1),(p_2+q_1)}]^T \end{aligned} \tag{21}$$

$\mathbf{r}_{i,j}$ in (21) is a vector with $(p_1+1)(p_2+1)+(q_1+1)(q_2+1)-1$ elements

$$\begin{aligned} \mathbf{r}_{i,j} = [&r_{yy}(i,j-1),\cdots,r_{yy}(i,j-p_2),r_{yy}(i-1,j),\cdots,r_{yy}(i-1,j-p_2),\cdots, \\ &r_{yy}(i-p_1,j),\cdots,r_{yy}(i-p_1,j-p_2),r'(i,j),\cdots,r'(i,j-p_2), \\ &r'(i-1,j),\cdots,r'(i-1,j-p_2),\cdots,r'(i-p_1,j),\cdots,r'(i-p_1,j-p_2)] \end{aligned}$$

ARMA parameters are finally estimated by computing the solution of (19), that is $c=(R^TR)^{-1}R^Tg$.

3.2 Algorithm Comparison

First, Zhang and Cheng 's algorithm needs solving nonlinear equations. Some numerical methods, such as Newton-Raphson method, may be used to solve this nonlinear equations, but the convergence conditions for their solutions could not be satisfied. Thus, Zhang and Cheng 's algorithm has a problem of computation implementation. In Kizilkaya and Kayran's algorithm, $q_1^2+q_1+2$ matrices with $(L_1+1)\times(L_2+1)$ dimension are constructed and operated to obtain MA parameters from (8) and AR parameters from (9). In contrast, our proposed algorithm is required to solve a system of $(p_1+q_1+1)\times(p_2+q_2+1)$ linear

equations. So, the proposed algorithm has a low computational cost and thus a faster speed. Next, unlike Zhang and Cheng 's and Kizilkaya and Kayran's algorithms, our proposed algorithm considers the noisy observation and is more suitable for practical applications.

4 Simulation

In this section, we give an example to demonstrate the effective performance of the proposed algorithm by comparing with two standing algorithms :Zhang and Cheng's algorithm [9] [13] and Kizilkaya and Kayaran's algorithm[14]. The simulation is conducted in MATLAB. The samples' size are chosen to be $(N_1, N_2) = (60, 60)$ and $(N_1, N_2) = (128, 128)$. The random excitation $w(n_1, n_2)$ is taken as a Gaussian process with zero mean noise and variance $\sigma^2 = 1$. The observation data was corrupted with additive Gaussian noise at SNR of 10 dB.

Similar to [14], we define the *Frobenius*(F)-norm error between estimated and original ARMA parameters as the measurement. In addition, spectral density of the 2-D ARMA(p_1, p_2, q_1, q_2) process, is computed by

$$P(e^{jw_1}, e^{jw_1}) = \sigma_w^2 |\frac{A(e^{jw_1}, e^{jw_1})}{D(e^{jw_1}, e^{jw_1})}|^2 = \sigma_w^2 |\frac{\sum_{h=0}^{q_1}\sum_{i=0}^{q_2} a_{h,i} e^{-j(w_1 h + w_2 i)}}{1 + \sum_{\substack{m=0 \\ (m,n)\neq(0,0)}}^{p_1}\sum_{n=0}^{p_2} d_{m,n} e^{-j(w_1 m + w_2 n)}}|^2 \quad (22)$$

Table 1. Computed results of estimated parameters

True value		Proposed algorith	Algorithm [9] [13]	Algorithm [14]	Proposed algorith	Algorithm [9] [13]	Algorithm [14]
		$(N_1, N_2) = (60, 60)$			$(N_1, N_2) = (128, 128)$		
$a_{0,1}$	0.2000	0.1649	0.1593	0.2411	0.1945	0.1944	0.2906
$a_{0,2}$	0.2300	0.2260	0.1906	0.2047	0.2254	0.1944	0.2286
$a_{1,0}$	0.1500	0.1386	0.1421	0.1434	0.1485	0.1455	0.2393
$a_{1,1}$	0.1800	0.1386	0.1421	0.1610	0.1385	0.1640	0.1923
$a_{1,2}$	0.1600	0.1491	0.1383	0.1288	0.1500	0.1444	0.1541
$a_{2,0}$	0.1700	0.1513	0.1505	0.1945	0.1607	0.1471	0.2154
$a_{2,1}$	0.2400	0.2202	0.2067	0.1933	0.2265	0.2154	0.2072
$a_{2,2}$	0.2100	0.1805	0.1750	0.1245	0.1976	0.1918	0.1557
F-norm error		0.0698	0.0804	0.1871	0.0483	0.0488	0.0588
$d_{0,1}$	0.1500	0.1179	0.1120	0.2868	0.1468	0.1460	0.1723
$d_{0,2}$	0.0900	0.1039	0.0705	0.1315	0.0995	0.0856	0.1057
$d_{1,0}$	0.1000	0.0922	0.1427	0.1775	0.0989	0.1612	0.1219
$d_{1,1}$	-0.1000	-0.1132	-0.0729	-0.1535	-0.1159	-0.0905	-0.1252
$d_{1,2}$	0.0500	0.0650	0.0516	0.0037	0.0504	0.0389	0.0526
$d_{2,0}$	-0.0500	-0.0426	-0.0540	-0.0638	-0.0381	-0.0546	-0.377
$d_{2,1}$	0.1300	0.1324	0.1072	0.1156	0.1308	0.1051	0.1188
$d_{2,2}$	0.0750	0.0648	0.0580	0.1312	0.0777	0.0695	0.0888
F-norm error		0.0430	0.0722	0.1152	0.0225	0.0683	0.1498

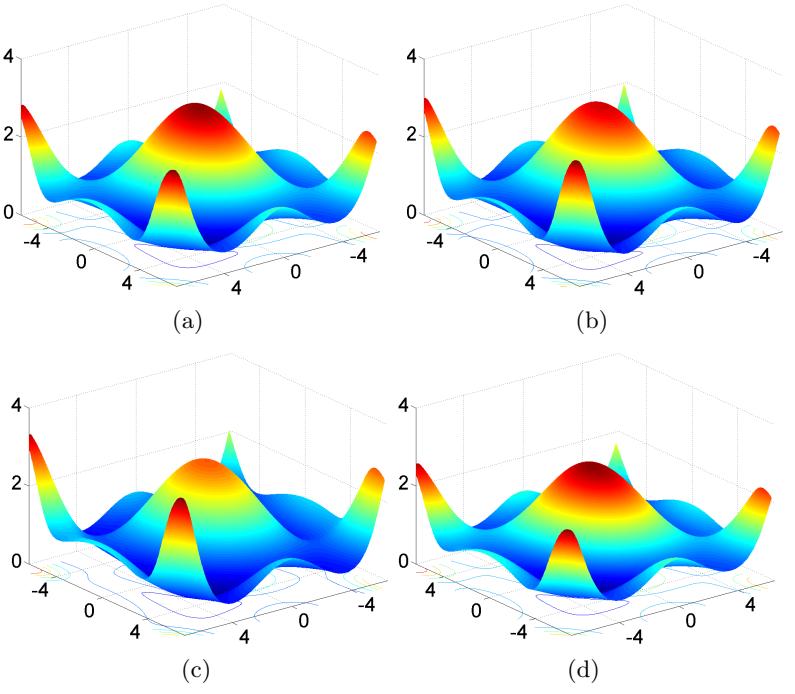


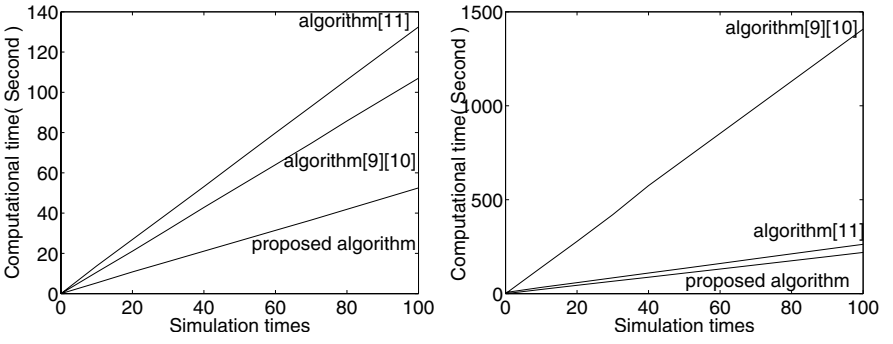
Fig. 1. Magnitude power spectrums with sample size $(N_1, N_2) = (128, 128)$;(a).original power spectrum;(b).estimated power spectrum by algorithm[9,13]; (c).estimated power spectrum by algorithm[14]; (d).estimated power spectrum by proposed algorithm

in order to obtain the power spectrums. In our simulation, we take w_1 and w_2 for $[-\pi, \pi]$ and draw the power spectrum figure.

Consider the parameter estimation of mixed-type 2-D ARMA model [14]. This ARMA model consists of wide-band and narrow-band ARMA process. 2-D ARMA(2,2,2,2) model of this kind having the transfer function:

$$H(z_1, z_2) = \frac{[1\ z_1^{-1}\ z_1^{-2}] \begin{bmatrix} 1 & 0.2 & 0.23 \\ 0.15 & 0.18 & 0.16 \\ 0.17 & 0.24 & 0.21 \end{bmatrix} \begin{bmatrix} 1 \\ z_1^{-1} \\ z_1^{-2} \end{bmatrix}}{[1\ z_2^{-1}\ z_2^{-2}] \begin{bmatrix} 1 & 0.15 & 0.09 \\ 0.1 & -0.1 & 0.05 \\ -0.05 & 0.13 & 0.075 \end{bmatrix} \begin{bmatrix} 1 \\ z_2^{-1} \\ z_2^{-2} \end{bmatrix}} \tag{23}$$

In this example, we take $L_1 = 7, L_2 = 6$ for $(N_1, N_2) = (60, 60)$, and $L_1 = 8, L_2 = 9$ for $(N_1, N_2) = (128, 128)$. These two kinds of selection has the optimal estimation for algorithm [14]. We perform three algorithms mentioned above, where in Zhang and Cheng's algorithm, we use the well-known Newton-Raphson method where the algorithm will stop after performs 500 iterations. Computed results is listed in Table 1 where the computed results were obtained by averaging 100 independent Monte Carlo simulations for different sample size



(a) $(N_1, N_2) = (60, 60)$ (b) $(N_1, N_2) = (128, 128)$

Fig. 2. Comparing of consume time

and the Frobenius-norm error represents difference between estimated parameters and original parameters. From Table 1 we can see that proposed algorithm obtains a good estimation result with a smaller F-norm error than other two algorithms in AR and MA parameters. Moreover, Figure 1 show that the proposed algorithm also estimates better power spectrum. Furthermore, Figure 2 shows that proposed algorithm takes less time than other two algorithms [13] [14].

From the simulation above, we can see that proposed algorithm is better in accuracy than other two algorithms. Moreover, the proposed algorithm takes less time than the other two algorithms as the ARMA size increase.

5 Conclusion

In this paper, we proposed a fast algorithm for 2-D ARMA parameter estimation. We transform the 2-D ARMA system into a equivalent 2-D AR system by using a expression of the random excitation. A system of linear equations with a low order is obtained for ARMA parameter estimation. Unlike Zhang and Cheng 's and Kizilkaya and Kayran's algorithms for 2-D ARMA parameters estimation, the proposed algorithm considers noisy observation in practical applications. Moreover, the proposed algorithm has a low computational cost and thus has a faster speed. Simulation result shows that proposed algorithm can obtain more accurate estimates than other two estimation algorithms with a fast speed.

Acknowledgements. This work was partly supported by the National Natural Science Foundation of China under Grant No. 60875085 and the National Natural Science Foundation of Fujian Province of China under Grant No. 2008J0019.

References

1. Kaufman, H., Tekalp, A.M.: Survey of Estimation Techniques in Image Restoration. IEEE Control Systems Magazine 11, 16–24 (1991)
2. Kaufman, H., Tekalp, A.M.: Image Identification and Restoration: A survey. In: 26th IEEE Conference on Decision and Control, pp. 1189–1195. Institute of Electrical and Electronics Engineers Press, California (1987)
3. Tekalp, A.M., Kaufman, H., Woods, J.W.: Identification of Image and Blur Parameters for the Restoration of Noncausal Blurs. IEEE Transactions On Acoustics Speech And Signal Processing 34, 963–972 (1986)
4. Hall, T.E., Giannakis, G.B.: Bispectral Analysis and Model Validation of Texture Images. IEEE Transactions on Image Processing 4, 996–1009 (1995)
5. Tao, K.M.: Adaptive Image Smoothing Algorithms for Edge and Texture Preservation. In: Proc.IEEE Int.Conf.Acoust,Speech,Signal Processing(ICASSP 1984), Alaska, vol. 9, pp. 287–290 (1984)
6. Sayood, K., Schekall, S.M.: Use of ARMA Predictors in the Differential Encoding of Images. In: IEEE Transactions on Acoustics,Speech and Signal Processing, vol. 36, pp. 1791–1795. Institute of Electrical & Electronics En, USA (1988)
7. Chung, Y.S., Kanefsky, M.: On 2-D Recursive LMS Algorithms Using ARMA Prediction for ADPCM Encoding of Images. IEEE Transactions on Image Processing 1, 416–422 (1992)
8. Chaparoo, L.F., Luo, L.: Identification of two-dimensional systems using sum-of-cumulants. In: 1992 IEEE International Conference on Acoustics,Speech,and Signal Processing, pp. 481–484. Institute of Electrical and Electronics, USA (1992)
9. Zhang, X.D., Cheng, Y.: High Resolution Two-dimensional ARMA Spectral Estimation. IEEE Transactions on Signal Processing 39, 765–769 (1991)
10. Zheng, W.X.: Autoregressive Parameter Estimation from Noise Data. IEEE Transactions on Circuits and systems 47, 71–75 (2000)
11. Rojo-Alvarez, J.L., Martinez-Ramon, M., Prado-Cumplido, M., de Artes-Rodriguez, A., Figueiras-Vidal, A.R.: Support Vector Method for Robust ARMA System Identification. IEEE Transactions On Signal Processing 52, 155–164 (2004)
12. Xia, Y.S., Kamel, M.S.: A Generalized Least Absolute Deviation Method for Parameter Estimation of Autoregressive Signals. IEEE Transactions on Neural Networks 19, 107–118 (2008)
13. Zhang, X.D.: On the Estimation of Two-Dimensional Moving Average Parameters. IEEE Transactions on automatic control 36, 1196–1199 (1991)
14. Kizilkaya, A., Kayran, A.H.: Estimation of 2-D ARMA Model Parameters by Using Equivalent AR Approach. Journal of the Franklin Institute 342, 39–67 (2005)

Grey Neural Network Based Predictive Model for Multi-core Architecture 2D Spatial Characteristics

Jingling Yuan, Tao Jiang, Jingjing He, and Luo Zhong

Computer Science and Technology School, Wuhan University of Technology, Wuhan, China 430070
yjl@whut.edu.cn

Abstract. The trend toward multi-/many- core processors will result in sophisticated large-scale architecture substrates that exhibit increasingly complex and heterogeneous behavior. Existing methods lack the ability to accurately and informatively forecast the complex behavior of large and distributed architecture substrates across the design space. Grey neural network is an innovative intelligent computing approach that combines grey system model and neural network. Grey neural network makes full use of the similarities and complementarity between grey system model and neural network to overcome the disadvantage of individual method. In this paper, we propose to use grey neural network to predict 2D space parameters produced by wavelet analysis,which can efficiently reason the characteristics of large and sophisticated multi-core oriented architectures during the design space exploration stage with less samples rather than using detailed cycle-level simulations. Experimental results show that the models achieve high accuracy while maintaining low complexity and computation overhead.

1 Introduction

Early design space exploration is an essential ingredient in modern processor development. It significantly reduces the time to market and post-silicon surprises [1]. The trend toward multi-/many-core processors will result in sophisticated large-scale architecture substrates with self-contained hardware components proximate to the individual cores but globally distributed across all cores. As the number of cores on a processor increases, these large and sophisticated multi-core-oriented architectures exhibit increasingly complex and heterogeneous characteristics. Processors with two, four and eight cores have already entered the market. Processors with tens or possibly hundreds of cores may be a reality within the next few years. In the upcoming multi-/many- core era, the design, evaluation and optimization of architectures will demand analysis methods that are very different from those targeting traditional, centralized and monolithic hardware structures. To enable global and cooperative management of hardware resources and efficiency at large scales, it is imperative to analyze and exploit architecture characteristics beyond the scope of individual cores and hardware components. Recently, various predictive models have been proposed to cost-effectively reason processor performance and power characteristics at the design exploration stage. A common weakness of existing analytical models is that they assume centralized and monolithic hardware

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 887–892, 2009.

structures and therefore lack the ability to forecast the complex and heterogeneous behavior of large and distributed architecture substrates across the design space. This limitation will only be exacerbated with the rapidly increasing integration scale. Therefore, there is a pressing need for novel and cost-effective approaches to achieve accurate and informative design trade-off analysis for large and sophisticated architectures in the upcoming multi-/many core eras.

This paper applies grey neural network predictive models and 2D wavelet transform techniques which can efficiently reason the characteristics of large and sophisticated multi-core oriented architectures during the design space exploration stage without using detailed cycle-level simulations. Grey neural network is mainly employed to predict 2D space parameters produced by wavelet decomposition [1,2] for achieving better prediction accuracy with less samples.

2 Combining Wavelets and Grey Neural Network for Architecture 2D Spatial Characteristics Prediction [1,2]

The 2D spatial characteristics yielded on large and distributed architecture substrates can be modeled as a nonlinear function of architecture design parameters. Instead of inferring the spatial behavior via exhaustively obtaining architecture characteristics on each individual node/component, we employ wavelet analysis to approximate it and then use a neural network to forecast the approximated behavior across a large architecture design space. Previous work shows that neural networks can accurately predict the aggregated workload behavior across varied architecture configurations. Nevertheless, monolithic global neural network models lack the ability to informatively reveal complex workload/architecture interactions at a large scale and training of neural network needs a lot of samples. To overcome this disadvantage, we propose grey neural networks to predict 2D wavelet coefficients that incorporate multi-resolution analysis into a set of grey neural networks for spatial characteristics prediction of multi-core oriented architecture substrates. The 2D wavelet transform is a very powerful tool for characterizing spatial behavior since it captures both global trend and local variation of large data sets using a small set of wavelet coefficients. The local characteristics are decomposed into lower scales of wavelet coefficients (high frequencies) which are utilized for detailed analysis and prediction of individual or subsets of cores/components, while the global trend is decomposed into higher scales of wavelet coefficients (low frequencies) that are used for the analysis and prediction of slow trends across many cores or distributed hardware components. Collectively, these wavelet coefficients provide an accurate interpretation of the spatial trend and details of complex workload behavior at a large scale. Our methods use a separate GRBF neural network to predict individual wavelet coefficients. The separate predictions of wavelet coefficients proceed independently. Predicting each wavelet coefficient by a separate GRBF simplifies the training task of each sub-network. The prediction results for the wavelet coefficients can be combined directly by the inverse wavelet transforms to synthesize the spatial patterns on large-scale architecture substrates.

Our proposed techniques consist of the following steps:

Step 1. Architecture 2D characteristics are decomposed into a series of wavelet coefficients using 2D discrete wavelet transform;

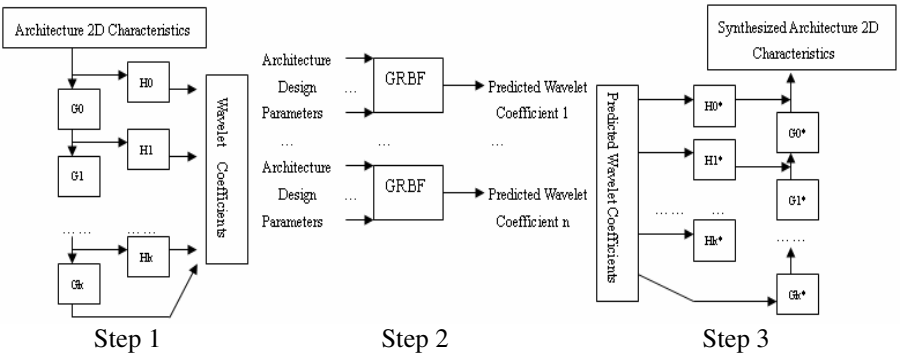


Fig. 1. Using wavelet transform and GRBF for forecasting architecture 2D characteristics

Step 2. Each wavelet coefficient is predicted by a separate GRBF;

Step 3. Architecture 2D characteristics are reconstructed by an inverse 2D wavelet transform on predicted wavelet coefficients.

Figure 1 shows our scheme for architecture 2D spatial characteristics prediction with GRBF neural network and 2D wavelet analysis. Given the observed spatial behavior on training data, our aim is to predict the 2D behavior of large-scale architecture under different design configurations. The hybrid scheme basically involves three stages. In the first stage, the observed spatial behavior is decomposed by wavelet multi-resolution analysis. In the second stage, each wavelet coefficient is predicted by a separate GRBF. In the third stage, the approximated 2D characteristics are recovered from the predicted wavelet coefficients. Each GRBF neural network receives the entire architecture design space vector and predicts a wavelet coefficient. The training of a GRBF network involves determining the center point and a radius for each GRBF, and the weights of each GRBF that determine the wavelet coefficients.

3 Grey Neural Network Predictive Model

Grey network model and neural network both have their own limitations. On one hand, some grey neural networks such as GM (0, N) can only deal with linear model. When the relationship between dependent and independent variables is nonlinear or uncertain, its accuracy of prediction falls unacceptably. On the other hand, neural networks have very strong ability in approaching nonlinear functions and are not easy to fall into local minimum when solving problems. However it also has an evident shortage with its precision and stability of results strongly depend on the number of samples [4]. The combination of these two methods, the Grey RBF (GRBF) Prediction Model[3], takes the advantages from both of them and can be used to efficiently solve the problem of predicting on small volume of sample data. The GRBF-based predictive models, which combine GM(0,N) and RBF, are applied to predict 2D space parameters produced by wavelet analysis.

The three steps used to predict 2D wavelet coefficients are described as follows:

Step 1: The Modification of Original Sample Data

In the tripolar coordinate system from samples, the coordinate values have both positive and negative signs. Thus, they could not be applied to GM (0, N) Model,

which requires nonnegative-ascending samples as its input. Therefore, the original data need to be modified to meet the requirement. To modify the original data, one can first add the absolute value of the minimum item in the sequence to all items in that sequence, i.e.,

$$b = \min\left\{x^{(0)}(i,j)\right\} \qquad i = 1,2,\ldots,N;\ j = 1,2,\ldots,n$$

$$y^{(0)}(i,j) = x^{(0)}(i,j) + |b|$$

After the above process, all data are nonnegative, but not necessarily ascending. Consequently, 1-AGO (once Accumulated Generating Operation) operation should be performed to the new nonnegative sequences, producing the final nonnegative-ascending data sequence. Moreover, when the original prediction values are needed, the Inverse Accumulated Generating Operation (IAGO) and reverse translation on coordinates should be operated to acquire them.

Step 2: The Training of RBF Neural Networks

The RBF takes 1-AGO sequence of correlative factors and featured data as input and output respectively and is trained by ROLS algorithm. Furthermore, on considering the quantitative differences of samples, data normalization should be performed to help the training of network.

Step 3: Data Prediction

When predicting, one could use 1-AGO sequence of featured data of desired data as input to the trained RBF, then 2-IAGO is needed to obtain the final predicted values.

To build a representative design space, one needs to ensure that the sample data sets disperse points throughout the design space but keep the space small enough to keep the cost of building the model low. To achieve this goal, we use a variant of Latin Hypercube Sampling (LHS) [4] as our sampling strategy since it provides better coverage compared to a naive random sampling scheme. Table 1 lists the prediction results, residual values and relative errors of 10 sets of testing data to show predictive results using our GRBF. Table 2 shows predictive results using RBF and GM individually.

Table 1. Prediction results with GRBF

No.	Original data	Prediction Value(GRBF)	Residual Value	Relative error (%)
1	12.2755	12.2906	-0.0151	0.12
2	11.9775	12.4433	-0.4658	3.89
3	12.6961	12.4112	0.2849	2.24
4	12.7196	12.4363	0.2797	2.20
5	12.7461	12.4524	0.2937	2.30
6	12.7186	12.3804	0.3382	2.66
7	12.3029	12.4705	-0.1676	1.36
8	12.7667	12.4305	0.3362	2.63
9	12.0971	12.3905	-0.2995	2.48
10	11.9588	12.4953	-0.5365	4.49

Table 2. Prediction results with RBF and GM(0,N)

No.	Original data	Prediction Value(RBF)	Relative error (%)	Prediction Value(GM)	Relative error (%)
1	12.2755	12.9000	5.07	11.0286	10.16
2	11.9775	11.6399	2.82	13.2144	10.33
3	12.6961	13.8622	9.18	12.3919	2.40
4	12.7196	12.1089	4.80	13.7860	8.38
5	12.7461	13.1151	2.90	12.2849	3.62
6	12.7186	13.0111	2.30	12.9554	1.86
7	12.3029	12.5124	1.70	13.5035	9.76
8	12.7667	12.5424	1.76	13.1576	3.06
9	12.0971	12.3834	2.37	12.4536	2.95
10	11.9588	11.8670	0.77	14.8408	24.10

The GRBF prediction model combined with 2D wavelet analysis has been applied in the 2D space parameters prediction for multi-core architecture successfully and proved to be an easy, convenient and accurate method. Figure 2 shows the performance of GRBF. As can be seen, the predicted values from RBF have the largest relative error since it requires large volume of samples. The predicted values from GM (0, N) are fairly close to the actual values, with average relative error of 7.67%. The predicted values from GRBF have the highest accuracy, with some points almost overlapped the actual values and the average relative error is 3.45%. Therefore, we conclude that the new GRBF-based predictive model works effectively and has the highest prediction accuracy of 96.55%.

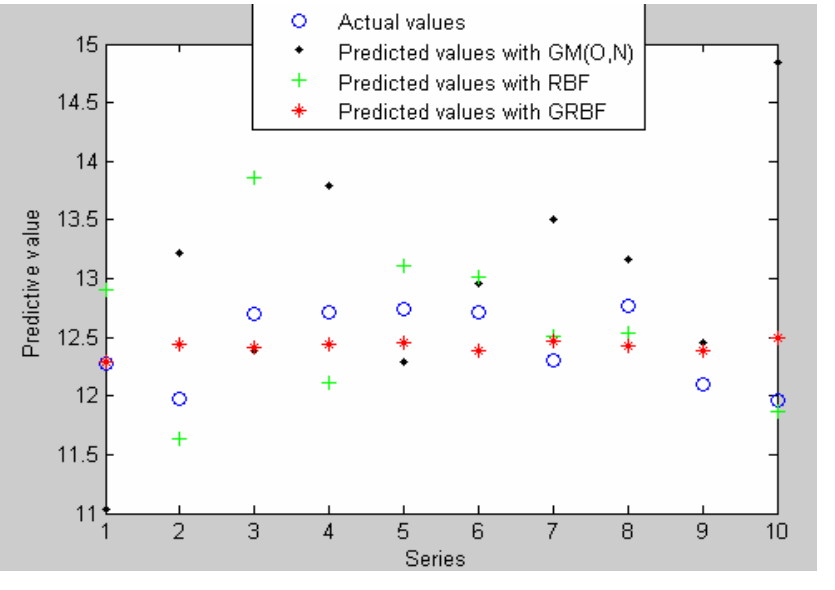


Fig. 2. Prediction performances among GRBF, GM(0, N), RBF

5 Conclusion

Grey neural computation model is a kind of computation model based on integrating grey system and neural network. It makes full use of the existing similarity of these two technologies on information processing and the existing complementarity on model characteristics, and it can make up the disadvantage of only using grey model or neural network to solve problems. Our proposed techniques employ 2D wavelet multiresolution analysis and grey neural network regression modeling. Experimental results show that the models achieve high accuracy while maintaining low complexity and computation overhead.

References

1. Cho, C.B., Zhang, W.Y., Li, T.: Informed Microarchitecture Design Space Exploration using Workload Dynamics. In: International Symposium on Microarchitecture (MICRO) (December 2007)
2. Cho, C.B., Zhang, W.Y., Li, T.: Characterizing the Effect of Microarchitecture Design Parameters on Workload Dynamic Behavior. In: IEEE International Symposium on Workload Characterization (IISWC) (September 2007)
3. Yuan, J.L., Zhong, L., Tong, Q.W.: The Modeling of Metabolic GM(1,1) Prediction Model and Its Application. In: Progress in Intelligence Computation and Application, ISICA 2005, pp. 718–722 (2005)
4. Yuan, J.L., Zhong, L., Jiang, Q.: A Study on Grey RBF Prediction Model. In: Proceedings of IEEE, ICML 2005, pp. 4140–4143 (2005)
5. Yuan, J.L., Zhong, L.: The Dynamic Grey Radial Basis Function Prediction Model and its Applications. In: Proceeding of IEEE ICICIC 2006, pp. 582–585 (2006)
6. Zhong, L., Yuan, J.L., Xia, H.X.: A Study on Gray Neural Network Modeling. In: Proceedings of the First International Conference on Machine Learning and Cybernetics, Beijing, China, pp. 2021–2023 (2002)

Neural Network Algorithm for Installation Error Identification Based on Bearing-only Target Motion Analyses

Lin Wen, Zhong Liu, Ya-song Luo, and Xue-zhi Fu

Electronics Engineering College, Naval University of Engineering,
Wuhan 430033, China

Abstract. It is quite strict with the degree of installation precision of photo-electricity device (PED) carried by unmanned aerial vehicle (UAV) when conducting bearing-only target motion analyses (BTMA), because of the existence of the installation error of PED, the precision of target location is bound to be affected. In order to solve this problem mentioned above, a neural network algorithm for installation error identification based on BTMA is put forward in this paper, consulting the target surveillance technology of PED to identify the installation error thought neural network algorithm, which is worthy to be put into practical applications.

Keywords: Unmanned aerial vehicle, Photo-electricity device, Neural network; Installation error identification.

1 Introduction

With the development of modern weapon technology, the range of fire has been considerably extended. However, the detection range of existing radar is confined within sight distance so the requirement for practical military operations cannot be satisfied. In this circumstance, various kinds of aerial platforms are invented to meet the demand. Aerial platform carries optical devices or video cameras to conduct precise remote reconnaissance at long time, providing real-time change of battlefield for the commander, supporting over-the-horizon attack for the weapon system.

Unmanned aerial vehicle (UAV) is familiar among aerial platform as shown in [1]. At present, there are about 48,000 UAVs on a globe scale. Being under the control of electronic devices, UAV needs no pilot to complete automatic aerial navigation. In the UAV, the space can be fully utilized for the installation of important equipments without concerning about the pilot. Additionally, the problem about injuries and deaths of the pilots can be ignored.

When UAV carries photo-electricity device (PED) to conduct bearing-only target motion analyses (BTMA), it is strict with the degree of installation precision of PED: The base of PED must run parallel with the horizontal surface of UAV; the line from beginning to the end of PED must run parallel with the head-to-tail line of UAV. But in

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 893–900, 2009.

practical installation, the precision cannot be ensured and the target location is bound to be affected.

In order to eliminate the negative effect of installation error and raise the degree of location precision, this paper designs experimentation, establishes optimal model based on BTMA of aerial platform, wields neural network to identify the installation error, make sure that the difference between real error and result is acceptable. The neural network algorithm for installation error identification based on bearing-only target motion analyses as shown in [2] that put forward from this paper is fast converging and fault tolerant to a certain degree judged from final simulation result.

2 Basic Theory of Neural Network Algorithm for Installation Error Identification

The error of installation characterizes itself mainly in two aspects: The base of PED is not parallel to the horizontal surface of UAV and the error can be resolved into horizontal installation error (HIE) Δr and vertical installation error (VIE) $\Delta\theta$; the angle between the line from beginning to the end of PED and head-to-tail line of UAV can be denoted as rolling error $\Delta\eta$ (RE). Granted that the azimuth angle to the target is F_{bj} and the pitching angle to the target is ε_{bj},which are measured by PED; the horizontal rolling angle is r_j,vertical rolling angle is θ_j and the navigating course is c_j which come from the inertia navigation device of UAV.

The essence concept of this algorithm is:

(1) Take a vehicle installed with GPS receiver as a target vehicle to be placed under surveillance by UAV installed with GPS receiver as well;

(2) Integrate the data including longitude, latitude and altitude measured by these two GPS receivers and convert the data into azimuth angle and pitching angle based on geography coordinate system of UAV ;

(3) The data measured by inertial navigation device and PED also have to be converted into azimuth angle and pitching angle based on geography coordinate system of UAV;

(4) By comparing these two data groups of azimuth angle and pitching angle, establish optimal model of neural network to figure out installation error.

The procedure of this algorithm refers to figure 1:

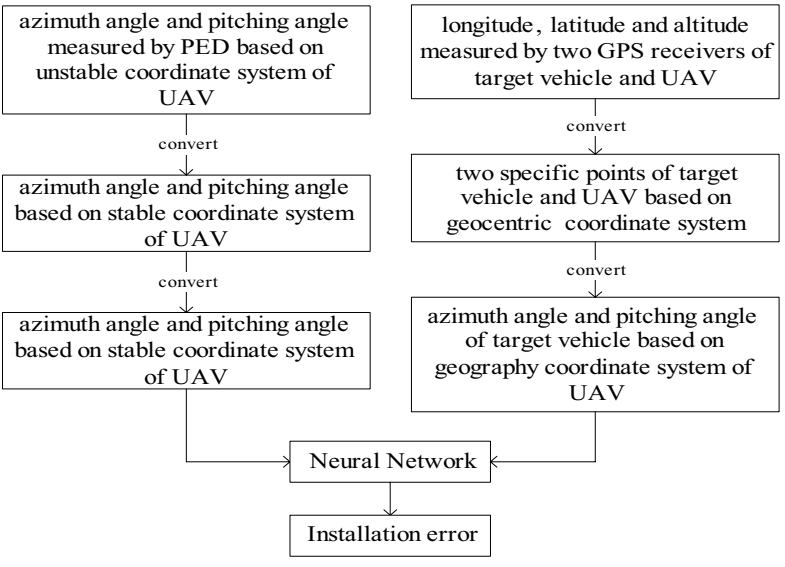


Fig. 1. The procedure of the algorithm。

Firstly, the azimuth angle F_{bj} and the pitching angle ε_{bj} based on unstable coordinate system of UAV should be converted into F_{wj} and ε_{wj} based on stable coordinate system of UAV. The conversion formula:

$$\begin{cases} F_{wj} = \arcsin(u_3^{'}) \\ \varepsilon_{wj} = arctg(\dfrac{u_1^{'}}{u_2^{'}}) \end{cases}$$

where $U^{'} = TU$ and $T = ABC$ and

$$U = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} \cos\varepsilon_{bj} \sin F_{bj} \\ \cos\varepsilon_{bj} \cos F_{bj} \\ \sin\varepsilon_{bj} \end{bmatrix}, A = \begin{bmatrix} \cos(r_j + \Delta r) & 0 & -\sin(r_j + \Delta r) \\ 0 & 1 & 0 \\ \sin(r_j + \Delta r) & 0 & \cos(r_j + \Delta r) \end{bmatrix},$$

$$B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_j + \Delta\theta) & \sin(\theta_j + \Delta\theta) \\ 0 & -\sin(\theta_j + \Delta\theta) & \cos(\theta_j + \Delta\theta) \end{bmatrix}, C \begin{bmatrix} \cos\Delta\eta & \sin\Delta\eta & 0 \\ -\sin\Delta\eta & \cos\Delta\eta & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

Moreover, F_{wj} and ε_{wj} are converted into azimuth angle β_j and pitching angle ε_j based on geography coordinate system of UAV.

$$\begin{cases} \beta_j = F_{wj} + C_j \\ \varepsilon_j = \varepsilon_{wj} \end{cases} \tag{1}$$

Actually, the obtained azimuth angle β_j and the pitching angle ε_j contains the affection of installation error, so they are $\beta_j(\Delta r, \Delta\theta, \Delta\eta)$ and $\varepsilon_j(\Delta r, \Delta\theta, \Delta\eta)$.

The data of longitude, latitude and altitude which measured by these two GPS receivers need to be converted into specific point coordinates based on geocentric coordinate system:

$$\begin{cases} x_e = (N+H)\cos B\cos L \\ y_e = (N+H)\cos B\sin L \\ z_e = (N(1-e^2)+H)\sin B \end{cases} \tag{2}$$

L, B, H is separately longitude, latitude and altitude, e represents the compression of the earth, $e = \sqrt{\dfrac{a^2-b^2}{a^2}}$, a and b is longer and shorter axis of the earth. N represents the curvature of The Prime Vertical Circle, $N = \dfrac{a}{\sqrt{1-e^2\sin^2 B}}$.

Additionally, according to the horizontal rolling angle, vertical rolling angle and the navigating course come from inertial navigation device of UAV, it is possible to work out azimuth angle β_j and pitching angle ε_j based on geography coordinate system of UAV.

Artificial neural network as shown in [3] is artificial intelligence technology simulating the biology process of human brain that has been developing from many years ago. It is turned to be a complex nonlinear system by comprehensively connections of large quantity of simply process units (nerve cell). It has strong nonlinear mapping ability and needs no prior knowledge to conclude inherence logic from

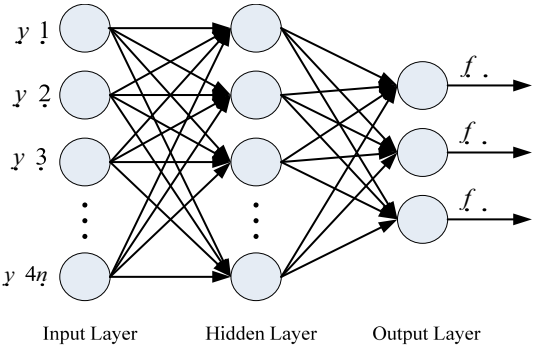


Fig. 2. The structure of the three-layer FNN。

existing data. Academic investigations show that: Three-layer feed-forward neural network (FNN) as shown in [4] with one single hidden layer has the ability of mapping arbitrary functions including nonlinear functions. Based on practical applications, this paper establishes three-layer FNN as figure 2:

This FNN is composed of input layer, hidden layer and output layer. There is no connection between the nerve cells at the same layer; but for different layers, nerve cells are completely connected, existing weight w at every connected route. The input of the input layer is $4n$ dimensions vector, it contains $\beta_j, \varepsilon_j, \beta_j^{'}, \varepsilon_j^{'}, j = 1, 2..n$, the output layer has three output nerve cells, they represent HIV Δr VIE $\Delta\theta$ and RE $\Delta\eta$.

3 Realization and Simulation of Identification Algorithm

After the structure is confirmed, the FNN needs to be trained via input and output samples. Adjust the threshold quantity and weight to make the network to possess the ability of mapping for given input and output data.

The process of adjustment has two phases:

(1) Input known samples, calculate every nerve cell's output from the first to last layer of the network by using the threshold quantity and weight which were set by previous iteration;

(2) Adjust threshold quantity and weight according to the degree of influence calculated from the last to the first layer of the network to the overall error.

Repeat these two processes above until calculation converge. Based on the principle of steepest descent back-propagations (SDBP) set k to be the time of iteration, modify threshold quantity and weight according to the formula as below:

$$x(k+1) = x(k) - \alpha g(k) \tag{3}$$

$E(k)$ is an overall error of the k^{th} iteration, in the MATLAB tool box of neural network, its default value as shown in [5] is mean square error (MSE).

$x(k)$ represents threshold quantity or weight of the k^{th} iteration.

$g(k) = \dfrac{\partial E(k)}{\partial x(k)}$ represents the grads vector of the overall error of neural network to threshold quantity or weight of the k^{th} iteration. The minus sign expresses the direction opposite to the grads, the steepest descent direction.

α is rate of adjustment, usually constant 0.01 in the MATLAB tool box of neural network.

SDBP modifies weight along the most precipitous direction, it quickly minimizes the overall error, but it is not always the first to converge. This paper deals the problem

referred above with a dual-phase method. At earlier stage of adjustment, SDBP is adopted to make overall error quickly descend into a given range, and then at later stage conjugate gradient back-propagation (CGBP) FR algorithm is used to make the precision to fit the demand.

FR algorithm is put forward by R.Fletcher and C.M.Reeves. The searching direction of first iteration of this algorithm is the steepest descent direction:

$$p(0) = -g(0) \tag{4}$$

Then, the optimal searching direction is decided by following formula:

$$\begin{cases} x(k+1) = x(k) + \alpha p(k) \\ p(k) = -g(k) + \beta(k) p(k-1) \\ \beta(k) = \dfrac{g^T(k) g(k)}{g^T(k-1) g(k-1)} \end{cases} \tag{5}$$

$p(k)$ is the searching direction of the $k+1$ th iteration.

Concluded from academic and practical experiments, the result figured out from BTMA algorithm is acceptable when installation error is less than 0.0333 degree. The input of the FNN is $4n$ dimensions vector, it contains $\beta_j, \varepsilon_j, \beta_j^{'}, \varepsilon_j^{'}, j = 1, 2..n$, the output vector has three components, they are HIV Δr VIE $\Delta\theta$ and RE $\Delta\eta$. The simulation made in this paper firstly sets precision degree to be 0.1 and utilizes SDBP to adjust threshold quantity and weight of FNN, when the result figured out from FNN reaches into the given range, modifies the precision degree to be 0.01 and utilizes CGBP strategy to adjust, finally complete the generation of FNN though the two phases of adjustment.

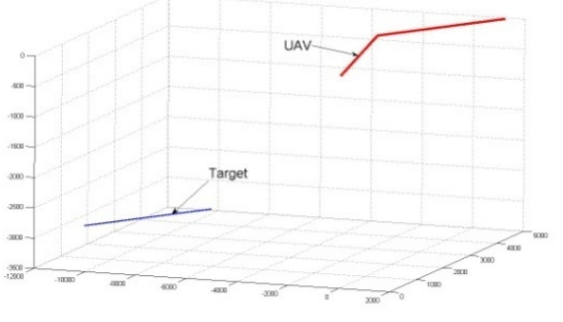


Fig. 3. The situation of UAV and target vehicle

According to experimental data, the initial position including longitude, latitude and altitude of UAV is set to be [121.9111, 29.7546, 3000], speed is 20m/s, navigating course is 330 degrees and 240 degrees after 3 minutes flying. The initial position of target vehicle is [121.7947, 29.7650, 0], speed is 10m/s, navigating course is 60 degrees. The sampling interval is 1 second, simulation lasts 6 minutes. Their situation is illustrated as figure 3.

Before every simulation, the installation error is already set for the generation of the 4n dimension vector for the input layer of the FNN to figure out the identification value. Add no noise to the azimuth angle F_{bj} and the pitching angle ε_{bj} that measured by UAV, the result is showed as Table 1. Then add white Gauss noise to F_{bj} and ε_{bj}, the result is showed as following tables.

From the simulation result above, the algorithm can identify the error at relatively high level of precision with no noise added in, the maximum discrepancy between real value and identification value is 0.001 degree, minimum is 0 degree; when added white

Table 1. Simulation result: Adding no noise to F_{bj} and ε_{bj}

Real Value			Identification Value		
HIE	VIE	RE	HIE	VIE	RE
-1	-1	1	-1.0001	-1.0000	1.0000
5	3	1	4.9999	3.0005	0.9996
15	15	15	15.0002	14.9998	15.0006
-5	-10	15	-4.9990	-10.0004	15.0002

Table 2. Simulation result: Adding MSE 0.1° white Gauss noise to F_{bj} and ε_{bj}

Real Value			Identification Value		
HIE	VIE	RE	HIE	VIE	RE
-1	-1	1	-0.9749	-1.0055	0.9833
5	3	1	5.0052	2.9980	0.9955
15	15	15	15.0113	15.0076	14.9981
-5	-10	15	-4.9987	-10.0199	14.9947

Table 3. Simulation result: Adding MSE 0.2° white Gauss noise to F_{bj} and ε_{bj}

Real Value			Identification Value		
HIE	VIE	RE	HIE	VIE	RE
-1	-1	1	-0.9762	-0.9870	1.0036
5	3	1	5.0022	3.0304	1.0138
15	15	15	14.9971	15.0112	15.0298
-5	-10	15	-4.9759	-9.9947	14.9809

Gauss noise of which MSE is 0.1 degree, maximum is 0.0251 degree and minimum is 0.0013 degree; when added white Gauss noise of which MSE is 0.2 degree, maximum is 0.0304 degree and minimum is 0.0022 degree.

4 Conclusion

The neural network algorithm for installation error identification based on bearing-only target motion analyses that put forward from this paper, consulting the target surveillance technology of PED to identify the installation error thought neural network. It satisfies the strict demand for PED installation and can be easily carried out, providing precision support for passively tracing and measuring system. Neural network algorithm is suit for high extent of precision and fast calculation as for this problem and fault tolerant to a certain degree judged from final simulation result.

References

1. Liu, K., Yan, M., Yuan, F.: Maneuver Detecting Algorithm Used for Bearings-only Maneuver Target Tracking. Command Control And Simulation 28, 30–34 (2006)
2. Dong, Z.: Nonlinear Least-Square Algorithms Used for TMA in Bearing-only System-The Engineering Mathematic Model and Algorithms. Information Command Control System And Simulation Technology 27, 4–7 (2005)
3. Jin, X.: Study on New Algorithm for Feed-forward Neural Network and its Simulation. Journal of Harbin University of Commerce Natural Sciences Edition 20, 24–27 (2004)
4. Li, H., Wan, B.: An Efficient Learning Algorithm For Large-Scale Feedforward Neural Networks And Its Application. Information and Control 32, 403–406 (2003)
5. Cao, Q., Zhou, J.: The Application of MATLAB in Design of Neural Networks. Journal of East China Jiaotong University 21, 86–88 (2004)

Nonlinear Time Series Prediction by Using RBF Network

Liqiang Zhu

School of Mechanical, Electronic and Control Engineering,
Beijing Jiaotong University, Beijing 100044, China
lqzhu@bjtu.edu.cn

Abstract. This paper describes a numerical algorithm for short-term prediction of nonlinear time series by using time-delay embedding and radial basis function (RBF) neural networks. Unlike the existing RBF algorithms with centers preselected during training process and fixed during prediction process, the proposed method utilizes a simple selection algorithm to dynamically change the center positions, resulting in a local RBF model with time varying parameters. Analysis and methodology are detailed in the context of the Leuven competition. Results show that the proposed local dynamical RBF network performed remarkably well.

Keywords: Time series predication, RBF, Time-delay embedding.

1 Introduction

The desire to understand the past and predict the future is throughout scientific research. The prediction of a nonlinear time series is a challenging problem. Although the local instability of nonlinear dynamical systems prohibits accurate long-term predictions, short-term predictions can be made due to the inherent determinism.

There exist successful methods of linear time series prediction, such as ARMA models. These conventional methods are generally ineffective for the prediction of nonlinear time series. Over recent years, several nonlinear time series models [1], [2], [3], [4], [5] have been proposed, where two techniques are utilized widely. One is the time-delay coordinates embedding [6],[7], which can be used to reconstruct the state space of a dynamical system. The other one is the machine learning, typified by artificial neural networks (ANNs), which can adaptively explore a large space of potential models. Previous studies based on these two developments have shown promising results. In general, the existing methods consist of two steps: (1) resolves all the model parameters from a training data set; (2) with model parameters fixed, predict the next state from a given state.

If multi-step predictions are needed, then the outputs on the previous step is fed back to the model, in which the model acts as an autonomous system, emulating the dynamic behavior of the system that generated the nonlinear time series. According to the complexity of the underlying system, many ANN models

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 901–908, 2009.

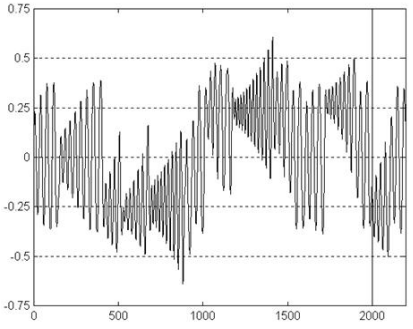


Fig. 1. The Leuven competition data set. The first 2000 points were given, 2001-2200 points were to be predicted.

can be selected, such as feed forward multilayer perceptrons (MLPs), recurrent neural networks (RNNs), radial basis function (RBF) neural networks, and their variations. No matter what structure is selected, one difficulty of applying ANN is to train the network. Since ANN can explore a large space of potential models and there exists abundant local minima in the cost function of optimization criterion due to the nonlinearity of the underlying system, it is possible for the training process to end at a suboptimal solution. In this situation, some randomness must be added to the training process by using, e.g., simulated annealing, genetic algorithm, or simply different initial conditions of ANNs at least. Unlike the back propagation-type algorithms in MLPs and RNNs, if the centers of RBFs are predetermined, the training of a RBF network has a closed-form optimal solution. So when the size of the training data set is relatively small, the RBF network can be easily constructed by using each point in the training data set as an RBF center. However, if the data size is not small, this simply method will lead to an impractically large network. Thus, the selection algorithms of RBF centers have attracted much attention in the literature [8], e.g. k-means clustering algorithm, the stochastic gradient approach, orthogonal least squares, genetic algorithm, etc.. Again, due to the nonlinearity, a global optimal solution can not be always guaranteed.

This paper proposes a prediction method based on time delay embedding and RBF networks with dynamically selected centers. The method first reconstruct the time series in a space large enough to unfold the dynamical attractor by using time-delay coordinate embedding. Then, within this space, a local RBF network model is built for one-step prediction. A small set of the training data points near the input state are selected as RBF centers. For the prediction of the next step, a new local RBF network is to be built with new centers near the new input state or the output of the RBF network of the previous step. Because the size of the center set in each step can be very small, the construction of the RBF network is very fast. The proposed method is especially efficient for cases when the training data set is small or the underlying system is time varying, in which only recent data samples are important. Analysis and methodology of the method are detailed in the context of the K.U. Leuven competition. The

competition data set consisted of 2000 points generated by a chaotic system. The task is to predict the next 200 points. The data set is shown in Fig. 1.

2 The Time-delay Embedding

When we observe an unknown dynamical system, if we can measure all the state variables of the system, then prediction can be made directly from previous measurements by using any interpolation model, as shown in Fig. 2.a. In practice, however, usually we can measure only a few of the state variables of the system. As shown in Fig. 2.b, the trajectory will intersect with itself, as a result, accurate prediction can't be made directly with interpolation methods. In the most limited case, we might be in the position of having available only the measurement of a single state variable as a function of time $y(t)$. Since the measurement depends only on the system state, we can represent such a situation by $y(t) = f(\mathbf{x}(t))$, where f is the single measurement function, evaluated when the system is in state $\mathbf{x}(t)$. The technique of time-delay embedding is used to reproduce the compact finite-dimensional set of dynamical states of the system using vectors derived from $f(t)$. Let A denote this set. We assign to $\mathbf{x}(t)$ the delay coordinate vector

$$\mathbf{b}(t) = \mathbf{F}(\mathbf{x}(t)) = [y(t-\tau), ..., y(t-m\tau)] \tag{1}$$

Takens [6] and Sauer et al [1] have shown that $\mathbf{F}$ is invertible, or has a one-to-one property, if the embedding dimension m is greater than twice the box-counting dimension of A, and the delay time τ is chosen properly.

The one-to-one correspondence is useful because the state of a deterministic dynamical system, and thus its future evolution, is completely specified by the corresponding time-delay vector. Thus, we can treat the time-delay space as the true state space of the system and make prediction by using interpolation methods directly.

Time delay embeddings are widely used as the input vector to dynamic models, both linear and nonlinear. The theorem provides a sound theoretical basis

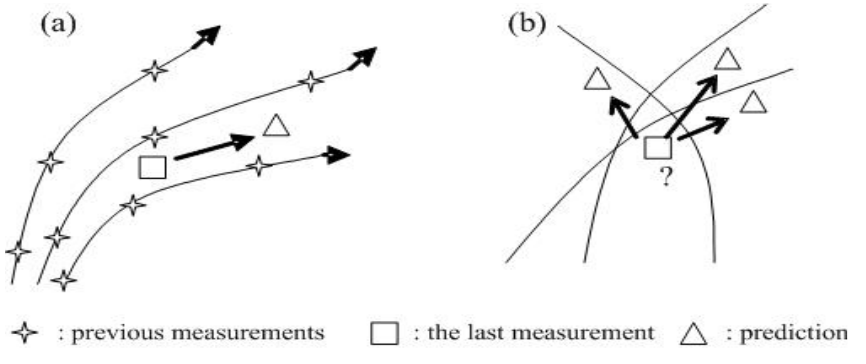


Fig. 2. (a) Trajectory in the full-dimensional state space, (b) Trajectory projected in a lower-dimensional space

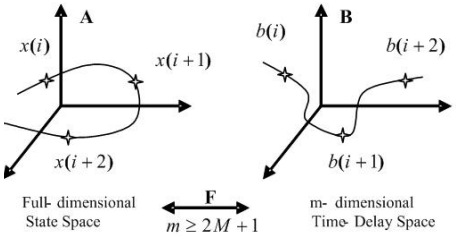


Fig. 3. Time-delay embedding. M is the box-counting dimension of the attractor A, m is the embedding dimension.

for this approach and has been applied successfully in many applications. The selection of m and τ may critically affect how accurately the embedding reconstructs the state of the system. Many researchers have proposed methods to find m and τ [9], [10]. There are free software's available now. For the date set of the competition, $m \geq 9$ and $\tau \approx 2$ are found.

3 RBF Network Models

As mentioned in the last section, in order to make a good prediction from the state space reconstruction, interpolation in this multidimensional time-delay space is needed. Radial-basis function (RBF) network is a good model for the multivariate interpolation problem. A RBF network can be expressed as simple as followings:

$$\mathbf{F}(\mathbf{x}_j) = w_0 + \sum_{i=1}^{k} w_i \varphi(||\mathbf{x}_j - \mathbf{b}_i||) \tag{2}$$

where w are the weights, φ is the basis function, typically bell-shaped with width d, $\mathbf{b}$ is the centers of RBFs, and $||\cdot||$ represents the distance function, typically Euclidean distance or any other meaningful function. For strict interpolation, given $\mathbf{b}$ and φ with d, then $w = \Phi^{-1}\mathbf{x}$, where $\Phi = \{\varphi_{ji}|(j,i) = 1,2,...,N\}$, N is the number of input points $\mathbf{X}$. When input points are noisy, some suitable regularization techniques are necessary to ensure good generalization [8].

In general, we can use RBF network as a predictor in a global form or a local form.

3.1 Global RBF Network Modeling

The idea of global modeling is straightforward: first choose $\mathbf{b}$ and φ with d, and solve w, then we model the true function of the whole attractor (the set of states). When input data set is not huge, like the competition data set, we can simply choose all the delay vectors as the centers of RBFs. When input data set is huge, we have to invert a huge matrix Φ when solving w and compute a huge network when making every prediction, which is impractical. Then we need choose a small number of centers carefully. For d, usually we use cross-validation

to select one value for all centers. During training, the 2000-points data set is divided into 1800-points training set and 200-points validation set, which are used to select an optimal value for the width of centers. And then construct the RBF network using the whole 2000 points to make one step of prediction. Use the predicted vector as input to make next step prediction, and so on.

Fig. 4 shows the result using regression RBF network in MATLAB Neural Network Toolbox with the method mentioned above. As shown in Fig. 4, the prediction can follow the true orbit roughly for the first 100 points. The possible cause of the large error after 100 points may be due to the fact that the competition date set is really small. Especially, there are not enough transitional points. In Fig. 1, we can find there are approximate three scrolls in the time series, around -0.25, 0 and 0.25 respectively. There is only one transition from -0.25 to 0.25 scroll in the training set. Unfortunately, from about the 80th point in prediction region, there is a transition from -0.25 to 0.25 scroll. So the densities around the centers (time-delay vectors formed from time series) are highly different in the state space. If we use the same width for all the centers, the big error must occur. There are two methods to solve this problem, one is to carefully select the locations of the centers and make them with the same density. Due to the small size of the date set, doing this will lose important details of input data, which is not good for prediction. Another method is to use different width for different center. A supervised learning algorithm in MATLAB is used to learn the widths, however, no obvious improvement can be observed in the simulation (not shown in this paper). In fact there is a straightforward way to determine the width. We can assign width proportional to the density around every center. When widths are chosen properly, for any input, only several centers close to it may be "fired". That is indeed a local model, but in a global form! And as we will discuss in the next section, using an explicit local form can give us more flexibility and accuracy.

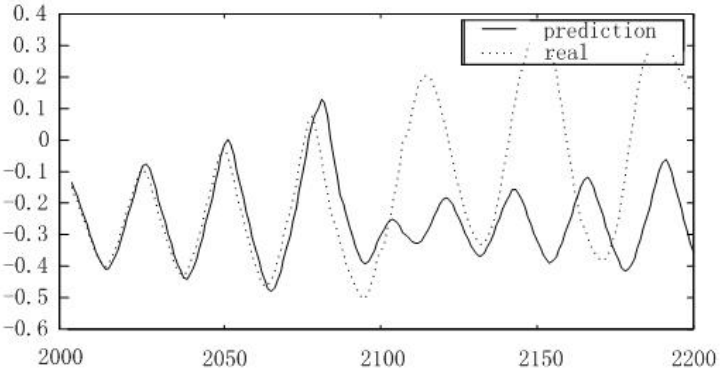


Fig. 4. The prediction produced by global RBF network model. A regression RBF network in MATLAB Neural Toolbox is used. The parameters are $(m, \tau, d) = (9, 2, 0.03)$, $MSE = 0.0683$.

3.2 Local RBF Network Modeling

Local models make predictions by finding local neighbors that are close to the end point of the time series in state space. The prediction is an estimate of the average change that occurred immediately after these neighbors. Like any numerical simulation, in which the smaller the step size is, the more accurate result can be achieved, local model can make better prediction using time series with a higher sampling rate. Although up-sampling the time series (or interpolating the time series) can't create new information, it does help to improve performance. For a global model, this method is impractical because of the significantly increased computational cost. Generally, the effective sampling rate is 5-10 times faster than the Nyquest rate [3].

The proposed prediction method using local RBF network model is as follows:

1. Up-sampling the time series.
2. Construct time-delay vectors from the time series with embedding dimension m and delay time τ.
3. Find the k-nearest neighbors of the last time-delay vector $\mathbf{x}$ using any efficient neighbor-searching algorithm. Use the minimum distance between $\mathbf{x}$ and neighbors as the width of the centers.
4. Construct a regression RBF network using these k-nearest neighbors as centers and input, and their corresponding immediate time-delay vectors as output.
5. Make a prediction using $\mathbf{x}$ as input.
6. Use the previous prediction as the input for next step, and go back to 3.

To use local models for time series prediction, there are several decisions that one must make, such as how many neighbors should be found, how the width of every center should be calculated. Often a coarse optimization is performed by measuring the model accuracy for several values of the model parameters. The selection of the extent of this optimization is limited by designers' experience and computational resources and, as a result, model accuracy is sacrificed. Using a RBF network as a local model, unlike the models used before, i.e. local averaging model [3], the accuracy of the model is much less sensitive to these parameters. This property is gained by virtue of the generalization capacity of RBF network using regularization technique. The network can generate prediction with similar accuracy over a large range of model parameters.

As Fig. 5 shows, the local model can make more accurate prediction than the global model, but still makes big error after 90 points. The possible cause of the large error may be still due to the fact that the competition date set is really small. Especially, there are not enough transitional points in the training data set. As we mentioned in previous section, from about the 80th point in prediction region, there is a transition from -0.25 to 0.25 scroll. Because of the lack of close neighbors, in this region accurate prediction is really hard to achieve. If there were more sampling points in the data set, this model should give better prediction.

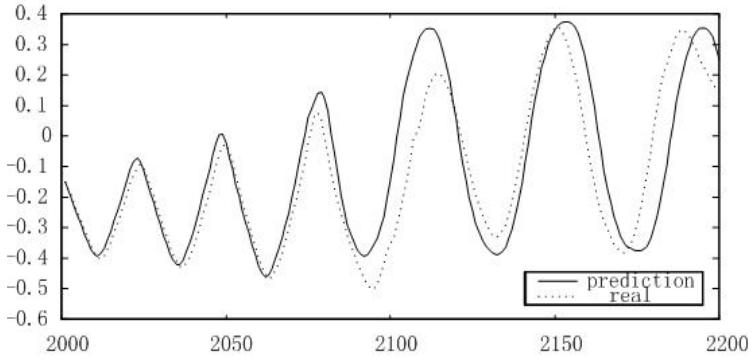


Fig. 5. The prediction produced by local RBF network. A regression RBF network in MATLAB Neural Toolbox is used. The parameters are $(m, \tau, k) = (9, 2, 8)$, $MSE = 0.0166$.

4 Conclusions

This paper introduced a new method for nonlinear time series prediction based on time-delay embedding and RBF network. Global and local regression RBF network models are presented. Results based on Leuven competition data set show that local RBF model with dynamically changed centers outperform traditional global models and at the same time, is more efficient to include any new observations. This property is very useful in situations where available data samples are limited. For example, during the construction and the use of underground tunnel, it is desired to predict the tunnel deformation using daily observed deformation data. It is well known that earth mass is a highly complex nonlinear dynamical system which is affected by many factors, such as rock property, underground water, earthquake and even human behaviors. Thus it is difficult to build a theoretical model in advance to predict deformation accurately. Especially, during the beginning period of the life time of a tunnel, its deformation data are limited and any new data have to be taken into account immediately. The proposed method is well suitable for this type of situations.

Acknowledgments. The work was supported by the Science & Technology Program of Beijing Municipality under the grant No. D07050601770705.

References

1. Sauer, T.: Time Series Prediction by Using Delay Coordinate Embedding. In: Proceedings of the NATO Advanced Research Workshop on Comparative Time Series Analysis, Santa Fe, New Mexico (1992)
2. Wan, E.: Time Series Prediction by Using a Connectionist Network with Internal Delay Lines. In: Proceedings of the NATO Advanced Research Workshop on Comparative Time Series Analysis, Santa Fe, New Mexico (1992)

3. McNames, J.: A Nearest Trajectory Strategy for Time Series Prediction. In: Proceedings of the International Workshop on Advanced Black-Box Techniques for Nonlinear Modeling, pp. 112–128. Katholieke Universiteit Leuven, Belgium (1998)
4. Song, A., Lu, J.: Evolving Gaussian RBF Network for Nonlinear Time Series Modeling and Rrediction. Electronics Letters 34(12), 1241–1243 (1998)
5. Medeiros, M.C., Veiga, A.: A Hybrid Linear-Neural Model for Time Series Forecasting. IEEE Trans. Neural Networks 11, 1402–1412 (2000)
6. Takens, F.: Detecting Strange Attractors in Turbulence, Dynamical Systems and Turbulence. Lecture Notes in Mathematics, vol. 898, pp. 336–381 (1981)
7. May, P., Ehrlich, H.C., Steinke, T.: ZIB Structure Prediction Pipeline: Composing a Sauer, Yorke, and Casdagli, "Embedology". Journal of Statistical Physics 65(3), 579–616 (1991)
8. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice-Hall, Englewood Cliffs (1999)
9. Buzug, P.: Comparison of Algorithms Calculating Embedding Parameters for Delay Time Coordinates. Physica D 58, 127–137 (1992)
10. Kantz, S.: Nonlinear Time Series Analysis. Cambridge University Press, Cambridge (1997)

Traveling Wave Solutions in a One-Dimension Theta-Neuron Model

Guoguang Wen[1], Yongguang Yu[1], Zhaoxia Peng[2], and Wei Hu[1]

[1] Department of Mathematics, Beijing Jiaotong University, Beijing 100044, P.R. China
[2] Department of Mathematics, Beijing University of Technology, Beijing 100124, P.R. China

Abstract. This paper mainly investigates traveling wave solutions in a one dimension theta-neuron model. We derive an analytical lower bound of synaptic coupling strength for traveling waves to exist. Using the numerical simulation methods, we verify some related results on the existence of traveling waves and its dependence on parameters, and give the solutions of traveling waves numerically. Furthermore, the change of the solutions curve of traveling wave is investigated corresponding to the variance of each parameter. Finally, there is an interesting phenomenon that the curve of the solution jumps with the increase of each parameter.

Keywords: Traveling waves, Theta-neuron, Synaptic coupling, One-dimension.

1 Introduction

Recently, there has been a great deal of interests in the propagation of waves in neural networks [1-7]. A variety of models have been developed to describe neuronal dynamics in detail. These models range from continuum firing rate models [12] to simplified spiking models [5, 7, 8, 9, 10, 11]. Some models which are derived from biophysical principles provide the most complete and accurate description of neuronal behavior. However, their complexity may preclude mathematical analysis. The theta model, which is more mathematically tractable, arises in certain limits from a class of biophysically based neural models for cells near an activity threshold [5, 6, 9].

R.Osan et al. [9] use continuous dependence of solutions of differential equation on the parameter for a one-dimensional network of theta neurons, and discuss the existence of traveling waves theoretically, but they only obtains the multiple-spike traveling waves, that is, the cell will continue to fire spikes after firing the first spike and generating a traveling wave. R.Osan et al. also mention the possibility of single-spike traveling waves, but it needs to modify the interactional mechanism among cells reasonably. Based on the above-mentioned results, Wu [5] gives an appropriate actional mechanism which improves the model for the one-dimensional network of theta neurons, and obtains the existence condition of single-spike traveling waves theoretically. However, it's a pity that he didn't give the lower bound of synaptic coupling strength. In this paper, on the basis of the work of reference[5] and considering with the characteristic of the theta neurons network model, an analytical lower bound of synaptic coupling strength for

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 909–918, 2009.

existence of traveling waves is derived. Besides, some related results on the existence of traveling waves and its dependence on parameters are verified using the numerical simulation methods, and the solution of traveling waves is presented numerically. Furthermore, the change of the solution curve of traveling wave is investigated with respect to the variance of each parameter. Finally, there is an interesting phenomenon that the curve of the solution jumps with the increase of each parameter.

The structure of this paper is organized as follows: in section 2, we describe the theta model in more detail and introduce additional simplifications. In sections 3 and 4, we present some important results on the existence of solution of single spike traveling waves and parameter-dependence of solutions on the velocity for fixed synaptic coupling strength. In section 5, we mainly derive a lower bound of the synaptic coupling strength for traveling waves to exist. In section 6, some results of simulations are presented. The paper ends with concluding remarks in section 7.

2 Theta Model

For a single cell, the theta model takes the form [5, 6, 9]:

$$\frac{d\theta}{dt} = 1 - \cos\theta + (1 + \cos\theta)(\beta + I(t)), \tag{1}$$

where θ is a phase variable, $\beta \in (-1, 0)$ is a bias parameter which controls the excitability of the cell, and $I(t)$ denotes the time-dependent inputs to the neuron. When θ increases to achieve the value $(2l + 1)\pi, l \in Z$, it is said that the theta neuron fires a spike.

As the input $I(t)$ is fixed, if $\beta + I < 0$, there are two critical points of the system (1), given by $\theta_{rest} = -\cos^{-1}((1 + \beta)/(1 - \beta))$ and $\theta_T = \cos^{-1}((1 + \beta)/(1 - \beta))$. The former is stable, and the latter is unstable. If $\beta + I = 0$, these coalesces in a saddle-node bifurcation on a limit cycle. If $\beta + I > 0$, the neuron fires spike with period $\pi/\sqrt{\beta + \mathrm{I}}$ [5, 6, 9]. The phase circle for this neuron is shown in Fig. 1.

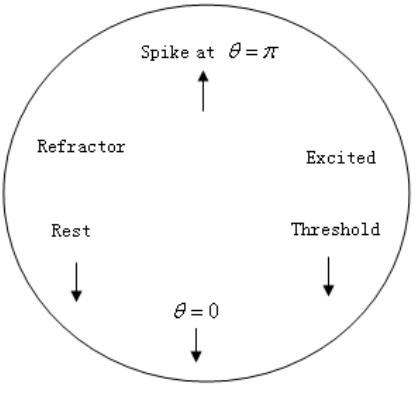


Fig. 1. The phase circle for the theta model (1)

In this paper, we consider a network of synaptically coupling neurons on a continuous special domain Ω, i.e. the time-dependent inputs $I(t)$ takes the form $I(t) = g_{syn}\int_{\Omega} J(x-y)s(y,t)dy$, where g_{syn} denotes the maximal synaptic coupling strength or a parameter measuring the overall coupling strength. Here, $g_{syn} > 0$ means that the connections among neurons are excitatory, while $g_{syn} < 0$ means that the connections are inhibitory. We mainly consider the situation $g_{syn} > 0$. $J(x-y) > 0$ describes the relative synaptic coupling strength from the neuron at y to the neuron at x. $s(y,t)$ measures the synaptic transmission from the neuron located at y and it satisfies an ordinary differential equation depending on time. Then a model of synaptically connected neurons on a continuous spacial domain Ω takes the form:

$$\begin{aligned}\tfrac{\partial\theta(x,t)}{\partial t} &= 1-\cos\theta(x,t)+(1+\cos\theta(x,t))(\beta \\ &\quad + g_{syn}\textstyle\int_{\Omega} J(x-y)s(y,t)dy).\end{aligned} \tag{2}$$

Employing the same traveling wave formulation used in [5, 9], we want to find traveling wave solutions of the above equation, that is $\theta(x,t) = \theta(ct-x)$. Denoting $ct - x \equiv \xi$, then $\theta(x,t) = \theta(\xi)$ and $s(x,t) = s(\xi)$. In [5,9] $s(\xi)$ is simply taken as a function $\alpha\left(t - {}^{x}\!/_{c}\right) = \alpha\left({}^{\xi}\!/_{c}\right)$. Assume that each cell fires spike at $\xi = 0$, the corresponding traveling waves equation takes the form

$$\begin{aligned}c\tfrac{d\theta}{d\xi} &= (1-\cos\theta)+(1+\cos\theta)\,[\beta \\ &\quad + g_{syn}\textstyle\int_{0}^{\infty} J(\xi'-\xi)\alpha\left({}^{\xi'}\!/_{C}\right)d\xi', \xi\in(-\infty,\infty)],\end{aligned} \tag{3}$$

then the ordinary differential equation on the phase θ is achieved.

According to the characteristics of single spike traveling waves, they should satisfy the following boundary conditions [5]:

$$\begin{aligned}&(i)\ \theta\to\theta_{rest}, \quad if\ \xi\to-\infty,\\ &(ii)\ \theta(0)=\pi,\\ &(iii)\ \theta\to\theta_{rest}+2\pi, \quad if\ \xi\to\infty.\end{aligned}$$

Also, Ref.[5] proposed two assumptions for the fire spiking mechanism of the neural cell:

1. The effect among cells is single direction in one-dimensional network of theta neurons.

2. The effect among cells only exist in finite time.

Furthermore, assume $J(x)$ and $\alpha(t)$ have the following form respectively:

$$J(x)=\begin{cases}e^{-x}, x\geq 0\\ 0, \quad x<0\end{cases}, \alpha(t)=\begin{cases}1, 0\leq t\leq\tau\\ 0, t<0\ or\ t>\tau\end{cases},$$

where $\tau > 0$ is constant.

Let $f(\theta) = (1-\cos\theta)+(1+\cos\theta)\beta, g(\theta) = 1+\cos\theta$. Substituting the above formulations into equation (3), obtain a new form of the model of synaptic coupling neural network

$$c\frac{d\theta}{d\xi}=\begin{cases}f(\theta)+g_{syn}g(\theta)\left(1-e^{-c\tau}\right)e^{\xi}, \xi\in(-\infty,0],\\ f(\theta)+g_{syn}g(\theta)\left(1-e^{\xi-c\tau}\right), \quad \xi\in[0,c\tau],\\ f(\theta), \qquad\qquad\qquad\qquad \xi\in[c\tau,+\infty).\end{cases} \tag{4}$$

3 Existence of Single Spike Traveling Wave Solutions

In order to study the existence of single spike traveling wave solutions which satisfy the boundary conditions (i), (ii), and (iii) on interval $(-\infty, +\infty)$, we make a study of equation (4) on intervals $(-\infty, 0]$, $[0, c\tau]$, $[c\tau, +\infty)$, respectively.

First, in order to guarantee the existence of traveling waves which satisfy the boundary condition (i), (ii) on interval $(-\infty, 0]$, the synaptic coupling strength and the wave velocity must vary in certain domain respectively.

Thus, for an arbitrarily fixed value of τ, we consider system (4) on the interval $(-\infty, 0]$, i.e., the following ordinary differential equation:

$$c\frac{d\theta}{d\xi} = f(\theta) + g_{syn}g(\theta)\left(1 - e^{-c\tau}\right)e^{\xi}. \tag{5}$$

To simplify the equation, we compactify from the infinite domain $-\infty < \xi < 0$ to the compact one $0 \leq \eta \leq 1$ using the transformation $\eta = e^{\xi}$, the above equation becomes

$$\begin{cases} c\frac{d\theta}{d\xi} = f(\theta) + g_{syn}g(\theta)\left(1 - e^{-c\tau}\right)\eta \\ \frac{d\eta}{d\xi} = \eta \end{cases}, \tag{6}$$

it is equivalent to the following equation

$$c\frac{d\theta}{d\eta} = \frac{1}{\eta}f(\theta) + g_{syn}g(\theta)\left(1 - e^{-c\tau}\right). \tag{7}$$

In the following, there are some important results on the existence of traveling waves:

Theorem 1.[5] The system (1) exists two traveling wave solutions with different velocity for sufficiently large g_{syn}, if

$$J(x) = \begin{cases} e^{-x}, x \geq 0 \\ 0, \quad x < 0 \end{cases}, \alpha(t) = \begin{cases} 1, 0 \leq t \leq \tau \\ 0, t < 0 \; or \; t > \tau \end{cases},$$

where $\tau > 0$ is constant.

Further, we request the existed traveling waves are single spike waves on interval $[0, c\tau]$. Based on the feature of system (4), we control the net current input of the cell by limit its interactional time τ, to guarantee that it will not fire spikes again within the time limitation.

Theorem 2.[5] If the interactional time τ among cells satisfies:

$$0 < \tau < \frac{\pi + \theta_{rest}}{2\left(1 + g_0\right)},$$

where $\theta_{rest} = -\cos^{-1}\frac{1+\beta}{1-\beta}, \beta \in (-1, 0), \beta \; is \; constant.$ then, system (1) exists two single spike waves on the interval $(-\infty, 0]$, a fast wave and a slow wave.

Finally, we can kown that cells are no longer affected by coupling interaction on interval $[c\tau, +\infty)$ via system (4). Under this condition, cells will naturally degenerate and finally stop at the rest state $\theta_{rest} + 2\pi$ after a longer time.

Therefore, we obtain the single spike waves which satisfy the boundary conditions.

4 Parameter-Dependence of Solutions on the Velocity c for Fixed g_{syn}

Consider equation (7) of $\theta \in [0, 2\pi)$. For all $\xi \in (-\infty, 0]$ we have $\eta = e^{\xi} \in [0, 1]$. Here we call the solution of $\frac{d\theta}{d\eta} = 0$ θ-nullcline and denote its knee-point by $(0, \eta_a(c))$, so we have

$$c\frac{d\theta}{d\eta} = \frac{1}{\eta} f(\theta) + g_{syn} g(\theta)\left(1 - e^{-c\tau}\right) = 0, \tag{8}$$

and further

$$\eta_a = \frac{-\beta}{g_{syn}(1 - e^{-c\tau})}. \tag{9}$$

Under the flow of (6), we follow the branch of the unstable manifold of $(\theta, \eta) = (\theta_{rest}, 0)$ that points into $\{(\theta, \eta) : \theta \geq \theta_{rest}, \eta > 0\}$. We call this branch $(\theta^u(\xi), \eta^u(\xi))$. When $(\theta^u(\xi), \eta^u(\xi))$ passes through $(\pi, 1)$, we say (θ^u, η^u) is a traveling wave.

The unstable manifold (θ^u, η^u) is bounded to the left-up side of the θ-nullcline , with $\theta^u < 0$, in (θ, η)-plane until $\eta > \eta_a(c)$. Thus, a traveling wave can exist only for c satisfying $\eta_a(c) < 1$. For every such c value, define $\theta_a(c$ as the θ-value such that $\theta^u = \theta_a(c)$, when $\eta^u = \eta_a(c)$. In what follows, we use $(\theta_a(c), \eta_a(c))$ as a reference point and consider how the evolution of phase θ depends on velocity c from that point onwards. The following results state that, for larger c, solutions are farther from the firing phase $\theta = \pi$ when they reach $(\theta_a(c), \eta_a(c))$.

Theorem 3. [5] $\theta_a(c)$ is a monotone decreasing function of $c > 0$.

Theorem 4. [5] The time uses for the θ-coordinate of the unstable manifold (θ^u, η^u) varies from $\theta_a(c)$ to π, denoted by $\xi_{\theta_a \to \pi}(c)$, is a monotone increasing function of c.

Theorem 5. [5] The time that η uses when it increases from η_a to 1, $\xi_{\eta_a \to 1}$, is a monotone increasing function of parameter c.

5 Lower Bounds on the Synaptic Coupling Strength Required for Traveling Waves to Exist

As the above mentioned, we observe an interesting feature of system (1) that if the coupling strength g_{syn} is small, the traveling wave solutions will not exist. Therefore, only if the coupling strength g_{syn} is sufficiently large, there will exist two traveling wave solutions with different velocities, one is fast and the other is slow. Furthermore, there exists a critical value g_{crit}, the infimum of g_{syn}. Based on the model, we can get the infimum of g_{syn} which satisfies the existence of traveling wave theoretically. However, we just obtain a lower analytical form bound for the coupling strength not its infimum.

Let θ_f be the value of the θ-coordinate of the unstable manifold (θ^u, η^u) when $\eta = 1$, set $r = -\beta$. Since the infimum cannot be achieved, we derive a lower bound of g_{syn} approximatively by increasing θ_f. If the value of g_{syn} is too small such that the increased θ_f is less than π for all c, then no waves exist.

We split the tracking of (θ^u, η^u) into subintervals. The first interval to consider is $\eta \in [0, \eta_a]$, where η_a was defined in (9). We know that the unstable manifold is confined to the left of the θ-nullcline in (θ, η)-space. As a consequence, $\theta_a = \theta(\eta_a) < 0$. We thus make the approximation $\theta(\eta_a) = 0$, which will increase θ_f.

Starting at $\theta(\eta_a) = 0$, we next integrate (7) over $[\eta_a, \eta_T]$, such that $\theta(\eta_T) = \theta_T$. Since $f^{'}(\theta) > 0$, $f(0) = 2\beta$, and $f(\theta_T) = 0$ on interval $[0, \theta_T]$, we have $f(\theta) < 0$. Consequently we can increases θ_f while neglecting the term $f(\theta)$. Thus from (7), we obtain

$$\frac{d\theta}{(1+\cos\theta)} = g_{syn}\frac{(1-e^{-c\tau})}{c}d\eta. \tag{10}$$

Further, by separation and integration from 0 to θ_T in θ and from η_a to η_T in η, it is easy to obtain:

$$\int_0^{\theta_T} \frac{d\theta}{(1+\cos\theta)} = \int_{\eta_a}^{\eta_T} g_{syn}\frac{(1-e^{-c\tau})}{c}d\eta,$$

$$\eta_T = \eta_a + \frac{c\sqrt{r}}{g_{syn}(1-e^{-c\tau})}. \tag{11}$$

Note: if $\eta_T = 1$, then $\theta_f = \theta_T < \pi$, Substituting $\eta_T = 1$ into (11)we obtain :

$$g_{syn} = \frac{r + c\sqrt{r}}{(1-e^{-c\tau})}. \tag{12}$$

Substituting Taylor expansion of $e^{-c\tau}$: $e^{-c\tau} = 1 - c\tau + \frac{1}{2}(c\tau)^2 + O((c\tau)^3)$ into (12), we have

$$\frac{1}{2}(c\tau)^2 - c\tau + \frac{c\sqrt{r}}{g_{syn}} + \frac{r}{g_{syn}} = 0, \tag{13}$$

$$c_{1,2} = \frac{(g_{syn}\tau - \sqrt{r}) \pm \sqrt{(g_{syn}\tau - \sqrt{r})^2 - 2g_{syn}\tau^2 r}}{\tau^2 g_{syn}}, \tag{14}$$

i.e. we get two velocity c_1, c_2 for at $\eta_T = 1$. The existence of traveling waves requires $\theta_f > \theta_T$, i.e. $\eta_T < 1$, and that is

$$\frac{1}{2}(c\tau)^2 - c\tau + \frac{c\sqrt{r}}{g_{syn}} + \frac{r}{g_{syn}} < 0,$$

so $c \in (c_1, c_2)$.

If $c_1 = c_2$, we can get a lower bound on the strength g_{syn} required for traveling waves to exist,satisfying

$$(g_{syn}\tau - \sqrt{r})^2 - 2g_{syn}\tau^2 r = 0,$$

$$g_{syn} = \frac{(\tau\sqrt{r} + \tau^2 r) \pm \sqrt{(\tau\sqrt{r} + \tau^2 r)^2 - \tau^2 r}}{\tau^2}.$$

So, we take the larger as the lower bounded of the synaptic strength g_{syn}:

$$g_{syn} = \frac{(\sqrt{r} + \tau r) + \sqrt{(\sqrt{r} + \tau r)^2 - r}}{\tau}. \tag{15}$$

6 Numerical Experiments

In this section, we verify the existence of solutions of the traveling wave, its dependence on the parameter c and the dependence on the parameter c, g_{syn}, τ on θ of while guaranteeing the existence of traveling waves numerically.

6.1 Existence of Traveling Wave Solution

In order to find traveling waves, let $\tau = 0.1$, $g_{syn} = 100$, $\beta = -0.1$. Fig. 2 shows how θ changes with c. In Fig. 2, θ can achieve π when c varies from 4.51 to 4.517, therefore traveling wave with large velocity can be found.

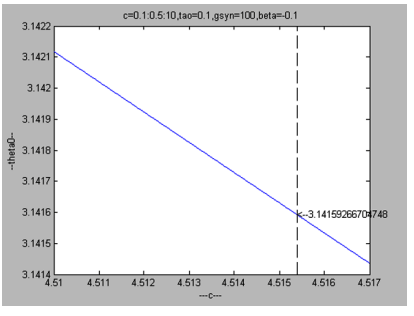


Fig. 2. The curve of θ when c varies from 4.51 to 4.517

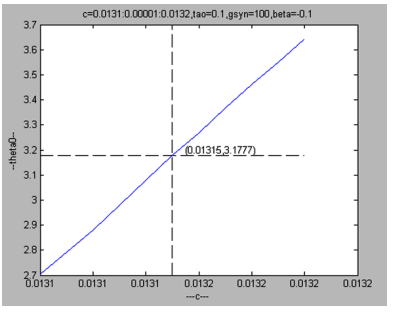


Fig. 3. The curve of θ when c varies from 0.0131 to 0.0132

From Theorem 1, we know that traveling wave with small velocity exists, therefore we can calculate it further. In Fig. 3, θ can achieve π when c varies varies from 0.0131 to 0.0132, therefore traveling wave with small velocity is found.

6.2 Dependence of Traveling Waves on the Velocity

Fig. 4 shows the variation of the θ-nullcline with c. We see that the knee-point of the θ-nullcline decreases while c increases. Further, the smaller value of c, the greater range of the curve.

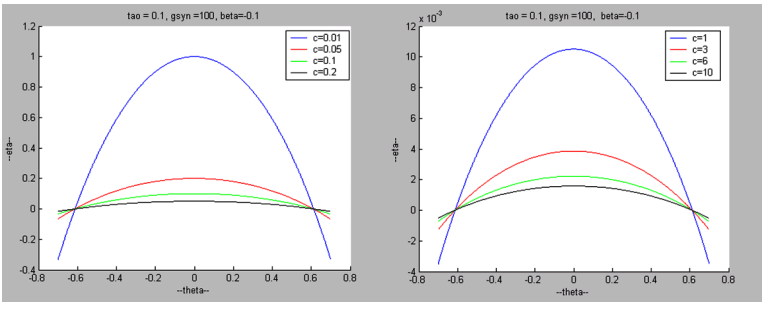


Fig. 4. θ-nullcline of c

Figs. 5 and 6 show the curves of $\theta_a(c)$, $\eta_a(c)$ for $c > 0$, the left hand is $\theta_a(c)$ and the right hand is $\eta_a(c)$. The values of the parameters in both figures are $\tau = 0.1$, $g_{syn} = 100$, $\beta = -0.1$, $\theta_{rest} = -0.6126$. The difference between the two figures is that the c varies from 0.01 to 1 in Fig. 5 and from 1 to 10 in Fig. 6. It can be seen that both $\theta_a(c)$, $\eta_a(c)$ decrease with c increases. Further, the smaller value of c, the greater range of $\theta_a(c)$, $\eta_a(c)$.

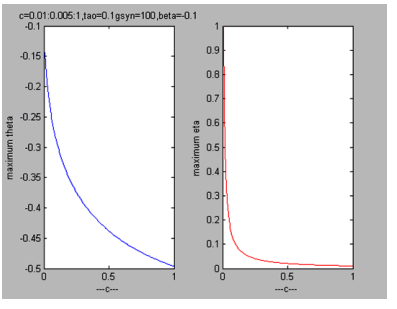


Fig. 5. The curves of $\theta_a(c)$,$\eta_a(c)$ when c varies from 0.01 to 1

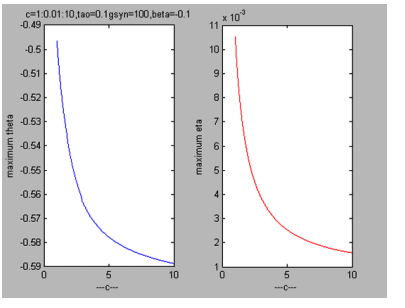


Fig. 6. The curves $\theta_a(c)$, $\eta_a(c)$ when c varies from 1 to 10

6.3 Parameter-Dependence of Traveling Wave Solutions on the Condition of Guaranteeing the Existence of Traveling Waves

Fig. 7 shows the variation of the traveling wave solution with respect to the wave speed c, when g_{syn}, τ is fixed. Fig. 8 shows the variation of the traveling wave solution with respect to the synaptic coupling strength g_{syn}, when c, τ is fixed. Fig. 9 shows the variation of the traveling wave solution with with respect to the interactional time τ among cells, when c, g_{syn}, is fixed.

It can be seen from Figs. 7, 8 and 9 that there is an interesting phenomenon that when c, g_{syn}, τ increase to some value respectively, the curve will jump accordingly. Further, when c, g_{syn}, τ continue increasing to some large value respectively, the curve will jump too, and θ will decrease by 2π each time when the curve jumps. This phenomenon

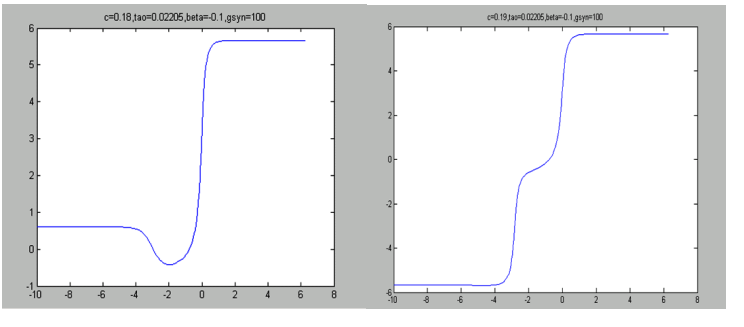


Fig. 7. The dependence of travelling waves on c

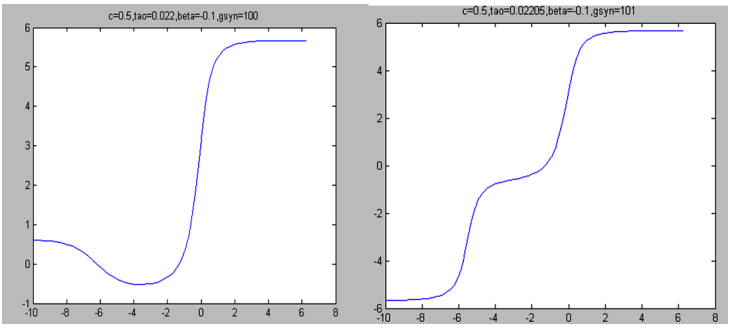


Fig. 8. The dependence of traveling waves on g_{syn}

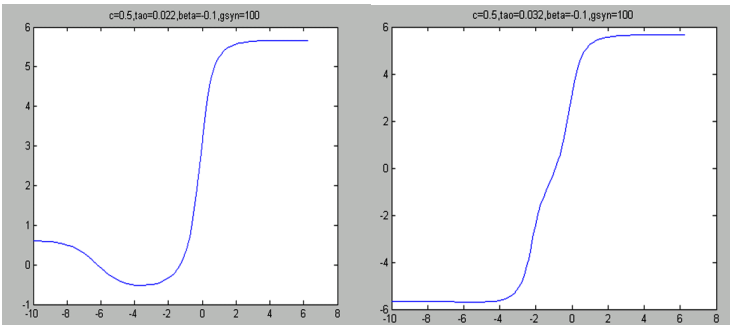


Fig. 9. The dependence of travelling waves on τ

just shows the dependence of generation of n-spiking travelling waves on the values of c, g_{syn}, τ which needs further research.

7 Conclusions

In this paper, we derive an analytical lower bound of synaptic coupling strength g_{syn} for traveling waves to exist. Using the numerical simulation methods, we verify some related results on the existence of traveling waves and the dependence of existence of traveling waves on parameters, and give the solution of traveling waves numerically. Furthermore, we investigate the change of the traveling waves solution curve while the parameters c, g_{syn}, τ change respectively. Finally, there is a phenomenon that the curve of θ jumps with the increase of parameters, and this needs further research.

As further extensions of this study, there exist a number of topics for future work. In this paper, we mainly study the single firing spike traveling waves. The situation on n-spiking traveling waves needs to be further studied. The system is assumed to be noiseless in this paper. In real world, however, the scheme clearly has to be implemented to be robust to noise. Furthermore, since the system is naturally delayed, the study of the system with delay needs more attention too.

Acknowledgments. The work was supported by the National Nature Science Foundation of China under the grant No.10771012, the Research Fund for the Doctoral Program of Higher Education under the grant No.20070004030.

References

1. Krishnaiah, D., Prasad, D.M.R., Bono, A., Pandiyan, P.M., Sarbatly, R.: Application of Ultrasonic Uaves Coupled with Functional Link Neural Network for Estimation of Carrageenan Concentration. International Journal of Physical Sciences 3, 90–96 (2008)
2. Dahlem, M.A., Schneider, F.M., Panchuk, A., Hiller, G., Scholl, E.: Control of Sub-excitable Waves in Neural Networks by Nonlocal Coupling. In: Proceedings of the Workshop NetWorks, Aranjuez, pp. 1–15 (2007)
3. Tolman, H.L., Krasnopolsky, V.M., Chalikov, D.V.: Neural Network Approximations for Nonlinear Interactions in Wind Wave Spectra-direct Mapping for Wind Seas in Deep Water. Ocean Modelling 8, 253–278 (2005)
4. Ruktamatakul, S., Bell, J., Lenbury, Y.: Wave-front Solution Behaviour for Continuous Neural Networks with Lateral Inhibition. IMA J. Applied Mathematics 71, 544–564 (2006)
5. Wu, G.: Regular Single-spike Traveling Waves in A One-dimensional Netwok of Theta Neurons. Master Thesis, Inner Mongolia University, China (2006)
6. Katriel, G.: Periodic Traveling Wave in the Theta Model for Synaptically Connected Neurons (2004)
7. Osan, R., Rubin, J., Xcurtu, R., Ermentrout, B.: Mutiple-spike Wave in One-dimensional Integrate-and-Fire Neural Network. J. Math. Biol. 48, 243–274 (2004)
8. Rubin, J.: A Nonlocal Eigenvalue Problem for the Ttability of A Traveling Wave in Neuronal Medium. Discrete and Continuous Dynamical Systems 10, 925–940 (2004)
9. Osan, R., Rubin, J., Ermentrout, B.: Regular Traveling Waves in A One-dimensional Network of Theta Neurons. SIAM J. Appl. Math 62, 1197–1221 (2002)
10. Ermentrout, G.B., Kopell, N.: Parabolic Bursting in An Excitable System Coupled with A Slow Oscillation. SIAM J. Appl. Math 46, 223–253 (1986)
11. Hoppensteadt, F.C., Izhikevich, E.: Weakly Connected Neural Network. Springer, New York (1997)
12. Pinto, D.J., Ermentrout, G.B.: Spatially Structured Activity in Synaptically Coupled Neuronal Networks: I. Traveling Fronts and Pulses. SIAM J. Appl. Math 62, 206–225 (2001)

Seismic Responses Prediction of Nonlinear Building Structures Based on Multi-Branch BP Neural Network

Linsheng Huo[1], Hongnan Li[1], and Bing Li[2]

[1] State Key Laboratory of Coastal and Offshore Engineering,
Dalian University of Technology, Linggong Road 2, Ganjingzi District,
Dalian 116023, P.R. China
{lshuo,hnli}@dlut.edu.cn
[2] School of Civil Engineering, Shenyang Jianzhu University,
Hunnan East Road 9, Hunnan New District, Shenyang 110168, P.R. China
bingleesy@sohu.com

Abstract. In this paper, a multi-branch back propagation neural network (BPNN) is adopted to predict the nonlinear seismic responses of an eccentric three-story reinforced concrete building. First of all, the network is trained in batch by the vibration table test data of the structure with the maximum acceleration of ground motion in 0.4g. Then, the trained network is used for structural responses prediction. The nonlinear structural acceleration responses of the each story are evaluated by the trained network for the maximum acceleration of ground motion in different amplitudes. Compared with the experimental results, it turns out that the trained network can accurately predict structural future dynamic responses.

Keywords: Structural dynamic system, Artificial neural network, System identification, Seismic responses.

1 Introduction

Building structures are subjected to various kinds of dynamic loadings, such as earthquakes and wind. The dynamic responses can result in uncomfortable, and even seriously dangerous circumstances for buildings. The inclusion of vibration absorbers in tall buildings can be a successful method for mitigating the effects of these dynamic responses. A structural control system is commonly classified by its device type, which results in three general types: passive, active and semi-active. Active control systems have the ability to adapt to different loading conditions and to control vibration modes of the structure. However, Classic control theory and modern control theory are both based on accurate mathematical model. And the validity, robustness and stability of control system directly depend on the precision of the model. Building structure, in which there are both structural and non-structural members, is a complicated nonlinear system whose energy dissipation mechanism has not been totally uncovered. And nonstructural members are usually ignored when structures are designed. It has been proved that, it may result in great control deviation, deteriorate control performance even destabilize the controlled structure if designed control system based on simplified mathematic model is applied to real structure. System modeling, which plays a very important role in automatic control, directly determines

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 919 – 928, 2009.

control quality. It has been turned from modeling by mechanism to modeling by statistics and system identification. System identification is an important branch of the study on control theory. It is the basis for designing control system [1].

Artificial neural networks (ANN) have attracted considerable attention and shown great potential for modeling complex non-linear relationships in recent years [2–6]. ANNs are derived through a modeling of the human brain and are composed of a number of interconnected artificial neurons that are similar to the biology neural networks. The function of single cell is simple and limited, but the networks composed by numerous neural cells can complete complicated tasks. A significant benefit of using an ANN is its ability to learn relationships between variables with repeated exposure to those variables. Therefore, instead of deriving an analytical relationship from mechanical principles to model a system, the ANN learns the relationship through an adaptive training process. One can construct a non-linear mapping function from multiple input data to multiple output data within the network in a distributed manner through a training process. The trained network has a feature of the so-called 'generalization', i.e. a kind of interpolation, such that the well-trained network estimates appropriate output data even for untrained patterns. Parallel data processing is also one of the main features of ANNs. In the system identification based on ANNs , a set of samples, usually structural response or seismic input data, are used to train the network to continuously adjust the link weights between neurons [7-8]. As a result, the relationship between the samples is approximated by updating system model or structural dynamic characteristics. ANNs provide an efficient approach for system identification and have been applied to structural control [9-14].

A dynamic structural system is inherently a multi-input multi-output system, and structure state variables and seismic inputs have different influences on its dynamic responses [16]. To construct an identified model that can precisely reflect structural dynamic characteristics is the key factor to realize a valid vibration control with high quality. A kind of multi-branch error back propagation neural network (BPNN) identification model has been proposed [1]. In a BPNN model, to enhance training efficiency and prediction precision, structural state vector and earthquake inputs, that affect structural dynamic responses, are treated as separate branches of the network and are input into the model. In this paper, the nonlinear seismic responses of an eccentric building are predicted using the multi-branch BPNN. The network is trained using the collected experimental data from the shaking table test with the maximum earthquake input in 0.4g. Then, the trained network is used to predict the nonlinear responses of the structure subjected to the earthquake with the amplitude of 0.5g, 0.6g, 0.7g, 0.8g and 1.0g.

2 Multi-branch BPNN System Identification Model

The objective of system identification is to establish an equivalent mathematical model for a system to be identified based on the observations of the inputs and outputs of the system. There are three key factors in system identification including input and output data, class of model and equivalent criteria. The task of system identification is to extract a model (M) of system (S) from the inputs and outputs, while satisfying the following requirements:

$$\left\| y-t \right\| = \left\| M(x)-S(x) \right\| \le \varepsilon , \tag{1}$$

where y is the system response corresponding to input x, and t is model response corresponding to input x; ε is the preset precision for identification.

Set e=y-t, then equivalent criteria leads to

$$\min_f \left\| e \right\| = \min_f \left\| y-t \right\| . \tag{2}$$

The multi-branch BPNN system identification model has been proposed as shown in Fig. 1 [1]. In the model, structural state variables and earthquake inputs, that affect structural dynamic responses, are treated as separate branches and are input into the model.

The input of the model are

$$\text{Input} = [\ddot{U}_{x_g}^T, \quad \ddot{U}_{y_g}^T, \quad \dot{U}_{x_1}^T, \quad \cdots, \quad \dot{U}_{\theta_n}^T, \quad U_{x_1}^T, \quad \cdots, \quad U_{\theta_n}^T]^T , \tag{3}$$

where

$$\begin{aligned}
\ddot{U}_{xg} &= \left[\ddot{u}_{xg}(k), \quad \cdots, \quad \ddot{u}_{xg}(k-q+1) \right]^T , \\
\ddot{U}_{yg} &= \left[\ddot{u}_{yg}(k), \quad \cdots, \quad \ddot{u}_{yg}(k-q+1) \right]^T , \\
\dot{U}_{x1} &= \left[\dot{u}_{x1}(k), \quad \cdots, \quad \dot{u}_{x1}(k-r+1) \right]^T , \\
\dot{U}_{\theta n} &= \left[\dot{u}_{\theta n}(k), \quad \cdots, \quad \dot{u}_{\theta n}(k-r+1) \right]^T , \\
U_{x1} &= \left[u_{x1}(k), \quad \cdots, \quad u_{x1}(k-r+1) \right]^T , \\
U_{\theta n} &= \left[u_{\theta n}(k), \quad \cdots, \quad u_{\theta n}(k-r+1) \right]^T
\end{aligned} \tag{4}$$

The outputs of the model are

$$\text{Output} = Y(k+1) = [y_1(k+1), \ y_2(k+1), \ \cdots \ y_{s3}(k+1)]^T . \tag{5}$$

In Fig. 1, there are one q-dimensional input sub-branch of seismic wave in x direction, one q-dimensional input sub-branch of seismic wave in y direction, n r-dimensional input sub-branches of floor velocity response in x direction, n r-dimensional input sub-branches of floor velocity response in y direction, n r-dimensional input sub-branches of floor velocity response in θ direction, n r-dimensional input sub-branches of floor displacement response in x direction, n r-dimensional input sub-branches of floor displacement response in y direction, and n r-dimensional input sub-branches of floor displacement in θ direction. There are totally 2+6n input sub-branches and the dimension of inputs is l=2q+ 6nr (n is the number of layers). However, in actual applications, sensors could not be deployed at all degrees of freedom to measure the responses of a structure. Because, a trained ANN has the ability of generalization, it is not necessary to use all the structural state data to train the networks, thus the number of inputs in Fig. 1 will be reduced considerably and the training efficiency is improved.

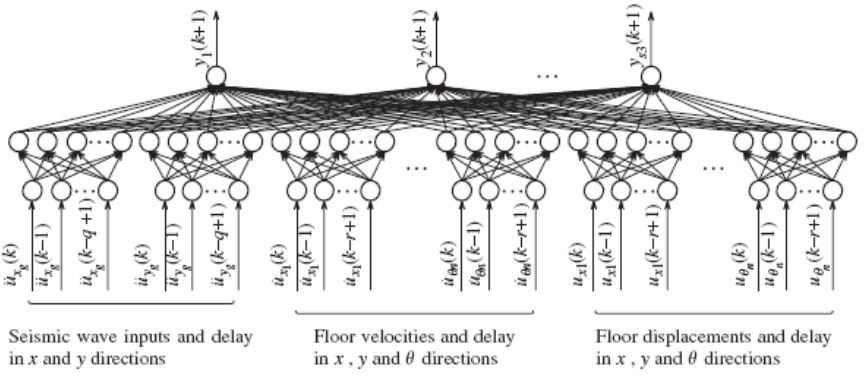


Fig. 1. The multi-branch BPNN identification model

Assume that a structural dynamic system takes the following form:

$$X(k+1) = f[X(k),\ \cdots,\ X(k-r+1),\ \ddot{U}_g(k),\ \ddot{U}_g(k-q+1)]^T, \tag{6}$$

where $X(k+1)$ represents the state vector of the structure at time $k+1$, that is determined by former r state vectors and q seismic inputs in two directions. $\ddot{U}_g(k)$ represents the seismic inputs at time k. Since the multi-branch BPNN model is used in the identification of a dynamic structure, the inputs and outputs of the multi-branch BPNN model can be chosen to take the form similar in Eq. (6). The form of such a multi-branch BPNN model is

$$Y(k+1) = \hat{f}[X(k),\ \cdots,\ X(k-r+1),\ \ddot{U}_g(k),\ \ddot{U}_g(k-q+1)]^T, \tag{7}$$

where $Y(k+1)$ represents the output vector of a multi-branch BPNN model at time $k+1$, i.e. the state vector of the identified model at time $k+1$. $\hat{f}$ that is an approximation of f, represents the mapping of the inputs onto the outputs. The multi-branch BPNN model is essentially a static model, but the introduction of tapped delay line (TDL) provides it with some dynamic features, which allows for a more accurate approximation of the relationship between inputs and outputs. The principle of system identification based on multi-branch BPNN model is self-explained in Fig. 2.

The purpose of utilizing the system identification based on a multi-branch BPNN model is to predict the dynamic behavior of a structure after the model has been sufficiently trained. After an identification model is established, it will be trained by seismic sample data and corresponding responses to learn the law of structural vibration, and then store the trained link weights between the neurons in the network. The process of identification is to minimize the error between actual outputs and the expected ones of the network through adjusting the link weights.

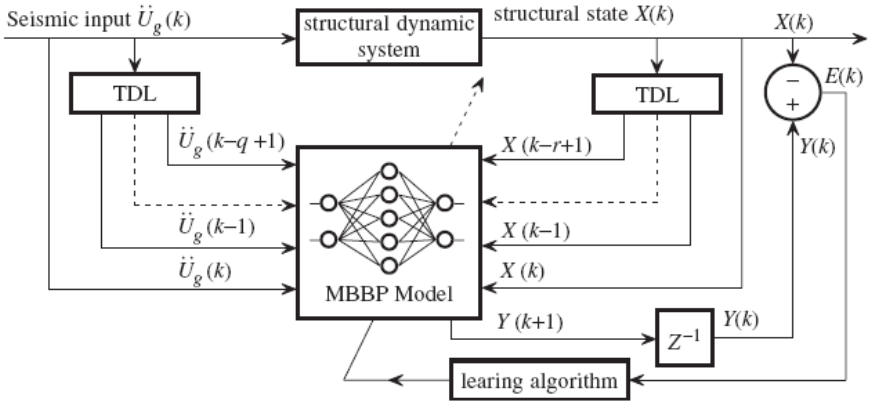


Fig. 2. Principle of system identification based on multi-branch BPNN model

3 Experimental Setup and Procedure

The experiment was carried out with the earthquake simulator facility of the State Key Laboratory of Coastal and Offshore Engineering, Dalian University of Technology in China. The facility includes a 4m×3m steel platform, driven by servo-hydraulic actuators and a MTS analogue electronic control system, which makes fine feedback over accelerations, velocities and displacements. With a maximum horizontal and vertical displacement of ±75mm and 50mm respectively, the shaking table has about 10-ton of payload capacity and a frequency range from 0.1Hz to 50Hz.

An experimental model is designed and constructed for this study, i.e. a three-story reinforced concrete frame-shear wall structure with an asymmetric shear wall distribution (see Fig. 3). The height of the structural model is 3.1m, and the total mass of the model is nearly 30kN. The cross-section area of the column and beam is 0.08m × 0.08m and 0.06m × 0.1m, respectively. The thickness of the shear wall is 0.03m. The data were acquired simultaneously at the rate of 500Hz in 16 channels. Accelerometers and load cells were used to measure the acceleration at each story, and shear forces on the columns of the first story. Fig. 4 shows the accelerometer placements on each floor of the building model. The base excitations were the 1940 El Centro earthquake NS acceleration time series and were scaled with the increasing Peak Ground Acceleration (PGA) to be 0.4g, 0.5g, 0.6g, 0.7g, 0.8g and 1.0g. The tested structure was excited by six consecutive horizontal acceleration processes with increasing intensities on the shaking table and without repairing or strengthening among processes.

No cracks were observed in the structure with PGA to be 0.4g and 0.5g, which means that the structure is still in elastic state. Obvious cracks can be found in the columns of the first floor when the structure subjected to the ground motion with PGA 0.6g and the structural responses became nonlinear. The cracks developed much when the PGA increased to be 0.7g and 0.8g. The width of cracks became larger and the "plastic hinges" were formed in the columns and beams when the structure subjected the ground motion with PGA 1.0g. The structure was a strong nonlinear dynamic system at this time.

Fig. 3. Experimental setup for shaking table tests

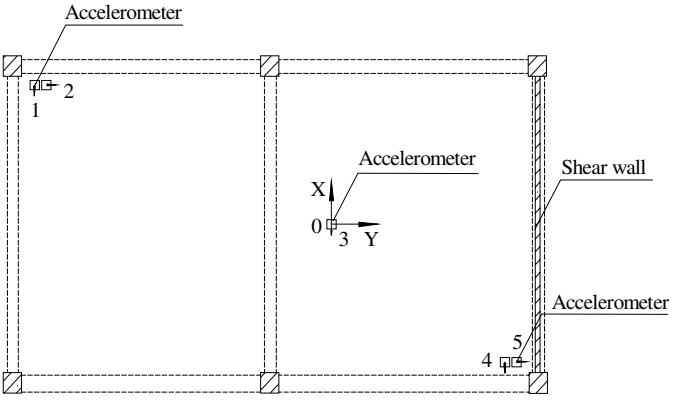


Fig. 4. Experimental setup for shaking table tests

4 Prediction of the Nonlinear Seismic Responses

In this paper, the accelerations of the three-story building is predicted using the proposed multi-branch BPNN. The multi-branch BPNN is essentially a static model, delay is added in the inputs to accurately describe the dynamic behavior of the structure. The inputs of BPNN include the ground motion data at the current time and previous two steps, the acceleration response of the three floors at the current time and previous three steps. The outputs are the accelerations of the building model at each story. In summary, there are totally 15 inputs and 3 outputs for the BPNN. To facilitate the training, the sample data are transformed into values between -1 and 1 to shorten the learning time of the network as follows:

$$X^* = 2\frac{X - X_{\min}}{X_{\max} - X_{\min}} - 1, \quad X^* \in [-1, 1], \tag{8}$$

where X is the sample data, $X_{\min}$ and $X_{\max}$ are the maximum and minimum of the sample data, respectively, and X^* is the transformed value.

The chance of "over-fitting" is rather little once the number of training sample data is far more than that of the neurons of hidden layers. Therefore, 1000 observations are chosen as the training data ensemble for structural responses with ground motion PGA 0.4g. To improve the convergence speed, training efficiency and algorithm stability, an acceleration factor and an adaptive learning algorithm are adopted to train the model and the maximum training step is set as 500. Then, a numerical optimization algorithm called Levenberg–Marquardt is adopted to train the model and the maximum training step is set as 200. The training function was the generally used mean square error as in Eq. (9), and this function is modified by regularization to improve its ability of generalization. As a result, Eq. (10) is used as the performance training function.

$$mse = \frac{1}{N}\sum_{i=1}^{N}(t_i - y_i)^2 \tag{9}$$

$$msereg = \gamma \cdot mse + (1-\gamma)msw, \tag{10}$$

where γ is a scalar ratio, msw is the average of all the squared weights, i.e.

$$msw = \frac{1}{N}\sum_{j=1}^{N} w_j^2 \tag{11}$$

It should be noted that, after a large amount of training, that the ability of training efficiency and generalization is best when the nodes of two hidden layers is 10 and 15, respectively. The trained network is used to predict the nonlinear responses of the structure subjected to the earthquake with different amplitudes of PGA. The mean of squared errors for the prediction are listed in Table 1. Due to the paper limitation, only the predicted time histories of the third floor acceleration with PGA 0.6g, 0.8g and 1.0g are given here to show the training results, as shown in Fig. 5, Fig. 6 and Fig. 7.

It could be seen, from the training results and prediction ability, that the overall performance of the proposed model is satisfied.

Table 1. The mean of squared errors for predicted accelerations of the structure

PGA	First Floor	Second Floor	Third Floor
0.4g	1.3669e-5	2.7144e-6	7.2738e-5
0.5g	9.5711e-4	7.2963e-6	1.0413e-4
0.6g	4.8775e-5	3.4402e-6	1.1140e-4
0.7g	1.5076e-4	1.8667e-5	8.3947e-4
0.8g	7.1419e-4	3.1866e-5	8.6780e-4
1.0g	0.0016	7.0186e-5	9.2791e-4

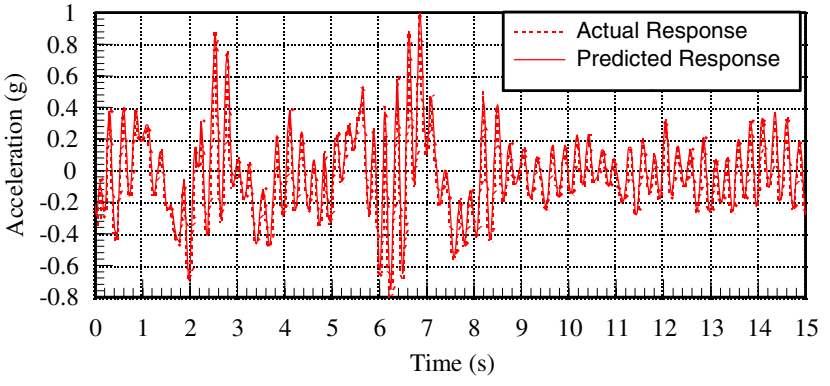


Fig. 5. Comparisons of predicted and actual acceleration of the third floor (PGA 0.6g)

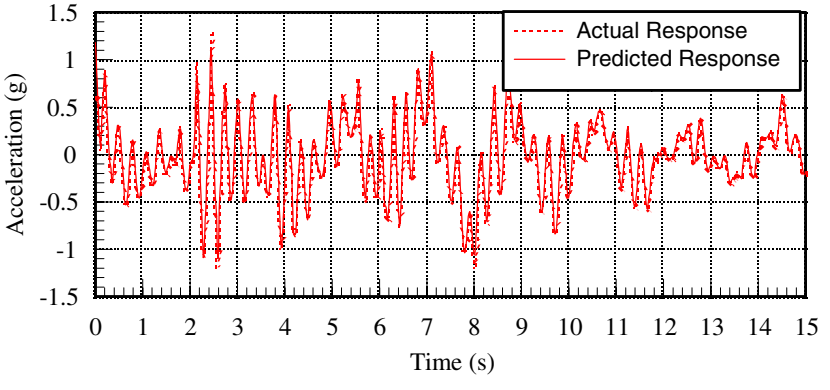


Fig. 6. Comparisons of predicted and actual acceleration of the third floor (PGA 0.8g)

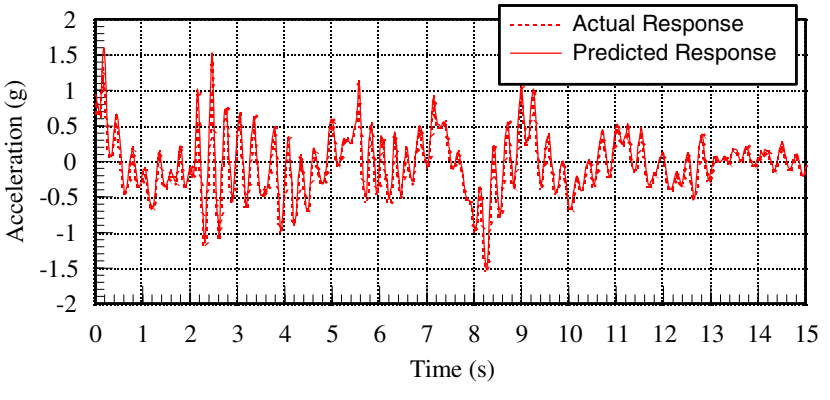


Fig. 7. Comparisons of predicted and actual acceleration of the third floor (PGA 1.0g)

5 Conclusions

A kind of multi-branch BPNN model is used to identify the multi-input multi-output nonlinear structural dynamic system in this paper. The network is trained by the experimental data from a shaking table test of a reinforced concrete building structures. In addition, the trained model was used to predict structural responses under the seismic wave with different amplitudes inputs to test the ability of generalization of the identified model. Numerical simulation shows that the proposed multi-branch BPNN model can accurately identify the structural nonlinear characteristics, and this model has a high precision in predicting structural future dynamic responses. Such high precision of model identification and response prediction can provide a solid foundation for further structural control.

Acknowledgements

This research is financially supported by the National Natural Science Foundation of China (No. 50708016). This support is greatly appreciated.

References

1. Li, H.N., Yang, H.: System Identification of Dynamic Structure by the Multi-Branch BPNN. Neurocomputing 70, 835–841 (2007)
2. Brown, A.S., Yang, H.T.Y.: Neural Networks for Multiobjective Adaptive Structural Control. J. Struct. Eng. 127, 203–210 (2001)
3. Brown, A.S., Yang, H.T.Y., Wrobleski, M.S.: Improvement and Assessment of Neural Networks for Structural Response Prediction and Control. J. Struct. Eng. 131, 848–850 (2005)
4. Kim, D.H., Seo, S.N., Lee, I.W.: Optimal Neurocontroller for Nonlinear Benchmark Structure. J. Eng. Mech. 130, 424–429 (2004)
5. Masri, S.F., Smyth, A.W., Chassiakos, A.G., Caughey, T.K., Hunter, N.F.: Application of Neural Networks for Detection of Changes in Nonlinear Systems. J. Eng. Mech. 126, 666–676 (2000)
6. Rajasekaran, S.: Functional Networks in Structural Engineering. J. Comput. Civil Eng. 18, 172–181 (2004)
7. Chen, H.M., Qi, G.Z., Yang, J.C.S., Amini, F.: Neural Networks for Structural Dynamic Model Identification. J. Eng. Mech. 121, 1377–1381 (1995)
8. Masri, S.F., Nakamura, M., Chassiakos, A.G., Caughey, T.K.: Neural Network Approach to Detection of Changes in Structural Parameters. J. Eng. Mech. 122, 350–360 (1996)
9. Ghaboussi, J., Joghataie, A.: Active Control of Structures Using Neural Networks. J. Eng. Mech. 121, 555–567 (1995)
10. Hung, S.L., Kao, C.Y., Lee, J.C.: Active Pulse Structural Control Using Artificial Neural Networks. J. Eng. Mech. 126, 838–849 (2000)
11. Kim, D.H., Lee, I.W.: Neuro-Control of Seismically Excited Steel Structure Through Sensitivity Evaluation Scheme. Earthquake Eng. Struct. Dyn. 30, 1361–1377 (2001)
12. Kim, J.T., Jung, H.J., Lee, I.W.: Optimal Structural Control Using Neural Networks. J. Eng. Mech. 26, 201–205 (2000)

13. Li, H.N., Huo, L.S.: Semi-Active TLCD Control of Fixed Offshore Platforms Using Artificial Neural Networks. China Ocean Eng. 17, 277–282 (2003)
14. Tang, Y.: Active Control of SDF Systems Using Artificial Neural Networks. Comput. Struct. 60, 695–703 (1996)
15. Li, B.: Nonlinear Analysis of R/C Frame-Wall Structures to Multiple Earthquake Excitations and Experimental Research. Ph. D. Dissertation, Dalian University of Technology, China (2005)
16. Xu, Z.D., Shen, Y.P., Guo, Y.Q.: Semi-active Control of Structures Incorporated with Magnetorheological Dampers Using Neural Networks. Smart Mat. Struct. 12(1), 80–87 (2003)

Coupling Analysis of Manufacturing Characteristics and Mechanics Property of Microminiature Gear Mechanism Based on Neural Network

Xin Jin, Zhijing Zhang, Fuchang Zuo, and Zhongxin Li

School of Mechanical and Vehicular Engineering, Beijing Institute of Technology, Beijing 100081, China
{goldking,zhzhj,zfch2008,lizx30526}@bit.edu.cn

Abstract. A coupling analysis method of manufacturing characteristics and mechanics property of microminiature gear mechanism based on BP neural network was proposed. By use of the existing finite element model with manufacturing characteristics, output data as BP neural network training set of samples was obtained. Through a comparative study of the effects of different network parameters settings on the precision of network model, the optimal network structure and parameters were determined and the neural network model which can approximate the mechanics property microminiature gear mechanism with high precision. This shows the nonlinear coupling relationships between input manufacturing characteristics and the output mechanical characteristics, and verifies the accuracy of the model.

Keywords: Neural Network, Manufacturing Characteristics, Mechanics property, Coupling Analysis, Microminiature Gear Mechanism.

1 Introduction

As reduction of size of micro-mechanical systems and components, the size that has no obvious effect inconventional systems will have strengthen role in the micro field. For example, effects on the performance of parts of manufacturing characteristics (which refers to an action of the manufacturing process given geometry and physical, reflecting the effect of the choice of process parameters and ability of process systems, It is mainly characterized by surface topography parameters, assembly errors, size errors, form and position errors, material properties, heat treatment hardness.) factors such as processing errors, the surface quality, assembly error can not be ignored [1-3]. The study of the effect of manufacturing characteristics of micromechanical systems on the relationship between manufacturing characteristics and mechanics property will have great significance on the evaluation of running performance of system. The coupling between manufacturing characteristics and mechanical characteristics is non-linear [4]. There is no explicit expression of function. To solve this kind of highly nonlinear problems, FEM software is often used. But because of the need to consider many manufacturing characteristics factors

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 929 – 936, 2009.

and many values of each factor, the data of parametric modeling is a huge and the calculation is time consuming. So it is not conducive to the engineering application. ANN is a black box modeling tool. Through analysis of the actual system input and output parameters, we create a neural network model. BP neural network is commonly used, and one of the important applications is starting with the training samples to make non-linear approximation for unknown mapping function. In this paper, a coupling analysis method of manufacturing characteristics and mechanics property of microminiature gear mechanism based on BP neural network was proposed. By use of the existing finite element model with manufacturing characteristics and data, we got the training set of samples of BP neural network. BP neural network model which is used for coupling relationship analysis of manufacturing characteristics of microminiature gear and mechanics property was established. This realizes to express the nonlinear relationships between manufacturing characteristics and mechanics property by use of explicit function. The verification result shows that the neural network model is correct and the analysis error is less than 9%.

2 The Structure of BP Neural Network

Structurally, BP neural network is a typical multilayer network. There are three neuron layers in standard BP model. The lowest layer is input layer, the middle layer hidden layer, and the top one is output layer. Between layers the whole connecting way is used. The units do not connect with each other in the same layer, as shown in Figure 1.

The basic idea of BP neural network model is: to study forward propagation of the signals and the back propagation of errors. In the process of forward propagation, input information is calculated on after another layer by layer from input layer to output layer. If ideal results have not been obtained in the output level, the changed value of its error is calculated, and then modifying the value of linking points reversely to achieve the desired goal. According to Robert Hecht-Nielson we can

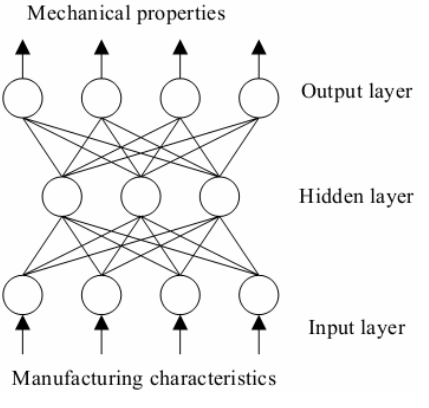


Fig. 1. Three-layer BP neural network

prove BP theorem that any continuous function in this interval of can be approached by a BP network containing a hidden layer. So a three-layer BP network can complete n to m dimensions nonlinear mapping relationships with any required precision [5-7]. There are n inputs which are $X_1, X_2, \cdots, X_n$. The hidden layer has m neurons. The corresponding output neurons are $y_1, y_2, \cdots, y_m$. ω_{il} is the weight value at the i input node in the l hidden layer, where $i = 1,2\cdots, n, l = 1,2\cdots, m$. v_{lk} is the connection weight value from hidden layer to output layer, b_k, b_l are the deviations. The relationship between input and output is as follows:

$$y_l = f\left(\sum_{i=1}^{n} \omega_{il} x_i - b_l\right) \tag{1}$$

$$O_k = f\left(\sum_{l=1}^{m} v_{lk} y_l - b_k\right) \tag{2}$$

3 The Coupled Model and Verification Based on BP Neural Network

3.1 Coupled Modeling Based on BP Neural Network

The basic idea of modeling of the coupled model between manufacturing characteristics and mechanics property is that to treat the specific manufacturing characteristics parametric variables such as dimension errors, material properties, temperature, form errors, and location errors as input variables. And treat the concerned structural characteristics of the output indicators such as stress, deformation and so on as output variables. Through study of engineering examples, the neural network obtains the non-linear mapping relations between input and output variables. The specific ways of establishment are as follows:

① Through a series of as few as possible deterministic test or structural finite element numerical calculation, obtain the output mechanics property of the structures under certain working conditions, and use them as training samples of a neural network model;

② By using of BP neural networks, optimize the value and study the sample, to set up the mapping relations between the input parameters and output characteristics indicators;

③ Corresponding to the way of normalization, transfer function, number of neural network layers, the number of neurons in each layer in the structure of network, the weight value and deviation will be trained successfully and then use them as known coefficient to substitute into the formula (1) and (2) for conversion. So the coupling relationship between the output parameters and mechanics property is obtained.

The network of the coupled model has three layers. There are 7 neurons in the input layer, which are position errors of three pairs of meshing gear respectively. T_1 is the position error of the first level driving gear, T_2 is the position error of the pinion in

first level, T_3 is the position error of the bull gear in second level, T_4 is the position error of the pinion in second level, T_5 is the position error of the bull gear in third level, T_6 is the position error of the pinion in third level, T_7 is the tooth trace errors of gears (the same for all the gears). There are 6 neurons in the output layer, which are the maximum equivalent stress σ_1, σ_2, σ_3, σ_4, σ_5 and σ_6 of the six gears respectively. The transfer function between hidden layer and input layer is hyperbolic S-transfer function sigmoid, which has the function as non-linear magnification factor. The function can map the input of neurons from $(-\infty,+\infty)$ to (-1, +1). The *S* activation function can be used to deal with and approximate non-linear relationship between the input and output. The linear transfer function *purelin* is used between hidden layer and output layer, so that any output value can get in the network. The structure of network is shown in Figure 2.

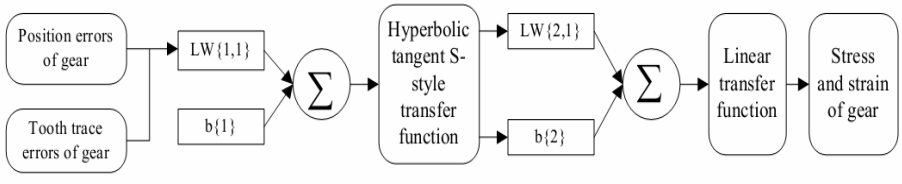


Fig. 2. Structure of BP neural network

3.2 Parameter Design of BP Neural Network Model

The design and selection of parameters of BP neural network models includes: choosing the training methods, determing the number neurons in hidden layer, selecting the initial weight value, and determining the study rate and the selection of expectation error.

3.2.1 Training Method and Process of Network

The selection of neural network training method lies on a number of factors, such as the complexity of the mapping, the number of training set samples, the number of network weight value and threshold, the error goal, the use of the network and so on. At the same time the speed of training is difficult to predict. However, LM algorithm is usually the fastest for function containing hundreds of weight value approximation of the network. In many cases, using the training function *trainlm* of the LM algorithm can get a smaller square error than any other method. This network has 7 input parameters and 6 output parameters as well as more than 100 weight values. So it is relatively complex and requires a faster training method. *Trainlm* is *Levenberg - Marquardt* optimization method which has high computing speed. So we use *trainlm* as a training method of network.

The input training data requires preprocessing, that is to normalize it in the interval (-1, +1), when it is inputted. In this paper, there are a total of 16 orthogonal experiments, 7 input elements, and 6 output elements. The basic computing unit of MATAB is matrix, so the input and output of training set of network are written in the

form of matrix and are 7 × 16 and 6 × 16 matrix respectively. The input row vector of the matrix is the value of the 16 tests of each factor (position and tooth trace errors of the gears) and column is the value of 7 factors for each test. The output row vector of the matrix line is the result of each of the variables (equivalent stress of gears) for 16 tests of the results and the column vector is the result of 7 dependent variables for each test. The input matrix is denoted by p and output matrix is t. In addition, the input parameters don't require normalization because of they are all in interval (-1, +1) as well as the output results do not need anti-normalization.

3.2.2 To Determine the Number of Neurons in Hidden Layer

The number of neurons in hidden layer is determined by experiments. There are many input parameters, output parameters and training arrays, so more hidden neurons are needed to achieve accurate approximation and fast convergence. By setting different numbers of hidden neurons, we select the most appropriate number of hidden neurons through a series of tests and comparing the speed of convergence. The results are shown in Table 1:

Table 1. Training results of different number of neurons in hidden layer

Number of neurons in hidden layer	Target errors	Time of training
18	277.74	20000
20	8.7	20000
22	228.37	20000
24	255.43	20000
26	366.63	20000
27	0.0996	13073
28	0.0892	111
29	0.0085	631
30	0.0079	3653

We can see that the appropriate number of hidden layer neurons is between 28 and 29 with fast convergence and fast training. The training error curves are shown in Figure 3 and Figure 4. Considering precision demand of the network, and the two networks on a group of its known function of the value of the parameters of training, we trained the parameters with known functions using the two networks. Through comparing the training results of the two networks, we know that the result is closer to the actual value of the function when the number of hidden neurons is 29.

3.2.3 Determine of Other Parameters

The initial weight values of the network use the default values, which is the random number in interval (-1,+1). When the characteristics of different networks are

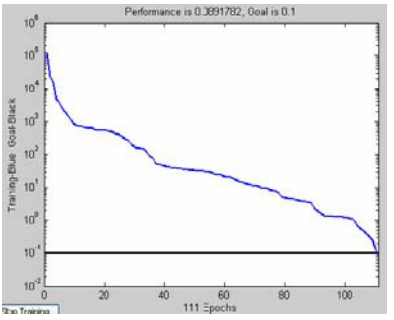


Fig. 3. Training curve when S=28

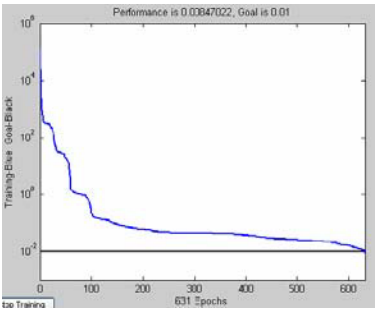


Fig. 4. Training curve when S=29

compared, the network needs to initialize. MATLAB statement is as follows [8]: *net = init (net)*. Learning rate is 0.1, though smaller learning rate will lead to relatively slow convergence, it can guarantee the stability of the system. Through a series of pilot training, the learning rate is set to be 0.1 and mean square error (MSE) is set to be 0.01. There are 112 sets of data in this network, and all the data in the 10^2 orders of magnitude. Setting MSE to 0.01 can sufficiently meet the requirements of precision. The maximum time of training is set to 20000, the training will automatically stop when the error is within the expections error.

3.3 Model Verification of the Network

The validity of the BP neural network model is verified according to the FEM results of a microminiature gear mechanism. Table 2 shows the input data for verification of network testing set and Table 3 shows the comparison between results of the neural network model(Y_1) and the FEM(Y_0).

Table 3 shows that the errors of solution of equivalent stress are less than 9%. The precision of the neural network model is relatively high and can be used as the explicit function of the coupling relationship between manufacturing characteristics and mechanics property of microminiature gear mechanism. It can be applied to solve the largest equivalent stress of gears meshing point.

Table 2. Input data for verification of the network

Manufacturing characteristics / Input data set	Position errors of each level (μm)						T_7
	T_1	T_2	T_3	T_4	T_5	T_6	(°)
1	3	3	0	0	0	0	0
2	0	0	3	3	0	0	0
3	0	0	0	0	6	6	0
4	0	0	0	0	0	0	0.01
5	4	3	3	3	6	6	0.01

Table 3. Comparison between results of the neural network model and the FEM（MPa）

Input data set \ Equivalent stress		σ_1	σ_2	σ_3	σ_4	σ_5	σ_6
1	Y_0	381	570	365	193	126	116
	Y_1	408	599	392	210	134	123
	$\lvert(Y_1 - Y_0)/Y_0\rvert \times 100$	7.0%	5.1%	7.5%	8.5%	6.5%	6.0%
2	Y_0	383	567	381	203	138	126
	Y_1	386	587	405	203	145	132
	$\lvert(Y_1 - Y_0)/Y_0\rvert \times 100$	0.7%	3.6%	6.2%	0.1%	4.8%	5.1%
3	Y_0	385	576	379	198	129	119
	Y_1	378	575	361	196	130	112
	$\lvert(Y_1 - Y_0)/Y_0\rvert \times 100$	1.7%	0.1%	4.6%	0.8%	0.3%	5.5%
4	Y_0	398	591	427	222	186	186
	Y_1	408	598	432	225	202	200
	$\lvert(Y_1 - Y_0)/Y_0\rvert \times 100$	2.5%	1.2%	1.1%	1.6%	8.3%	7.7%
5	Y_0	415	596	462	261	220	213
	Y_1	414	596	449	242	201	195
	$\lvert(Y_1 - Y_0)/Y_0\rvert \times 100$	0.4%	0.1%	2.9%	7%	8.5%	8.5%

4 Conclusions

Through analysis of output characters of microminiature gear mechanism, combining with neural network modeling approach, a three-layer BP neural network model with 7 input nodes and output nodes is established. With the known finite element output data as a sample of training, through a comparative study of effects of different parameters of the network settings on the accuracy of the network model, the best network structure and parameters are determined. Explicit function expression of the nonlinear coupling relationship between manufacturing characteristics and mechanics property of microminiature gear mechanism is achieved. The results show that the neural network model can approximate the results of finite element analysis with errors of less than 9%.

References

1. Sun, Y.Z., Liang, Y.C., Cheng, K.: Micron and Middle-scale Machinery. Journal of Mechanical Engineering 40(5), 1–6 (2004) (in Chinese)
2. Chen, Z.C., Tang, R.Z.: Micro and Digital Manufacturing. Electrical and Mechanical 18(5), 1–3 (2001) (in Chinese)

3. Zhang, Z.J., Jin, X., Zhou, M.: Precision Micro-manufacturing Theory, Technology and Its Applications. Journal of Mechanical Engineering 43(1), 49–61 (2007) (in Chinese)
4. Jin, X.: Coupling Relationship between Manufacturing Characteristics and Mechanics Property of Micro-mechanical Systems. Beijing Institute of Technology, Beijing (2005) (in Chinese)
5. Li, S.Y.: Fuzzy Control, Neural Control and Intelligent Control Theory. Harbin Institute of Technology Press, Harbin (1996) (in Chinese)
6. Funahashi, K.I.: On the Approximate Realization of Continuous Mapping by Neural Nerworks. Neural Networks (1989) (in Chinese)
7. Zhang, J., Xiao, N.C., Cheng, J.: Application of Evidentization of Implicit Form based on Neural Network in Structural Reliability Analysis. Quarterly Mechanics 28(1), 135–141 (2007) (in Chinese)
8. Chen, J.: Guide Book of MATLAB. Electronics Industry Press, Beijing (2007)

Gas Concentration Forecasting Based on Support Vector Regression in Correlation Space via KPCA

Jian Cheng, Jian-sheng Qian, Guang-dong Niu, and Yi-nan Guo

School of Information and Electrical Engineering, China University of Mining and Technology, 221116, Xu Zhou, China
chjpaper@126.com

Abstract. Gas concentration is one of key factors which influence safe production of coal mine, and it is very important to forecast gas concentration accurately for ensuring the coal mine safety. A novel approach is presented to forecast gas concentration based on support vector regression (SVR) in correlation space reconstructed by kernel principal component analysis (KPCA). A two-stage architecture is proposed to improve its prediction accuracy and generalization performance for gas concentration forecasting. In the first stage, KPCA is adopted to extract features and obtain kernel principal components, so the correlation space of gas concentration is reconstructed according to the accumulative contribution ratio. Then, in the second stage, support vector regression (SVR) is employed for forecasting gas concentration, which hyperparameters are selected by adaptive chaotic cultural algorithm (ACCA). The approach is compared with the forecasting model in whole space of gas concentration. The simulation shows that SVR model in correlation space using KPCA performs much better than that without correlation analysis.

1 Introduction

Gas concentration is a very complex dynamic phenomenon which is affected by many factors, and is also one of key factors which influence safe production of coal mine. It is very important to forecast gas concentration accurately for ensuring the coal mine safety. Gas concentration is nonlinear and chaotic time series in essence, so it can be forecasted using the method of chaotic time series model[1]. In fact, there has inter-influencing correlation in all the gas concentrations, so we can observe the key gas concentration through other gas concentrations, in other words, the key gas concentration should be forecasted by the others. Therefore, a multi-inputs and single output model is built to realize gas concentration forecasting via correlation analysis.

Nonlinear and chaotic time series prediction is a practical technique which can be used for studying the characteristics of complicated systems based on recorded data[2]. As a result, the interests in chaotic time series prediction have been increased, however, most practical time series are of nonlinear and chaotic nature that makes conventional, linear prediction methods inapplicable[3]. Although the neural networks is developed in chaotic time series prediction[3, 4], some inherent drawbacks such as the multiple local minima problem, the choice of the number of hidden units

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 937 – 945, 2009.

and the danger of over fitting and so on, would make it difficult to put the neural networks into some practice.

In recent years, support vector regression (SVR) has been proposed as a novel technique in time series prediction[5]. SVR is a new approach established on the unique theory of the structural risk minimization principle to estimate a function by minimizing an upper bound of the generalization error via the kernel functions and the sparsity of the solution[6]. SVR usually achieves higher generalization performance than traditional neural networks that implement the empirical risk minimization principle in solving many machine learning problems[6, 7]. Another key characteristic of SVR is that training SVM is equivalent to solving a linearly constrained quadratic programming problem so that the solution of SVR is always unique and globally optimal[8].

In developing a SVR model for gas concentration forecasting, the first important step is correlation analysis. Kernel principal component analysis (KPCA) is one type of nonlinear principal component analysis (PCA) developed by generalizing the kernel method into PCA, which first maps the original input space into a high dimensional feature space using the kernel method and then calculates PCA in the high dimensional feature space[9]. The linear PCA in the high dimensional feature space corresponds to a nonlinear PCA in the original input space. KPCA can also extract nonlinear principal components up to the number of training data points[10].

The paper proposes a correlation space reconstruction method based on KPCA with correlation analysis in order to improve quality of input space and accuracy of gas concentration modeling. On the basis of KPCA, some kernel principal components are chosen according to their correlation degree. The restructured correlation space is then used as the inputs of SVR to solve gas concentration forecasting problems. By examining the gas concentration, the simulation shows that SVR by correlation analysis using KPCA performs much better than that without correlation analysis.

The rest of this paper is organized as follows. Section 2 presents the correlation space reconstruction of gas concentration. In section 3, the theory of SVR is presented. The architecture and algorithm of model are given in Section 4. Section 5 presents the results and discussions on the experimental validation. Finally, some concluding remarks are drawn in section 6.

2 Correlation Space Reconstruction of Gas Concentration

2.1 Correlation Space Reconstruction via KPCA

Given a set of centered input vectors x_k ($k = 1,2,\cdots,l$, and $\sum_{k=1}^{l} x_k = 0$), each of which is of m dimension $x_k = (x_k(1), x_k(2), \cdots, x_k(m))^T$ (usually $m < l$), The basic idea of KPCA is to map the original input vectors x_k into a high dimensional feature space $\Phi(x_k)$ and then to calculate the linear PCA in $\Phi(x_k)$. By mapping x_k into $\Phi(x_k)$ whose dimension is assumed to be larger than l , KPCA solves the eigenvalue problem (1).

$$\tilde{\lambda}_i \alpha_i = K\alpha_i,\ i = 1,2,\cdots,l \tag{1}$$

where K is the $l \times l$ kernel matrix. The value of each element of K is equal to the inner product of two high dimensional feature vector $\Phi(x_i)$ and $\Phi(x_j)$. That is, $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$. The advantage of using K is that one can deal with $\Phi(x_k)$ of arbitrary dimensionality without having to compute $\Phi(x_k)$ explicitly, as all the calculations of the dot product $\Phi(x_i) \cdot \Phi(x_j)$ are replaced with the kernel function $K(x_i, x_j)$. This means that the mapping of $\Phi(x_k)$ from x_k is implicit. $\tilde{\lambda}_i$ is one of the eigenvalues of K, satisfying $\tilde{\lambda}_i = l\lambda_i$. α_i is the corresponding eigenvector of K, satisfying $u_i = \sum_{j=1}^{l} \alpha_i(j)\Phi(x_j)$ ($\alpha_i(j)$, $j = 1,2,\cdots,l$, are the components of α_i).

For assuring u_i is of unit length, each α_i must be normalized using the corresponding eigenvalue by

$$\tilde{\alpha}_i = \alpha_i \Big/ \sqrt{\tilde{\lambda}_i},\ i = 1,2,\cdots,l \tag{2}$$

Based on the estimated $\tilde{\alpha}_i$, the principal components for x_k is calculated by

$$s_k(i) = u_i^T \Phi(x_k) = \sum_{j=1}^{l} \tilde{\alpha}_i(j) K(x_j, x_k),\ i = 1,2,\cdots,l \tag{3}$$

In addition, for making $\sum_{k=1}^{l} \Phi(x_k) = 0$, in (4) the kernel matrix on the training set K and on the testing set K_t are respectively modified by

$$\tilde{K} = (I - \frac{1}{l} 1_l 1_l^T) K (I - \frac{1}{l} 1_l 1_l^T) \tag{4}$$

$$\tilde{K}_t = (K_t - \frac{1}{l} 1_{l_t} 1_l^T K)(I - \frac{1}{l} 1_l 1_l^T) \tag{5}$$

where I is l dimensional identity matrix. l_t is the number of testing data points. 1_l and 1_{l_t} represent the vectors whose elements are all ones, with length l and l_t respectively. K_t represents the $l_t \times l$ kernel matrix for the testing data points.

From above equations, it can be found that the maximal number of principal components extracted by KPCA is l. If only the first several eigenvectors sorted in descending order of the eigenvalues are considered, the number of principal components in s_k can be reduced.

The popular kernel functions includes norm exponential kernel, Gaussian kernel function, sigmoid kernel, polynomial kernel, etc.. Gaussian kernel function is chosen in this paper.

$$K(x, x_k) = \exp(-\left\| x - x_k \right\| \Big/ \sigma^2) \tag{6}$$

2.2 Reducing the Dimension of the Input Space Based on Correlation Analysis

The kernel principal components s_k in feature space can be computed as Section 2.1, which are denoted by $H^1, H^2, \cdots, H^l$ ($H^i = (H_1^i, H_2^i, \cdots, H_l^i)^T$ for $i = 1,2,\cdots,l$) in this part for convenience.

Where H_j^i is the i^{th} principal component of j^{th} sample. The first q principal components are chosen such that their accumulative contribution ratio is big enough, which form reconstructed the phase space. As formula(7), training sample pairs for gas concentration modeling can be formed as below,

$$X = \begin{bmatrix} H_1^1 & H_1^2 & \cdots & H_1^q \\ H_2^1 & H_2^2 & \cdots & H_2^q \\ \vdots & \vdots & \ddots & \vdots \\ H_l^1 & H_l^2 & \cdots & H_l^q \end{bmatrix}, \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_l \end{bmatrix} \tag{7}$$

Modeling for gas concentration, which is based on KPCA correlation space reconstruction, is to find the hidden function $\tilde{f}$ between input X and output Y such that $y_i = \tilde{f}(x_i)$ 。 The above KPCA-based correlation space reconstruction choose the first q principal components successively according to their accumulative contribution ratio (their accumulative contribution ratio must be big enough so that they can stand for most information of original variables).

3 The Support Vector Regression

The support vector regression (SVR) approach is used to approximate an unknown function from a set of (input, output) dada $\{x_k, y_k\}_{k=1}^N$ with input data $x_k \in R^n$ and output data $y_k \in R$. Assuming that a set of basis functions $\{g_s(x)\}_{s=1}^l$ is given, there exists a family of functions that can be expressed as a linear expansion of the basis functions. That is, the problem of function approximation transforms into that finding the parameters of the following basis function linear expansion:

$$f(x,\theta) = \sum_{s=1}^{l} \theta_s g_s(x) + b \tag{8}$$

Where $\theta \in (\theta_1, \cdots, \theta_l)$ is parameter vector to be identified and b is a constant. Usually, the solution for this problem is to find f that minimizes the following empirical risk function:

$$R_{emp}[f] = \frac{1}{N} \sum_{k=1}^{N} L(y - f(x,y)) \tag{9}$$

Where $L(y-f(x,y))$ is the loss function measuring the different between the desired y and the estimated output $f(x,y)$ for given input x. Although having the advantage of being relatively easy to compute and being uniformly consistent hypothesis classes with bounded complexity, the attempt to minimize $R_{emp}[f]$ may directly lead to the phenomenon of overfitting and thus, poor generalization occurs in the case of a high model capacity in f. To reduce the overfitting effects, a regulation term is added into $R_{emp}[f]$, and (9) is modified as

$$R_{sv}[f] = R_{emp}[f] + \gamma \cdot \|\theta\|^2 \tag{10}$$

Where $\gamma > 0$ is a regular constant. The idea of adding the regulation term is to keep the weight vector θ as small as possible in the approximation process.

In SVR, the ε-insensitive loss function, which was first introduced in the originally support vector machines (SVM). The ε-insensitive loss function is to define the loss function as

$$L(e) = \begin{cases} 0, & for\,|e| \le \varepsilon \\ |e| - \varepsilon & otherwise \end{cases} \tag{11}$$

A ε zone is defined that if the e value within zone, the loss is zeros. Otherwise, the loss is the magnitude of the difference between the absolute value of e and ε zone.

By using the Lagrange multiplier method, the minimization of (9) leads to following dual optimization problem, minimize

$$\begin{aligned} Q(\alpha,\alpha^*) &= \varepsilon\sum_{u=1}^{N}(\alpha_u + \alpha_u^*) - \sum_{v=1}^{N} y_v(\alpha_v^* - \alpha_v) \\ &+ \frac{1}{2}\sum_{u,v=1}^{N}(\alpha_u^* - \alpha_u)(\alpha_v^* - \alpha_v)[\sum_{s=1}^{l} g_s(x_u) g_s(x_v)] \end{aligned} \tag{12}$$

Subject to the constraint

$$\sum_{u=1}^{N}\alpha_u^* = \sum_{u=1}^{N}\alpha_u\ ,\ 0 < \alpha_u, \alpha_u^* < \gamma\text{, for } u = 1,\cdots,N \tag{13}$$

Where α_u, α_v, α_u^* and α_v^* are the Lagrange multipliers. It was shown that the inner product of basis functions $g_s(x)$ in (12) is replaced by the kernel function

$$K(x_u, x_v) = \sum_{s=1}^{l} g_s(x_u) g_s(x_v) \tag{14}$$

Where x_u and x_v are any two input vector. The kernel function determines the smoothness properties of solutions and should reflect a prior knowledge on the data. In this paper, the Gaussian function is selected as kernel function that is defined as equation (6). Hence, the optimization of (12) is rewritten as

$$Q(\alpha,\alpha^*) = \varepsilon\sum_{u=1}^{N}(\alpha_u + \alpha_u^*) - \sum_{v=1}^{N} y_v(\alpha_v^* - \alpha_v) + \frac{1}{2}\sum_{u,v=1}^{N}(\alpha_u^* - \alpha_u)(\alpha_v^* - \alpha_v)K(x_u, x_v) \tag{15}$$

It was shown that the solution of SVR approach is in the form of the following linear expansion of kernel functions

$$f(x,\alpha,\alpha^*)=\sum_{k=1}^{N}(\alpha_k^*-\alpha_k)K(x,x_k)+b \tag{16}$$

Where the constant b be taken as

$$\frac{1}{2}\{\min_k(d_k-\sum_{k=1}^{N}(\alpha_k^*-\alpha_k)K(x,x_k))+\max_k(d_k-\sum_{k=1}^{N}(\alpha_k^*-\alpha_k)K(x,x_k))\} \tag{17}$$

Note that only some of $(\alpha_k^*-\alpha_k)$'s are not zeros and the corresponding vectors x_k 's are called the support vectors (SVs).

4 The Proposed Architecture and Algorithm

The Basic idea is to use KPCA to reconstruct the correlation space and apply SVR for gas concentration forecasting. Fig.1 shows how the model is built.

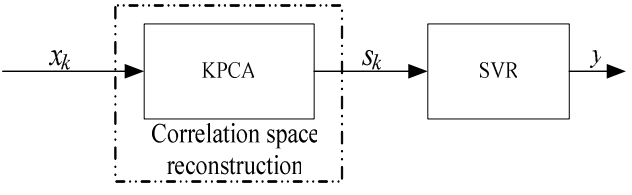


Fig. 1. The architecture of model for gas concentration forecasting

Up to here the process of forecasting gas concentration is completed. The detailed step of algorithm is illustrated as the following:

Step 1. For a gas concentration original input data x_k , KPCA is applied to assign the reduced dimension with the accumulative contribution ratio. The principal components s_k , whose number of dimension is less than x_k , is obtained

Step 2. s_k is used for the input vector of SA, and select appropriate threshold θ according to the satisfied result. The dimension of the final input space of forecasting model is assigned.

Step 3. In the reconstructed input space, the structure of SVR model is built, trained and tested by the partitioned data set respectively to determine the kernel parameters σ^2, γ and ε of SVR with Gaussian kernel function as shown in equation(6). Choose the most adequate SVR that produces the smallest error on the testing data set for gas concentration forecasting.

5 Results and Analysis of Experiment

From the Luling coal mine, located in north of Anhui province, 10 gas concentrations are researched, including ll816-working face T0 (y), ll816-working face T1 (x_1), ll 816-working face T2 (x_2),ll816-4#stone lane T2 (x_3),ll817-3#hole T1 (x_4),ll817-1#hole T1 (x_5),ll818-1#hole T1 (x_6), ll818-2#hole T1 (x_7),ll825-working face T0 (x_8),ll825-4#stone lane T2 (x_9). 2010 samples of each gas concentration are collected from online sensor underground after eliminating abnormal data in this study.

For choosing the optimal number of principal components in s_k , the σ^2 value from 0.1 to 100 are all investigated. So $\sigma^2 = 15$ is adopted as shown in Fig.2, and number of principal components is 5 through correlation analysis with $\theta = 0.90$ as shown in Table 1. The decreasing order of eigenvalues is x_1 , x_2 , x_3 , x_6 , x_5 , x_8 , x_7 , x_9 , x_4 , so the final input space (s_k) concludes x_1 , x_2 , x_3 , x_6 and x_5 .

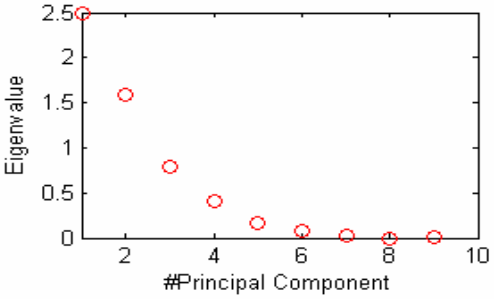


Fig. 2. The eigenvalues in KPCA with $\sigma^2 = 15$

Table 1. The accumulative contribution ratio of former 5 principal in KPCA

Former #Principal Component	1(x_1)	2(x_2)	3(x_3)	4(x_6)	5(x_5)
The accumulative contribution ratio	39.26%	59.51%	72.79%	85.57%	90.79%

Then the model based on SVR is constructed to realize the following:

$$y = f(s_k) = f(x_1, x_2, x_3, x_6, x_5) \tag{18}$$

The prediction performance is evaluated using by the root mean squared error (*RMSE*) and the normalized mean square error (*NMSE*) as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{19}$$

$$NMSE = \frac{1}{\delta^2 n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \ , \ \delta^2 = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2 \tag{20}$$

where n represents the total number of data points in the test set, y_i, $\hat{y}_i$, $\bar{y}$ are the actual value, prediction value and the mean of the actual values respectively.

From the gas concentration series $\{y, s_k\}$, we extracted 1000 input-output data pairs. The first 500 pairs is used as the training data set, the second 500 pairs is used as testing data set for testing the predictive power of the model. To assure there is the best prediction performance in the SVR model, through adaptive chaotic cultural algorithm (ACCA) to determine the hyperparameters of SVR, σ^2, γ and ε are, respectively, fixed at 0.45, 15 and 0.04. The simulation results are shown in Table 2.

Table 2. The results of gas concentration forecasting

Model		SVR	KPCA+SVR
#Principal Component		9	5
RMSE	Training	0.0147	0.0105
	Testing	0.0154	0.0116
NMSE	Training	0.0382	0.0297
	Testing	0.0765	0.0433

From Table 2, it can be observed that the KPCA+SVR forecast more closely to actual values than SVR, So there are correspondingly smaller absolute prediction errors in the KPCA+SVR than the SVR.

6 Conclusions

This paper describes a novel methodology, a correlation analysis SVR based on KPCA, to model and predict gas concentration series. The correlated SVR model is developed by correlation analysis with KPCA algorithm in a two-stage architecture inspired by the idea of reducing dimension modeling method. Through this investigation, there are several advantages in the correlative SVR. In the first, KPCA is a nonlinear PCA by generalizing the kernel method into linear PCA, which is adopted to extract features of gas concentration, reflecting its nonlinear characteristic fully. Secondly, on the basis of KPCA, the correlative dimension of the input space of gas concentration is reduced according to the accumulative contribution ratio, so the model precision is improved greatly. The proposed model has been evaluated by real gas concentration from coal mine. Its superiority is demonstrated by comparing it with the model without correlation analysis. The simulation results show that the proposed model in the paper can achieve a higher prediction accuracy and better generalization performance than that with out correlation analysis.

On the other hand, there are some issues that should be investigated in future work, such as how to ascertain the accumulative contribution ratio of KPCA and confidence

threshold of correlation analysis which affect deeply the performance of the whole model, how to construct the kernel function and determine the optimal kernel parameters, etc.

Acknowledgements

This research is supported by National Natural Science Foundation of China under grant 70533050, Young Science Foundation of CUMT under grant 2006A010.

References

1. Cheng, J., Bai, J.Y., Qian, J.S., Li, S.Y.: Short-term forecasting method of coal mine gas concentration based on chaotic time series. Journal of China University of Mining and Technology 10(2), 231–235 (2008) (in chinese)
2. Cao, L.J., Tay, F.E.H.: Support vector machine with adaptive parameters in financial time series forecsting. IEEE Transactions on Neural Networks 14(6), 1506–1518 (2003)
3. Han, M., Xi, J.H., Xu, S.G., Yin, F.L.: Prediction of chaotic time series based on the recurrent predictor neural network. IEEE Transactions on Signal Processing 52(12), 3409–3416 (2004)
4. Leung, H., Lo, T., Wang, S.: Prediction of noisy chaotic time series using an optimal radial basis function neural network. IEEE Transactions on Neural Networks 12(5), 1163–1172 (2001)
5. Wei, X.K., Li, Y.H., et al.: Analysis and Applications of Time Series Forecasting Model via Support Vector Machines. System Engineering and Electronics 27(3), 529–532 (2005)
6. Vapnik, V.N.: An Overview of Statistical Learning Theory. IEEE Transactions Neural Networks 10(5), 988–999 (1999)
7. Chuang, C.C.: Fuzzy weighted support vector regression with a fuzzy partition. IEEE Transactions on System, Man, and Cybernetics 37(3), 630–640 (2007)
8. Li, D., Mersereau, R.M., Simske, S.: Blind image deconvolution through support vector regression. IEEE Transactions on Neural Networks 18(3), 931–935 (2007)
9. Twining, C.J., Taylor, C.J.: The use of kernel principal component analysis to model data distributions. Pattern Recognition 36(1), 217–227 (2003)
10. Jade, A.M., Srikanth, B., Jayaraman, V.K., Kulkarni, B.D., Jog, J.P., Priya, L.: Feature extraction and denoising using KPCA. Chemical Engineering Science 58(19), 4441–4448 (2003)

A CMMS-Based Formal Conceptual Modeling Approach for Team Simulation and Training

Jian Wang and Hongwei Wang

Systems Engineering Institute, Huazhong University of Science and Technology, Wuhan 430074, China
Key Laboratory of Image Processing and Intelligent Control, Wuhan Hubei 430074, China
National Simulation Training Center for National Economy Mobilization, Wuhan Hubei 430074, China
wj0826_can@126.com, hwwang@mail.hust.edu.cn

Abstract. For the conceptual modeling of team simulation and training based on EBAT (Event-Based Approach to Training), it is crucial to solve the problem of effective communication between subject matter experts, training experts and simulation technologists, and the problem of conceptual modeling quality and reuse. By analyzing modeling elements of team training, a layered logical structure of the simulation and training conceptual model based on CMMS (Conceptual Model of the Mission Space) and a formal description approach are proposed. Based on the workflow patterns and interactive features of team training, the event causality patterns and formal description approach are presented for the process of event response. The results of the formal conceptual modeling of training cases show the validity of the proposed formal conceptual modeling approach.

Keywords: Team simulation and training, EBAT, Conceptual modeling, CMMS, Workflow patterns.

1 Introduction

Team training is an approach with some training strategies and instruments to enhance the trainee's knowledge, kills, and attitudes related to teamwork[1]. Because team simulation and training offers a realistic, safe, cost-effective, and flexible environment in which to learn the requisite competencies for the job, commercial aviation and the military have invested heavily in the use of which[2]. For the simulation-based training environments, Oser, et al.[3] propose a event-based approach to training (EBAT), which is a team training approach focuses on scenario-based training design. The EBAT has been tested empirically and demonstrated in a variety of team simulation and training environments[4]. In the fields of research on team simulation and training based on EBAT, there is a lack of research on the problem of simulation and training formal conceptual modeling.

In the military simulation field, there are some researches on the formal simulation conceptual modeling. Wang, et al.[5] propose an abstract description of task, space,

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 946 – 955, 2009.

mission space, military mission space, CMMS. He, et al.[6] present a formal description of dynamic conceptual model based on the elements of task, decision, synchronization, control, condition, which considers the execution mechanisms of CMMS. He, et al.[7] propose a lightweight formal description and verification method of CMMS based on the elements of task, entity, relationship, event, and condition. The proposed approaches mainly focus on the description of modeling elements and the relationships of which. And, the event causality has not been considered yet, which includes the sequence and trigger conditions of events.

In this paper, Section 2 analyses the conceptual modeling elements of team simulation and training. A layered logic structure of conceptual model is proposed in section 3. The formal description approach of conceptual model is presented in section 4. Section 5 implements the formal conceptual modeling of training cases. The conclusion follows in section 6.

2 Logic Structure of Conceptual Model

The layered logical structure of simulation and training conceptual model is shown in Fig.1.

Based on the core concept of CMMS, a layered conceptual model is designed which includes the task space, the content space and the response space. The task space includes the task entities and the objective entities, reflecting the overall objectives of training. The content space includes the scenario entities for each training objective, and the related assessment and feedback entities. The response space describes the

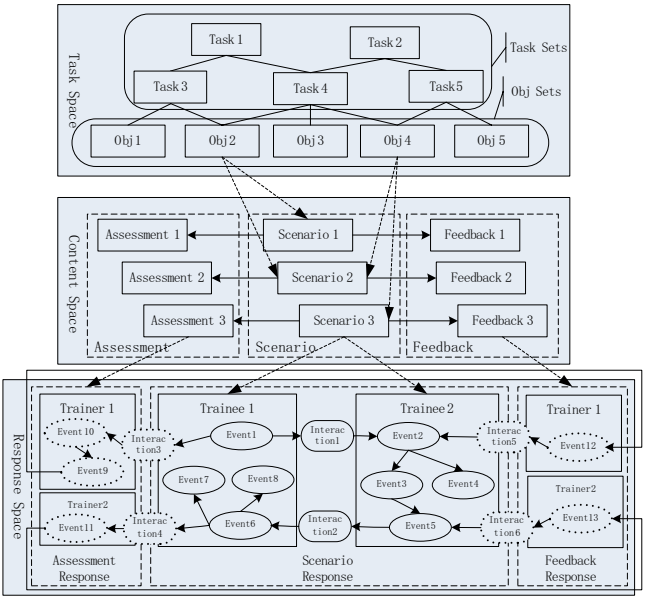


Fig. 1. Logic structure of simulation and training conceptual model

scenario entities, the assessment and feedback entities in the content space with the responder entity, the event and the interaction. The task space can be divided into the sub-spaces of task and objective, and the content space and the response space can be divided into three sub-spaces of scenario, assessment and feedback.

3 Formal Description Approach

The formal description based on the layered model, can not only describe more clearly the layered structure of the training task, content and the event response process entities, and the event response processes of these training content. Moreover, it can fully realize the modular description and the reuse of the description of training task, the content and the event response process.

1. Task space

The formal description of task space is mainly aimed at the characteristics of training task and objective entities, and the relationship between the entities. The detail descriptions are as follow:

$$\begin{aligned} CM_T &= \{T_t,\ O_o,\ SET_T,\ SET_O,\ \delta_{TT},\ \delta_{TO}\}\ \ t = 1, 2, ..., T_{\max},\ o = 1, 2, ..., O_{\max}. \\ \delta_{TT} &= \left\{\delta_{tt} \middle| \delta_{tt} = (T_{\sup}, T_{\inf}), T_{\sup}, T_{\inf} \in SET_T, T_{\sup} \neq T_{\inf}\right\}.\ \ \delta_{TO} = \left\{\delta_{to} \middle| \delta_{to} = (T_t, O_o), T_t \in SET_T, O_o \in SET_O\right\}. \end{aligned} \tag{1}$$

CM_T: The task space. T_t: The task entity, t is the task number, $T_{\max}$ is the total number of tasks. O_o: The objective entity, o is the objective number, $O_{\max}$ is the total number of objective. SET_T: The task set, $T_t \in SET_T$. SET_O: The objective set, $O_o \in SET_O$. δ_{TT}: The set of task relationship. δ_{tt} is the superior subordinate relationship of the task, $T_{\sup}$ is the higher level task, $T_{\inf}$ is the inferior level task. A task can have a number of inferior level tasks, the task can also belong to a number of higher tasks. δ_{TO}: The set of the relationship between the objectives and tasks. δ_{to}: The affiliation relationship of the tasks and objectives. A task can have multiple objectives; a number of objectives can also be attached to a task.

2. Content space

The formal description of content space is mainly aimed at the characteristics of scenario, assessment and feedback entities, the relationship between the entities.

$$\begin{aligned} CM_C &= \{S_s,\ E_e,\ F_f,\ SET_S,\ SET_E,\ SET_F,\ \delta_{SE},\ \delta_{SF}\}\ \ s = 1, 2, ..., S_{\max},\ e = 1, 2, ..., E_{\max},\ f = 1, 2, ..., F_{\max}. \\ \delta_{SE} &= \left\{\delta_{se} \middle| \delta_{se} = (S_s, E_e), S_s \in SET_S, E_e \in SET_E\right\}.\ \ \delta_{SF} = \left\{\delta_{sf} \middle| \delta_{sf} = (S_s, F_f), S_s \in SET_S, F_f \in SET_F\right\}. \end{aligned} \tag{2}$$

CM_C: The content space. S_s: The scenario entity, s is the scenario number, $S_{\max}$ is the total number of scenarios. E_e: The assessment entity, e is the assessment number, $E_{\max}$ is the total number of assessment. F_f: The feedback entity, f is the feedback number, $F_{\max}$ is the total number of feedback. SET_S: The set of the scenario, $S_s \in SET_S$. SET_E: The set of assessment, $E_e \in SET_E$. SET_F: The set of feedback, $F_f \in SET_F$. δ_{SE}: The

set of the relationship of the scenario and assessment. δ_{se} is the relationship of the scenario and assessment, whose mapping is one-to-one. δ_{SF} is the set of the relationship between the scenario and assessment. δ_{sf} is the relationship between the scenario and the feedback , whose mapping is also one-to-one.

3. Response space

The formal description of response space is divided into the entities relationship and the event response processes.

The relationship of entities

The description of entities relationship abstracts the event response processes of the scenario, assessment and feedback into the process entities, and describes the characteristics of the entities of the scenario, assessment and feedback processes, the relationship between the entities, and the relationship between the entities of the content space and the response space.

$$CM_A = \{AS_{as}, AE_{ae}, AF_{af}, SET_{AS}, SET_{AE}, SET_{AF}\} \ as = 1, 2, ..., AS_{\max}, \ ae = 1, 2, ..., AE_{\max}, \ af = 1, 2, ..., AF_{\max}. \quad (3)$$

CM_A : The response space. AS_{as} : The scenario process entity, as is the scenario process number, $AS_{\max}$ is the total number of the scenario processes. AE_{ae} : The assessment process entity, ae is the number of the assessment process, $AE_{\max}$ is the total number of the assessment processes. AF_{af} : The feedback process entity, af is the number of the feedback process, $AF_{\max}$ is the total number of feedback processes. SET_{AS} : The set of the scenario processes, $AS_{as} \in SET_{AS}$. SET_{AE} : The set of evaluation processes, $AE_{ae} \in SET_{AE}$. SET_{AF} : The set of feedback processes, $AF_{af} \in SET_{AF}$.

The event response process

The formal description of event response process mainly describes the event response and interaction processes of the scenario, assessment and feedback, namely the responders to carry out the training content, the event response and interaction processes in the layered logical structure of the simulation and training conceptual mode.

① The event causality patterns

In order to describe the event causality, the workflow patterns are adopted to extend the formal conceptual modeling approach based on CMMS. The event causality patterns includes the inside causality patterns and the interaction causality patterns. The inside causality patterns is aimed at the event response processes responded by only one responder, without the interaction with others. The interaction causality patterns refer to the event response processes across many responders; there are interactions in the event response process. The type of event causality patterns is shown in Table 1.

It can be seen from Table 1, in fact, the inside causality patterns choose part of the workflow patterns, and merge many workflow patterns, and describe the causality through the trigger conditions for the front and sequence events. In the interaction causality patterns, events connect through the interaction, namely to trigger the sequence events not only depends on whether the front events response have

completed, but also depends on whether the front events have sent the interaction. Meanwhile, the front and the sequence events may be responded by different responders. The implementation process of the event response described by the causality patterns is shown in Figure 2 and Figure 3.

Table 1. Event causality patterns

Patterns Type	Patterns Name	Description
Inside causality patterns	Sequence	Sequence pattern in workflow patterns.
	Split	Parallel Split(AND-Split), Muti-Choice(OR-Split)、 Exclusive Choice(XOR-Split)patterns in workflow patterns.
	Join	Synchronization(AND-Join), Simple Merge(OR-Join), Discriminator(XOR-Join)patterns in workflow patterns.
Interaction causality patterns	Sequence Interaction	Sequence-I.
	Split Interaction	AND-Split-I、 OR-Split-I、 XOR-Split-I.
	Join Interaction	AND-Join、 OR-Join-I、 XOR-Join-I.

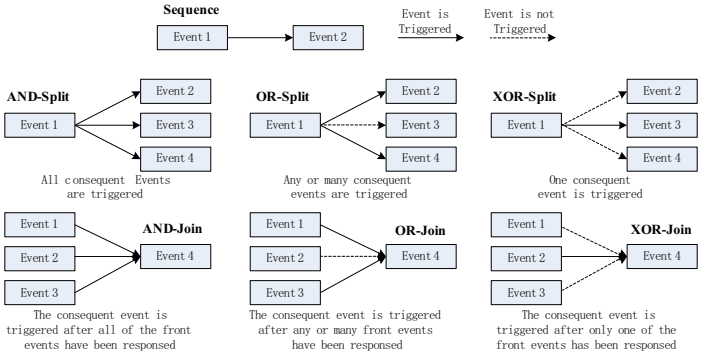


Fig. 2. Inside causality patterns

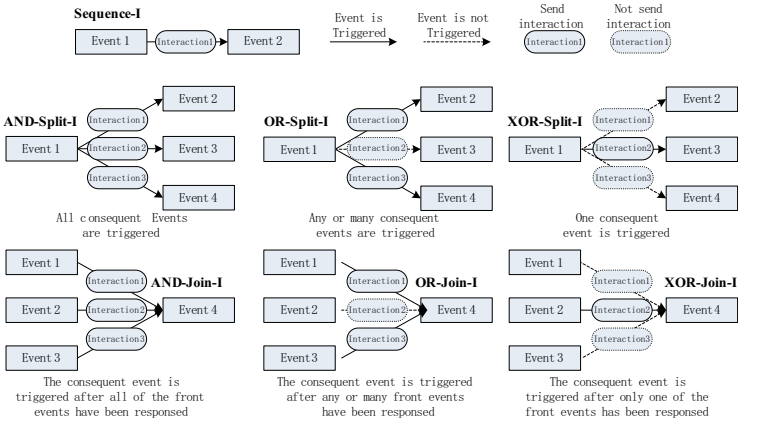


Fig. 3. Interaction causality patterns

② The description of the causality patterns

The description of causality includes the common expressions and the patterns' bound parameters. The common expressions describe the basic elements of various types of patterns; the different types of patterns are limited by different parameters. The common expressions are as follows:

$$Pattern(type) = \{SET_{PRE},\ SET_{POST},\ SET_{EXR},\ SET_I,\ R_{TRIG},\ R_{EXR-PRE},\ R_{EXR-POST},\ R_I\},$$
$$type = \{Sequence,\ Sequence-I,\ Split,\ Split-I,\ Join,\ Join-I\}. \tag{4}$$

$Pattern(type)$: The event causality pattern, $type$: the type of the pattern. SET_{PRE} : The set of the front events. SET_{POST} : The set of sequence events. SET_{EXR} : The set of responders. SET_I : The set of the interactions. R_{TRIG} : The set of event trigger conditions, namely the vector of the causality. $R_{EXR-PRE}$: The relationship vector between the front events and the responders. $R_{EXR-POST}$: The relationship vector between the sequence events and the responders. R_I : The relationship vector between the front events and the interactions.

The bound parameters of the patterns are shown in Table 2.

Table 2. Patterns' bound parameters

Patterns Type	Parameters
Sequence	$\lvert SET_{PRE}\rvert = 1,\ \lvert SET_{POST}\rvert = 1,\ \lvert SET_{EXR}\rvert = 1,\ SET_I = \varnothing,\ R_{EXR-PRE} = \varnothing,\ R_{EXR-POST} = \varnothing,\ R_{TRIG} = \varnothing,\ R_I = \varnothing$ $\lvert xxx\rvert$ is the amount of elements in the set.
Sequence-I	$\lvert SET_{PRE}\rvert = 1,\ \lvert SET_{POST}\rvert = 1,\ \lvert SET_{EXR}\rvert = 2,\ \lvert SET_I\rvert = 1, R_{EXR-PRE} = (Exr_{pre}),\ R_{EXR-POST} = (Exr_{post}),\ R_{TRIG} = \varnothing,\ R_I = \varnothing$ Exr_{pre} , Exr_{post} are the responders of front and consequent events. $Exr_{pre} \neq Exr_{post}$ means the interaction should occur between different responders.
Split	$\lvert SET_{PRE}\rvert = 1,\ \lvert SET_{POST}\rvert = n\ (n > 1),\ \lvert SET_{EXR}\rvert = 1,\ SET_I = \varnothing, R_{EXR-PRE} = \varnothing,\ R_{EXR-POST} = \varnothing,\ R_{TRIG} = (r_1\ r_2\ \dots\ r_n),\ R_I = \varnothing$ r_i is the flag of whether event i should be triggered, when the response state of front event is i . r_i=1 means to trigger the event. AND-Split: R_{TRIG}=(1 1 … 1) . OR-Split: R_{TRIG}=(1 0 … 1) . XOR-Split: R_{TRIG}=(0 1 … 0) .
Split-I	$\lvert SET_{PRE}\rvert = 1,\ \lvert SET_{POST}\rvert = n\ (n > 1),\ \lvert SET_{EXR}\rvert = k\ (1 < k \le n+1),\ SET_I = l\ (n > l > 1),$ $R_{EXR-PRE} = (Exr_{pre}),\ R_{EXR-POST} = (Exr_1\ Exr_2\ \dots\ Exr_k),\ R_{TRIG} = (r_1\ r_2\ \dots\ r_n),\ R_I = (r'_1\ r'_2\ \dots\ r'_n)$ Exr_{pre} , Exr_i are the responders of front and consequent events, different consequent events can be responded by the same responder. $Exr_{pre} \neq Exr_i$ means the interaction should occur between different responders. r_i is the flag of whether event i should be triggered, when the response state of front event is i . r_i=1 means to trigger the event. AND-Split: R_{TRIG}=(1 1 … 1) . OR-Split: R_{TRIG}=(1 0 … 1) . XOR-Split: R_{TRIG}=(0 1 … 0) . r'_i is the flag of whether interaction r'_i should be sent, when the response state of front event is i . r_i=1 means to trigger the event. Different events can send the same interaction.
Join	$\lvert SET_{PRE}\rvert = m\ (m > 1),\ \lvert SET_{POST}\rvert = 1,\ \lvert SET_{EXR}\rvert = 1,\ SET_I = \varnothing, R_{EXR-PRE} = \varnothing,\ R_{EXR-POST} = \varnothing,\ R_{TRIG} = (r_1\ r_2\ \dots\ r_m),\ R_I = \varnothing$ r_i is the flag of whether to trigger consequent events should wait for front event i has been responded. r_i=1 means need to wait. AND-Join: R_{TRIG}=(1 1 … 1) . OR-Join: R_{TRIG}=(1 0 … 1) . XOR-Join: R_{TRIG}=(0 1 … 0) .

Table 2. (*continued*)

Patterns Type	Parameters
Join-I	$\lvert SET_{PRE}\rvert = m\ (m > 1)$, $\lvert SET_{POST}\rvert = 1$, $\lvert SET_{EXR}\rvert = k\ (1 < k \le n+1)$, $SET_I = l\ (m > l > 1)$, $R_{EXR-PRE} = (Exr_1\ \ Exr_2\ \ \dots\ \ Exr_k)$, $R_{EXR-POST} = (Exr_{post})$, $R_{TRIG} = (r_1\ \ r_2\ \ \dots\ \ r_m)$, $R_I = (r'_1\ \ r'_2\ \ \dots\ \ r'_m)$ Exr_i, Exr_{post} are the responders of front and consequent events, different front events can be responded by the same responder. $Exr_i \neq Exr_{post}$ means the interaction should occur between different responders. r_i is the flag of whether event i should be triggered, when the response state of front event is i. r_i=1 means to trigger the event. AND-Join: R_{TRIG}=(1 1 … 1). OR-Join: R_{TRIG}=(1 0 … 1). XOR-Join: R_{TRIG}=(0 1 … 0). r'_i is the flag of whether interaction r'_i should be sent, when front event i has been responded. r_i=1 means to trigger the event. Different events can send the same interaction. Different events can send the same interaction.

③ The description of event response process

The description of event response process describes the relationship of entities in detail, namely describe the set of the event causality patterns and the combination relationship of the patterns included in the event response process entities. The description of event response process includes the set of events, the set of responders, the set of interactions, the set of causality patterns, as well as the set of processes.

$$CM_{A'} = \{A_a,\ I_i,\ Exr_e,\ Pattern(type)_p,\ A\text{-}PROCESS_{ap},\ SET_A,\ SET_{EXR},\ SET_I,\ SET_{A-PATTERN},$$

$$SET_{A-PROCESS},\ \delta_{PATTERN-PROCESS}\}\ a=1,2,...,A_{\max},\ i=1,2,...,I_{\max},$$

$$e = 1, 2, ..., E_{\max},\ \ p = 1, 2, ..., P_{\max},\ \ ap = 1, 2, ..., AP_{\max}.$$

$$\delta_{PROCESS-PATTERN} = (A\text{-}PROCESS_{ap},\ \delta_{PATTERN-PATTERN}) \tag{5}$$

$$\delta_{PATTERN-PATTERN} = \left\{ \delta_{pp} \middle| \begin{array}{l} \delta_{pp} = (Pattern(type)_{pre}, \\ Pattern(type)_{post}), \\ Pattern(type)_{pre}, \\ Pattern(type)_{post} \in SET_{A-PATTERN} \end{array} \right\}.$$

$CM_{A'}$: The response process of events. A_a: The event entity, a is the number of event, $A_{\max}$ is the total number of events. I_i: The interaction entity, i is the number of interaction, $I_{\max}$ is the total number of interactions. Exr_e: The responder entity, e is the number of responder, $E_{\max}$ is the total number of responders. $Pattern(type)_p$: The event causality pattern entity, p is the number of pattern entity, $P_{\max}$ is the total number of pattern entities. $A\text{-}PROCESS_{ap}$: The process entity, ap is the number of process, $AP_{\max}$ is the total number of responders. SET_A: The set of events, $A_a \in SET_A$.

SET_{EXR}: The set of responders, $Exr_e \in SET_{EXR}$. SET_I: The set of interactions, $I_i \in SET_I$.

$SET_{A-PATTERN}$: The set of patterns, $Pattern(type)_p \in SET_{A-PATTERN}$. $SET_{A-PROCESS}$: The set of processes, $A\text{-}PROCESS_{ap} \in SET_{A-PROCESS}$. $\delta_{PROCESS-PATTERN}$: The set of processes and patterns. δ_{pp}: The relationship of the patterns, $Pattern(type)_{pre}$ is the front process, $Pattern(type)_{post}$ is the sequence process. The relationship between process patterns is

many-to-many. In the description of event response process, the process does not distinguish between scenario, assessment and feedback processes. That is the sets of various processes in the entity relationship are described as different types of process sets.

4 Training Cases Modeling

Taking the provincial and municipal institutions as the example, the training needs of the organization establishment are described as follows:

First of all, the highest headquarters issues the mobilization startup order in advance, and then starts the process of provincial and municipal organization establishment. Secondly, the mobilization headquarters of the provincial and municipal establish the mobilization headquarters in wartime, which is divided into two types: to establish the wartime institutions by the national defense mobilization command staff, or to establish the wartime institutions by the national defense and economy mobilization command staff together. Third, the mobilization headquarters at all levels organize the subordinate headquarters to hold a video conference, announce the wartime adjustment order. Finally, the mobilization headquarters at all levels send a notice of completion of the organization establishment.

The formal description of task space, content space, and response space is shown in Table 3. As the description processes of scenario, assessment and feedback processes are the same, and scenario 3 is the most complex one, so only the description of scenario 3 is presented.

Table 3. The formal description of training cases

Parameters	Values
	Task space
SET_T	1:Establish wartime institutions, 2:Establish provincial wartime institutions, 3:Establish municipal wartime institutions.
SET_O	1:Issue an order of mobilization startup, 2:Establish institutions rapidly, 3:Hold a video conference.
δ_{TT}	(1,2),(1,3).
	Content space
SET_S	1:Issue an order of mobilization startup, 2:Establish institutions by national defence mobilization commander, 3:Establish institutions by national defence and economy mobilization commander, 4:Hold a video conference.
SET_E	1:Veracity of the order of mobilization startup, 2:Rationality and cost of institutions established by national defence mobilization commander, 3:Rationality and cost of institutions established by national defence and economy mobilization commander, 4:Rationality of video conference.
SET_F	1:Feedback of the veracity of the order of mobilization startup, 2:Feedback of the rationality and cost of institutions established by national defence mobilization commander, 3:Feedback of rationality and cost of institutions established by national defence and economy mobilization commander, 4:Feedback rationality of video conference.
δ_{SE}	(1,1),(2,2),(3,3),(4,4).
δ_{SF}	(1,1),(2,2),(3,3),(4,4).

Table 3. (*continued*)

Parameters	Values
	Response space
	Basis sets
SET_A	1:Issue an order of mobilization startup, 2:Receive an order of mobilization startup, 3: Establish institutions, 4:Report the results of institutions establishment, 5:Receive the results of institutions, 6:Collect the results of institutions establishment, 7:Send a notice of institutions establishment finish, 8: Receive a notice of institutions establishment finish, 9:Send a notice of video conference, 10:Receive a notice of video conference, 11:Enter the video conference.
SET_{EXR}	1:Provincial national defence mobilization commander, 2:Municipal national defence mobilization commander, 3: Provincial economy mobilization commander, 4:Municipal economy mobilization commander.
SET_I	1: Order of mobilization startup, 2:Results of institutions, 3:Summary results of institutions, 4:Notice of video conference.
	Event causality patterns ($SET_{A-PATTERN}$ - $Pattern(type)$ **)**
Sequence	Issue an order of mobilization startup: 1.1:({1},{1},{1},null,null,null,null,null) (startup action) Establish institutions: 1.2:({1},{3},{1},null,null,null,null,null),1.3:({2},{3},{2},null,null,null,null,null) 1.4:({2},{3},{3},null,null,null,null,null),1.5:({2},{3},{4},null,null,null,null,null) Enter a video conference: 1.6:({9},{11},{1},null,null,null,null,null), 1.7:({9},{11},{2},null,null,null,null,null) 1.8:({10},{11},{3},null,null,null,null,null), 1.9:({10},{11},{4},null,null,null,null,null) Report the results of institutions establishment: 1.9: ({3},{4},{3},null,null,null,null,null), 1.10: ({3},{4},{4},null,null,null,null,null)
Sequence-I	Issue an order of mobilization startup: 2.1:({1},{2},{1,2},{1},null,(1),(2),null),2.2:({1},{2},{1,3},{1},null,(1),(3),null) 2.3:({1},{2},{1,4},{1},null,(1),(4),null) Report the results of institutions establishment: 2.4:({4},{5},{1,3}},{2},null,(3),(1),null),2.5:({4},{5},{2,4}},{2},null,(4),(2),null) Send a notice of institutions establishment finish: 2.6:({7},{8},{1, 3},{3},null,(1),(3),null),2.7:({7},{8},{2, 4},{3},null,(2),(4),null) Send a notice of video conference: 2.8:({9},{10},{1, 3},{4},null,(1),(3),null), 2.9:({9},{10},{2, 4},{4},null,(2),(4),null)
Split	Collect the results of institutions establishment: 3.1:({6},{7,9},{1},(1,1),null,null,null,null),3.2:({6},{7,9},{2},(1,1),null,null,null,null)
Join	Collect the results of institutions establishment: 4.1:({3,5},{6},{1},(1,1),null,null,null,null),4.2:({3,5},{6},{2},(1,1),null,null,null,null)
	Event response process
$SET_{A-PROCESS}$	1:Issue a provincial order of mobilization startup, 2:Establish provincial institutions, 3:Hold a provincial video conference, 4:Establish municipal institutions, 5: Hold a municipal video conference.
$\delta_{PATTERN-PROCESS}$	Establish provincial wartime institutions (as an example): (2, {(1.1, 1.2), (1.1, 2.2), (2.2, 1.4) , (1.4, 1.9), (1.9, 2.4), (2.4, 4.1), (1.2, 4.1) , (4.1, 3.1), (3.1, 1.6), (3.1, 2.8) , (3.1, 2.6), (2.8, 1.8) }).
δ_{SAs}	(3,1),(3,2),(3,3),(3,4),(3,5).

5 Conclusions

For the problem of conceptual modeling of team simulation and training based on EBAT, the modeling elements and a layered logical structure of conceptual model are proposed, according to the conceptual modeling approach of CMMS. On this basis, the formal description approach is presented. The results of conceptual modeling of training cases show that the proposed formal description approach can enhance the communication between subject matter experts, training specialists and simulation technologists, and can improve the quality and reuse of conceptual models efficiently.

References

1. Kristi, D., Willbanks, B.S.: Relationship of Team Training Components to Perceptions of Team Performance. University Of North Texas, 1–46 (2003)
2. Salas, E., Wilson, K.A., Burke, C.S., Priest, H.A.: Using Simulation-Based Training to Improve Patient Safety: What Does It Take? Jt Comm. J. Qual. Saf. 31, 363–371 (2005)
3. Fowlkes, J.E., Dwyer, D.J., Oser, R.L., Salas, E.: Event-based Approach to Training (EBAT). International Journal of Aviation Psychology 8, 209–221 (1998)
4. Chapman, R.J., Ryder, J., Bell, B., Wischusen, D., Benton, D.: An Integration of Cognitive Models and Virtual Environments for Mobile Scenario Based Training. In: Human Factors and Ergonomics Society Annual Meeting Proceedings, Special Sessions, pp. 2094–2098 (2007)
5. Wang, X.L., Guo, Q.S., Ding, S.Y.: Discussion and Formal Abstract Description of Conceptual Model of Mission Space. Journal of System Simulation 15, 1408–1409 (2003)
6. He, X.Y., Xu, P.D., Sha, J.C.: Prelim Inary Study on Execution of CMMS. Computer Simulation 22, 78–80 (2005)
7. He, X.Y., Xu, P.D., Sha, J.C.: Lightweight Formal Verification Method of Mission Space Conceptual Model Research. Journal of System Simulation 18, 1108–1109 (2006)

SOSBP: An Efficient Bargaining Protocol for E-Market

Liu Hong[1,3], Haigang Song[2], Xueguang Chen[1,3], and Qihua Zhang[4]

[1] Institute of System Engineering, Huazhong University of Science and Technology, Wuhan 430074, P. R. China
[2] Basic Research Service of the Ministry of Science and Technology of the P. R. China, Beijing 100862, P. R. China
[3] Key Lab. for Image Processing & Intelligent control, Huazhong University of Science and Technology, Wuhan 430074, P. R. China
[4] Semiconductor Manufacturing International Corporation, Shanghai 201203, P. R. China
newtorrent@sina.com.cn

Abstract. Alternating-Offer Protocol (AOP) is the most predominant way for one-to one bargaining. In this paper, we provide an alternative protocol, Sealed-Offer Synchronous Bargaining Protocol (SOSBP), which can be useful for the automated bargaining between self-interested agents for Open E-Marketplaces. SOSBP introduces an intermediary agent who monitors the bargaining process and hides buyer's offers from seller, and vice versa. Using SOSBP, agents could free submit their offers for evaluating because the intermediary agent has no preference on both buyer and seller. We design some mechanisms to facilitate the success of bargaining, such as Minimum Concession, Concession Reward, Surplus Quote and etc, and present the basic framework of SOSBP. From the result of strategy analysis and experimental study, we concluded that SOSBP has better performance than AOP when agents negotiate under specified situation.

Keywords: Automated negotiation, MAS, Bargaining strategy, SOSBP, Alternating-Offer Protocol.

1 Introduction

In MAS, when autonomous agents were applied into the E-Commerce or E-marketplace, the problem of how to negotiate effectively and efficiently between autonomous agents becomes an essential issue [1, 2]. Incorporating with theoretic results derived from the bargaining model in economics, MAS experts and scholars have endeavored much to explore and design special mechanism for negotiations in agent world, see [3-5]. Despite various different assumptions made for analysis convenience, there exists one common feature in traditional bargaining protocol---- almost all literatures on bargaining model use AOP, by which agents make offer and counter-offer in turn [6]. To avoid private info losing in using AOP, in this paper, we present a novel bargaining mechanism: Sealed-Offer Synchronous Bargaining Protocol (SOSBP).

This paper was organized as follows: in section 2, we will describe main activities occurred under SOSBP. This includes the initiation of bargaining, what kind of

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 956 – 966, 2009.

information being exchanged in the process, in what situation the bargaining reached an agreement or terminated, etc. What kinds of strategy should the agents employ are analyzed in section 3. Section 4 discussed the results of our experimental study; summarization and some future research avenues were concluded in the last section.

2 General Description of SOSBP

2.1 Symbols and Definitions

We will adopt the similar symbols introduced in [4] for our study.

Let B and S denote respectively the negotiating agent Buyer and Seller, and M be the fore-mentioned intermediary agent. $p^t_{a\to M}$ denotes the price that a ($a \in \{B,S\}$) submit to M at time (or stage) t. $t^a_{\max}$ is the deadline of a. $sur(t)$ denotes the surplus generated from the bargaining at time t.

Definition 1: Concession The difference between the offer at time t and the previous offer at time *t-1* is the concession agent a made at time t, i.e.

$$c_a(t) = \mid p^t_{a\to M} - p^{t-1}_{a\to M} \mid, a \in \{B,S\}, t \in N \tag{1}$$

Definition 2: Concession Distance At any time t, the difference between the offers of B and S is the concession distance at time t, that is:

$$c_{dis}(t) = p^t_{B\to M} - p^t_{S\to M} \tag{2}$$

Definition 3: Concession Reward At time t, if $p^t_{B\to M} \geq p^t_{S\to M}$, according to the efforts (concessions) bargainers have made for eliminating the concession distance of previous stage *t-1*, both agents will get some rewards defined as follows:

$$c^a_{rew}(t) = \frac{c_a(t)}{c_{dis}(t-1)} \cdot \mid c_B(t) - c_S(t) \mid, a \in \{B,S\} \tag{3}$$

But SOSBP will make the rewards invisible to the agents for the reason of possible counter speculation. The rewards will be added or subtracted on agents next offers, denoted by:

$$p^t_{B'} = p^t_{B\to M} - c^B_{rew}(t) \quad and \quad p^t_{S'} = p^t_{S\to M} + c^S_{rew}(t) \tag{4}$$

Definition 4: Concession Pressure if at time $t, t > 1$, then agent B's concession pressure is the difference between S's initial offer and S's offer at time t:

$$c^B_{pre}(t) = p^0_{S\to M} - p^t_{S\to M} \tag{5}$$

Similarly, agent S's concession pressure at time t is:

$$c^S_{pre}(t) = p^t_{B\to M} - p^0_{B\to M} \tag{6}$$

Definition 4 is used to compute how much concession pressure B or S has endured for achieving the agreement if it is possible. This parameter will decide how to divide the surplus between B and S

Definition 5: Surplus Quote if at time $t, sur(t) >= 0$, the surplus quote of B is defined as follows:

$$s^B_{quo}(t) = \frac{c^B_{pre}(t)}{c^B_{pre}(t) + c^S_{pre}(t)} \tag{7}$$

Clearly,

$$s^S_{quo}(t) = \frac{c^S_{pre}(t)}{c^B_{pre}(t) + c^S_{pre}(t)} = 1 - s^B_{quo}(t) \tag{8}$$

Given the above definitions, we can define the final price of the agreement as:

$$p = p^t_{B\to M} - s^B_{quo}(t) \cdot sur(t) \; or \; p = p^t_{S\to M} + s^S_{quo}(t) \cdot sur(t) \tag{9}$$

Or with concession rewards:

$$p = p^t_{B'} - s^B_{quo}(t) \cdot sur(t) \; or \; p = p^t_{S'} + s^S_{quo}(t) \cdot sur(t) \tag{10}$$

2.2 The Bargaining Process under SOSBP

Generally speaking, a negotiating process can be separated into three main stages: Initiation, negotiation and termination [7, 8].

1. B or S request M to initiate the bargaining session.
2. M will ask B and S for their initial offers, sincere price constraints and their deadline information.
3. At the time t, both B and S have two behavior options to choose:
   1) Submit new offers: They must satisfy that $\lambda_a \le c_a(t), a \in \{B, S\}$;
   2) Send termination message: due to reservation limit was reached or deadline was up or other possible reason.
4. According to the information received from B and S at the time t, M has following possible actions:
   1) If both *B* and *S* submit new offers and $p^t_{B\to M} \ge p^t_{S\to M}$, *M* will terminate the process and inform *B* and *S* the final price with following outcome:

$$p = \begin{cases} p^t_{B'} - s^B_{quo}(t) \cdot sur(t) & p^t_{B'} \ge p^t_{S'} \\ p^t_{B\to M} - s^B_{quo}(t) \cdot sur(t) & p^t_{B'} < p^t_{S'} \end{cases}, \text{otherwise,}$$

   2) *M* will save the information of this time round, and send a message to *B* and *S* that they need to make some more concession to reach an agreement. And the process goes back to step 3 for a new bargaining round.
   3) If *B* or *S* or both submit termination message, *M* terminates the bargaining process without any delay and sends notification of the reason to both sides. The process is then over.

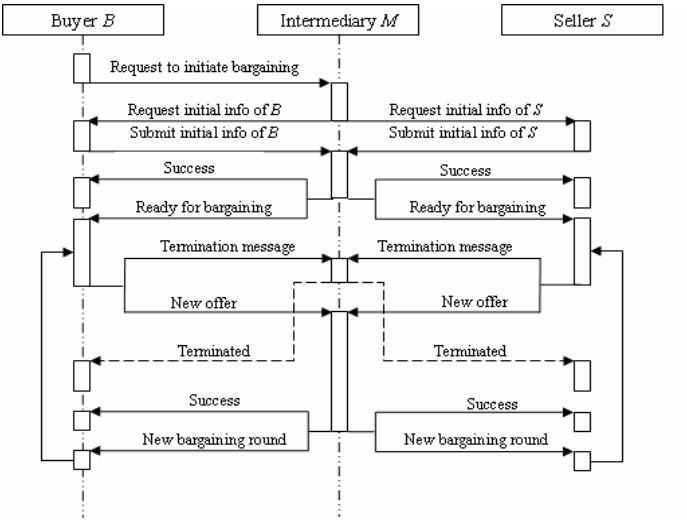


Fig. 1. The interactions under SOSBP

3 Bargaining Strategy Analysis under SOSBP

3.1 Buyer's Strategy Analysis

At stage *t*, *B*'s utility *u* can be classified as follows:

(1) If *B* choose to terminate the bargaining, then $u = -v_B$;

(2) If $t = d_B + 1$, then *B* has to quit and $u = -v_B$;

(3) If $t = d_S + 1$, then *S* will quit and $u = -v_B$;

(4) If $t <= d_s$, then *u* is determined by the relation between p_B^t and p_S^t.

We can summarize *B*'s bargaining strategy as follows:

(1) At the beginning, B submits its initial offer O and the sincere-price constraint r' which S's initial offer must satisfy.

(2) If the bargaining goes into stage 2, and d=1, then B will terminate the process; otherwise, he will choose an offer for stage 2 according to proposition 3.

(3) At any stage t (2<t<d), if the previous offer is r, then terminate; otherwise, choose a new offer according to proposition 5.

(4) When t=d, r is B's last offer.

(5) When t=d+1, B will terminate the bargaining.

From above discussion we can see that, to the buyer B, if there is no other external information which can modify B's beliefs about S, his strategy for each bargaining round could be predetermined because the parameters o, r, λ have already been decided after first stage.

3.2 Seller's Strategy Analysis

Now we will point out the main differences in the utility functions of *B* and *S*:

- For any offer *x* of *B* or *S*, if deal is made, then *u* is *r-x-v* for *B*; but for *S*, *u* is *x-r-v*; or *B*; while for *S*, it's the probability of $t = d_B + 1$;
- At stage *t*, ρ is the probability of $t = d_S + 1$ f
- For both *B* and *S*, μ is the probability of $p_B^t \geq p_S^t$.

We can conclude S's bargaining strategy as follows:

(1) At the first stage, *S* submits its initial offer *O* and the sincere-price constraint *r'* which *B*'s initial offer must satisfy.

(2) *S*'s strategy for second stage is decided by the proposition 7 when the deadline of *S* is greater than 1.

(3) If $t = d$, *S*'s last offer is *r*; If $t = d + 1$, *S* will quit the bargaining.

(4) At any stage *t* ($2 < t < d$), if the latest offer is *r*, then terminate, otherwise, choose new offer according to proposition 9.

Just like *B*, if no other external information can change *S*'s beliefs about *B*, the strategy for each bargaining round will be set up just after the first stage, as the parameters *o*, *r*, λ have been preset.

Hereto we have completed the basic analysis of *B* and *S'* bargaining strategy for some special assumptions on bargainers' beliefs. In the next section, we did an experimental study to show that, under such kind of beliefs, the result of bargaining under SOSBP is better than AOP on several aspects such as time saved, fairness, average deal rate etc.

4 Experimental Study for SOSBP

4.1 Generating Data

We design the functions for generating test data as follows:

Table 1. Functions for generating test data

Common Price P_c	$p_c = p_l + (p_h - p_l) \cdot rand$
Initial offer *O*	$o_B = p_c \cdot (1 + randn), r_B = o_B \cdot (1 + randn)$
Reservation Limit *r*	$r_S = p_c \cdot (1 + randn), o_S = r_S \cdot (1 + randn)$
Sincere-price constraint *r'*	$r_B' = r_B \cdot (1 + randn), r_S' = r_S \cdot (1 - randn)$
Common Deadline D_c	$D_c = D_l + (D_h - D_l) \cdot rand$
Deadline *d*	$d = D_c \cdot (1 + randn)$
Belief of opponent's deadline *d'*	$d' = D_l + (D_h - D_l) \cdot rand$

In all functions, the *rand* function is to generate random real numbers uniformly distributed in (0, 1) and *randn* is to generate data which are normally distributed with mean 0 and variance 1.

Table 2. Preconditions and limits

Fore-mentioned parameters	$p_l = 500, p_h = 5000, D_l = 2, D_h = 30$
The *randn* for *S*'s initial offer and *O*'s reservation limit	$(0.5 + 0.1 \cdot randn) + 0.1 \cdot randn$
The *randn* for *B* and *S*' sincere-price constraint	$(0.3 + 0.1 \cdot randn) + 0.1 \cdot randn$

And the *randn* for their deadlines is: $0.2 \cdot randn$. It means the generated deadlines are normally distributed with a mean of zero and a variance of 0.04. For the illustration, we only show the generated data of the first 20 sessions of the first group within following figures:

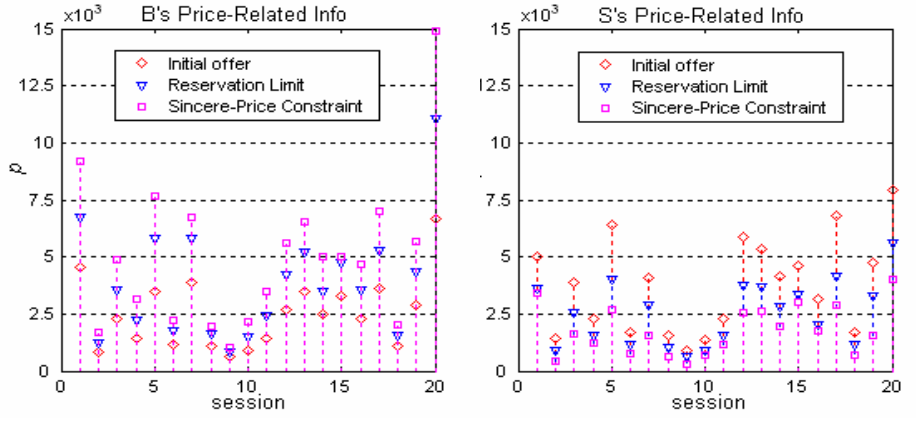


Fig. 2. *B* and *S*'s price-related Info

Fig.2. illustrates the randomly generated *B* and *S*'s price-related info. We take the third session as a typical example (see table 3) to confirm that the generated data is feasible for the experimental study.

As we can see in the Table 3, because $O_B > r_S$' and $O_S < r_B$ ', both bargainers satisfy each other's sincere-price constraints, so their sincerities can be justified for

Table 3. Price-Related info of the third session

	Initial Offer	Reservation Limit	Sincere-Price Constraint
B	*2263.04*	*3548.84*	*4898.67*
S	*3863.91*	*2574.27*	*1618.56*

this reason. On the other hand, although the initial concession distance ($O_S - O_B = 1600.87$) seems a little high, and $O_S > r_B, O_B < r_S$, their deal opportunity (the agreement zone) is not too bad because of the difference between their reservation limits ($r_B - r_S = 974.57$).

In the Fig.3, bargainers' deadline info was presented respectively. Again, we choose the third session as the example for the confirmation of feasibility. Table 4 contains the deadline info of the third session.

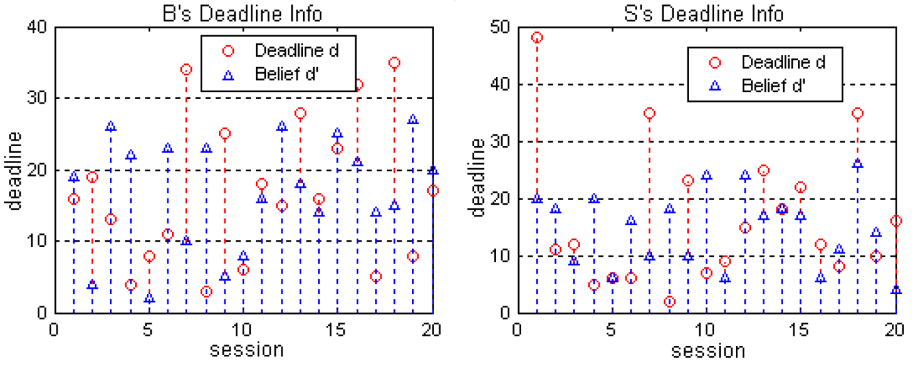


Fig. 3. Deadline info of the bargainers

In *B*'s belief, *S*'s lowest possible deadline d_B' is *26,* it is greater than d_B, so to the buyer, there is no need to consider the possibility of $t = d_S + 1$ during the bargaining session. Seller's situation is different, because if the bargaining goes into stage t which is greater than *9*, then he must keep changing his belief on d_B till the bargaining is over.

Table 4. Deadline info of the third session

	Deadline	Belief on opponent's deadline
B	*13*	*26*
S	*12*	*9*

4.2 Results

The experiment is performed in completely same environment except the different bargaining protocol. In section 3, we have analyzed what strategy the bargainers should take for a given condition when using SOSBP. However, when using AOP, how to bargain? To answer this question, two things must be considered: which bargainer should take the first step? And how to update beliefs after receiving opponent's offer?

As for the first one, we classify the problem into three cases: Buyer Move First (BMF), Seller First (SMF) and Random one Move First (RMF). To the second, we

assume their beliefs will be updated in a manner that is similar to the assumption in section 3. New received offer will be used to changing bargainer's beliefs.

For example (suppose in BMF case and the bargaining info is from table 3), at first stage, *B* offers *2263.04*. As it is less than $r_S = 2574.27$, S will not change his belief on this info and his counter offer will be his most desired price *3863.91*. Now, the bargaining goes into second stage. Because $3863.91 > 3548.84 = r_B$, *B* need not to change his belief at this moment, so he will choose

$$p_B^2 = (o_B + r_B)/2 = 2905.04 \tag{11}$$

as new offer (Basing on analysis in section 3). What shall *S* do right now? At first, before he receive the p_B^2, his new offer for stage 2 will be

$$p_S^2 = (o_S + r_S)/2 = 3219.09 \tag{12}$$

If $p_B^2 \geq p_S^2$, the *S* will accept *B*'s offer, but now, since $p_B^2 < p_S^2$ and $p_B^2 > r_B$, *S*'s rational action is to change r_S to *2905.04*, thus,

$$p_S^2 = (o_S + r_S)/2 = 3384.48 \tag{13}$$

In this manner, the bargaining will continue until the agreement is reached or the earlier deadline is up.

4.2.1 Time-Saved Rate

We record all the time-saved rates of all sixteen different cases in table 5 provide a clearer view. Apparently, under SOSBP, the time-saved rate of each data group is much higher than its counterparts under the traditional protocol. Therefore, it's reasonable to say that, under the given environment, bargainers will reach agreement more quickly using SOSBP, that's to say, more time could be saved if bargainers adopt the SOSBP as the rule of negotiation.

Table 5. Time-Saved rate

	40	120	360	1080
SOSBP	0.647	0.678	0.698	0.716
BMF	0.437	0.519	0.549	0.541
SMF	0.410	0.503	0.545	0.541
RMF	0.435	0.489	0.537	0.544

4.2.2 Surplus-Division Rate

Table 6 provides us the data on some aspects of surplus-division rate. Except Rsd, we also calculated the percentage of surB and surS respectively, because that who is the first mover is a key factor for surplus division under the traditional protocol.

Our experimental results provide some helpful evidences which indicate, under AOP, that who will take the first move is an important factor for dividing the surplus. After reviewing the cases of BMF and SMF in table 6, we found that the first mover

Table 6. Surplus-Division rate

	40			120			360			1080		
	R_{sd}	sur_B	sur_S	R_{sd}	sur_B	sur_S	R_{sd}	sur_B	sur_S	R_{sd}	sur_B	sur_S
SOSBP	0.001	0.499	0.501	0.022	0.511	0.489	0.015	0.507	0.493	0.020	0.510	0.490
BMF	0.116	0.442	0.558	0.125	0.438	0.562	0.057	0.471	0.529	0.095	0.453	0.547
SMF	0.242	0.621	0.379	0.135	0.567	0.433	0.200	0.600	0.400	0.204	0.602	0.398
RMF	0.109	0.554	0.446	0.008	0.496	0.504	0.061	0.531	0.469	0.049	0.524	0.476

always got the less part of the whole surplus no matter what scale of the group is. Due to the higher R_{sd} in these two rows, taking the first move means getting much less surplus from the deal, thus, for both bargainers, it's reasonable to refuse to be the first mover. In our view, this could be a main obstacle to a smooth bargaining process under AOP. Although RMF may be the feasible solution of this problem, as we mentioned earlier, its fairness level is not as good as SOSBP.

4.2.3 Deal Rate

Finally, we show the results about deal rate in the table 7. For each group of sessions, R_{1st}, R_{2nd} and R_{end} have also been provided for more discussions. From the table 7, we can see the deal rates seem a little too high in all cases (over 0.7), the possible reasons may rest on the way of data generation, or our assumptions that bargainers are honest and their motivations are pure.

Table 7. Deal rate

	40				120				360				1080			
	R_d	R_{1st}	R_{2nd}	R_{end}	R_d	R_{1st}	R_{2nd}	R_{end}	R_d	R_{1st}	R_{2nd}	R_{end}	R_d	R_{1st}	R_{2nd}	R_{end}
SOSBP	0.900	0.028	0.472	0.111	0.817	0.020	0.500	0.071	0.833	0.043	0.537	0.053	0.840	0.034	0.563	0.042
BMF	0.775	0.226	0.258	0.581	0.750	0.133	0.422	0.489	0.728	0.191	0.424	0.443	0.756	0.147	0.441	0.443
SMF	0.875	0.257	0.171	0.629	0.758	0.231	0.319	0.495	0.756	0.265	0.327	0.460	0.769	0.225	0.353	0.448
RMF	0.825	0.242	0.212	0.606	0.800	0.177	0.344	0.521	0.747	0.223	0.375	0.457	0.752	0.190	0.401	0.437

Three sub measures considered here help us getting more understandings about the two kinds of protocol. First, R_{1st} reflects the situation where the deal is made instantly after one-stage bargaining. In each group of sessions, R_{1st} under SOSBP is very tiny. It means that the traders' most desired prices will seldom match each other instantly. While under traditional protocol, R_{1st} is much higher no matter who will take the first move. This is because the bargainer who takes the second move uses its optimal price, rather than the reservation limits, to compare with the first mover's offer.

5 Conclusion

Conventionally, AOP is the most predominant way for one-to-one bargaining. Almost all past theoretical or practical researches on bargaining issue took it as the

default-trading rule. Regardless of its entrenched place in people's mind, we provide an alternative protocol SOSBP that can be useful in autonomous bargaining between self-interested agents. The idea comes from one weak point of the traditional protocol. When agents begin to bargain, both of them are unwilling to show their reservation limit or even ordinary offer to the opponent, because he may take advantage of the revelation of private info. With SOSBP, agent could feel free to reveal their offer to the intermediary agent who has no any preference on both sides. Additionally, since the bargainer does not know what offers his opponent have ever made, the ex-post regret are also excluded by this mean, i.e. if the agreement is reached, the bargainer should not feel that he may get more out from his opponent. Through the minimum concession constraint, concession reward and surplus quote, SOSBP facilitates the success of bargaining by stimulating the bargainers to make more concessions.

Besides the traditional bargaining protocol, we believe that SOSBP could be an alternative for automated bargaining if only buyer and seller agree with each other to employ it. SOSBP should be built in bargaining applications and co-exist with existent protocol. One important issue is that where the intermediary agent should be located. In [9], MAGNET provides an explicit intermediary infrastructure for agent negotiation. We anticipate that the intermediary can be added into MAGNET as bargaining service provider. For this purpose, we need to identify the proper interface between SOSBP and MAGNET so that they can work together smoothly.

Perhaps it is somewhat imprudent to claim that SOSBP has great advantage before it can be applied in practical systems. Many aspects and potential problems need to be investigated carefully. Therefore, our research will continue to be hold in the laboratory till it can be released as a practical trading protocol.

Acknowledgments. Great thanks for the support from National Natural Science Foundation of China (NSFC, Grant 70572034), National Natural Science Foundation of China (NSFC, Grant 60773188) and China Postdoctoral Science Foundation (CPSF Grant 20080430961).

References

1. Lomuscio, A., Wooldridge, M., Jennings, N.R.: A Classification Scheme for Negotiation in Electronic Commerce. In: Sierra, C., Dignum, F.P.M. (eds.) AgentLink 2000. LNCS, vol. 1991, pp. 19–33. Springer, Heidelberg (2001)
2. Jiang, W.J., Xu, Y.S.: A Novel Multi-agent Automated Negotiation Model Based on Associated Intent. Agent Computing and Multi-Agent Systems 4088, 608–613 (2006)
3. Rubinstein, A.: Perfect Equilibrium in a Bargaining Model. Econometrica 50, 97–109 (1982)
4. Fatima, S., Wooldridge, M.J., Jennings, N.R.: Optimal Negotiation Strategies for Agents with Incomplete Information. In: Meyer, J.J., Tambe, M. (eds.) Intelligent Agent series VIII: Proc. 8th Intern. Workshop on Agent Theories, Architectures, and Languages, ATAL (2001)
5. Sandholm, T., Vulkan, N.: Bargaining with Deadlines. In: AAAI 1999, pp. 44–51, Orlando, FL (1999)

6. Winoto, P., McCalla, G.I., Vassileva, J.: Non-monotonic-offers Bargaining Protocol. Autonomous Agents and Multi-Agent Systems 11, 45–67 (2005)
7. Rahwan, I., McBurney, P., Sonenberg, L.: Towards a Theory of Negotiation Strategy (A Preliminary Report). In: Proceedings of the Workshop on Game Theoretic and Decision Theoretic Agents (GTDT), Melbourne, Australia (2003)
8. Jennings, N.R., Faratin, P., Lomuscio, A.R., Parsons, S., Sierra, C., Wooldridge, M.: Automated Negotiation: Prospects, Methods and Challenges. International Journal of Group Decision and Negotiation 10, 199–215 (2001)
9. John, C., Ben, Y., Scott, J., Bamshad, M., Maria, G.: A Market Architecture for Multi-agent Contracting. In: Proc. of the Second Int'l Conf. on Autonomous Agents, pp. 285–292 (1998)

Implicit Camera Calibration Based on a Nonlinear Modeling Function of an Artificial Neural Network

Dong-Min Woo and Dong-Chul Park

Image Processing Lab., Information Engineering Department, Myongji University, Yongin, Gyeonggido 449-728, Korea
dmwoo@mju.ac.kr

Abstract. Most calibration methods are based on the camera model which consists of physical parameters of the camera including position, orientation, focal length, and optical center. In this paper, we propose a new approach which is based on the neural network model instead of the physical camera model. The neural network employed in this paper is primarily used as a nonlinear modeling function between 2D image points and points of a certain space in 3D real world. The neural network model implicitly contains all the physical parameters, some of which are very difficult to be estimated in the conventional calibration methods. In order to show the performance of the proposed method, images from two different cameras with three different camera angles were used for calibrating the cameras. The performance of the proposed neural network approach is compared with the well-known Tsai's two stage method in terms of calibration errors. The results show that the proposed approach gives much more stable and acceptable calibration error over Tsai's two stage method regardless of camera camera angle.

Keywords: camera calibration, image coordinate, real world coordinate, camera model, focal length, neural network.

1 Introduction

Camera calibration methods have been mostly studied on the estimation of camera physical parameters including position, orientation, focal length, and optical center [1,2,3,4]. This kind of the traditional method explicitly evaluates the physical parameters, and referred to be an explicit calibration method. However the main objective of camera calibration is to obtain the correlation between camera image coordinate and 3D real world coordinate. In this context we can use an implicit calibration method, where the nonlinear mapping model functions as an implicit model which can give us a transformation between 2D image points and points of a certain space in 3D real world. This model is called as an implicit model, and the calibration method using this model is an implicit calibration method [5,6].

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 967–975, 2009.

This paper is concerned with an implicit calibration method in terms of an artificial neural network (ANN). ANNs have been shown to have the ability to model an unspecified nonlinear relationship between input patterns and output patterns. This nonlinear mapping ability can be utilized to address some physical parameters in implicit camera calibration that cannot be readily estimated by the existing calibration methods. The ANN-based camera calibration approach does not estimate camera physical parameters. However, this is not an issue when the objective of the camera calibration process is to obtain the correlation between the camera image coordinates and the 3D real world coordinates. The implicit camera calibration approach, which can calibrate a camera without explicitly computing its physical parameters, can be used for both the 3D measurement and the generation of image coordinates.

2 Explicit Calibration Method

Most conventional calibration methods are based on the explicit estimation of camera parameters of the camera model initially established. Tsai's two stage method (TSM)[4] is one of the widely used explicit calibration methods. This paper choose Tsai's two stage method as a reference state of art method for the purpose of performance comparison. The TSM first obtains the transformation parameters with the assumption that there exists no distortion in the camera. The TSM then refines the transformation parameters with the distortion of the camera by using a nonlinear search. That is, first, the camera model is assumed to be ideal for the camera calibration by neglecting the lens distortion.

Fig. 1 shows the camera model used in TSM. A point P is an object of the real world coordinate(X_w, Y_w, Z_w) and (x,y,z) is a 3D camera coordinate. The center of the camera coordinate is the optical center O and (X,Y) is the image coordinate with the center of O_i. The distance between O and O_i is f, the focal length of the camera. (X_u, Y_u) is the corresponding point with the assumption of no lens distortion. (X_u, Y_u) is then translated to (X_f, Y_f), which is a point in computer image coordinate on the image buffer and is expressed in pixel numbers. The basic geometry of the camera model can be written as the transformation of the two coordinates with the following displacement and orientation:

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \tag{1}$$

with

$r_1 = \cos\psi \cos\theta$
$r_2 = \sin\psi \cos\theta$
$r_3 = -\sin\theta$
$r_4 = -\sin\psi \cos\theta + \cos\psi \sin\theta \cos\phi$
$r_5 = \cos\psi \cos\theta + \sin\psi \sin\theta \sin\phi$
$r_6 = \cos\theta \sin\phi$

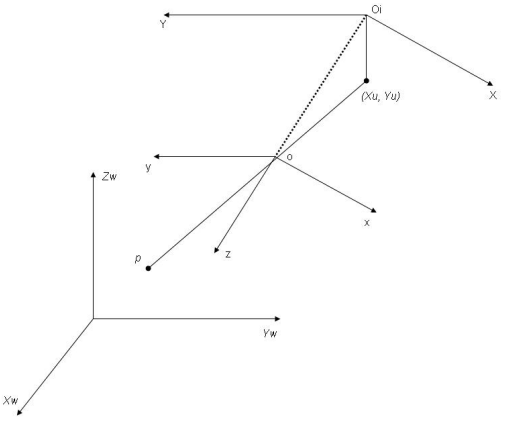


Fig. 1. Camera model defined in TSM

$r_7 = \sin\psi \sin\phi + \cos\psi \sin\theta \cos\phi$
$r_8 = -\cos\psi \sin\phi + \sin\psi \sin\theta \cos\phi$
$r_9 = \cos\theta \cos\phi$

where θ, ϕ and ψ represent yaw, pitch and roll, respectively.

As can be seen from the above equations, there are six extrinsic parameters: θ, ϕ, and ψ for rotation, and three components for the translation vector T. The problem of camera calibration is to find the six parameters $\theta, \phi, \psi, T_x, T_y$, and T_z by using the number of points measured in the (X_w, Y_w, Z_w) coordinate.

In the second stage of the TSM, a distortion parameter is considered. The relations between the computer image coordinate with distortion and the real world coordinate can be derived as follows:

$$S_x(X_f - C_x)(1 + G(X_d^2 + Y_d^2)) = f\left(\frac{r_1 x_w + r_2 y_w + r_3 z_x + T_x}{r_7 x_w + r_8 y_w + r_9 z_w + T_x}\right) \tag{2}$$

$$S_x(X_f - C_y)(1 + G(X_d^2 + Y_d^2)) = f\left(\frac{r_4 x_w + r_5 y_w + r_6 z_x + T_x}{r_7 x_w + r_8 y_w + r_9 z_w + T_x}\right) \tag{3}$$

where (X_f, Y_f) is the image coordinate of the frame grabber, (C_x, C_y) is the image center, S_x and S_y are components of the translating scale of the x-axis and y-axis when the A/D transform is performed, (X_d, Y_d) is a distorted coordinate by lens distortion, and G is the distortion parameter. TSM obtained the solution by using a gradient-based nonlinear search method. In an explicit calibration, the calibration is performed with extrinsic parameters. However, the distortion parameters cannot include all the parameters involved in the distortion of the image. Even with the assumption of perfect inclusion of distortion parameters, there still remains room for errors in finding the right solution for such parameters.

3 Camera Calibration Using ANN

3.1 Implicit Calibration Method

Suppose that there is a calibration plane and the center of the calibration plane is defined as O. In the calibration plane, we have N points. A point, P:$(X_i, Y_i) \in w_i$, $i = 1, 2, \cdots, N$, in the world plane is ideally projected to $\bar{p} : (\bar{x_i}, \bar{y_i})$ in the camera CCD plane. However, because of the distance of the camera lens, the point of the world plane is projected to a distorted point, p:(x_i, y_i). This point is observed through the frame buffer coordinate $p(u_i, v_i)$ in pixels.

For a back-projection problem, a transformation from the image coordinates in the frame buffer to the world coordinates in the calibration plane is required. For this purpose, an ANN is adopted in the proposed ANN-based calibration approach, where the input and the output of the ANN are the image coordinates and the world coordinates, respectively. After proper training of the ANN with training points, the ANN can map the relation of two planes. Owing to the nonlinear system modeling capability of the ANN, it is not necessary to utilize all the physical parameters involved with the camera calibration, including the lens distortion and the focal length of the camera.

With the coordinate system shown in Fig. 2, (x_1, y_1, z_1) and (x_2, y_2, z_1) are defined as two points on the calibration plane $Z = z_1$, and (x_1', y_1', z_2) and

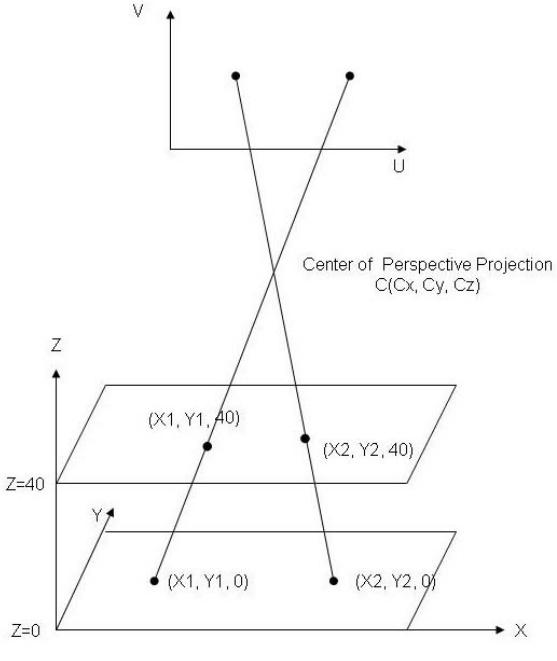


Fig. 2. The center of a perspective projection

(x'_2, y'_2, z_2) are two other points on the plane $Z = z_2$. The line equations that pass each of the two points can be expressed by the following equations:

$$\overrightarrow{\boldsymbol{P}} = (x_1, y_1, z_1) + t(x'_1 - x_1, y'_1 - y_1, z_2 - z_1) \tag{4}$$

$$\overrightarrow{\boldsymbol{Q}} = (x_2, y_2, z_2) + t(x'_2 - x_2, y'_2 - y_2, z_2 - z_1) \tag{5}$$

$$\overrightarrow{\boldsymbol{P}} = \overrightarrow{\boldsymbol{Q}} \tag{6}$$

Since the equations given by Eq.(1) and Eq.(2) meet at the point C, i.e., Eq.(3), this point can be considered as the perspective center of the image, as shown in Fig. 2.

By using the perspective center of an image, the estimation of the image coordinates of any 3D world point P can be obtained. In this case, an ANN that is trained with the real world coordinates of points on $Z = z_1$ as inputs and the image plane coordinates for the corresponding points as targets is given. It should be noted that the input and target for the ANN in this case are different from those of the back-projection problem. When the image coordinate of a point (P_1) on any calibration plane Z is needed, the line equation that passes the point(P_1) in the calibration plane Z and the perspective center of a camera(C) is first obtained. The line equation can produce P_0 on the calibration plane $Z = z_1$. By using P_0 as the input to the trained ANN, we can obtain the image coordinates of the point $\hat{p}$. This process is shown in Fig. 2.

3.2 ANN Structure for Camera Calibration

The ANN model adopted in this paper is a standard MultiLayer Perceptron Type Neural Network (MLPNN) and an error back-propagation algorithm is used for training the MLPNN. After several experiments, the architecture of the MLPNN is selected as $2 \times 10 \times 8 \times 2$, as shown in Fig. 3. Note that the selection of a

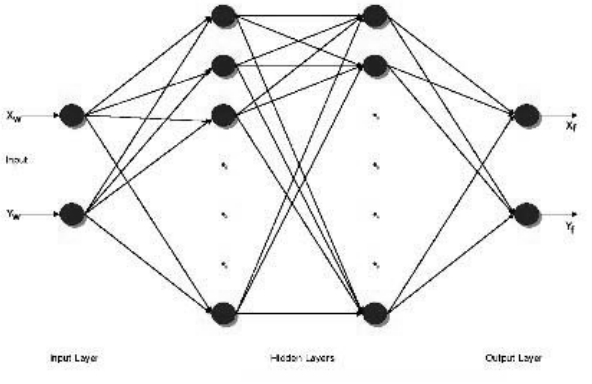


Fig. 3. ANN structure for camera calibration

specific architecture is a state of art and other architectures can be also used without any degradation of the resulting performance. With the architecture chosen, no overfitting problem was experienced with 5,000 training epochs. Note that proper numbers of training epochs are dependent on the complexity of the given problem and the number of training data. Note that the neurons in the input and output layers represent the 2D coordinates. More detailed information on the MLPNN and error back-propagation algorithm can be found in [7].

Unlike the explicit camera calibration method, the proposed ANN-based method finds the direct relation between the world coordinates and the image coordinates. The ANN adopted in this implicit calibration approach can incorporate all the extrinsic parameters of the camera and the distortion parameters when the ANN is trained properly.

4 Experiments and Results

4.1 System Environment for Experiments

The specifications of the image acquisition tool for our simulation environment are summarized in Table 1.

Table 1. The specification of image acquisition

Image aquisition tool	Specification
Frame grabber	Horizontal resolution 512 Vertical resolution 512
CCD Image censor	Scale of cell(X-axis) 8.4 μm Scale of cell (Y-axis) 9.8 μm
lens	Focal length (F 1.4) 16mm

Images are acquired at three different orientations. The performance of camera calibration results using artificial neural networks is compared and analyzed with that of Tsai's two stage method, the most widely used approach for explicit camera calibration. In this paper, the average error between the calibrated image coordinates and real world coordinates is used to compare the performance of the camera calibration methods. The average error in pixels (AEIP) is defined as follows:

$$AEIP = \frac{1}{N}\sum_{i=1}^{N}[(X_{fi} - \hat{X}_{fi})^2 + (Y_{fi} - \hat{Y}_{fi})^2]^{1/2} \tag{7}$$

where $(\hat{X}_{fi}, \hat{Y}_{fi})$ is the estimated image coordinate, which is computed by using calibrated variables from the real coordinate point (X_{wi}, Y_{wi}, Z_{wi}) corresponding to the computer image coordinate (X_{fi}, Y_{fi}).

The images used for the experiments are obtained by positioning the camera in the real world coordinate. The positions of the camera are also rotated for obtaining image data with different orientations. Each image is composed of 99 calibration points (11×9), which have an interval of 25mm between columns

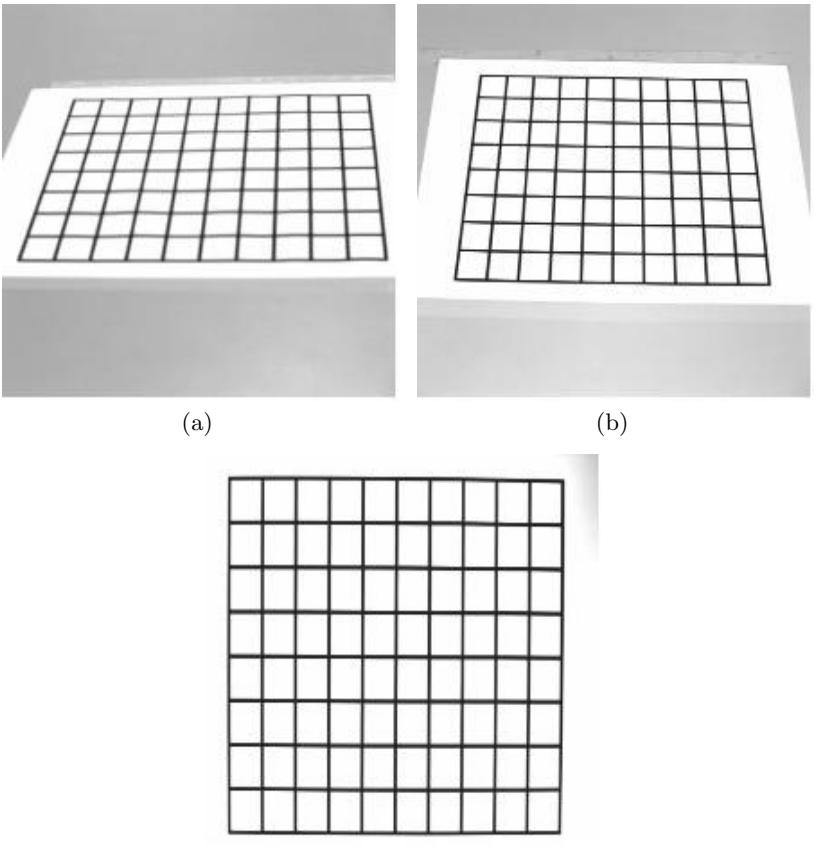


Fig. 4. The calibration points at different angles: (a) angle = 30 °, (b) angle = 60 °, and (c) angle = 90 °

and an interval of 20mm between rows. Among the calibration points acquired from two images including 99 calibration points for each different heights, 79 randomly selected calibration points in each image are used for training the ANN and the remaining 20 points are used for evaluation of the trained ANN. Fig. 4 shows the images with different heights used in our experiments.

4.2 Experiments on Image Coordinate Projection

The proposed method is compared with Tsai's two stages method, which finds the physical parameters of the camera using the interrelation between the image coordinates and the known 3D space coordinates. For the calculation of the physical parameters for Tsai's method and training ANN, 10 sets of 79 randomly chosen calibration points are collected. For each set of calibration points, the remaining 20 points are used for testing the performance of both methods. Table 2

Table 2. Comparison of estimation errors in AEIP

Case	Tsai's two stage method	ANN-based method
angle = 30 °	0.84	0.47
angle = 60 °	0.63	0.45
angle = 90 °	0.43	0.45
Average	0.63	0.46

shows the test results for both methods. As shown in Table 2, the average improvement of the proposed ANN-based method over Tsai's method in terms of AEIP is 27 %.

4.3 Experiments on 3D Real World Coordinate Reconstruction

The real space coordinate obtained by estimating the 3D space coordinate at an arbitrary height can be reconstructed after training the ANN with points on two calibration plans, i.e., Z = 0 and Z = 40, as follows: select a certain point of the image and then find the point of the real space coordinate of Z=0 and Z=40 calibration plane corresponding to the selected image point. Using Eq. (1) - Eq. (3), the perspective center of the image can be found. The real points for the space coordinate of the $Z = 0$ and $Z = 40$ calibration plane corresponding to the selected image points are then found. By using Eq. (1) - Eq. (3), ten linear equations connecting points of $Z = 0$ plane with points of the $Z = 40$ plane are formulated for estimating the coordinate of the 3D space on the $Z = 40$ calibration plane. The average error on 3D world point reconstruction comes up with 0.65, which is accurate enough for this calibration system to be efficiently used in many computer vision applications.

5 Conclusion

This paper proposed a camera calibration method using an artificial neural network. The proposed ANN-based implicit method is applied to the estimation of 2D coordinates of an image world with given 3D space coordinates. The proposed method has advantages over Tsai's two stage method in real-time applications as it can be operated in real time after proper training while Tsai's two stage method requires somewhat time consuming procedures for calculating proper parameters for a given task. The proposed method is also more flexible than Tsai's two stage method since it is not affected by camera position, illumination or distortion of the camera lens. More importantly, the proposed ANN-based method is not affected by the quality of the camera lens in finding the mapping function between the image coordinates and the real coordinates whereas Tsai's method is considerably affected by the quality of the camera. In comparison to the conventional approach Tsai's two stage method, the proposed ANN-based

method shows promising results for calibrating camera when issues including practical applicability, flexibility, and real-time operation are relevant.

Acknowledgment. This work was supported by the Korea Research Foundation Grant funded by the Korea Government (MOEHRD-KRF-2005-042-D00265).

References

1. Heikkila, J.: Geometric Camera Calibration Using Circular Control Points. IEEE Trans. on Pattern Anal. Mach. Intell. 22, 1066–1077 (2000)
2. Xu, Q., Ye, D., Che, R., Huang, Y.: Accurate Camera Calibration with New Minimizing Fuction. In: Proc 2006 IEEE ROBIO, pp. 779–784 (2006)
3. Douxchamps, D., Chihara, K.: High-Accuracy and Robust Localization of Large Control Markers for Geometric Camera Calibration. IEEE Trans. on Pattern Anal. Mach. Intell. 31, 376–383 (2009)
4. Tsai, R.: An Efficient and Accurate Camera Calibration Technique for 3-D Machine Vision. In: Proc. IEEE Int. Computer Vision and Pattern Recognition, pp. 364–374 (1986)
5. Martins, H.A., Birk, J.R., Kelley, R.B.: Camera Models Based on Data from Two Calibration Plane. Computer Graphics and Image Processing 17, 173–180 (1981)
6. Mohr, R., Morin, L.: Relative Positioning from Geometric Invariant. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 139–144 (1991)
7. Rumelhart, D., Hinton, G., Williams, R.: Parallel Distributed Processing. MIT Press, Cambridge (1986)

Credit Risk Assessment Model of Commercial Banks Based on Fuzzy Neural Network

Ping Yao[1], Chong Wu[2], and Minghui Yao[3]

[1] Shool of Economics & Management , Heilongjiang Institute of Science and Technology, Harbin 150027, China
[2] Shool of Management, Harbin Institute of Technology, Harbin 150001, China
[3] Shool of Management, FuDan Uiverstiy, Shanghai 200433, China

Abstract. A commercial bank credit risk assessment model based on fuzzy neural network has been established using the credit assessment index system established for commercial banks. This network is a 6 layered structure with 4 factor inputs and one output measuring the credit risk of commercial banks. The fuzzy rule layer has the capability of making necessary adjustments in accordance with specific conditions of problems. The operation of this model is much better than the totally black-box operation of a neural system. A substantiation analysis has been made with 167 observations as sample data; training results indicate that the network prediction has less error.

Keywords: Commercial bank, credit risk assessment, fuzzy neural network, factor analysis.

1 Introduction

Without doubt credit evaluation is an important topic for research in the field of financial risk management [1-3].Due to the important of credit risk evaluation, there is an increasing research stream focusing upon credit risk assessment and credit scoring. After that logit regression [4-6] or probit regression [7] have widely adopted in subsequent work. Nevertheless empirical results have shown that most of financial ratios violate the assumptions of the multivariate statistical model used in these previous studies. Recent studies have revealed that emerging artificial intelligent techniques, such as artificial neural networks (ANNs)[2,3,8-11,12], evolutionary computation (EC) and genetic algorithm (GA)and support vector machine (SVM) are advantageous to statistical analysis and optimization models for credit risk evaluation in terms of their empirical results[13].

Although almost all classification methods can be used to evaluate credit risk, the use of a technique depends on the complexity of the institution, and the size and the type of loan. Analytical models, such as empirically derived credit scoring systems (based on historical data), use the probability of default to predict the relative creditworthiness of a loan applicant. But, the credit-scoring model does not completely eliminate the human element. The section of cutoff scores is a subjective

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 976–985, 2009.

decision. Moreover, the evaluation of applicants that have scores between the accept-scores and the reject-scores is quite subjective. Thus, to be more objective in evaluating loan applications, many institutions are exploring the use of artificial intelligence techniques such as artificial neural systems and fuzzy logic.

This study investigates the classification of neural-fuzzy systems to consumer loan applications. The rest of this paper is organized as follows. In Section 2, the basic ideas of modeling are described in detail; Section 3 describes the data and variables used in this study; explains the design of a neural fuzzy system model and analyzes the empirical examples, and finally, In Section 4, some concluding remarks are drawn.

2 Basic Ideas of Modeling

In this section, the basic ideas of methods for credit risk assessment are provides: (1) the structure of neuro-fuzzy system; (2) hybrid learning algorithm (Gradient-Descent and the LSE).

2.1 Neuro-fuzzy System Structure

Once the input and output variables are identified the neuro-fuzzy system is realized using a six-layered network: The input, output, and node functions of each layer are explained in the subsequent paragraphs.

(1) ***Layer 1* (*input layer*)**
Each node in layer 1 represents the input variables. This layer simply transmits these input variables to the fuzzification layer.

(2) ***Layer 2* (*fuzzification layer*)**
The fuzzification layer describes the membership function of each input fuzzy set. Membership functions are used to characterize fuzziness in the fuzzy sets. The output of each node i in this layer is given by $\mu_{Ai}(x_i)$, where the symbol $\mu_{Ai}(x_i)$ is the membership function. Its value on the unit interval [0,1] measure the degree to which element x belongs to the fuzzy set A, x_i is the input to node i and A_i is the linguistic label for each input variable associated with this node.

Gradient methods can be used easily for optimizing their design parameters. Thus in this model, we have replaced the triangular fuzzy memberships with Gaussian functions. Gaussian function is specified by a set of two fitting parameters $\{a,b\}$ as

$$\mu_A(x) = e^{-\left(\frac{x-a}{b}\right)^2} \tag{1}$$

(3) ***Layer 3* (*inference layer*)**
The third layer is the inference layer. Each node in this layer is a fixed node and represents the IF part of a fuzzy rule. This layer aggregates the membership grades using any fuzzy intersection operator which can perform fuzzy AND operation [23]. The fuzzy intersection operators are commonly referred to as T-norm (triangular

norm) operators. Most frequently used T-norm operators are min or product operators. For instance

Rule 1: IF x is A_1 AND y is B_1 THEN $f_1 = p_1 x + q_1 y + r_1$;

Rule 2: IF x is A_2 AND y is B_2 THEN $f_2 = p_2 x + q_2 y + r_2$;

The output of ith node in layer 3 is given as

$$O_{3,i} = wi = \mu_{A_i}(x)\mu_{B_i}(y) \tag{2}$$

(4) ***Layer 4* (*normalization layer*)**
The ith node of this layer is also a fixed node and calculates the ratio of the ith rule's firing strength in inference layer to the sum of all the rule's firing strengths

$$O_{4,i} = \overline{w}_i = \frac{w_i}{w_1 + w_2 + \cdots + w_R} \tag{3}$$

where $i = 1,2,\ldots,R$ and R is total number of rules. The outputs of this layer are called normalized firing strengths.

(5) ***Layer 5* (*output layer*)**
This layer represents the THEN part (i.e., the consequent) of the fuzzy rule. The operation performed by the nodes in this layer is to generate the qualified consequent (either fuzzy or crisp) of each rule depending on firing strength. Every node i in this layer is an adaptive node. The output of the node is computed as

$$O_{5,i} = \overline{w}_i f_i \tag{4}$$

where $\overline{w}_i$ is a normalized firing strength from layer 3 and f_i is a linear function of input variables of the form $(p_i x_1 + q_i x_2 + r_i)$, where $\{p_i, q_i, r_i\}$ is the parameter set of node i, referred to as consequent parameters or f may be a constant. If f_i is linear function of input variables then it is called first order Sugeno fuzzy model and if f_i is a constant then it is called zero order Sugeno fuzzy model.

(6) ***Layer 6* (*defuzzification layer*)**
This layer aggregates the qualified consequents to produce a crisp output. The single node in this layer is a fixed node. It computes the weighted average of output signals of the output layer as

$$O_{6,1} = \sum_i O_{5,i} = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{5}$$

2.2 Hybrid Learning Method

For the ANFIS architecture above, when the values of the premise parameters are fixed, the overall output can be expressed as a linear combination of the consequent parameters. In symbols the output f can be written as

$$\begin{aligned} f &= \frac{w_1}{w_1+w_2} f_1 + \frac{w_2}{w_1+w_2} f_2 \\ &= \overline{w}_1 (p_1 x + q_1 y + r_1) + \overline{w}_2 (p_2 x + q_2 y + r_2) \\ &= (\overline{w}_1 x) p_1 + (\overline{w}_1 y) q_1 + (\overline{w}_1) r_1 + (\overline{w}_2 x) p_2 + (\overline{w}_2 y) q_2 + (\overline{w}_2) r_2 \end{aligned} \tag{6}$$

which is linear in the consequent parameters p_1, q_1, r_1, p_2, q_2 and r_2. From this observation, we have

S set of total parameters,
S_1 set of premise (non-linear) parameters,
S_2 set of consequent (linear) parameters.

Using the above notation, the Hybrid Learning Algorithm is discussed below.

Assuming that the adaptive network under consideration has only one output represented by

$$O = f(i, S) \tag{7}$$

where i is the vector of input variables, S is the set of parameters, and f is the overall function implemented by the adaptive network. If there exists a function H such that the composite function $H \circ f$ is linear in some elements of S, then these elements can be identified by the least-squares method. As illustrated before, if the parameter set S can be divided into two sets

$$S = S_1 \oplus S_2 \tag{8}$$

(where $\oplus$ represents the direct sum) such that $H \circ f$ is linear in the elements of S_2, then upon applying H to Eq.(7), we have

$$H \circ O = H \circ f(i, S) \tag{9}$$

which is linear in the elements of S_2. Assuming H is identity, Eq.(6) and Eq.(9) are equivalent. Given values of elements of S_1, we can put P training data in Eq.(9) and obtain a matrix equation

$$A\theta = y \tag{10}$$

Where θ is an unknown vector whose elements are parameters in S_2. This is a standard linear least-squares problem, and the best solution for θ, which minimizes $\|A\theta - y\|^2$, is the least-squares estimator (LSE) θ^*:

$$\theta^* = (A^T A)^{-1} A^T y \tag{11}$$

where A^T is the transpose of A and $(A^TA)^{-1}A^T$is the pseudoinverse of A, if A^TA is non-singular. Further, we can use the recursive LSE formula to calculate θ^*. if the ith row vector of matrix A is $a_i^T$and the ith element of y is y_i^T, then θ can be calculated iteratively as follows:

$$\theta_{i+1} = \theta_i + P_{i+1} a_{i+1} (y_{i+1}^T - a_{i+1}^T \theta_i)$$
$$P_{i+1} = P_i - \frac{P_i a_{i+1} a_{i+1}^T Pi}{1 + a_{i+1}^T P_i a_{i+1}} \quad i = 0,1,\ldots,P-1 \tag{12}$$

where the LSE θ^* is equal to θ_p. The initial conditions needed to bootstrap Eq.(12) are $\theta_0 = 0$ and $P_0 = \gamma I$, where γ is positive large number and I is the identity matrix of dimension $M \times M$.

The hybrid learning algorithm combines the Gradient-Descent Model and the LSE to update the parameters in an adaptive network. The ANFIS employs an external reference signal, which acts like a teacher and generates an error signal by comparing the reference with the obtained response. Based on error signal, the network modifies the design parameters to improve the system performance. It uses gradient descent method to update the parameters. The input/output data pairs are often called as training data or learning patterns. They are clamped onto the network and functions are propagated to the output unit. The network output is compared with the desired output values. The error measure E^P, for pattern P at the output node in layer 6, may be given as

$$E^P = \frac{1}{2}(T^P - O_6^P)^2 \tag{13}$$

where T^P is the target or desired output and O_6^P the single node output of defuzzification layer in the network. Further, the sum of squared errors for the entire training data set is

$$E = \sum_P E^P = \frac{1}{2}\sum_P (T_P - O_P^6)^2 \tag{14}$$

The error measure with respect to node output in layer 6 is given by delta (δ)

$$\delta = \frac{\partial E}{\partial O_6} = -2(T - O_6) \tag{15}$$

This delta value gives the rate which the output must be changed in order to minimize the error function. The delta value for layer 5 is given as

$$\frac{\partial E}{\partial O_5} = \frac{\partial E}{\partial O_6}\frac{\partial O_6}{\partial O_5} \tag{16}$$

Similarly, for Kth layer, the delta value may be calculated using the chain rule as

$$\frac{\partial E}{\partial O_K} = \frac{\partial E}{\partial O_{K+1}}\frac{\partial O_{K+1}}{\partial O_K} \tag{17}$$

Now, if α is a set of design parameters of the given adaptive network, then

$$\frac{\partial E}{\partial \alpha} = \sum_{O' \in P} \frac{\partial E}{\partial O'} \frac{\partial O'}{\partial \alpha} \tag{18}$$

Where P is the set of adaptive nodes whose output depends on α. Thus, update for parameters α is given by

$$\Delta \alpha = -\eta \frac{\partial E}{\partial \alpha} \tag{19}$$

where η is the learning rate and may be calculated as

$$\eta = \frac{k}{\sqrt{\sum_{\alpha} \left(\frac{\partial E}{\partial \alpha} \right)^2}} \tag{20}$$

Where k is the step size. The value of k must be properly chosen as the change in value of k influences the rate of convergence.

3 Empirical Analysis

3.1 Variables Selection

3.1.1 Variables Selection Principles

(1) It is feasible in China. In the present capital market of China, it is very common and serious for the forgery and fabrication of accounting documents and accounts, and fraudulent financial statement preparation.

(2) The internal credit rating results are taken as benchmarks according to New Basel Capital Accord. All the requirements are taken into consideration in variable selection and econometric model determination so as to make the model practical.

(3) The matured experience and international standards in the same western industry are concerned.

(4) The predictive principle. Variables should reflect the future development trend.

3.1.2 Credit Risk Assessment Index System Establishment

Based on the comprehensive concerns on each impact factor of credit risk, 16 variables are finally determined in the commercial bank credit assessment: (1) Net sales/ Total assets; (2) Total assets turnover; (3) Current assets turnover; (4) Fixed assets turnover; (5) Inventory turnover; (6) Accounts receivable turnover; (7) Current ratio; (8) Operation capital / Total assets; (9) Quick ratio; (10) Conservation quick ratio; (11) Debt ratio; (12) Cost profit rate; (13) Sales net profit rate; (14) Return on assets; (15) Return on net assets; (16) Type of loan.

3.2 Sample Data Acquisition

The sample data from the short term loan of same industry is employed to evade the problem. The data herein is from the credit department of the headquarter of Industrial

and Commercial Bank of China (ICBC), and ICBC Heilongjiang Branch, Nangang Subbranch of Harbin City.

Data acquired retrieval conditions are as follows:

(1) Sample industrial scope: Type C — manufacturing industry;
(2) Loan type: short term loan within 1 year;
(3) Loan granted date: January 1, 1998 to January 31, 1998;
(4) Loan balance closing date: August 13, 2001;
(5) Loan modes: credit, guarantee, mortgage, and pledge according to the General Provisions of Loans of the People's Bank of China;
(6) Loan amount: actual loan granting amount;
(7) Loan balance: loan balance recognized as loss until August 13, 2001;
(8) Loan state: the present state of the loan;
(9) Business full name and code: the only identification code of the business
(10) Financial statements date and statements: corporate balance sheet and income statement on the date of December 31, 1997.

3.3 Model Application and Results Analysis

First, we work on the stability of the samples .Because of the large capacity sample of data, the broader range of indicators and relative smoothness of data, twice or three times standard deviation abnormal test is used to select data, and ultimately 167 sample data is obtained. Then SPSS is used for the factor analysis on the data. These 16 variables were grouped under factor analysis with varimax rotation. The list of financial ratios and the factor loading is summarized in Table 1.

Table 1. The varimax rotated factor matrix

Variable	Factor							
	1	2	3	4	1	2	3	4
1 Net sales / Total assets	0.966				0.195	0.052	−0.033	0.032
2 Total assets turnover	0.966				0.195	0.052	−0.033	0.032
3 Current assets turnover	0.953				0.192	0.022	−0.038	−0.009
4 Fixed assets turnover	0.843				0.170	0.076	−0.029	0.064
5 Inventory turnover	0.819				0.165	0.025	−0.045	0.104
6 Accounts receivable turnover	0.755				0.152	−0.037	−0.014	−0.227
7 Current ratio		0.899			−0.032	0.276	0.009	−0.053
8 Operation capital / Total assets		0.891			−0.040	0.273	0.014	−0.136
9 Quick ratio		0.782			−0.038	0.240	0.004	0.249
10 Conservation quick ratio		0.750			−0.024	0.230	0.007	0.230
11 Debt ratio		−0.566			0.012	−0.174	0.004	0.338
12 Cost profit rate			0.879		0.048	0.005	0.325	0.037
13 Sales net profit rate			0.843		0.050	0.005	0.312	0.045
14 Return on assets			0.823		0.009	0.009	0.304	−0.117
15 Return on net assets			0.694		0.007	−0.022	0.257	−0.065
16 Type of loan				0.821	−0.004	−0.028	0.037	0.786

It can be seen from table 1, 16 variables are divided into four interpretation factors, and the detailed results used as sample data of this paper (We have selected 100 observations as training samples and 67 observations as testing samples.). The network's second layer is fuzzy layer, and its role is to fuzzy the input variables. Fuzzy processing is the transformation of input in form of digit into the sequence of a specified code often expressed in a language value. This paper chooses Gaussian function.

From the above, we can see that network is a 6-layer network structure where the input layer has four nodes, the output layer one node. The parameters of network research are as follows:

(1) Epochs: 300;
(2) Goal of error: 0.01;
(3) Learning algorithm: hybrid learning method (a combination of least square estimation and back propagation for membership function parameter estimation);
(4) Use noise measurement to collect data;
(5) The weight of each rule is 1;
(6) The training data is 100 observations, and the testing data is 67 observations;

Editor GUI of Fuzzy Logic Toolbox ANFIS in MATLAB 7.0 is used to train the fuzzy neural network. After training, we get a real fuzzy neural network structure which is a complicated fuzzy neural network consisting of four inputs, one output, a fuzziication layer of 12 neurons, and a inference layer of 81 rules . the structure of the fuzzy neural network is shown in Fig 1:

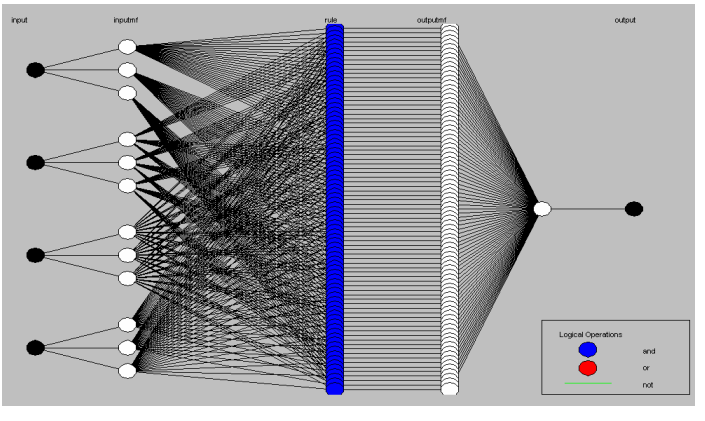


Fig. 1. Structure of fuzzy neural network

The Camber observer in GUI tools is used to show the dependence between one output and one or two inputs. That is to say that it generates or draws the output Camber mapping. As Input 1, 2 and 3, 4 are independent with each other, so Fig 2 can be formed. It is about Input 1 and 2 and the Camber map of output function after training (Figure on the left) and Input 3 and 4 and the Camber map of output function after training (Figure on the right).

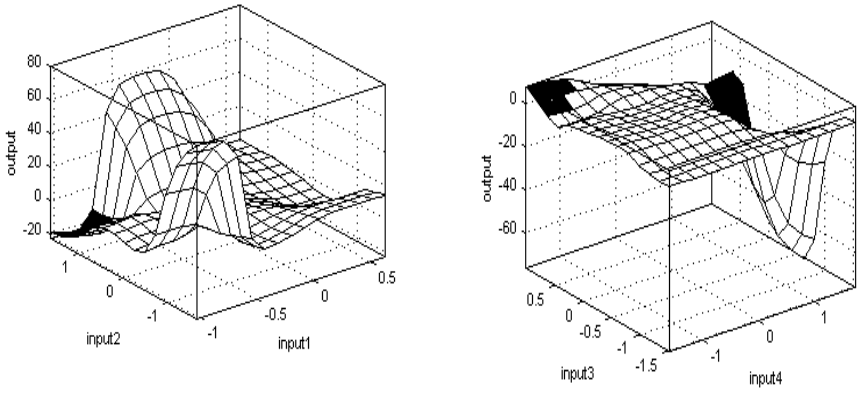


Fig. 2. Function Camber after training

Through rule editor, 81 fuzzy rules are generated. Because of the limited space, not all will be listed in detail. The following procedures can be got with MATLAB. We forecasted 67 test observations (appendix table 2) and got the average forecast error 0.0233. For the error curve are shown in Fig 3.

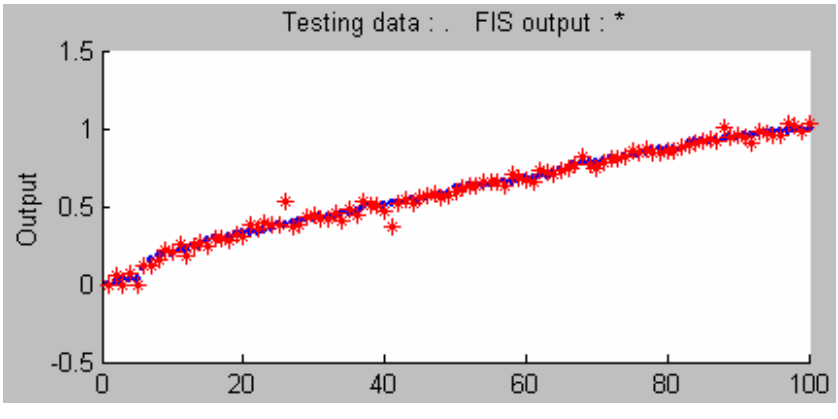


Fig. 3. Error curve after training

Seen from the results of training, the error is very small, and the results of training is satisfactory, so we can see that fuzzy neural network method is very suitable for assessing credit risk of commercial banks.

4 Conclusions

In this paper, indicator system to assess credit risks of commercial has been established; a credit risk assessment model of commercial banks based on the fuzzy neural network has been established. The results of the empirical study have shown that FNN is very suitable for assessing credit risk of commercial banks. There are few

errors in the prediction of network, and fuzzy rule layer also has the regulating ability on specific issues, which is superior to the characteristics of a complete black-box operation, and is suitable for credit risk assessment of commercial banks.

Acknowledgements. This work is partially supported by the grants from the National Natural Science Foundation of China (NSFC No: 70773029).

References

1. Wang, Y.Q., Wang, S.Y., Lai, K.K.: A New Fuzzy Support Vector Machine to Evaluate Credit Risk. IEEE Transactions on Fuzzy Systems 13, 820–831 (2005); Lecture Notes in Artificial Intelligence, vol. 4099, pp. 980–984 (2005)
2. Lai, K.K., Yu, L., Wang, S.Y., Zhou, L.G.: Credit Risk Analysis Using a Reliability-Based Neural Network Ensemble Model. In: Kollias, S.D., Stafylopatis, A., Duch, W., Oja, E. (eds.) ICANN 2006. LNCS, vol. 4132, pp. 682–690. Springer, Heidelberg (2006)
3. Lai, K.K., Yu, L., Wang, S.Y., Zhou, L.G.: Neural Network Metalearning for Credit Scoring. In: Huang, D.-S., Li, K., Irwin, G.W. (eds.) ICIC 2006. LNCS, vol. 4113, pp. 403–408. Springer, Heidelberg (2006)
4. Ohlson, J.A.: Financial Ratios and the Probabilistic Prediction of Bankruptcy. Journal of Accounting Research 18(1), 109–131 (1980)
5. Platt, H.D., Platt, M.B.: Development of a Class of Stable Predictive Variables: the Case of Bankruptcy Prediction. Journal of Business Finance and Accounting 17(1), 31–51 (1990)
6. Tseng, F.M., Lin, L.: A Quadratic Interval Logit Model for Forecasting Bankruptcy. OMEGA: The International Journal of Management Science 33(1), 85–91 (2005)
7. Zmijewski, M.E.: Methodological Issues Related to the Estimation of Financial Distress Prediction Models. Journal of Accounting Research 22, 59–82 (1984)
8. Min, J.H., Lee, Y.C.: Bankruptcy Prediction Using Support Vector Machine with Optimal Choice of Kernel Function Parameters. Expert Systems with Applications 28(4), 603–614 (2005)
9. Salcedo-Sanz, S., Fernandez-Villacanas, J.L., Segovia-Vargas, M.J., Bousono-Calzon, C.: Genetic Programming for the Prediction of Insolvency in Non-life Insurance Companies. Computers and Operations Research 32(4), 749–765 (2005)
10. Tam, K.Y.: Neural Network Models and the Prediction of Bankruptcy. OMEGA: The International Journal of Management Science 19(5), 429–445 (1991)
11. Tam, K.Y., Kiang, M.Y.: Managerial Applications of Neural Networks: The Case of Bank Failure Predictions. Management Science 38(7), 926–947 (1992)
12. Molhotra, R., Malhotra, D.K.: Evaluating Consumer Loans Using Neural Networks. Omega 31, 83–96 (2003)
13. Chen, M.C., Huang, S.H.: Credit Scoring and Rejected Instances Reassigning through Evolutionary Computation Techniques. Expert System with Applications 24, 433–441 (2003)

Multi-information Fusion and Identification System for Laser Welding

Ming Zhou, Wenzhong Liu, and Lei Wan

Department of Control Science and Engineering
Huazhong University of Science and Technology
Wuhan 430074, China
zhouming@hust.edu.cn

Abstract. In this paper, we proposed a laser welding quality monitoring system based on vector machine, which uses the light and sound sensors access to the various signals of the welding and Gabor transform features to extract the vector. The basic features of Laser welding is the formation of small holes, pool and photo-induced plasma, and the result is quality of the welding of light, sound and the potential of poor signal. On this basis, making use of the advantage and objectivity of machine leaning, SVM classification and more comprehensive information to determine the parameters of laser welding quality of the welding seam.

Keywords: SVM, Laser welding, Feature extraction.

1 Introduction

Compared with the traditional welding, the Laser welding and soldering with high-performance, high-quality and other advantages is widely used in the electronics, automotive, aviation and other industry.. However, because the possibility that some accidental instability leads to change in the state of welding, consequently welding defects occur, it will not be able to meet the quality requirements of the modern industrial process. The real-time monitoring of the quality is of great significance during the laser welding process [1, 2].

The basic features of Laser welding process is the formation of small holes, pool and photo-induced plasma production, which result in characterization of the welding quality of light, sound and the potential of poor signal [3, 4]. This paper takes the plasma reflect the state of visible light, infrared light and radiation bath plasma welding quality of the voice signal to monitor the signal of the laser welding process. Through appropriate algorithm which can extract the features of the vector, and with the help of computer to establish a set of real-time monitoring system for laser welding.

SVM (Support Vector Machine) is proposed by V.Vapnik of AT & TBell Laboratory. SVM is proposed for the classification and regression problems of statistical learning theory [5]. Since the SVM has many advantages and promising experimental performance, it draws more and more attention and become a hot area of research [6-8].

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 986 – 992, 2009.

This article introduces the support vector machine (SVM) related to the theory and its application in laser welding process of classification.

2 Support Vector Machine

The machine learning which based on data is very important of modern intelligence technology. The main method is finding the law which is use to observe or predict of the future.

Support Vector Machine method is based on the VC-dimensional of the statistical learning theory and the principle of minimum risk. The SVM obtain the best way between the accuracy and capacity in order to obtain the best ability to promote (Generalization Ability).

SVM from the linear case can be divided into the optimal development of surface classification; the basic idea can be discussed by Fig. 1. In Fig. 1, solid and hollow-point on behalf of the two types of samples, H is the classification for the line, H1, H2 are various types of classification from the line in the most recent samples and parallel to the classification of straight lines, the distance between them is called interval classification (Margin). The best classification line requires that the classification line not only the right to separate two types of (training error rate of 0), but also between the largest classification. Classification equation for the line is $x \cdot w + b = 0$. According to normalizing, we can make on the linear set of samples: $(x_i, y_i), i = 1, \ldots, \mathrm{n}, \mathrm{x} \in \mathrm{R}^{\mathrm{d}}$, where $y \in \{+1, -1\}$, satisfy:

$$y_i \left[(w \cdot x_i) + b \right] - 1 \geq 0, \quad i = 1, \cdots, n. \tag{1}$$

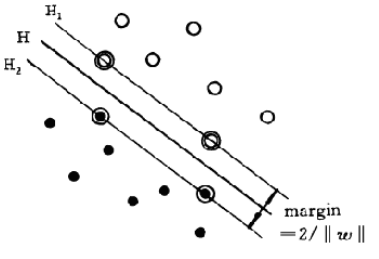


Fig. 1. The best line in linear classification used in Support Vector Machine

The interval of the classification is $2/\|\mathrm{w}\|$. The maximum interval is equivalent to the minimum of $\|\mathrm{w}\|^2$. The classification surface which satisfies the conditions (1) and makes the minimum $\frac{1}{2}\|\mathrm{w}\|^2$ is the best classification surface, and the samples of H1 and H2 are called Support Vector. Get the maximum intervals of the classification is to get the maximum of the capacity and this is one of the core idea of SVM.

For the non-linear conditions, the X samples can be mapped to a high-dimensional feature space H, and then use the functions of the original space to convert it to be a linear problem to another space. According to the relevant functional theory, as long as a core function can satisfy the conditions of Mercer, it corresponds to a inner product of a space.. Therefore the classification of the non-linear problem can be implemented by adopting appropriate inner product. In a nutshell, first the SVM transforms the non-linear to a high-dimensional space by the appropriate inner product, and then get the optimal classification of the surface.

SVM classification function is similar to the form of a neural network, the output is the linear combination of the middle nodes, each node in the middle is correspond to a support vector, as shown in Fig. 2.

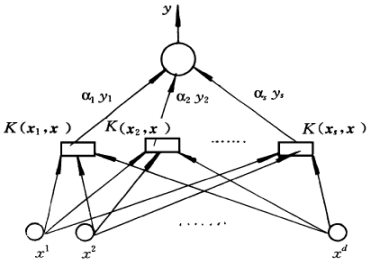


Fig. 2. Support Vector Machine (SVM) structure used in lasser welding indentification

The output (decision-making rules):

$$y = \mathrm{sgn}\left(\sum_{s} \alpha_i y_i K\left(x_i, x\right) + b \right) \tag{2}$$

And the weight variable $\alpha_i y_i$ is the inner product of $x_1, x_2, \cdots, x_s$ which support s vectors.

$$x = \left(x^1, x^2, \cdots, x^d \right).$$

Obviously it ensures that all the samples are classified appropriately, that's to say get the best performance of the promotion when the experience is 0 for Remp. If you want to experience the risks and seek to promote some kind of performance among all values through the introduction of the relaxation factor is wrong to allow the existence of sub-samples. At this time, bound by (1) change to

$$y_i \left[\left(w \bullet x_i \right) + b \right] - 1 + \xi_i \geq 0, \qquad i = 1, \cdots, n \,. \tag{3}$$

At the same time, add a punishment $C\sum_{i=1}^{n}\xi_i$ to the target which is obtain the smallest $\frac{1}{2}\|w\|^2$, and $C\sum_{i=1}^{n}\xi_i$ geometric loss of the wrong classification, which reflects the structural risk minimization (SRM) ideas.

3 Implementation

For the monitoring of the welding process, the researchers are focusing how to handle the signals between the melting and non-melting of the wielding line. Because the complexity of the welding we analyze the signal with Gabor transformation, then classify the state of the wielding line by SVM.

3.1 Extraction of Feature Vector

The difference of signal between the melting and non-melting states can be describe in two fields. One is the signal intensity of the statistical value of a difference; another is there are differences in the frequency of the signal. Here, take the blue-violet light as an example to extract feature vector. The waveform of Blue-violet light signal after filtering shown in Fig. 3.

In order to get the signal features of the frequency band, apply time-frequency analysis to the signal.. The signal which is transformed by Gabor transform is shown in Fig. 4.

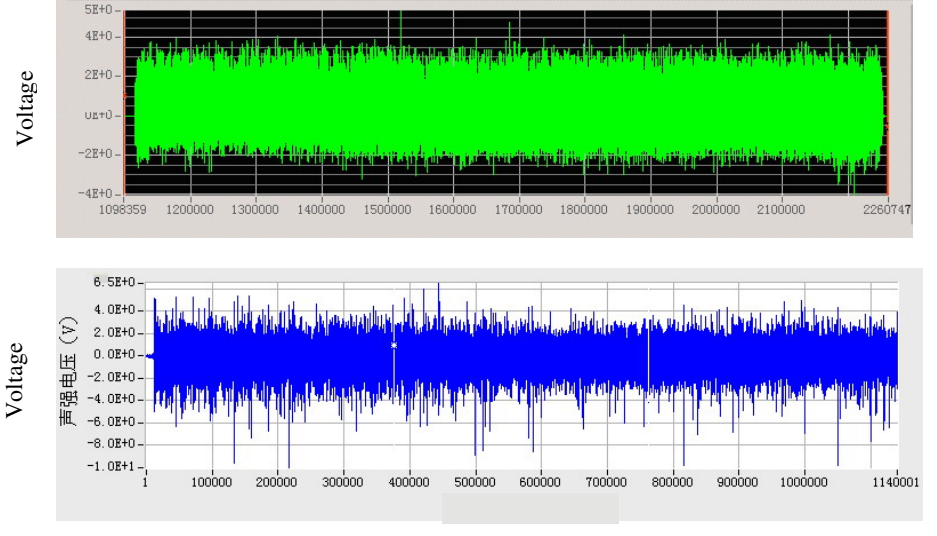


Fig. 3. The waveform of Blue-violet light for laser welding (unit: V)

According to Fig. 4, we can see that the blue-violet light has more energy in 2 ~ 3 kHz. It can be speculated that blue-violet light signals in the frequency of paragraph 2 ~ 3 kHz possible representation to the quality of the welding features.

Similarly, the frequency features of infrared light, as well as voice signals can be extracted. And combined with the mean, variance, as well as welding technology, and other parameters, so each of the selected feature vector takes 9 attributes as input vector of SVM, that's to say enter a 9-dimensional features Vector. As the difference between the feature vectors, it is necessary to map prior to the value of the property so normalized so that it is in a range of the same size, reduce the influence of individual features of the vector model.

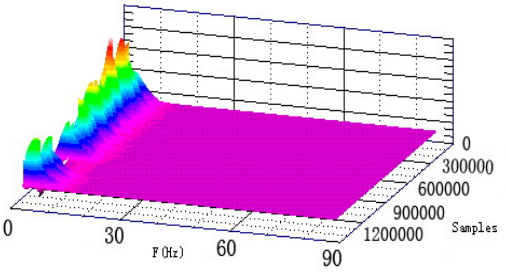


Fig. 4. The voice signal of laser welding after Gabor transformation

3.2 SVM Algorithm and Parameters

The different kernel functions of SVM will form different algorithms, currently there are three types of kernel functions:

①polynomial kernel function： $K(\mathbf{x}, \mathbf{x}_i) = \left[(x \bullet x_i) + 1\right]^q$

②gauss kernel function(RBF)： $K(\mathbf{x}, \mathbf{x}_i) = \exp\left\{-\frac{|x - x_i|^2}{\sigma^2}\right\}$

③Sigmoid kernel function： $K(x, x_i) = \tanh\left(v(x \bullet x_i) + c\right)$

Kernel functions are corresponding to the mapping function, and feature space. So kernel function and the parameters of them will change the complexity (dimensions) of the sub-space. And the dimensions of the sub-space decides the dimension of maximum VC thus it decides minimum error experience of the linear classification. Therefore choose the right kernel function is very important to promote their well-SVM classifier.

The punishment parameter C is used to determined the regulation of machine learning the scope of confidence and experience of sub-space, so that the proportion of the risk of machine learning to promote the ability of the best. Parameter C is different in different sub-optimal space, in the sub-space the error punishment is small when parameter C is small, the complexity of machine learning but the experience of

a small risk is great, and vice versa. For each sub-space, at least there is a suitable C makes the ability to promote the best, when the C above a certain value, SVM complexity reached. The experiences of the risks almost have no ability to promote change. However, at present there is no uniform method to determine the best C value. General common and more effective way is through continuous experiment to select the C to be satisfied with the results.

Currently the choice of kernel function are based on experience, the some a use repeated experiment to get them. Kernel choice becomes a dedicate research field called model selection, the choice of the kernel functions and parameters are the key of the SVM.

4 Simulation and Analyze

The kernel function of the paper is radial basis function, the rules for decision-making at this time (reference to the type (2))

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{N} \alpha K_{\gamma}\left(\left|x - x_i\right|\right) + b\right). \tag{4}$$

$K_{\gamma}\left(\left|x - x_i\right|\right)$ depends on the distance between the two vector, for a fixed γ, is a monotone function of the non-negative, and tends to infinity when $\left|x - x_i\right|$ tends to zero. According to the topic of the actual situation of this paper, and through a large number of experiments we get the appropriate punishment factor C and the width of the kernel parameters γ.

In order to verify the accuracy of SVM classification, we extracted a sample of 52 of the weld, where the penetration and non-state penetration 50% of each state. Arbitrary selecting 26 samples for the training model, the remaining 26 samples for testing model. In this case, the identification of the penetration obtains good results, and for the identification of the same seam has reached 100%.

For different Weld circumstances, we selected 4 typical welds and each seam select a 26 samples of the total 104 samples for the training model, and the penetration and non-state penetration are 50% of each state. Then select another to weld to verify the model, tested the performance of the model shown in Fig. 5.

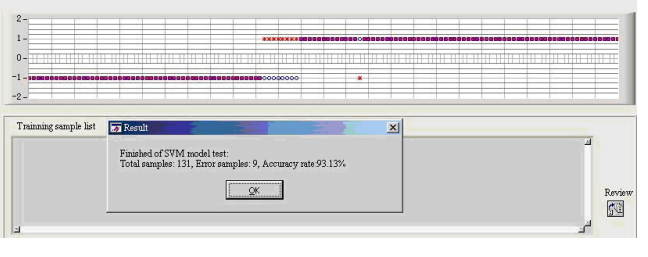


Fig. 5. The classification ability of SVM in several weld seams

It has a high requirement because the feature extraction and recognition are based on different parameters. In a test of 131 samples, there are 9 miss identifications. Compare with single weld recognition, the correct rate of classification decreases from 100% to 93.1%. Of course, after the algorithm is optimized for integration, it's able to get a better correct rate.

5 Conclusions

The statistical learning theory and support vector machine found a theory framework and general method for the machine learning of the limited samples. It is the fastest growing field of the research on the basis of theory.

Support vector machines is high precision, fast and ability to promote strong, especially in small samples in the classification reflects the neural network as opposed to a huge advantage. SVM algorithm, of course, due to the development of a relatively short time, there are still many unresolved issues in the application is started relatively late. For example, the system will be able to continue to optimize the standard of SVM algorithm to make some improvement. From the application perspective, through a large number of experiments to verify the feature vectors which attributes to the final status of the formation to play a key role, together with its on a larger weighted factor, to attribute the difference to the value of the final form Weld's status.

References

1. Duan, A.Q., Hu, L.J., Wang, Y.J.: Research on Laser Weld Penetration Monitoring with Laser Induced Plasma Signals. China Welding 02 (2004)
2. Sun, A., Kannatey, J.E., Cartner, M.: Laser Weld Porosity Monitoring Using Sensor Fusion. In: Proceedings of Mechatronics Forum International Conference, Georgia (2000)
3. Foley, D., Ali, A.: Laser Weld Penetration with Multiple Emission Signal Measurements. Journal of Laser Applications 11, 47–53 (1999)
4. Allen, S.S.: Multiple Sensor Monitoring of Laser Welding Michigan, University of Michigan, pp. 9–29 (2001)
5. Valdimir, N.: Vapnik: The Nature of Statistical Learning Theory. Springer, New York (2000)
6. National Instruments. Signal Processing Toolset User Manual. Austin: National Instruments Corporation (2001), http://www.ni.com
7. Filip, M.: Vapnik-Chervonenkis (VC) Learning Theory and Its Applications. IEEE Trans. on Neural Networks. 10(5), 985–987 (1999)
8. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley-Interscience, Hoboken (2000)

Integrating Generalized Linear Auto-Regression and Artificial Neural Networks for Coal Demand Forecasting

Ping Yao

School of Economics & Management, Heilongjiang Institute of Science and Technology, Harbin 150027, China

Abstract. In this study, a novel combined forecasting model integrating generalized linear auto-regression (GLAR) with artificial neural networks (ANN) was proposed to obtain accurate prediction results and ameliorate forecasting performances. We compare the performance of the new model with the two individual forecasting models-GLAR and ANN. Empirical results obtained reveal that the prediction using the combined model is generally better than those obtained using the individual model of GLAR and ANN in terms of the same evaluation measurements. Our findings reveal that the combined model proposed here can be used as an alternative forecasting tool for coal demand to achieve greater forecasting accuracy and improve prediction quality further.

Keywords: GLAR, ANN, Combined model, Forecasting, Coal demand.

1 Introduction

Coal demand modeling and forecasting has been a common research stream in the last few decades. Over this time, the research steam has gained momentum with the advancement of computer technologies, which have made many elaborate computation methods available and practical. However, it is not easy to predict coal demand due to its high volatility and noise. But the difficulty in forecasting coal demand is usually attributed to the limitation of many conventional forecasting models; this has encouraged academic researchers and business practitioners to develop more predictable forecasting models. As a result models using artificial intelligence such as artificial neural network (ANN) techniques have been recognized as more useful than conventional statistical forecasting models.

PENG [1] simulated the main factors of influencing the coal consumption and employed the artificial neural network method to forecast the coal consumption; WANG & SUN [2] set up GM(1,3) model based on grey system theory to forecast coal demand.; NING [3] gave a forecast of annual increase ratio of coal consumption by using superposition wavelet-neural network model. Recently, more hybrid forecasting models have been developed that integrate neural network neural network techniques with many conventional and burgeoning forecasting methods such as econometrical models and time series models to improve prediction accuracy.

This study proposes a novel nonlinear ensemble forecasting model for coal demand forecasting. This model utilizes the NN technique and PCA technique, integrates

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 993 – 1001, 2009.

GLAR with NN models, and takes full advantage of hybrid methods and combined techniques. The proposed novel nonlinear ensemble model is the PCA-NN nonlinear ensemble forecasting approach, which integrates GLAR and NN models as well as the single hybrid methodology. The aims of this study are two aspects: (1) to show how to predict coal demand using the proposed ensemble model and (2) to display how various methods compare in their accuracy in forecasting coal demand.

2 Basic Idea of Methods

The GLAR and the neural network BP models are summarized in the following as foundation to describe the ensemble forecasting model.

2.1 Generalized Linear Auto-Regression (GLAR) Model

The generalized linear auto-regression (GLAR), first introduced by Shephard [4], is equivalent to a system of reduced form equations relating each endogenous variable to lag endogenous (predetermined) and pertinent exogenous variables. That is, in a GLAR model, the future value of a variable is assumed to be a linear function of several past observations and random errors. Generally, the form of the GLAR model is given in the following.

$$y_t = \alpha + C(L)y_{t-1} + D(L)x_{t-1} + \varepsilon_t \tag{1}$$

Where y_t and ε_t are the actual value of endogenous variables and random disturbance at time t, x_{t-1} is the actual value of related exogenous variables, α is a constant term, $C(L)$ and $D(L)$ are the pth and qth order lag polynomial of endogenous variable and exogenous variables respectively in the lag operator L, such that $C(L)=C_0-C_1L-C_2L^2-\ldots-C_PL^P$ and $D(L)=D_0-D_1L-D_2L^2-\ldots-D_qL^q$. L denotes the lag operator, e.g., $Ly_t=y_{t-1}$, $L^2y_t=y_{t-2}$ and so on, and random disturbances, ε_t are assumed to be independently and identically distributed with a mean of zero and a constant variance of σ^2, i.e. $\varepsilon_t \sim \mathrm{IID}(0, \sigma^2)$.

Based on earlier works [4, 5, 6], the GLAR model involvers the following five-step iterative procedures:

(1) Stationary test of time series, such as unit root test;
(2) Identification of the GLAR structure, i.e. the model order is determined;
(3) Estimation of the unknown parameters;
(4) Model checks and diagnostics;
(5) Forecast future outcomes based on the known data.

This five-step model building process is typically repeated several times until a satisfactory model is finally selected. The final model selected can then be used for prediction purposes. The advantage of GLAR is that it is capable of receiving external information from exogenous variables. Furthermore, the GLAR model fits the linear characteristic of time series well. The disadvantage of the GLAR is that it cannot capture nonlinear patterns coal demand time series if nonlinearity exists.

2.2 Neural Network Approach

The BP neural network consists of an input layer, an output layer and one or more intervening layers also referred to as hidden layers. The hidden layers can capture the nonlinear relationship between variables. Each layer consists of multiple neurons that are connected to neurons in adjacent layers. Since these networks contain many interacting nonlinear neurons in multiple layers, the networks can capture relatively complex phenomena [7, 8].

A neural network can be trained by the historical data of a time series in order to capture the characteristics of this time series. The model parameters (connection weights and node biases) will be adjusted iteratively by a process of minimizing the forecast errors. For each training iteration, an input vector, randomly selected from the training set, was submitted to the input layer of the network being trained [9]. The output of each processing unit (or neuron) was propagated forward through each layer of the network, using the equation

$$\mathrm{NET}_t = \sum_{i=1}^{N} w_{ti} x_i + b_i \tag{2}$$

Where NET_t is an output of unit t; w_{ti} is the weight on connection from the ith to the tth unit; x_i is an imput data from unit i (input node) to t; b_t denotes a bias on the tth unit; and N is the total number of input units. A bias or activation of a proper magnitude can affect output activation in the same manner as imposing a limit on the network mapping function.

A sigmoid transformation was then applied to the summation for each unit in a hidden layer, using the equation

$$y_t = f(\mathrm{NET}_t) = 1/(1+e^{-\mathrm{NET}_t}) \tag{3}$$

The activity of each output unit was also computed by Eq.(2), using the weights on the connections from the last hidden layer. But, unlike any output from the hidden unit activity, NET_j was not transformed by the sigmoid function. An error $\delta_j^{(L)}$ for the jth output unit was calculated by

$$\delta_j^{(L)} = T_j - \mathrm{NET}_j \tag{4}$$

Where L denotes the number of the output layer (Eq.(4)). For example, L=3, j=1,2,…,k, k is the number of output unit, and T is the target or the desired activity of the output unit. This error was propagated back to the lower hidden layers as follows:

$$\delta_t^{(l)} = \sum_{i=1}^{N} \delta_i^{l+1} w_{ti}^{(l)} f'(\mathrm{NET}_t^{(l)}) \tag{5}$$

Where $w_{ti}^{(l)}$ is the weight from the ith unit in layer l to the tth unit in layer (l+1), l=1,2,…,L-1, and $f'(\cdot)$ is the first derivative of the sigmoid function.

In order for the network to learn, the value of each weight had to be adjusted in proportion to each unit's contribution to the total error in Eq.(5). The incremental change in each weight for each learning iteration was computed by (Eq.(6) and (7))

$$\Delta w_{ti}^{(l)} = c_1 \delta_t^{l+1} w_{ti}^{(l)} f'(\mathrm{NET}_t^{(l)}) + c_2 m_{ti}^{(l)} \tag{6}$$

where c_1 is a learning constant that controls the rate of learning; c_2 is a positive constant that, being less than 1.0, is the momentum term to smooth out the weight changes, and

$$m_{ti}^{(l)} = \Delta w_{ti}^{(l-1)} \tag{7}$$

The procedures for developing the neural network BP model are as follows [10]:

(1) Normalize the learning set;
(2) Decide the architecture and parameters: i.e., learning rate, momentum, and architecture. There are no criteria in deciding the parameters except on a trial-and-error basis;
(3) Initialize all weights randomly;
(4) Training, where the stopping criterion is either the number of iterations reached or when the total sum of squares of error is lower than a pre-determined value;

The major advantage of NN models is their flexible nonlinear modeling capability. They can capture the nonlinear characteristics of time series well. However, using NN to model linear problems may produce mixed results [11, 12, 13]. Therefore, we can conclude that the relationship between GLAR and NN is complementary. To take full advantage of the individual strengths of two models, it is necessary to integrate the GLAR and NN models.

2.3 The Hybrid Methodology Integrated GLAR with ANN

Time series forecasting problem such as coal demand prediction is far from simple due to high volatility, complexity, and irregularity and noisy out environment. Furthermore, real-world time series are rarely pure linear or nonlinear. They often contain both and nonlinear patterns. If this is the case, there is no omnipotent model that is suitable for all kinds of time series data. Although both GLAR and NN models have achieved success in their own linear or nonlinear domains, neither GLAR nor NN can adequately model and predict time series since the linear models cannot deal with nonlinear relationships while the NN model alone is not able to handle both linear and nonlinear patterns equally well. On the other hand, as previously mentioned, for time series forecasting the relationship between GLAR and NN is complementary. GLAR is a class of linear models that can capture time series' linear characteristics, while NN models trained by back-propagation with hidden layers are a class of general function approximates capable of modeling nonlinearity and which can capture nonlinear patterns in time series. Commixing the two models may yield a robust method, and more satisfactory forecasting results may be obtained by incorporating a time series model and a NN model. Therefore, we propose a hybrid model integrating GLAR and NN for coal demand forecasting. In view of the works of [14], the proposed hybrid model is a two-phase forecasting procedure, which incorporating GLAR model with NN model in an adaptive manner. The motivation of hybrid methodology is to create a synergy effect that further improves the prediction power.

In the first phase, a GLAR model is used to fit the linear component of time series, which is assumed to be $\{y_t, t = 1,2,\cdots\}$, and generate a series of forecasts, which is defined as $\{\hat{L}_t\}$. However, since some time series, such as coal demand, often contain more complex patterns than those of linear regression models also need to discover the nonlinear relationship of the time series in order to improve prediction accuracy. By comparing the actual value y_t of the time series and forecast value $\hat{L}_t$ of the linear component, we can obtain a series of nonlinear components, which is assumed to be $\{e_t\}$, that is,

$$e_t = y_t - \hat{L}_t \tag{8}$$

Thus, a nonlinear time series is obtained. The next step is how to fit the nonlinear component of series.

The second phase of the hybrid methodology, a neural network model is used to model the above nonlinear time series. By training the NN model using previously generated nonlinear time series as inputs, the trained NN model is then used to generate a series of forecasts of nonlinear components of time series, defined by $\{\hat{N}_t\}$.

In order to obtain the synergetic forecast results, the final forecasting results, defined by $\{\hat{y}_t\}$,are calculated as

$$\hat{y}_t = \hat{L}_t + \hat{N}_t \tag{9}$$

In summary, the proposed hybrid methodology consists of two phases or four steps as follows:

(1) A GLAR model should be identified and the corresponding parameters should be estimated;
(2) The nonlinear components are computed from the GLAR model;
(3) A neural network model is developed to model the nonlinear components;
(4) Then combined forecast results are obtained from Eq.(9).

2.4 Forecasting Evaluation Criteria

Yokum and Armstrong [15] conducted an expert opinion survey about evaluation criteria to select forecasting techniques. Clearly, accuracy was the most important criterion, with the other one being the cost savings generated from improved decisions. In addition, execution issues such as ease of interpretation and ease of use were also highly rated. In this study, three criteria of forecasting accuracy were used to make comparisons of the forecasting capabilities.

The first measurement is the mean square error (MSE; Eq.(10)),

$$\text{MSE} = \frac{1}{T}\sum_{t=1}^{T}(\hat{y}_t - y_t)^2 \tag{10}$$

Where $\hat{y}_t$ is the predicted value at time *t*; y_t is the actual value at time *t*; and *T* is the number of predictions.

The second criterion is the mean absolute error (MAE; Eq (11)),

$$\mathrm{MAE} = \frac{1}{T}\sum_{t=1}^{T}\left|\hat{y}_t - y_t\right| \tag{11}$$

The third criterion is the mean absolute percentage error (MAPE; Eq (12)),

$$\mathrm{MAPE} = \frac{1}{T}\sum_{t=1}^{T}\left|\frac{\hat{y}_t - y_t}{y_t}\right| \tag{12}$$

3 Empirical Study

3.1 Data Description

On the consideration of the economic development, characteristics of the energy structure and changes of the coal consumption, we choose three exogenous variables: GDP growth rate (%), coal proportion of the energy consumption (%) and coal

Table 1. Variable values and prediction results

Year	Domestic Coal demand	GDP growth rate	Coal proportion of the energy consumption	Coal consumption growth rate	GLAR model	ANN model	Combined model
1979	585.16	7.60	71.31	3.45	___	___	___
1980	606.24	7.80	72.15	3.60	___	___	___
1981	605.65	4.50	72.47	-0.10	633.37	609.20	605.60
1982	641.26	8.50	73.67	5.88	613.26	626.20	641.00
1983	674.85	10.20	74.16	5.24	681.95	691.10	674.40
1984	744.18	15.20	75.27	10.27	714.17	732.40	743.20
1985	814.16	13.50	75.81	9.40	816.62	816.80	815.90
1986	861.89	8.80	75.83	5.86	888.12	845.10	862.30
1987	920.40	11.60	76.21	6.79	916.46	932.20	916.20
1988	982.76	11.30	76.17	6.78	985.46	959.80	986.80
1989	1000.42	4.10	75.81	1.80	1052.13	1002.70	1000.60
1990	1025.55	3.80	76.20	2.51	1029.77	1009.30	1025.50
1991	1104.32	9.20	76.10	7.68	1061.96	1092.70	1103.80
1992	1140.85	14.20	75.70	3.31	1190.10	1153.50	1140.80
1993	1209.93	13.50	74.60	6.06	1189.08	1207.90	1210.90
1994	1286.83	11.80	75.00	6.36	1288.40	1266.40	1282.80
1995	1315.24	10.50	74.60	2.21	1373.38	1345.90	1315.70
1996	1345.93	9.60	74.70	2.33	1358.50	1324.80	1345.60
1997	1310.70	8.80	71.50	-2.62	1391.66	1277.00	1310.60
1998	1294.90	7.80	69.60	-1.21	1296.59	1294.20	1294.90
1999	1263.56	7.10	67.10	-11.19	1298.07	1261.00	1263.70
2000	1245.37	8.00	67.80	8.26	1252.32	1247.40	1245.60
2001	1262.11	7.50	66.70	1.37	1245.73	1263.30	1262.10
2002	1366.05	8.30	66.30	8.24	1294.15	1394.50	1366.00
2003	1637.32	9.50	68.40	19.83	1478.01	1676.00	1637.00
2004	1890.00	9.50	68.00	13.90	1903.61	1899.00	1890.10

consumption growth rate (%). These three exogenous variable values from 1979 to 2004 are listed in table 1, in order to save space, the prediction results are also listed in table 1.

3.2 Empirical Results

In this study, all GLAR models are implemented via the Eviews software package, which is produced by Quantitative Micro Software Corporation. The individual ANN model, the combination model with minimum-error method is built using the Matlab software package, which is produced by Mathworks Laboratory Corporation. In the ANN model, a three-layer BPNN with the architecture of 3-6-1 is used. That is, it has 3 input neurons, 6 TANSIG neurons in the hidden layer and one PURELIN neuron in the output layer. The network training function is the TRAINSCG. Besides, the learning rate and momentum rate is set to 0.15 and 0.30. the accepted average squared error is 0.005 and the training epochs are 3000. The above parameters are obtained by trial and error.

Fig. 1 give graphical representations of the forecasting results for coal demand using different models. Table 2 show the forecasting performance of different models from different perspectives. Form the graphs and tables; we can generally see that the forecasting results are very promising under the different forecasting evaluation criteria. Table 3 shows the coal demand forecasting of China from 2005 to 2013.

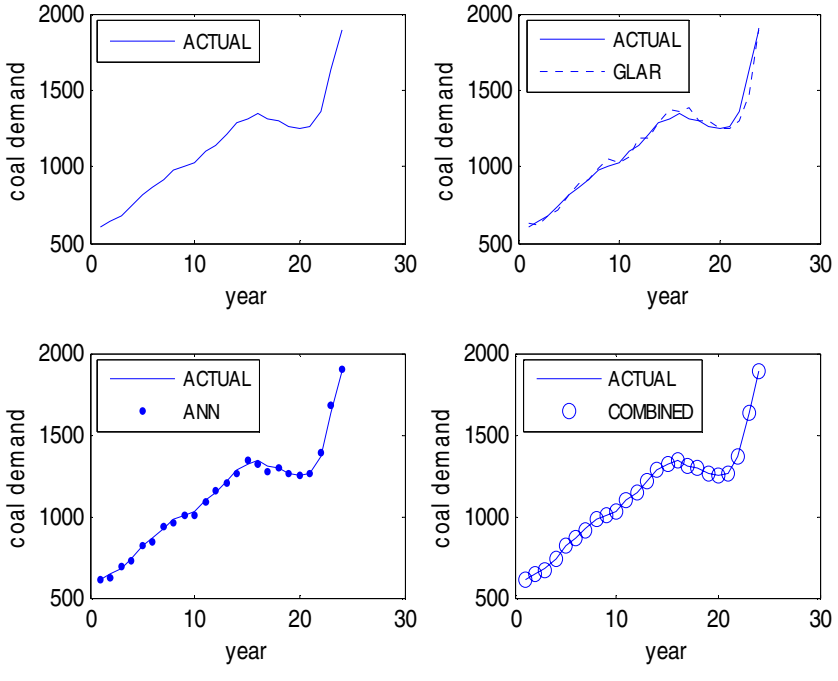


Fig. 1. A graphical comparison of coal demand forecasting results using different models (1981-2004)

Table 2. A comparison of error between different methods

Method	GLAR	ANN	Combined model
MSE	2204.4520	312.6939	2.3537
MAE	31.4225	13.9246	0.8196
MAPE	0.0273	0.0127	0.0008

GLAR: generalized linear auto-regression; ANN: artificial neural network;
MSE: mean square error; MAE: mean absolute error; MAPE: mean absolute percentage error.

Table 3. Coal demand forecasting of China. from 2005 to 2013

Year	2005	2006	2007	2008	2009	2010	2011	2012	2013
Coal demand	2003.5	2095.3	2192.0	2293.6	2400.1	2511.2	2627.3	2794.7	28762

4 Conclusions

The authors proposed to use a hybrid model that combines the time series GLAR model and the neural network BP model to predict coal demand of China. The results showed that the combined model is superior to the GLAR and ANN model. The MSE, MAE and MAPE were all the lowest for the combined model, indicating that the combined forecasting model can be used as a viable alternative solution for coal demand prediction.

References

1. Ning, Y.C.: A Superposition Wavelet-Neural Network Model of Coal Demand Forecast. Journal of China Coal Society 28(1), 108–112 (2003)
2. Wang, L.J., Sun, J.H.: Coal Required Quantity Forecasting Based on Grey System Theory. Journal of China Coal Society 27(3), 333–336 (2002)
3. Peng, J.L.: Study on Prediction of the Coal Consumption Based on the Method of Artificial Neural Network. Journal of China University of Mining & Technology 26(3), 63–67 (1997)
4. Shephard, N.: Generalized Linear Auto Regressions. Economics Working Paper 8, Nuffield College, Oxford (1995)
5. Firth, D.: Generalized Linear Models. In: Hinkley, D.V., Reid, N., Snell, E.J. (eds.) Statistical Theory and Modeling, pp. 55–82. Chapman & Hall, London (1990)
6. McCullagh, P., Nelder, J.A.: Generalized Linear Models, 2nd edn. Chapman & Hall, London (1989)
7. Hill, T., O'Connor, M., Remus, W.: Neural Network Models for Time Series Forecasts, Manage. Sci. 42(7), 1082–1092 (1996)
8. Chiang, W.C., Urban, T.L., Baldridge, G.W.: A Neural Network Approach to Mutual Fund Net Asset Value Forecasting, Omega. Int. J. Manage. Sci. 24(2), 205–215 (1996)
9. Liu, M.C., Kuo, W., Sastri, T.: An Exploratory Study of a Neural Network Approach for Reliability Data Analysis. Qual. Reliab. Eng. Int. 11, 107–112 (1995)

10. Su, C.T., Tong, L.I., Leou, C.M.: Combination of Time Series and Neural Network for Reliability Forecasting Modeling. J. Chin. Inst. Ind. Eng. 14(4), 419–429 (1997)
11. Zhang, G.P.: Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing 50, 159–175 (2003)
12. Denton, J.W.: How Good are Neural Networks for Causal Forecasting? Journal of Business Forecasting 14, 17–20 (1995)
13. Markham, I.S., Rakes, T.R.: The Effect of Sample Size and Variability of Data on the Comparative Performance of Artificial Neural Networks and Regression. Computers & Operations Research 25, 251–263 (1998)
14. Chen, A.S., Leung, M.T.: Regression Neural Network for Error Correction in Foreign Exchange Rate Forecasting and Trading. Computers & Operations Research 31(7), 1049–1068 (2004)
15. Yokum, J.T., Armstrong, J.S.: Beyond Accuracy: Comparison of Criteria Used to Select Forecasting Methods. Int. J. Forecast 11(4), 591–597 (1995)

On Momentum and Learning Rate of the Generalized ADLINE Neural Network for Time Varying System Identification

Wenle Zhang

Dept. of Engineering Technology
University of Arkansas at Little Rock,
Little Rock, AR 72204
wxzhang@ualr.edu

Abstract. On-line System identification of linear time-varying (LTV) systems whose system parameters change in time has been studied lately. One neural network based such on-line identification method was studied by the author with a generalized ADAptive LINear Element (ADALINE). Since the ADALINE is slow in convergence, which is not suitable for identification of LTV system, two techniques were proposed to speed up convergence of learning. One idea was to introduce a momentum term to the weight adjustment during convergence period. The other technique was to train the generalized ADALINE network with data from a sliding window of the system's input output data. The second technique took multiple epochs to train which was considered as a shortcoming. In this paper, simulation study towards optimizing the momentum term and learning rate parameter will be presented. Simulation results show that once the momentum factor and learning rate are tuned properly, time varying parameters of LTV systems can be identified quite effectively; which, in turn, sows that the fined tuned GADLINE is quite suitable for online system identification and real time adaptive control applications due to its low computational demand.

Keywords: System identification, Neural network, ADALINE, Tapped delay line feedback.

1 Introduction

Given the structure (or simply *order* here) of a single input single output (SISO) linear time varying (LTV) system is known, system identification reduces to a parameter estimation problem from the measurable input and output data of the system, which can be well solved by traditional identification methods, such as least squares method, maximum likelihood method and instrumental variable method (L. Ljung 1999, Söderström and Stoica 1989). Increased interests in system identification area based on neural network techniques (Haykin 1999, Mehrotra 1997) have been seen in the past decade. Most neural networks used for system identification are in the category called recurrent (feedback) neural networks, such as, Multi-Layer

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 1002 – 1013, 2009.

Perceptron (Rumelhart 1986) with feedback and Hopfield neural network (Hopfield, 1982). And most neural network methods studied for system identification are mainly for nonlinear systems based on MLP (Sjöberg et al. 1994, Narendra and Parthasarathy 1990, Qin et al. 1992). Other types of neural networks are also seen, Hopfield (Chu et al. 1990), Radial Basis Function (Valverde 1999), Support Vector (Gretton 2001) and Self-Organizing Map (Abonyi and Szeifert 2002).

In this paper, the author continues to study the online identification method based on the generalized ADaptive LINear Element (GADALINE) neural network, for LTV systems which can be described by a discrete time model. In such systems, the current output is dependent on past outputs and on both the current and past inputs. So the GADALINE needs to have output feedback and to remember past input/output data. The proposed neural network employed a Tapped Delay Line (TDL) for both input and output of the system to remember the past input/output data. The GADALINE learning included a momentum term for the weight adjustment. This momentum term is turned on only during convergence period and the learning curve is therefore smoothed by turning off the momentum once the learning error is within a given small number – epsilon. Extensive simulation study towards optimizing the momentum term and learning rate parameter is performed. Simulation results show that once the momentum factor and learning rate are tuned properly, time varying parameters of LTV systems can be identified quite effectively. The rest of the paper is organized as: in section 2, the author's previous work GADLINE applied to system identification is summarized; section 3 explains the idea of optimization of the momentum term and learning rate; section 4 presents simulation study and section 5 draws some conclusion.

2 The GADALINE and System Identification

Originally, ADALINE was developed by Widrow and Lehr (Widrow and Lehr 1990) to predict the next bit by recognizing binary patterns from reading streaming bits from a phone line. The structure of an ADALINE is shown in Fig. 1, where a threshold logic unit can also be attached to the output. The structure of a GADALINE is shown in Fig. 4.

2.1 Structure of GADALINE

In the GADALINE structure, the bias input is removed, and each input is expanded through a TDL to become a number of delayed inputs and the output feedback is introduced to the input, also passing through a TDL to become a number of inputs. If we still keep the notation for inputs as $u_1 \sim u_m$, then the output y can be found by,

$$y = \Sigma_1^{\,m} w_i u_i = \boldsymbol{u}^T \boldsymbol{w}$$

where $\boldsymbol{u}$ is the input vector and $\boldsymbol{w}$ is the weight vector,

$$\boldsymbol{u} = [u_1\ u_2\ \ldots\ u_m]^T$$

$$\boldsymbol{w} = [w_1\ w_2\ \ldots\ w_m]^T$$

An example GADALINE structure for linear SISO system identification is shown in Fig. 4.

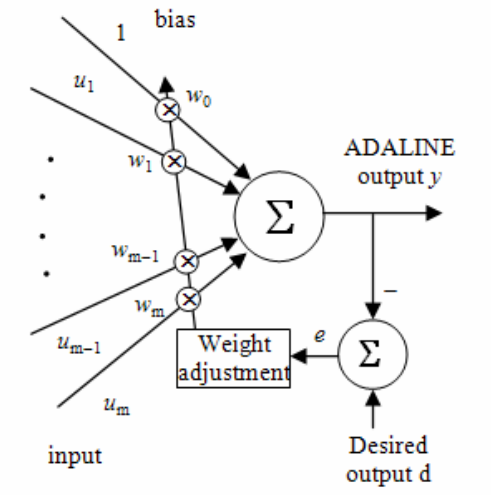


Fig. 1. An ADALINE unit

2.2 GADALINE Learning Algorithm

The GADALINE learning algorithm is based on the Widrow-Hoff rule with generalization by adding a momentum term to speed up convergence. The implication on the generalized learning will be discussed in Section 3.

Widrow-Hoff's delta rule or Least Mean Square algorithm (LMS), is based on minimizing the cost function,

$$E(\boldsymbol{w}) = 1/2\ e^2(t)$$

where t is the iteration index and $e(t)$ is given by,

$$e(t) = \mathrm{d}(t) - y(t)$$

where the desired output $\mathrm{d}(t)$ at each iteration t is constant.

According to LMS, the weight adjustment is

$$\Delta \boldsymbol{w}(t) = \boldsymbol{w}(t+1) - \boldsymbol{w}(t) = -\eta e(t)\boldsymbol{u}(t) \tag{1}$$

here the learning–rate is usually in the range of $0 < \eta < 1$.

It is well known (Haykin 1999), the LMS algorithm has a slow convergence speed. Especially for bigger learning parameter, the convergence trajectory exhibits a zigzag form. This can be stabilized partially by normalizing input vector to a unit vector in the weight adjustment equation (2) so that the weight adjustment magnitude is independent of the magnitude of the input vector. That is, the weight adjustment is given by,

$$\Delta \boldsymbol{w} = \eta\ e\ \boldsymbol{u}/\|\boldsymbol{u}\|$$

where $\|\boldsymbol{u}\|$ is the Euclidean norm of the input vector $\boldsymbol{u}$ or the length, which is calculated as,

$$\| \|\boldsymbol{u}\| = (u_1^2 + u_2^2 + \ldots + u_m^2)^{1/2}$$

In order to speed up the convergence and thus to increase the capability of tracking time varying system parameters, a momentum term is introduced into the weight adjustment equ. (1). The new weight adjustment now becomes,

$$\Delta \boldsymbol{w}(t) = \eta\, e(t)\boldsymbol{u}(t) + \alpha \Delta \boldsymbol{w}(t-1) \tag{2}$$

where $1 > \alpha \geq 0$ is a small non-negative number.

Because of this added momentum term, the weight adjustment continues after convergence, that is, when even the magnitude of the error term e is small. For linear system identification purpose, these weights correspond to system parameters. It is desirable to keep these parameters fixed once convergence is seen. So we should turn off the momentum term after convergence is detected, say $|e| < \varepsilon$, $\varepsilon > 0$ is a small number. However, when the LTV system parameters change at a later time, another convergence period will begin. Then it is better to turn the momentum back on.

GADALINE training is performed by presenting the training pattern set – a sequence of input-output pairs. During training, each input pattern from the training set is presented to the GADALINE and it computes its output y. This value and the target output from the training set are used to generate the error and the error is then used to adjust all the weights.

2.3 Identification Using GADALINE

Consider a discrete time SISO LTV system, the observable input and output data can be given in the form of a time series: $\{u(k\mathrm{T}), y(k\mathrm{T})\}$, T is the sample period, a constant, thus the data is often simply written as $\{u(k), y(k)\}$. Then the system can be modeled by the following difference equation,

$$\begin{aligned} & y(k) + a_1 y(k-1) + a_2 y(k-2) + \ldots + a_n y(k-n) \\ & = b_1 u(k-1) + b_2 u(k-1) + \ldots + b_m u(k-m) \end{aligned} \tag{3}$$

where n and m are system structure parameters, $m \leq n$, n is the order of the system; a_i and b_j are system parameters, $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$. If a_i and b_j are constants, the system is said to be time invariant, otherwise time varying. In the time varying case, a_i and b_j can be slowly changing with time or can be switched to different values at different time instants.

Then the system identification becomes a parametric estimation problem, that is, given n and m, to determine coefficients a_i and b_j from the input and output data of the system according to a certain criterion, such as minimizing an error function, which is a measure of how close the model is to the actual system. Fig. 2 shows a general system identification architecture, $y_u(k)$ is the deterministic output, $n(k)$ is the noise (white) and $e(k)$ is the error.

In the following, the paper will discuss how the model in the dashed line box in Fig. 2 can be approached by training a GADALINE, whose weights correspond to coefficients a_i and b_j.

As can be seen earlier, GADALINE is a simple linear neural network which can be trained to adaptively model a linear or near-linear system. Adaptive training (also called online/incremental training, as opposed to batch training) can be used so that

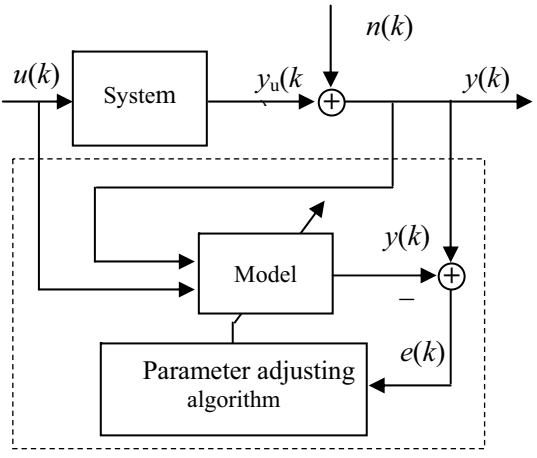


Fig. 2. System identification of a LTV system

GADALINE Learning Algorithm

Start with a randomly chosen weight vector $\boldsymbol{w}(0)$
Let $t = 1$
While % online version is an infinite loop
  Let $\boldsymbol{u}(t) = (u_1, .., u_m)$ be next input vector, for which d(t) is the desired output
  Calculate $e(t) = \mathrm{d}(t) - \boldsymbol{u}^{\mathrm{T}}(t)\boldsymbol{w}(t\text{-}1)$
  $\Delta\boldsymbol{w}(t) = \eta e(t)\boldsymbol{u}(t)$
  If $| e(t)| > \varepsilon$, **then**
    $\Delta\boldsymbol{w}(t) = \Delta\boldsymbol{w}(t) + \alpha\Delta\boldsymbol{w}(t\text{–}1)$
  Adjust the weight vector to
    $\boldsymbol{w}(t) = \boldsymbol{w}(t\text{-}1) + \Delta\boldsymbol{w}(t)$
  Increment t
End while

Fig. 3. The GADALINE learning

any change in parameters of the system also changes the weights of the neural network. Thus, at any instant of time, GADALINE can track changes in the parameters of a linear time varying systems. Fig. 3 summarizes the GADALINE learning algorithm.

The idea is to use a GADALINE in place of the block of Fig. 2 in the dashed line box. For system identification, the GADALINE can be configured as in Fig. 4, where the system input u_S is passed through a TDL to feed to the linear combiner of the

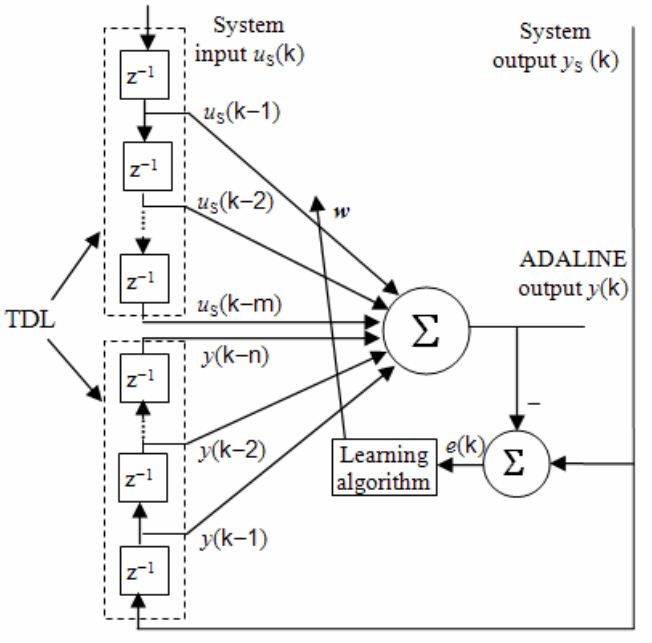


Fig. 4. GADALINE configuration

ADALINE. And the measured system output y_S is also taken to pass through a TDL as feedback. Then the input vector to the GADALINE can be set up as,

$$\boldsymbol{u}(k) = [u_S(k-1) \ldots u_S(k-m) -y_S(k-1) \ldots -y_S(k-n)]^T \quad (4)$$

and the weight vector

$$\boldsymbol{w}(k) = [w_1(k)\ w_2(k) \ldots w_m(k)\ w_{m+1}(k) \ldots w_{m+n}(k)]^T \quad (5)$$

corresponds to an estimate of the system parameters,

$$\boldsymbol{\theta} = [b_1\ b_2 \ldots b_m\ a_1\ a_2 \ldots a_n]^T$$

See the system model (3).

Then, we have the following identification procedure.

1) Experiment design: As in traditional system identification, the input signal should be selected such that it is "rich" enough to excite all modes of the system, this is called persistent exciting. For linear systems, input signal should excite all frequencies and amplitude is not so important. Common used persistent exciting signals that can be tried are: random signal, multi-sine signal and random binary sequence signal (RBS) – the signal switches between two levels with given probability.

2) Data collection: Measurement of input-output data $u_S(k)$ and $y_S(k)$ and to set up $\boldsymbol{u}(k)$.

3) Parameter estimation: Apply the training algorithms to estimate the parameters $\boldsymbol{\theta}(k) = \boldsymbol{w}(k)$.

3 Optimizing Momentum Term and Learning Rate

To see the effect of the momentum, we can expand the equation starting from $t = 0$,

$$\Delta \boldsymbol{w}(0) = \eta e(0)\boldsymbol{u}(0)$$

$$\Delta \boldsymbol{w}(1) = \eta e(1)\boldsymbol{u}(1) + \alpha\Delta \boldsymbol{w}(0) = \eta e(1)\boldsymbol{u}(1) + \alpha\eta e(0)u(0)$$

$$\Delta \boldsymbol{w}(2) = \eta e(2)\boldsymbol{u}(2) + \alpha\Delta \boldsymbol{w}(1) = \eta e(2)\boldsymbol{u}(2) + \alpha\eta e(1)\boldsymbol{u}(1) + \alpha^2\eta e(0)\boldsymbol{u}(0)$$

$$\begin{aligned}\Delta \boldsymbol{w}(t) &= \eta e(t)\boldsymbol{u}(t) + \alpha\Delta \boldsymbol{w}(t-1) \\ &= \eta e(t)\boldsymbol{u}(t) + \alpha\eta e(t-1)\boldsymbol{u}(t-1) + \ \ldots\ + \alpha^{n-1}\eta e(1)\boldsymbol{u}(1) + \alpha^n\, \eta e(0)\boldsymbol{u}(0) \\ &= \eta \sum_{\tau=0}^{t} \alpha^{t-\tau} e^{\tau} u(\tau)\end{aligned}$$

Remember $e(t)\boldsymbol{u}(t)$ is just $-\partial E(t)/\partial \boldsymbol{w}(t)$, then

$$\Delta \boldsymbol{w}(t) = \eta \sum_{\tau=0}^{t} \alpha^{t-\tau}\ \partial E(\tau)/\partial \boldsymbol{w}(\tau)$$

So, we have an exponentially weighted time series. We can then make the following observations,

i) If the derivative $\partial E(t)/\partial \boldsymbol{w}(t)$ has opposite sign on consecutive iterations, the exponentially weighted sum $\Delta \boldsymbol{w}(t)$ reduces in magnitude, so the weight $\boldsymbol{w}(t+1)$ is adjusted by a small amount, which reduces the zigzag effect in weight adjustment.

ii) On the other hand, if the derivative $\partial E(t)/\partial \boldsymbol{w}(t)$ has same sign on consecutive iterations, the weighted sum $\Delta w(t)$ grows in magnitude, so the weight $w(t+1)$ is adjusted by a large amount. The effect of the momentum term is to accelerate descent in steady downhill region.

4 Simulation Study

Simulation study is performed on several example systems. Based on a classic representative example, the comparison on convergence speed and capability of tracking time varying systems' parameters among the normal LMS training algorithm and the SWLMS training algorithms is presented in this section.

Consider a second order linear SISO system described by the following difference equation,

$$y(k) - 1.5\, y(k-1) + 0.7\, y(k-2)$$

$$= 1.0\, u(k-1) + 0.5\, u(k-2)$$

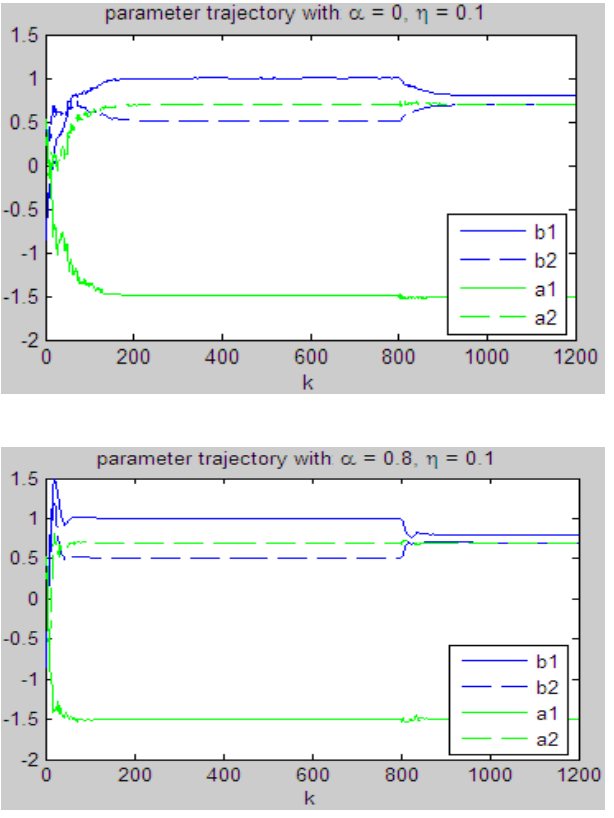


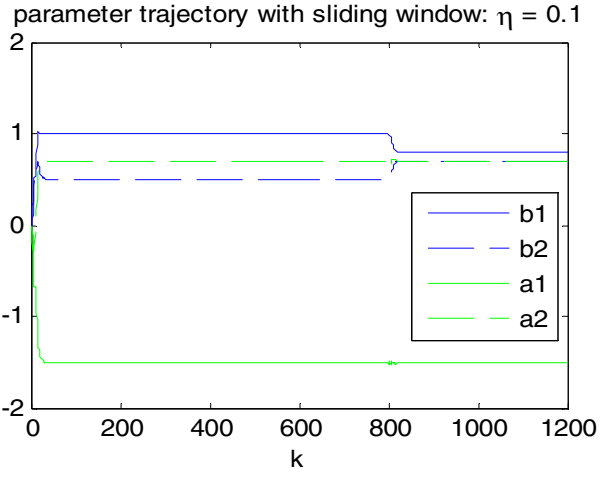
Fig. 5. MTMS trajectories for different α

The system is to be simulated with an RBS input signal switching between -1 and 1 with equal probability from k=1 to k=1200. Assume at k=800, that system parameters b_1 and b_2 are switched to 0.8 and 0.7, respectively. Fig. 5 shows the comparison of parameter trajectories obtained with different α values, i.e., α = 0 and α = 0.8.

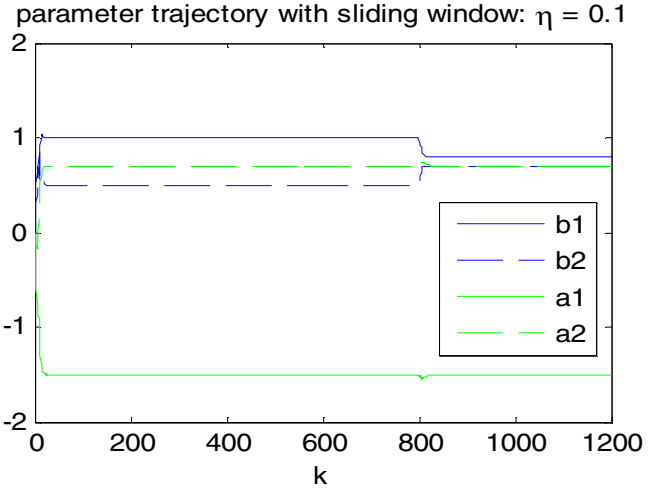
From Fig. 5, we can observe that the average convergence speed of all parameters is doubled and the capability of tracking time varying parameters is improved when α=0.8 as opposed to α=0.

Fig. 6 shows the parameter trajectories obtained by the sliding window LMS algorithm with different number of epochs and window size combinations. We observed much faster convergence speed and much better capability of tracking time varying parameters but the bigger sliding window Fig. 6(a) vs. 6(b) did not show apparent improvement.

The system is also simulated with the output contaminated with a white noise signal of variance σ^2 = 0.05, Fig. 7. Increased α improves performance while makes the parameter trajectories more fluctuating. Note that the learning rate parameter η is



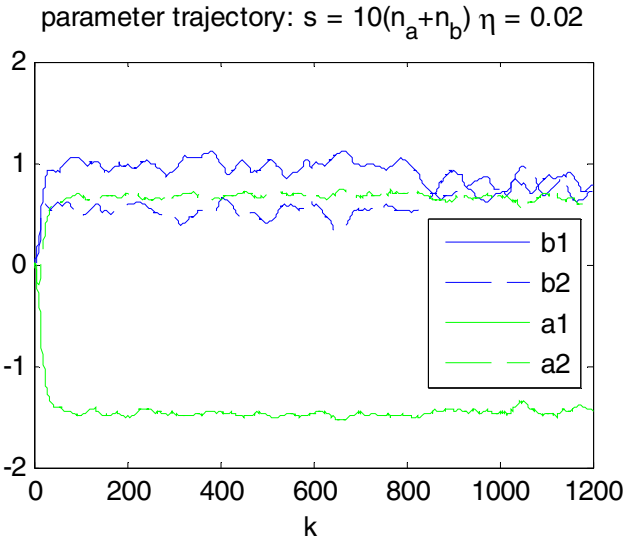
(a) n_e =1, $s = 4(n_a+n_b)$, $\eta = 0.1$



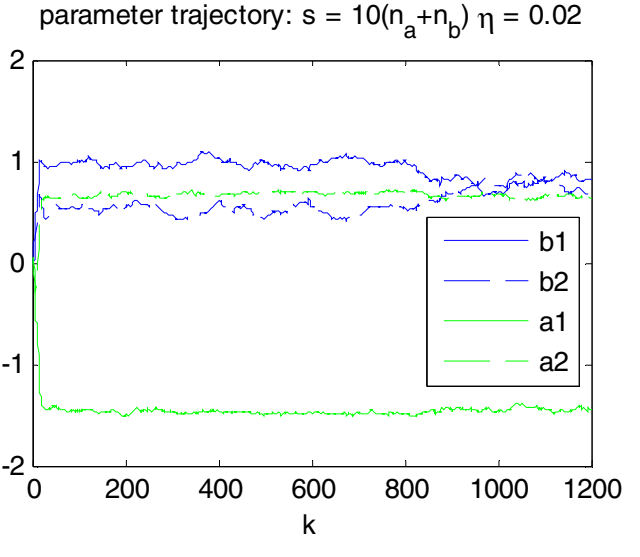
(b) n_e =4, $s = n_a+n_b$, $\eta = 0.1$

Fig. 6. SWLMS trajectories with different n_e and s

now 5 times smaller. Increased sliding window size smoothes the curves, Fig. 7(b) vs. (a). Again, increased number of epochs speeds up the convergence, but it also tracks the noise mixed in the system's output, Fig. 7(c). Therefore, tradeoff on the two tuning parameters should be considered.



(a) n_e =1, $s = 10(n_a+n_b)$, $\eta = 0.02$



(b) n_e =4, $s = 20(n_a+n_b)$, $\eta = 0.02$

Fig. 7. SWLMS trajectories for output with noise

5 Conclusions

The ADALINE neural network is generalized and used for adaptive system identification of linear time varying systems. Two techniques are proposed to improve the LMS learning algorithm. The momentum term helps to speed up the learning process and reduce the zigzag effect during convergence period of learning. The second training technique uses each set of samples for several epochs obtained from a sliding window of the system's input output data. The second technique took multiple epochs to train which was considered as a shortcoming. In this paper, simulation study towards optimizing the momentum term and learning rate parameter is presented. Optimal or near optimal range for momentum term combined with proper learning rate is determined. Simulation results show that once the momentum factor and learning rate are tuned properly, time varying parameters of LTI systems can be identified quite effectively; which, in turn, sows that the fined tuned GADLINE is quite suitable for online system identification and real time adaptive control applications due to its low computational demand.

References

1. Abonyi, J., Szeifert, F.: System Identification Using Delaunay Tessellation of Self-Organizing Maps. In: The 6th International Conference on Neural Networks and Soft Computing, ICNNSC, Poland, Zakopane, pp. 11–15 (2002)
2. Atencia, M., Sandoval, G.: Gray Box Identification with Hopfield Neural Networks. Revista Investigacion Operacional 25(1), 54–60 (2004)
3. Bhama, S., Singh, H.: Single Layer Neural Network for Linear System Identification Using Gradient Descent Technique. IEEE Trans. on Neural Networks 4(5), 884–888 (1993)
4. Chu, S.R., Shoureshi, R., Tenorio, M.: Neural Networks for System Identification. IEEE Control Systems Magazine, 31–34 (1990)
5. Gretton, A., Doucet, A., Herbrich, R., Rayner, P., Schölkopf, B.: Support Vector Regression for Black-Box System Identification. In: The 11th IEEE Workshop on Statistical Signal Processing (2001)
6. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice-Hall, Englewood Cliffs (1999)
7. Hopfield, J.: Neural Networks and Physical Systems with Emergent Collective Computational Abilities. Proc. Nat. Acad. Sci. 79, 2554–2558 (1982)
8. Ljung, L.: System Identification-Theory for the User. Prentice-Hall, Englewood Cliffs (1999)
9. Mehrotra, K., Mohan, C., Ranka, S.: Elements of Artificial Neural Networks. MIT Press, Cambridge (1997)
10. Narendra, K.S., Parthasarathy, K.: Identification and Control of Dynamical Systems Using Neural Networks. IEEE Transactions on Neural Networks 1, 1–27 (1990)
11. Qin, S.Z., Su, H.T., McAvoy, T.J.: Comparison of Four Neural Net Learning Methods for Dynamic System Identification. IEEE Trans. on Neural Networks 2, 52–262 (1992)
12. Rumelhart, H.D.E., Williams, R.J.: Learning Internal Representations by Error Propagation. In: Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. I. MIT Press, Cambridge (1986)

13. Sjöberg, J., Hjalmerson, H., Ljung, L.: Neural Networks in System Identification. In: Preprints 10th IFAC symposium on SYSID, Copenhagen, Denmark, vol. 2, pp. 49–71 (1994)
14. Söderström, T., Stoica, P.: System Identification. Prentice Hall, Englewood Cliffs (1989)
15. Valverde, R.: Dynamic Systems Identification Using RBF Neural Networks. Universidad Carlos III de Madrid. Technical Report (1999)
16. Widrow, B., Lehr, M.A.: 30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation. Proc. IEEE 78(9), 1415–1442 (1990)
17. Wenle, Z.: System Identification Based on a Generalized ADALINE Neural Network. In: Proceedings of the 2007 American Control Conference, New York City, pp. 4792–4797 (2007)

Inference of Differential Equations for Modeling Chemical Reactions

Bin Yang, Yuehui Chen, and Qingfang Meng

Computational Intelligence Lab.
School of Information Science and Engineering
University of Jinan, Jinan, 250022, China
yhchen@ujn.edu.cn

Abstract. This paper presents an evolutionary method for identifying a system of ordinary differential equations (ODEs) from the observed time series data. The structure of ODE is inferred by the Multi Expression Programming (MEP) and the ODE's parameters are optimized by using particle swarm optimization (PSO). The experimental results on chemical reaction modeling problems show effectiveness of the proposed method.

Keywords: Multi expression programming, Ordinary differential equations, Particle swarm optimization, Chemical reaction.

1 Introduction

In the fields of physics, chemistry, economics, bioinformatics etc. a lot of problems can be expressed in term of ordinary differential equations(ODEs). Weather forecasting, quantum mechanics, wave propagation, stock market dynamics and identification of biological systems are some examples [1]. For this reasion many methods were proposed for inferring ODEs during the last few years. The researches can be classified into two classes of methods: the one is to identify the parameters of the ODEs and the other is to identify the structure. The former is examplified by Genetic Algorithms (GA), and the latter by the Genetic Programming (GP) approach.

Cao and his colleagues use GP to evolve the ODEs from the observed time series in 1999 [2]. His main idea is to embed a genetic algorithm in genetic programming, where GP is employed to discover and optimize the structure of a model, while a GA is employed to optimize its parameters. They showed that the GP-based approach introduced numerous advantages over the most available modeling methods. H. Iba proposed a ODEs identification method by using the least mean square(LMS) along with the ordinary GP [3][14]. Some individuals were created by the LMS method at some intervals of generations and they replaced the worst individuals in the population. I.G.Tsoulos and I.E.Lagar proposed a novel method based on grammatical evolution [1]. The method forms generations of trial solutions expressed in an analytical closed form. The Bayesian

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 1014–1023, 2009.

inferential methodology provides a coherent framework with which to characterize and propagate uncertainty in such mechanistic models and this provides an introduction to Bayesian methodology as applied to system models represented as differential equations [15].

In this paper we propose a new method, in which the particle swarm optimization (PSO) is used along with Multi Expression Programming (MEP). We infer the structure of the right-hand sides of ODEs by MEP and optimize the parameters of ODEs by PSO. And the partitioning [4] is used in the process of identification of structure of system. Each ODE of the ODEs can be inferred separately and the research space reduces rapidly.

The paper is organized as follows. In Section2, we describe the details of our method. In section 3, the three examples are used to examine the effectiveness and veracity of the proposed method. Conclusions are drawn in Section4.

2 Method

2.1 Structure Optimization of Models Using MEP

Encoding. MEP is a relatively new technique in genetic programming that is first introduced in 2002 by Oltean [5] [6]. A traditional GP [7] encodes a single expression (computer program). By contrast, a MEP chromosome encodes several genes. Each gene encodes a terminal or a function symbol which is selected from a terminal set T or a function set F. The two sets for a given problem are pre-defined. A gene that encodes a function includes some pointers towards the function arguments. The number of the pointers depends on how many arguments the function has. The best of the encoded solution is chosen to represent the chromosome [8]. We use MEP to identify the form of the system of differential

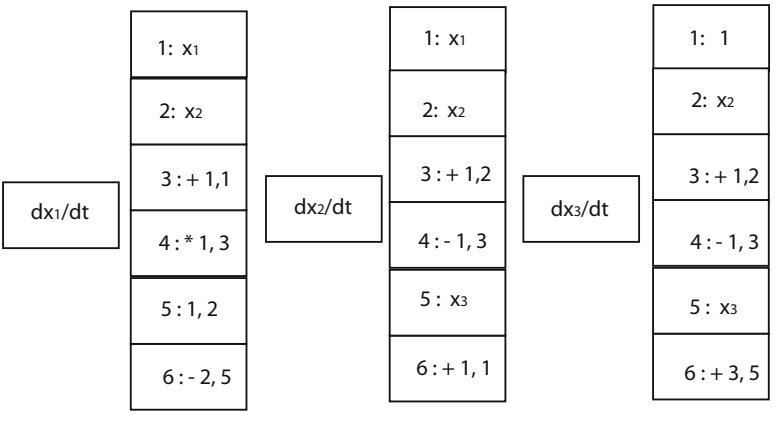


Fig. 1. Example of a ODEs

equations. For this purpose, we encode right-hand side of each ODE into a MEP chromosome. For example a ODEs model with the form of

$$\begin{cases} \dot{X}_1 = aX_1 + bX_2 \\ \dot{X}_2 = cX_1 \\ \dot{X}_3 = dX_2 + e \end{cases} \tag{1}$$

can be represented as three MEP chromosomes$\{E_3, E_6, E_3\}$ illustrated in Fig. 1, where the coefficients a, b, c, d, e are derived by PSO (described later in this paper).

We infer the system of ODEs with partitioning. Partitioning, in which equations describing each variable of the system can be inferred separately, significantly reducing the research space. When using partitioning, a candidate equation for a signal variable is integrated by substituting references to other variables with data from the observed time series [4].

Fitness Function. A MEP chromosome contains some expressions, so each expression E_i is calculated by root mean squared error(RMSE) or the sum of absolute error(SAM):

$$f(E_i) = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(x_{ji} - x'_j)} \tag{2}$$

$$f(E_i) = \sum_{j=1}^{n}\left|x_{ji} - x'_j\right| \tag{3}$$

Where x_{ji} is the time series by expression E_i and x'_j is the targeted time series. The fitness of a chromosome is equal to the best fitness of the expressions encoded.

Genetic Operators. The genetic operators used within MEP algorithm are crossover and mutation [5].

(1) Crossover. In this paper, we choose the one-point crossover. Firstly, two parents are selected according to the predefined crossover probability P_c. One crossover point is randomly chosen and the parents exchange the sequences at this point.
(2) Mutation. One parent is selected according to the predefined mutation probability P_m. One mutation point is randomly chosen. If the mutation position encodes a function symbol, it may be mutated into a terminal symbol or another function with arguments and parameters. And we can mutate the function arguments and parameters into random arguments and parameters.

2.2 Parameter Optimization of Models Using PSO

Encoding. At the beginning of this process, we check all the constants contained in each equation , namely count their number n_i and report their places. Distribution of parameters in each chromosome is illustrated in Fig. 2.

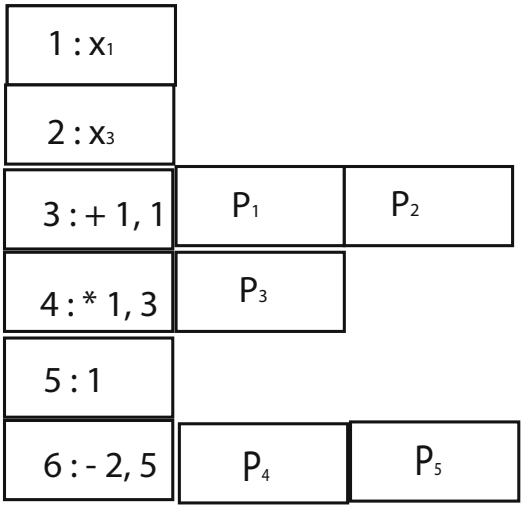


Fig. 2. Distribution of parameters in each chromosome

According to n_i, the particles are randomly generated initially. Each particle x_i represents a potential solution. A swarm of particles moves through space, with the moving velocity of each particle represented by a velocity vector v_i. At each step, each particle is evaluated and keep track of its own best position, which is associated with the best fitness it has achieved so far in a vector $Pbest_i$. And the best position among all the particles is kept as Gbest [9]. A new velocity for particle i is updated by

$$v_i(t+1) = v_i(t) + c_1 r_1(Pbest_i - x_i(t)) + c_2 r_2(Gbest(t) - x_i(t)) \tag{4}$$

where c_1 and c_2 are positive constant and r_1 and r_2 are uniformly distributed random number in [0,1]. Based on the updated velocities, each particle changes its position according to the following equation:

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{5}$$

2.3 Fitness Definition

The fitness of each variable is defined as the sum of squared error and the penalty for the degree of the equations:

$$fitness(i) = \sum_{k=0}^{T-1} \left(x_i'(t_0 + k\Delta t) - x_i(t_0 + k\Delta t)\right)^2 + a \tag{6}$$

where t_0 is the starting time, $\triangle t$ is the stepsize, T is the number of the data point, $x_i(t_0+k\triangle t)$is the actual outputs of i-th sample, and $x_i'(t_0+k\triangle t)$ is ODEs outputs. All outputs are calculated by using the approximate forth-order Runge-Kutta method. And a is the penalty for the degree of the equations. To reduce

the problem space, degrees of the individuals are limited to the stated range according a. When calculating the outputs, some individuals may cause overflow. In such cases, the individual's fitness becomes so large that it will be weeded out from the population.

2.4 Summary of Algorithm

The MEP for the optimal design of each ODE can be described as follows.

(1) Create a initial population randomly(structures and their corresponding parameters);
(2) Structure optimization is achieved by MEP as described in subsection 2.1;
(3) At some interval of generations, select the better structures to optimize parameters. Parameter optimization is achieved by PSO as described in subsection 2.2. In this process, the structure is fixed.
(4) If satisfactory solution is found, then stop; otherwise go to step (2).

If the parameters of ODEs have some error, we can use the standard fourth-order Runge-Kutta method to integrate the ODE to optimize parameters.

3 Experimental Results

We have prepared four tasks to test the effectiveness of our method. Experimental parameters are summarized in Table 1. Function and terminal sets F and T are follows:

$$\begin{aligned} F &= \{+, -, *\} \\ T &= \{X_1, ..., X_n, 1\} \end{aligned} \tag{7}$$

3.1 Experiment I: Chemical Reaction Model

The reaction equations [2]are described below

$$HCHO + (NH_2)_2CO \xrightarrow{k_1} H_2N \cdot CO \cdot NH \cdot CH_2OH \tag{8}$$

$$H_2N \cdot CO \cdot NH \cdot CH_2OH + (NH_2)_2CO \xrightarrow{k_2} (NH_2CONH)_2CH_2 \tag{9}$$

Table 1. Parameters for experiments

	Exp1	Exp2	Exp3
Population size	20	50	50
Generation	50	100	100
Crossover rate	0.7	0.7	0.7
Mutation rate	0.3	0.3	0.3
Time series	1	1	1
Stepsize	0.01	0.05	0.05
Data point	30	30	48

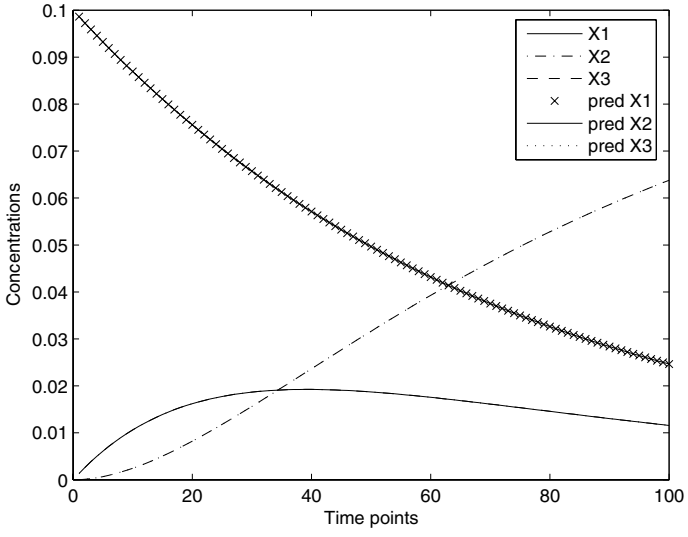


Fig. 3. Time series of the acquired model for chemical reaction

As a kind of typical consecutive reactions, the concentrations of the three components in the system satisfy the following system:

$$\begin{cases} \dot{X_1} = -1.4000X_1 \\ \dot{X_2} = 1.4000X_1 - 4.2X_2 \\ \dot{X_3} = 4.2000X_2 \end{cases} \tag{10}$$

A time series was generated for the above set of reactions with initial conditions $\{0.1, 0, 0\}$ for $\{X_1, X_2, X_3\}$. Experimental parameter for this task are shown in Table 1. We have acquired the system of eq.(11), which gave the sums of sums of absolute errors as (X_1, X_2, X_3)=$(3.6\times10^{-12}, 4.01\times10^{-12}, 8.79\times10^{-12})$. The time series generated is shown in Fig. 3 along with that of the target.

$$\begin{cases} \dot{X_1} = -1.400017X_1 \\ \dot{X_2} = 1.400044X_1 - 4.199987X_2 \\ \dot{X_3} = 4.199939X_2 \end{cases} \tag{11}$$

The best kinetic model acquired in [2] was as follows:

$$\begin{cases} \dot{X_1} = -1.400035X_1 \\ \dot{X_2} = 1.355543(X_1 + t) - 4.482911X_2 \\ \dot{X_3} = 4.069420X_2 + t - 0.002812 \end{cases} \tag{12}$$

Where the the sums of squared errors were(X_1, X_2, X_3)=$(1.6 \times 10^{-11}, 3.24 \times 10^{-8}, 3.025\times10^{-9})$. Note that the terminal set in [2] included the time variable t.

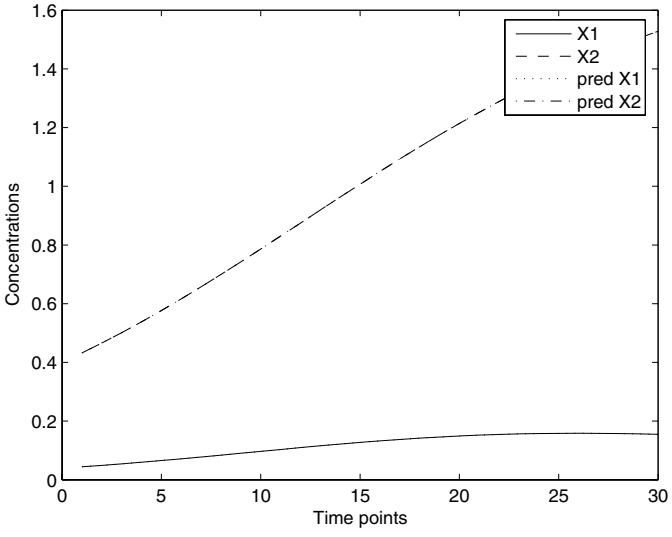


Fig. 4. Time series of the acquired model for Lotka-Volterra model

3.2 Experiment II: Two-Species Lotka-Volterra Model

The Lotka-Volterra model describes interactions between two species, i.e., predators and preys, in an ecosystem [10]. The following ODEs represent a two-species Lotka-Volterra model:

$$\begin{cases} \dot{X_1} = 3X_1 - 2X_1X_2 - {X_1}^2 \\ \dot{X_2} = 2X_2 - X_1X_2 - {X_2}^2 \end{cases} \tag{13}$$

A time series was generated for the above set of reactions with initial conditions {0.04, 0.4} for $\{X_1, X_2\}$. The time series generated is shown in Fig. 4. Experimental parameter for this task are shown in Table 1. We have acquired the system of eq. (14), which gave the sums of sums of absolute errors as (X_1, X_2)=(2.5×10^{-11}, 4.45×10^{-10}). In all runs,we have succeeded in getting almost the same ODEs.

$$\begin{cases} \dot{X_1} = 2.999998X_1 - 2.000081X_1X_2 - 0.9993\,{X_1}^2 \\ \dot{X_2} = 2.000005X_2 - 1.000064X_1X_2 - 0.999997{X_2}^2 \end{cases} \tag{14}$$

The best model acquired in [4]was eq.(15). Compared with it, structure is the same and the parameters of our model are closer to the targeted model.

$$\begin{cases} \dot{X_1} = 3.0014X_1 - 2X_1X_2 - {X_1}^2 \\ \dot{X_2} = 2.0001X_2 - X_1X_2 - {X_2}^2 \end{cases} \tag{15}$$

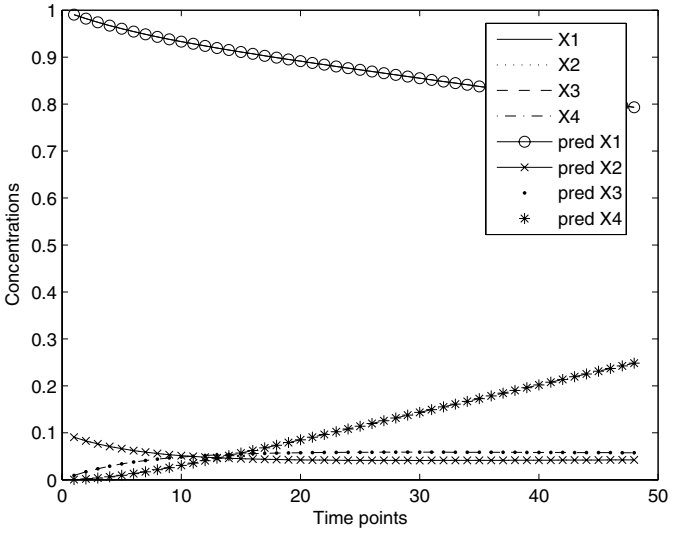


Fig. 5. Time series of the acquired model for bimolecular reaction

3.3 Experiment III: Bimolecular Reaction

The a bimolecular reaction equations [11] are described below:

$$X_2 + X_1 \xrightarrow{k_1} X_3 \tag{16}$$

$$X_3 \xrightarrow{k_2} X_4 + X_2 \tag{17}$$

The corresponding rate equations for all the four species are as follows:

$$\begin{cases} \dot{X_1} = -2X_1X_2 \\ \dot{X_2} = -2X_1X_2 + 1.2X_3 \\ \dot{X_3} = 2X_1X_2 - 1.2X_3 \\ \dot{X_4} = 1.2X_3 \end{cases} \tag{18}$$

A time series was generated for the above set of reactions with initial conditions $\{1, 0.1, 0, 0\}$ for $\{X_1, X_2, X_3, X_4\}$ which is shown in Fig. 5 along with targeted time series. Experimental parameter for this task are shown in Table 1.

We have acquired the system of eq.(20), which gave the sums of sums of absolute errors as (X_1, X_2, X_3, X_4)=(1.6×10^{-11}, 9.0×10^{-12}, 8.8×10^{-12}, 2.5×10^{-11}).

$$\begin{cases} \dot{X_1} = -1.9920X_1X_2 \\ \dot{X_2} = -1.1983X_1X_2 + 1.9920X_3 \\ \dot{X_3} = 1.9920X_1X_2 - 1.1983X_3 \\ \dot{X_4} = 1.1983X_3 \end{cases} \tag{19}$$

Compared with eq(19) [11], our predicted model and parameters are closer to the targeted system and the model is able to predict the standard enzyme kinetics scheme with rate parameters close to the generative values.

$$\begin{cases} \dot{X_1} = -1.99999 X_1 X_2 \\ \dot{X_2} = 1.20000 X_3 - 1.99999 X_1 X_2 \\ \dot{X_3} = -1.20000 X_3 - 2.00000 X_1 X_2 \\ \dot{X_4}\, t = 1.99999 X_3 \end{cases} \tag{20}$$

4 Conclusion

In this paper, a new approach for evolving ODEs is proposed. By several experiments, we succeeded in creating the systems of ODEs which are close to the targeted systems. The result shows the effectiveness and veracity of the proposed method. The method has following two advantages: (1) a MEP chromosome encodes several expressions, so we can acquire the best structure of the ODE only by a small population; (2) with partitioning, we can acquire the best system very fast.

In the future, we will apply our approach to the bioinformatics fields to solve some the real biological problems. We will work on the structure of the interactive system, which proposes the possible solutions.

Acknowledgement. This research was supported by the NSFC (60573065), the the Natural Science Foundation of Shandong Province (Y2007G33), and the Key Subject Research Foundation of Shandong Province.

References

1. Tsoulos, I.G., Lagaris, I.E.: Solving Differential Equations with Genetic Programming. Genetic Programming and Evolvable Machines 1, 1389–2576 (2006)
2. Cao, H., Kang, L., Chen, Y., Yu, J.: Evolutionary Modeling of Systems of Ordinary Differential Equations with Genetic Programming. Genetic Programming and Evolvable Machines 40, 309–337 (2000)
3. Iba, H., Mimura, A.: Inference of a Gene Regulatory Network by Means of Interactive Evolutionary Computing. Information Sciences 145, 225–236 (2002)
4. Bongard, J., Lipson, H.: Automated Reverse Engineering of Nonlinear Dynamical Systems. Proceedings of the National Academy of Science 104, 9943–9948 (2007)
5. Oltean, M., Dumitrescu, D.: Multi Expression Programming. Technical Report, UBB-01-2002, Babes-Bolyai University, Cluj-Napoca, Romania, www.mep.cs.ubbcluj.rog
6. Crina, G., Ajith, A., Sang, Y.H.: Multi-Expression Programming for Intrusion Detection System. In: Mira, J., Álvarez, J.R. (eds.) IWINAC 2005. LNCS, vol. 3562, pp. 163–172. Springer, Heidelberg (2005)
7. Andrew, H.W.: System Identification Using Genetic Programming. In: Proc. of 2nd Int. Conference on Adaptive Computing in Engineering Design and Control (1996)

8. Oltean, M., Grosan, C.: Evolving Digital Circuits Using Multi Expression Programming. In: Zebulum, R., et al. (eds.) NASA/DoD Conference on Evolvable Hardware, Seattle, June 24-26, pp. 87–90. IEEE Press, NJ (2004)
9. Chen, Y.H., Yang, B., Ajith, A.: Flexible Neural Trees Ensemble for Stock Index Modeling. Neurocomputing 70, 697–703 (2007)
10. Takeuchi, Y.: Global Dynamical Properties of Lotka-Volterra Systems. World Scientific, Singapore (1996)
11. Arkin, A.P., Ross, J.: Statistical Construction of Chemical Reaction Mechanisms from Measured Time-Series. J. Phys. Chem., 970–979 (1995)
12. Gennemark, P., Wedelin, D.: Efficient Algorithms for Ordinary Differential Equation Model Identification of Biological Systems. IET Syst. Biol. 1, 120–129 (2007)
13. Savageau, M.A.: Biochemical Systems Analysis: A Study of Function and Design in Molecular Biology. Addison-Wesley, Reading (1976)
14. Hitoshi, I.: Inference of Differential Equation Models by Genetic Programming. Information Sciences 178, 4453–4468 (2008)
15. Mark, G.: Bayesian Inference for Differential Equations. Theoretical Computer Science 408, 4–16 (2008)

Supply Chain Management with Revenue-Sharing Contract in a JIT Setting

Taigui Qin

College of Science, China Three Gorges University,
Yichang 443002, China
jmh1239@163.com

Abstract. This paper investigates supply chain management between the retailer and the supplier with revenue-sharing contract under uncertain demand by means of a single-period order within a Just-In-Time setting. And we also show how to effectively handle the demand uncertainty in a supply chain, both for the case of a centralized-decision-making system and the case of decentralized-decision-making system. The aim of the paper is to find the optimal order from the retailer in both systems in order that the total expected profits are maximal. The paper focuses not only on the overall performance but also on the allocation of the profit between the retailer and the supplier.

Keywords: Supply chain management; Demand uncertainty; Revenue-sharing contract.

1 Introduction

Coordination producer and supplier is one of the main issues of supply chain management (SCM) research, which has attracted many researcher and experts to study nowadays. SCM research has generated a significant body of literature. Most of SCM literature focuses on the criteria of minimizing expected cost or maximizing expected profits. Thus research in supply chain coordination is devoted to designing supply contracts that can improve the expected value of a given performance measure.

Uncertainty plays an important role in the modern SCM. Handling uncertainty in an efficient and effective way is becoming more and more important to the success of SCM. Uncertainties in SCM include many elements, such as the demand rate, the order cost, the holding cost rate, seller's price and so on. Yu[1],[2] has studied the classical economic order quantity (EOQ) model under significant uncertainties including the demand, the order cost and the holding cost. And the demand disruptions in SCM coordination is also studied

Traditionally, uncertainty is studied by stochastic models with some appropriate probability assumptions. For the related supply chain literature, we refer to M.A.Larivieve, R.Anupindi, H.Gurnami, and Y. Gerchak .

Generally speaking, the literature on supply chain management can be roughly split into two classes. The first considers the supply chain from the point of view of one decision maker and determines those decisions that minimize overall total cost. In

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 1024 – 1032, 2009.

these approaches the supply chain is considered as a fully vertically integrated firm where all information is common knowledge and the material flow is controlled by a single decision maker, where the production yield of the component is either deterministic or random .

In fact, most of the models presented so far in the literature have treated subsystems of SCM separately, and attempted to coordinate just parts of the SCM. A lot of research has been done considering multi-echelon inventory/distribution systems in this area. Models that coordinate the supply chain subsystems such as buyer-vendor and production-distribution subsystems ,have appeared in the last decade only. One of the representative analytical models that considers the whole SCM is developed by Cohen and Lee[3].

The second class considers the supply chain as a chain of antagonistically behaving companies and analyzes the implications of contract forms . There are only two decisive makers, buyer and supplier. One of the first studies in this area is that of Monaham[4].

In our paper, we consider a supply chain consisting of one supplier and one retailer, where the supplier has normal supplying quantity Q for the retailer. The order quantity from the retailer is determined by the market demands, which is stochastic. On the behalf of the retailer, he may order quantity is more than Q, which means supplier should build up the extra capacity at the higher expense, or less than Q, which means supplier should pay the holding cost of the un-ordering capacity. Then such supply chain has a coordination mechanism when maximizing the profit of the whole supply chain.

Otherwise, our paper differs from the existing literature in that we not only attempt to find contract parameters that allow the decentralized system to perform just as well as a centralized one, but we investigate more general coordination mechanisms which in addition give us higher flexibility in allocating profit.

In addition, for analytical simplicity, without loss of generality, we consider the system is controlled in the environment of just-in-time. We assume that there is little information delay between the retailer and its supplier[5]. Thus we don't consider any inefficiency due to on information distortion in the supply chain[6], [7].

The remainder of the paper is organized as follow. In the next section, we establish mathematical models to describe the supplier-retailer relationship briefed above. Details of the models and its assumptions are also elaborated. In section 3, we have considered two extreme cases analysis illustrates and centralized systems and is used in the following sections as a benchmark to evaluate the coordination mechanisms. In section 4, in which we give the kind of coordination mechanism in profit distribution after the joint operation. Finally, we give some examples and the further research's directions.

2 Models Formulation

In this section, we model a supply chain consisting of only one manufacturer and its supplier. Analytical and managerial implications from this sparse model can be extended to more complicated situation. The retailer faces a unknown demand X for a finished product which is assembled using a special component or primary product

that he orders in q from the supplier, here we suppose that retailer requires one unit of the component to produce one unit of the finished product. To the supplier, its normal capacity is denoted by Q. In order to satisfy the need of retailer, he should build up extra capacity ΔQ, for which is not in the plan, the supplier should use more manpower and material resource in special setting to finish it, then the unit cost of ΔQ is more than that of normal Q.

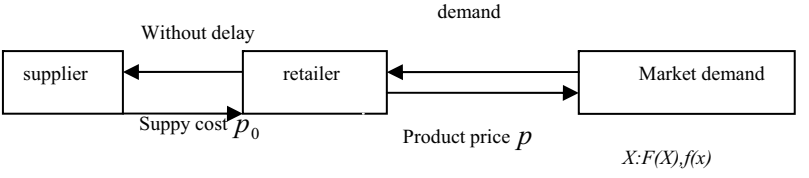


Fig. 1. Depicts the present model situation

As shown in Fig.1 the delivery quantity q depends on the normal quantity and capacity ΔQ. Here if ΔQ is positive, then the supplier should build up the extra capacity ΔQ, otherwise, the supplier should put the $|\Delta Q|$ as inventory. Before we give the detail of models, we denote the notations as follow.

q Demand information/order from the retailer.
Q Normal capacity of the supplier.
X Random demand for the finished product.
u Shortage cost per unit for the finished product.
v Holding (recycling) cost of the primary product per unit of retailer
h Cost for overcapacity (holding cost) of the supplier.
p Product price to the market demand;
p_0 Purchase cost of the primary product per unit/ supply cost;
ω Cost for increase the capacity by one unit;
$\Pi_r(q, X)$ The retailer's profit when the order is q and the demand is X;
$\Pi_s(q, X)$ The supplier's profit when the order is q and the demand is X;
$E(\Pi_r(q, X))$ The expected profit of the retailer when the order is q and the demand is X;
$E(\Pi_s(q, X))$ The expected profit of the supplier when the order is q and the demand is X;

Under the unknown market demand, the retailer faces the problem of determining the optimal order quality in order to maximum its profit. Here we first give the actual total profit of the retailer, and secondly determine the total expected profit depending on the random demand X.

The total profit model:

$$\Pi(q, X) = p\min\{X, q\} - u(X-q)^+ - v(q-X)^+ - p_0 q \tag{1}$$

In the profit function described above, the first term represents the total revenue and the second two terms describe the shortage cost and the hold cost respectively, the last term shows the procurement cost.

In this paper, we consider the demand X is stochastic, that is to say, we don't know the actual number. However through the market analysis, the distribution function $F(X)$, and the density function $f(X)$ are known to the system, and the same time we fix the bound of X e.g. $[D_1, D_2]$. Here D_1 is the below bound, and D_2 is the upper bound.

From the model (1), the total expected profit for the retailer is given by

$$E(\Pi_m(q,X)) = \int_q^{D_2} pqf(X)dX + \int_{D_1}^{q} Xpf(X)dX - u\int_q^{D_2}(X-q)f(X)dX - v\int_{D_1}^{q}(q-X)f(X)dX - p_0 q \tag{2}$$

The expected profit comprises the expected revenue (the first two terms), the expected customer penalty cost (the third term), the expected inventory cost (the forth term), and the expected procurement cost (the last term). And such the total expected profit can be expressed simply as follow.

$$E(\Pi_m(q,X)) = (p-p_0)q - u(D_2-q) - p\int_{D_1}^{q} F(X)dX + u\int_q^{D_2} F(X)dX - v\int_{D_1}^{q} F(X)dX \tag{3}$$

Here we only give the total expected profit of retailer, and we will come back to this point later in section3 when we discuss different kinds of supply chain.

In model (3), the retailer orders quantity q from the supplier. After receiving the order q for goods, the decision-maker of the supplier will check his inventory Q. He will build up extra capacity if the order quantity is larger than Q, and if the order quantity is less than the Q, the supplier does not use all of the available capacity to produce components which must be stored, thus cost may correspond to the cost for overshooting capacity, for instance to store unused raw materials. And the cost for building up extra capacity is fixed at ω per unit, at the same time, we suppose that the revenue of one unit of the component exceeds the cost of procuring this unit with extra capacity, that's to say, $p_0 > \omega$.

Then, for a given value of the order quantity q of the producer, the profit for the supplier is given as the following models.

$$\Pi_s(q) = p_0(Q+\Delta Q) - \omega(\Delta Q)^+ - h\,|\,(\Delta Q)\,|^- \tag{4}$$

or

$$\Pi_s(q) = p_0 q - \omega(q-Q)^+ - h(Q-q)^+ \tag{5}$$

In this model, the first term is denoted as the revenue, and the last two terms are cost to increase the capacity and cost for overcapacity respectively. From this model, we can see that the profit of supplier isn't directly connected with the market demand. Then the expected function is just same to the model (4) or (5).

3 Worst and Best Analysis

In section2, we have given the models of the retailer' profit as well as the supplier's separately. In real-life situations, the retailer and the supplier optimize their objective functions independently. And apart from the order quantity they don't exchange any further information, which is defined as the worst case in this paper. Excepting this case, there is another one, defined as the best case, in which there is only one decision maker choosing the optimal decision that yields maximize the total expected profit for the whole supply chain. So in the section, we will show how the models can be linked in different ways. We will also compare the results from the different two cases. At the same time, this section will have important relationship with the content of the next section. The worst and best case analysis have decided whether it is necessary to coordinate the supply chain.

3.1 The Worst Case

In the worst case, we assume that the retailer has no idea of the supplier's information. He will order the quantity which can maximize his expected profit. Base on this order quantity q, the supplier determines to product the extra capacity $q-Q$ or holding the surplus capacity $Q-q$.

According to the model (3), the optimal order quality can be calculated as follow. Set:

$$(d/dq)E(\Pi_r(X,q)) = p - p_0 + \pi - pF(q) - \pi F(q) - gF(q) = 0$$

We have the solution

$$q_w^* = F^{-1}(\frac{p - p_0 + \pi}{p + g + \pi}) \tag{6}$$

Since $(d^2/dq^2)E(\Pi_r(X,q)) < 0$, the function $E(\Pi_r(X,q))$ is strictly concave. So the order quality q_w^* can maximize the expected profit of retailer, and retailer's profit is $E(\Pi_r(X,q_w^*))$.

As to the supplier under the worst case, we have two cases to be considered after receiving the order q_w^*, obviously, the supplier's profit is represented as follows.

$$\Pi_s = \begin{cases} p_0 q - h(Q - q_w^*) & Q \geq q_w^* \\ p_0 q - w(q_w^* - Q) & Q < q_w^* \end{cases} \tag{7}$$

3.2 Best Cast: Joint Operation System

In this subsection, we will discuss the every member of supply chain management has the shared information, and common goals. The retailer cares about not only his profit, but also have fully known productivity or its order quality from the whole prospective. Depending on q and X the actual profit of the supply chain is given below.

$$\Pi_j(X,q) = p\min\{X,q\} - \pi(X-q)^+ - \min\{g,h\}(q-X)^+ \\ -w(q-Q)^+ - \min\{g,h\}(Q-q)^+ \tag{8}$$

The profit function consists of the total revenue, shortage cost for not meeting the external demand, holding cost and cost for building up extra capacity. Note that in the cases, the decision-maker of supply chain management will hold the overcapacity or over-order product in the place where has the lesser holding cost.

Then the expected total joint-profit is given as the following model.

$$E(\Pi_j(q,X)) = \int_q^{D_2} pqf(x)dx + \int_{D_1}^{q} xpf(x)dx - \pi\int_q^{D_2} xf(x)dx + \pi q\int_q^{D_2} f(x)dx$$

$$-\min\{g,h\}q\int_{D_1}^{q} f(x)dx + \min\{g,h\}\int_{D_1}^{q} xf(x)dx - w(q-Q)^+ - \min\{g,h\}(Q-q)^+$$

To this complex model, we can be formulated below simply.

$$E(\Pi_j(q,X)) = pq - p\int_{D_1}^{q} F(X)dX - \pi D_2 + \pi q - \min\{g,h\}\int_{D_1}^{q} F(X)dX$$

$$\pi\int_q^{D_2} F(X)dX - w(q-Q)^+ - \min\{g,h\}(Q-q)^+ \tag{9}$$

Since the total expected joint-profit function satisfies $(d^2/dq^2)E(\Pi_j(X,q)) < 0$, then the function $E(\Pi_j(X,q))$ is also strictly concave, and we can find the optimal solution through the following method.

Set $(d/dq)E(\Pi_j(X,q)) = 0$, i.e. the two cases below.

Case 1. when $q > Q$, we have

$$p - pF(q) + \pi - \min\{g,h\}F(q) - \pi F(q) - w = 0.$$

Then the solution is

$$q_1 = F^{-}\left(\frac{p+\pi-w}{p+\min\{g,h\}+\pi}\right)$$

If $q_1 > Q$, then the optimal order is q_1, otherwise set that the optimal order is Q for the function $E(\Pi_j(X,q))$ is strictly concave.

Case 2. when $q_1 \le Q$, we have

$$p - pF(q) + \pi - \min\{g,h\}F(q) - \pi F(q) + \min\{g,h\} = 0.$$

Then the solution is $q_2 = F^{-}(1)$

If $q_2 \le Q$, then the optimal order is q_2, otherwise set that the optimal order is Q for the function $E(\Pi_j(X,q))$ is strictly concave.

From the analysis above, we can draw the following conclusion.

$$\hat{q} = \begin{cases} F^{-}\left(\dfrac{p+\pi-w-\min\{g,h\}}{p+\min\{g,h\}+\pi}\right) & F^{-}\left(\dfrac{p+\pi-w-\min\{g,h\}}{p+\min\{g,h\}+\pi}\right) > Q \\ Q & F^{-}\left(\dfrac{p+\pi-w-\min\{g,h\}}{p+\min\{g,h\}+\pi}\right) \le Q \le F^{-}(1) \\ F^{-}(1) & F^{-}(1) < Q \end{cases} \tag{10}$$

4 Coordination Mechanisms

The worst /best case analysis shows that in a decentralized system without coordination the total expected profit of the entire supply chain is usually higher than in a centralized one. In the following, we will determine appropriate coordination mechanisms, which allow the decentralized system to achieve the same performance (the same maximal overall total expected profit) as a centralized supply chain. Here, we introduce the revenue-sharing contract, which can coordination the profits between supplier and retailer.

The revenue-sharing contract is flexible in the sense that it allows the supply chain profit to be divided between the supplier that the retailer according to any pre-negotiated φ within certain bounds as specified below.

In the following analysis, we develop a lower bound and an upper bound for the retailer's shave φ of the supply chain' profit.

The retailer's expected profit in absence of any contract cannot be greater than that in the revenue-sharing contract. That is,

$$\varphi(E(\Pi_j(X,\hat{q}))) \ge E(\Pi_m(q_w^*, X)) \tag{11}$$

Which gives a lower bound,

$$\frac{E(\Pi_m(q_w^*, X))}{E(\Pi_j(X,\hat{q}))}$$

Likewise, for the supplier,

$$(1-\varphi)E((\Pi_j(X,\hat{q}))) \ge \prod_s(q_w) \tag{12}$$

Then the upper bound,

$$\frac{E(\Pi_j(X,\hat{q})) - \Pi_s(q_w^*)}{E(\Pi_j(X,\hat{q}))}$$

So, In order to ensure that retailer and supplier have incentives to enter into a revenue-sharing contract, the retailer's share φ of the supply chain's profit should satisfy:

$$\frac{E(\Pi_m(q_w^*,X))}{E(\Pi_j(X,\hat{q}))} \le \varphi \le \frac{E(\Pi_j(X,\hat{q})) - \Pi_s(q_w^*)}{E(\Pi_j(X,\hat{q}))} \tag{13}$$

It is easy to see that if the supplier and the retailer have agreed to the revenue-sharing contract, then both will be benefited, and the profit of the supply chain, the supplier and the retailer will increased by amount

$$E(\Pi_j(X,\hat{q})) - E(\Pi_m(q_w^*,X)) - \Pi_s(q_w^*)$$

$$(1-\varphi)E((\Pi_j(X,\hat{q}))) - \Pi_s(q_w^*)$$

and

$$\varphi(E(\Pi_j(X,\hat{q}))) - E(\Pi_m(q_w^*,X))$$

respectively.

5 Discussion

In order to get a better understanding of the underlying contract mechanisms problem and to illustrate the flexibility of profit allocation using the described coordination we consider a numerical example. For a general and deeper discussion of the behavior of the contract mechanisms, and particularly the range over which the total cost of the supply chain can be allocated.

We assume the special case that the random market demand X follows a uniform distribution over the range $[2000, 3800]$, thus the density function $f(x)$ is given by

$$f(x) = \begin{cases} 1/1800 & if \quad 200 \le x \le 3800 \\ 0 & otherwise \end{cases}$$

The other parameters are specified as follows: The price charged by the supplier to the retailer for each unit of the supply product and the price for each unit to the marked are p_0 =\$10 and p =\$16, respectively. The retailer must store any unsold supply product at a value of v =\$4 per unit and pays a shortage cost for the final product, which is u =\$30 per unit. In order to illustrate the complexity of the problem, we consider three cases to the normal capacity of the supplier, which are

$Q = 3000, Q = 3600,$ and $Q = 4200$ respectively. To increase the capacity by one unit costs the supplier ω=\$4 and the holding cost is h=\$2 per unit. The analysis under different cases is given blow.

Case 1. When normal capacity of the supplier $Q = 3000$. Worst case compared with best case (φ:=0.22)**.**

the optimal order	q	$E(\Pi_s)$	$E(\Pi_r)$	$E(\Pi_j)$
Worst case	3296	31480	8328	39808
Best case	3537	32385	9134	41519

Case 2. When normal capacity of the supplier $Q = 3600$. Worst case compared with best case (φ:=0.22)**.**

the optimal order	q	$E(\Pi_s)$	$E(\Pi_r)$	$E(\Pi_j)$
Worst case	3296	32352	8328	8328
Best case	3600	34684	9783	44467

Case 3. When normal capacity of the supplier $Q = 4200$ .Worst case compared with best case (φ:=0.22)**.**

the optimal order	q	$E(\Pi_s)$	$E(\Pi_r)$	$E(\Pi_j)$
Worst case	3296	31152	8328	39480
Best case	3800	34164	9636	43800

References

1. Bard, J., Qi, X., Yu, G.: Supply Chain Coordination with Demand Disruptions. Omega 32(4), 301–312 (2004)
2. Xiao, T., Yu, G., Sheng, Z.: Reputation, Utility and Technological Innovation Strategies. Information Technology and Decision Making 3(1), 81–100 (2004)
3. Cohen, M.A., Lee, H.L.: Strategic Analysis of Integrateed Production-Distribution Systems: Models and Methods. Operations Research 36(2), 216–228 (1988)
4. Monahan, J.P.: A Quantity Discont Pricing Model to Increase Vendor Profits. Management Science 30(6), 720–726 (1984)
5. Whang, S.: Analysis of Interorganizational Information Sharing. Journal of Organizational Computing 3(3), 357–377 (1993)
6. Desiraju, R., Moorthy, S.: Managing a Distribution Channel under Asymmetric Information with Performance Requirements. Management Science 43(12), 28–34 (1997)
7. Kekre, S., Murthi, B.P.S., Srinivasan, K.: Operating Decisions, Supplier Availability and Quality: An Empirical Study. Journal of Operations Management 12(2), 387–396 (1995)

Hyper-Chaotic Mathematical Programming Method and Its Application to Dodecahedron Variable Geometry Truss Manipulator

Youxin Luo, Bin Zeng, and Zheming He

Department of Mechanical Engineering, Hunan University of Arts and Science, Changde 415000, China
LLYX123@126.com

Abstract. As a new robot mechanism variable geometry truss manipulator can be applied widely for its good performance. Forward displacement of variable geometry truss manipulator is always transformed into finding the solutions of nonlinear equations. Newton iterative method is an important technique to one dimensional and multidimensional variable and often used to solve nonlinear equations. Iterative process is sensitive to the initial point. The mathematical programming method is adopted when the iteration diverges with the Newton and quasi-Newton methods. A new method of finding all solutions of nonlinear questions is proposed, in which we combine mathematical programming method with hyper-chaotic neuron network system and utilize hyper-chaotic network system to obtain locate initial points. The numerical example in dodecahedron variable geometry truss manipulator synthesis shows that all solutions have been quickly obtained, and it also shows that the method is correct and effective. This provides a simple and new method for mechanism design.

Keywords: Hyper-chaotic system, Neuron network, Mathematical programming method, Dodecahedron variable geometry truss manipulator, Non-linear equations.

1 Introduction

As robot mechanism the variable-geometry truss mechanisms have their particular structure characteristic and good mechanical performance, which overcome the bad stiffness in the sequence robots and little workspace in the parallel manipulator. The variable-geometry truss mechanism has the extensive potential application foreground in space navigation, industrial automation and new robot mechanism and so on. Variable-geometry truss mechanism is composed of series of variable geometry fundamental unit, in which tetrahedron, octahedron, decahedron and dodecahedron [1] are their four basic units. In this kind of mechanism only pull or tension is loaded on every component, which leads to good stiffness, high carrying capacity, large workspace and good facility. Many researchers have done much work [1-10] about the kinematic analysis of this kind of mechanism, which is composed of four fundamental units. The configuration analysis of these mechanisms is about how to find the solutions of non-linear equations and it is a very tough theoretic problem. We often use

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 1033 – 1040, 2009.

numerical methods, analytic methods and numerical & analytic hybrid methods to find solutions. Numerical methods include optimization method, interval analysis method and Homotopy continuation method etc. and analytic methods include polynomial remainder method, Wu methods and Groebner base method etc.[6-8] Analytic methods can find all the solutions with the shortcoming of large number of middle dilations term and long calculation time and unable to solve the high dimensionality problem. Numerical methods often can only find one solution with a high demand of initial value. The Homotopy continuation method can also find all the solution, but the computation in range of complex number field, computation task is huge and is hard to create the original equations. Chaos is one of most achievement in 21st century. How to use the chaotic characteristic in the mechanism is very important. Using the chaotic method, Luo et al.[10] had found the positive solutions of 6-SPS mechanism in the real numbers rang, in which he considered Julia set point appears in the neighbor space of solution equations' Jacobian matrix determinant whose value is zero. But assumption is not proved and to the multi-variables Jacobian matrix determinant it is very difficult. Chaotic serials method is a new method, in which using chaotic and hyper-chaotic system to generate Newton initial value we can find all real solutions [11-13] in the mechanism synthesis. When divergent in the Newton and Quasi-Newton methods [14,15], we adopt hyper-chaos-based mathematic programming. The man-made neural network system is very complicated nonlinearity dynamics system that is the most perfect information processing with the function is up to now the strongest and the efficiency is the highest [16]. This paper provides a new method of mechanism synthesis, in which mathematical programming method combines hyper-chaos neural network system variable geometry truss manipulator.

2 Hyper-Chaotic Neural Network of Permanent Sustaining Chaos

Model of hyper-chaotic neural network of permanent sustaining chaos is as follows [16].

$$\begin{cases} x_i(t) = e^{-(\mu \cdot y_i(t))^2/2} \cos(5\mu \cdot y_i(t)) \\ y_i(t+1) = k \cdot y_i(t) + \alpha \cdot (\sum_{\substack{j \\ i \neq j}}^{n} W_{ij} x_i(t) + I_i) - z_i(t) \cdot (x_i(t) - I_0) \\ z_i(t+1) = (1-\beta) \cdot z_i(t) \end{cases} \tag{1}$$

Where, x_i is neuron output, y_i is internal condition of the i[th] neuron, W_{ij} is connect weights from the j[th] neuron to the i[th] neuron, I_i is biased of the i[th] neuron, I_0 is positive constant, k is damping factor of neuron （$0 \le k \le 1$）, β is the constant of simulated annealing rate （$0 \le \beta \le 1$） ; $z_i(t)$ is the constant term of self-control feedback, α is joint strength between neurons, μ is parameter of activation function, n is the number of neurons.

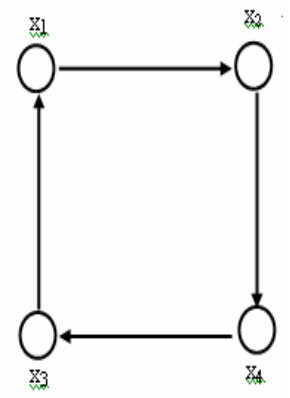


Fig. 1. Topological structure of uni-axial cycle connection

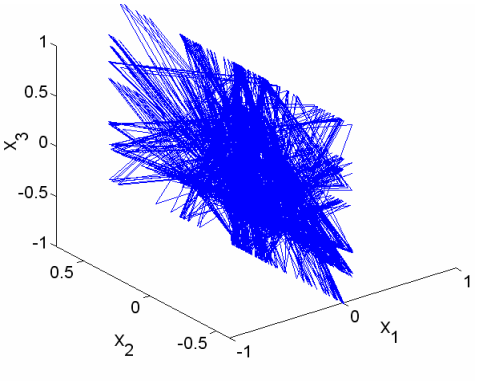


Fig. 2. Phase diagram of neuron

While $\beta = 0$, self-control feedback term is zero, chaos has the characteristic of permanent sustaining [16]. Figure 1 is the joint neuron network of four neurons (n=4), let $\mu = 2, z_i(t) = 0.3, I_0 = 0.3, k = 0.2,\quad \alpha = 0.7$, internal condition initial value $y_i(1) = [rand(1), rand(1), rand(1), rand(1)]^{\mathbf{T}}$, is generated randomly, neuron biased is $\mathbf{I} = [0.92181, 0.73821, 0.17627, 0.40571]^{\mathbf{T}}$, weights is

$$W = \begin{bmatrix} 0 & 0.95013 & 0 & 0 \\ 0 & 0 & 0.8913 & 0 \\ 0 & 0 & 0 & 0.82141 \\ 0.4447 & 0 & 0 & 0 \end{bmatrix}$$

After calculating Lyapunov exponents are , respectively

$$LE_1 = 0.380476, LE_2 = 0.201916$$
$$LE_3 = -0.0.019262, LE_4 = -0.594168$$.

The system has two positive Lyapunov exponents. The system is hyper-chaotic system and phase diagram is shown as Figure 2.

3 Mathematical Programming of Nonlinear Equations

Newton iterative method is often used and it is an important one-dimensional and multi-dimensional technique. We use mathematical programming method when Newton method and quasi-Newton method are not convergent [14,15].

If nonlinear equations are

$$f_i(\mathbf{x}) = 0(\mathbf{x} \in D \subset R^n, i = 1,2,\cdots,n) \tag{2}$$

Where $f_i(\mathbf{x})$ has continuous 2-rank derivative in convex set, and there is $\mathbf{x}^* \in D$ satisfying $f_i(\mathbf{x}^*) = 0(i = 1,2,\cdots,n)$.

Denoting $s_i = sign(f_i(x^0))$, where x^0 is initial value of D, because of function continuous, near $\mathbf{x}^0$, $\mathbf{x}$ satisfies

$$s_i f_i(\mathbf{x}) \geq 0(i = 1,2,\cdots,n) \tag{3}$$

Equations (2) can be transformed into

$$\begin{cases} \sum_{i=1}^{n} s_i f_i(\mathbf{x}) \to \min \quad (\mathbf{x} \in D) \\ subject.to: \quad s_i f_i(\mathbf{x}) \geq 0 \quad (i = 1,2,\cdots,n) \end{cases} . \tag{4}$$

Because $s_i f_i(\mathbf{x}) \geq 0$, so $\sum_{i=1}^{n} s_i f_i(\mathbf{x}) \geq 0$, substituting $f_i(\mathbf{x}^*) = 0$ into formula (4) to satisfy constraint condition and objective value is zero. So $\mathbf{x}^*$ are the solutions of formula (4). If there are no solutions for equations (2), the optimum point can not be the solutions of equations (2). If $f_i(\mathbf{x})$ has no continuous 2-rank derivative in convex set, we use numerical difference substituting derivative.

4 Solution Method for Nonlinear Equations Based on Hyper-Chaotic Mathematical Programming

The steps of solution for nonlinear equations based on mathematical programming of hyper-chaotic neural network are as follows.

Step 1. For equation (1), select corresponding parameters of neural network of four neuron connection and generate initial point of internal state y_i to create hyper-chaotic set $x0(i_1, j_1)$ ($i_1 = 1,2.,\cdots,n_1$, n_1 is the number of variables, $j_1 = 1,2,\cdots,N$, where N is the length of hyper-chaotic set, where N is changeable with the tried solution). In addition, if $n_1 > 4$ we should run many times to generate $x0(i_1, j_1)$.

Step 2. Take j_1^{th} chaotic sequence $x(:, j_1)$ as the initial value of mathematical programming method, after running N times, we can get all real solutions x^*.

5 Forward Displacement of Dodecahedron Variable Geometry Truss Manipulator

Dodecahedron variable geometry truss manipulator of six degrees is shown as figure 3. The mechanism is composed of ending platform FGH, six flex member (input bar), six stable length member and basic platform. With the change of input member length, ending platform will move compared to the basic platform and basic platform is fixed to the referent coordinate system.

The constraint equations of dodecahedron variable geometry truss manipulator can be expressed as follows [3].

$$\begin{cases} (G_x-F_x)^2+(G_y-F_y)^2+(G_z-F_z)^2=L_{GF}^2 \\ (G_x-H_x)^2+(G_y-H)^2+(G_z-H_z)^2=L_{GH}^2 \\ (G_x-E_x)^2+(G_y-E_y)^2+(G_z-E_z)^2=L_{GE}^2 \\ (G_x-C_x)^2+(G_y-C_y)^2+(G_z-C_z)^2=L_{GC}^2 \\ (G_x-A_x)^2+(G_y-A_y)^2+(G_z-A_z)^2=L_{GA}^2 \\ (G_x-D_x)^2+(G_y-D_y)^2+(G_z-D_z)^2=L_{GD}^2 \\ (B_x-D_x)^2+(B_y-D_y)^2+(B_z-D_z)^2=L_{BD}^2 \\ (B_x-F_x)^2+(B_y-F_y)^2+(B_z-F_z)^2=L_{BF}^2 \\ (B_x-H_x)^2+(B_y-H_y)^2+(B_z-H_z)^2=L_{BH}^2 \\ (B_x-E_x)^2+(B_y-E_y)^2+(B_z-E_z)^2=L_{BE}^2 \\ (C_x-E_x)^2+(C_y-E_y)^2+(C_z-E_z)^2=L_{CE}^2 \\ (E_x-H_x)^2+(E_y-H_y)^2+(E_z-H)^2=L_{EH}^2 \\ (H_x-F_x)^2+(H_y-F_y)^2+(H_z-F_z)^2=L_{HF}^2 \\ (D_x-F_x)^2+(D_y-F_y)^2+(D_z-F_z)^2=L_{FD}^2 \\ (D_x-A_x)^2+(D_y-A_y)^2+(D_z-A_z)^2=L_{DA}^2 \end{cases} \tag{5}$$

In equation (5), there are 15 constraint equations, given the bar length and every coordinate of joint points of basic platform, there are 15 variables for D,E,F,G,H coordinate.

Let $\mathbf{x}=[x_1,x_2,\cdots,x_{15}]^T=[F_x,F_y,F_z,H_x,H_y,H_z,E_x,E_y,E_z,D_x,D_y,D_z,G_x,G_y,G_z]^T$, equation (5) can be transformed into $\mathbf{F}(\mathbf{x})=[f_1,f_2,\cdots,f_{15}]^T=0$.

Given the geometry parameters

$$L_{GD}=L_{GF}=L_{FH}=L_{GH}=L_{GE}=L_{BD}=L_{BE}=L_{DA}=L_{CE}=1$$

$$A_x=1, A_y=A_z=B_x=B_y=B_z=C_z=0, C_x=0.5, C_y=\frac{\sqrt{3}}{2}$$

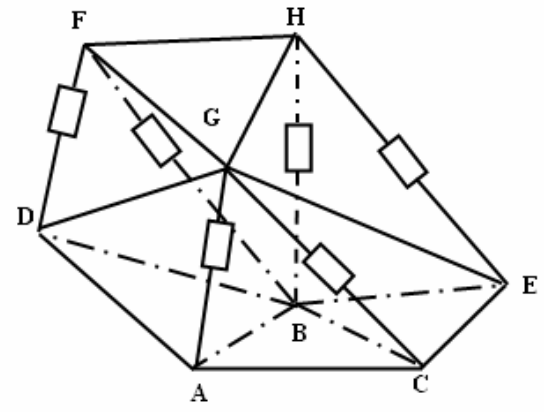


Fig. 3. Dodecahedron variable geometry truss manipulator of six degrees

Table 1. Forward displacement of dodecahedron variable geometry truss manipulator

NO	x_{41}	x_{42}	x_{43}	x_{31}	x_{32}	x_{33}	x_{21}	x_{22}
1	-0.0733	1.0942	-0.4872	0.6791	0.9762	0.1609	0.9585	0.0240
2	-0.0733	1.0942	0.4872	0.6791	0.9762	-0.1609	0.9585	0.0240
3	0.0982	0.6341	1.0141	0.5982	-0.2320	1.0141	0.0231	0.5640
4	0.0982	0.6341	-1.0141	0.5982	-0.2320	-1.0141	0.0231	0.5640
5	0.1206	0.7048	-0.9637	0.3253	1.1513	-0.0927	0.0664	0.5390
6	0.1206	0.7048	0.9637	0.3253	1.1513	0.0927	0.0664	0.5390
7	0.2523	1.0273	-0.5666	0.5958	0.9733	0.3710	0.9397	0.0348
8	0.2523	1.0273	0.5666	0.5958	0.9733	-0.3710	0.9397	0.0348
9	0.3115	1.0116	0.5654	0.5642	0.9800	-0.4016	0.9207	0.0458
10	0.3115	1.0116	-0.5654	0.5642	0.9800	0.4016	0.9207	0.0458
11	0.6889	0.9751	-0.1210	1.1889	0.1091	-0.1210	0.4956	0.2912
12	0.6889	0.9751	0.1210	1.1889	0.1091	0.1210	0.4956	0.2912
13	1.0205	0.4565	0.4360	0.5377	1.0530	-0.2051	-0.4131	0.8159
14	1.0205	0.4565	-0.4360	0.5377	1.0530	0.2051	-0.4131	0.8159
15	1.0933	0.1521	0.4707	0.6077	1.0010	0.2622	-0.3093	0.7559
16	1.0933	0.1521	-0.4707	0.6077	1.0010	-0.2622	-0.3093	0.7559
17	1.1308	-0.0014	-0.4016	1.0318	-0.2360	0.5654	0.9207	0.0458
18	1.1308	-0.0014	0.4016	1.0318	-0.2360	-0.5654	0.9207	0.0458
19	1.1408	0.0293	0.3710	1.0158	-0.2952	-0.5666	0.3400	0.3810
20	1.1408	0.0293	-0.3710	1.0158	-0.2952	0.5666	0.3400	0.3810
21	1.1597	-0.2940	0.0927	0.6707	-0.2480	0.9637	0.0664	0.5390
22	1.1597	-0.2940	-0.0927	0.6707	-0.2480	-0.9637	0.0664	0.5390
23	1.1707	0.0258	0.2622	0.6784	0.8708	0.4707	-0.3093	0.7559
24	1.1707	0.0258	-0.2622	0.6784	0.8708	-0.4707	-0.3093	0.7559
25	1.1808	-0.0609	-0.2051	0.9056	0.6555	0.4360	-0.0203	0.5891
26	1.1808	-0.0609	0.2051	0.9056	0.6555	-0.4360	-0.0203	0.5891
27	1.1850	0.1000	-0.1609	0.9109	-0.6106	0.4872	0.9585	0.0240
28	1.1850	0.1000	0.1609	0.9109	-0.6106	-0.4872	0.9585	0.0240

Table 1. (*continued*)

NO	x_{23}	x_{51}	x_{52}	x_{53}	x_{81}	x_{82}	x_{83}
1	0.2842	0.5000	0.8181	0.2842	0.5492	0.3171	-0.5798
2	-0.2842	0.5000	0.8181	-0.2842	0.5492	0.3171	0.5798
3	0.8254	0.5000	-0.2620	0.8254	0.9875	0.5701	0.5611
4	-0.8254	0.5000	-0.2620	-0.8254	0.9875	0.5701	-0.5611
5	-0.8397	0.5000	-0.2120	-0.8397	1.0196	0.5886	-0.5414
6	0.8397	0.5000	-0.2120	0.8397	1.0196	0.5886	0.5414
7	0.3403	0.5000	0.1039	-0.8598	1.1721	0.6767	-0.3905
8	-0.3403	0.5000	0.1039	0.8598	1.1721	0.6767	0.3905
9	-0.3876	0.5000	0.7745	-0.3876	1.2256	0.7076	0.2972
10	0.3876	0.5000	0.7745	0.3876	1.2256	0.7076	-0.2972
11	-0.8183	0.5000	0.2836	-0.8183	0.2209	0.1275	0.1292
12	0.8183	0.5000	0.2836	0.8183	0.2209	0.1275	-0.1292
13	-0.4046	0.5000	-0.3121	0.8078	0.2106	0.1216	-0.0455
14	0.4046	0.5000	-0.3121	-0.8078	0.2106	0.1216	0.0455
15	0.5770	0.5000	-0.6458	0.5770	0.2092	0.1208	0.0045
16	-0.5770	0.5000	-0.6458	-0.5770	0.2092	0.1208	-0.0045
17	-0.3876	0.5000	0.7745	-0.3876	1.2256	0.7076	0.2972
18	0.3876	0.5000	0.7745	0.3876	1.2256	0.7076	-0.2972
19	-0.8598	0.5000	0.7964	0.3403	1.1721	0.6767	-0.3905
20	0.8598	0.5000	0.7964	-0.3403	1.1721	0.6767	0.3905
21	0.8397	0.5000	-0.2120	0.8397	1.0196	0.5886	0.5414
22	-0.8397	0.5000	-0.2120	-0.8397	1.0196	0.5886	-0.5414
23	0.5769	0.5000	-0.6459	0.5769	0.2092	0.1208	0.0045
24	-0.5770	0.5000	-0.6458	-0.5770	0.2092	0.1208	-0.0045
25	0.8078	0.5000	-0.7657	-0.4046	0.2106	0.1216	-0.0455
26	-0.8078	0.5000	-0.7657	0.4046	0.2106	0.1216	0.0455
27	-0.2842	0.5000	0.8181	-0.2842	0.5492	0.3171	0.5798
28	0.2842	0.5000	0.8181	0.2842	0.5492	0.3171	-0.5798

And lengths of driven part are $L_{GA} = L_{GC} = 0.8, L_{BF} = L_{BD} = 1.2, L_{FD} = L_{FB} = 1$. To solve equation $\mathbf{F}(\mathbf{x}) = 0$ with the proposed method, all the real solutions are shown as Table 1 and the results are the same the reference [3], but the calculation is done in real number field.

6 Conclusions

The paper investigates the combination of mathematical programming and hyper-chaotic neural network system. The hyper-chaotic sequences generated by the hyper-chaotic neural network system and taken as initial values of mathematical programming methods, all solutions of nonlinear equations are found. The synthesis sample of dodecahedron variable geometry truss manipulator proves the method is correct and effective. It lays a good foundation for engineering and provides a good way for chaotic characteristic of other iterative method.

Acknowledgement

This research is supported by the grant of the 11th Five-Year Plan for the construct program of the key discipline (Mechanical Design and Theory) in Hunan province (XJT2006180), National Science Foundation of China (No. 50845038), Hunan Provincial Natural Science Foundation of China (07JJ3093), Hunan Province Foundation Research Program (2007FJ3030, 2007GK3058).

References

1. Arun, V., Reinholtz, C.F., Watson, L.T.: Enumeration and Analysis of Variable Geometry Truss Manipulators. In: Proceedings of 1990, ASME Mechanisms Conference, Chizaga, vol. DE-26, pp. 93–98 (1990)
2. Subramaniam, M., Kramer, S.N.: The Inversion Kinematic Solution of the Tetrahedron Besed Variable Geometry Truss Manipulators. ASME Journal of Mechanical Design 114, 443–447 (1992)
3. Liu, A.X., Yang, T.: Kinematics Design in Mechanical System. China Petrochemical Press, Beijing (1999)
4. Yao, J., Fang, H.R.: Forward Displacement Analysis of the Decahedral Variable Geometry Truss Manipulator. Robotics and Autonomous Systems 15, 173–178 (1995)
5. Griffs, M., Duffy, J.: A Forward Displacement Analysis of a Class Stewart Platforms. Journal of Robotic Systems 16, 703–720 (1989)
6. Hang, L.B., Wang, Y.: Forward Displacement Analysis of Octahedron Variable Geometry Truss Based on Groebner Basis. Mechanical science and Technology 23(6), 745–747 (2004)
7. Yao, J., Rong, H.: Forward Displacement Analysis of the Decahedron Variable Geometry Truss Manipulator. Journal Robotics and Autonomous System 15(3), 173–178 (1995)
8. Du, G.J.: A forward Displacement Analysis of a Class of Stewart Platforms. Journal of Robotic Sysems 6, 703–720 (1989)
9. Luo, Y.X., Guo, H.X.: Newton Chaos Iteration Method and Its Application to Mechanism Kinematics Synthesis. Journal of Harbin Institute of Technology 14, 13–17 (2007)
10. Luo, Y.X., Li, D.: Finding All Solutions to Forward Displacement Analysis Problem of 6-SPS Parallel Robot Mechanism with Chaos-iteration Method. Journal of Chinese Engineering Design 10(2), 95–101 (2003)
11. Luo, Y.X., Fan, X.F., Li, D.Z.: Hyper-chaotic Mapping Newton Iterative Method to Mechanism. Journal Of Mechanical Engineering 54, 372–378 (2008)
12. Luo, Y.X., Liao, D.G.: Coupling Chaos Mapping Newton Iterative Method and Its Application to Mechanism Accurate Points Movement Synthesis. Journal of Chinese Mechanical Transmission 31, 28–30 (2007)
13. Luo, Y.X., Li, X.F., Liao, D.G.: Chaos Mapping Newton Iterative Method and Its Application to Mechanism Synthesis. Journal of Chinese Mechanical Transmission 31, 35–36, 44 (2007)
14. Luo, Y.X.: Hyper-chaotic Mathematical Programming Method and Its Application to Mechanism Synthesis of Parallel Robot. Transactions of the Chinese Society for Agricultural Machinery 39, 133–136 (2008)
15. Shui, Y.K., Yao, W.Z.: Quadratic Programming Method for Nonlinear Equations and Its Application. Chinese Journal of Computational Mechanics 19, 245–246 (2002)
16. Xu, Y.Q., Sun, M., Yang, S.W.: Hyper-chaos in the Chaotic Neuron Network. Journal of Harbin University of Commerce 22, 54–57 (2006)

An Adaptive MO-HGA for Resource-Constrained Transport Task Scheduling

Jian Wang[1,2] and Hongwei Wang[1,2]

[1] Systems Engineering Institute, Huazhong University of Science and Technology, Wuhan Hubei 430074, China
[2] Key Laboratory of Image Processing and Intelligent Control, Wuhan Hubei 430074, China
JianWang@126.com

Abstract. This paper proposes an adaptive multi-objective hybrid genetic algorithm (MO-HGA) based on the serial scheduling method to solve the resource-constrained transport task scheduling problem (RCTTSP) with two optimal objectives. The proposed algorithm uses the serial scheduling method to initialize the population and evaluate the individual, and use the weighted sum method and the rank-based fitness assignment method to assign the individual fitness. Furthermore, an adaptive GA parameters tuning method based on fuzzy logic controller is implemented to improve the performance of the algorithm. Firstly, this paper describes the multi-objective RCTTSP and presents the principle of the adaptive MO-HGA, and then develops the algorithm to implement several experimental cases with different problem sizes, lastly the effectiveness and efficiency of the algorithm are compared. The numerical result indicates that the proposed adaptive MO-HGA can resolve the proposed multi-objective resource-constrained transport task scheduling problem efficiently.

Keywords: Multi-Objective, Resource-Constrained Transport Task Scheduling, Serial Scheduling, Hybrid Genetic Algorithm, Fuzzy Logic Controller.

1 Introduction

The multi-objective resource-constrained transport task scheduling problem considers a transport scheduling problem in a transport network of multiple transport entities distributed across a predefined geographical area, and each entity has different tasks need to be scheduled simultaneously. A transport task is defined as a request to load some cargo at a certain entity after a given release time, to move it to its destination entity and to unload the cargo before a given due time. Each entity has limited capacity of transport resources including vehicles, docks for loading and unloading, parking lot and cargo storage space. Any entity can rent the vehicle from other entities when ever there is a lack of the vehicle, and the vehicle must return to its owner when the cargo has been transported to the destination entity. The executive process of a transport task consists of a series of activities including empty vehicle travel, loading, transporting with cargo, parking, unloading, storage and empty vehicle return. The

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 1041–1052, 2009.

question is how to integrate these resource capacity restrictions in the scheduling process for the tasks of all entities. The result of scheduling is to determine the start time, the finish time and the operational cost of each activity, such that the precedence constraint and the resource constraint are obeyed. The optimal objectives include the total operational cost of all tasks is minimized and the total delay of all tasks is minimized.

The multi-objective RCTTSP can be modeled as a vehicle scheduling problem with time windows[1,2]. The literature on vehicle scheduling problem seldom considers the multiple terminal heterogeneous resource capacity restrictions and the transport activity scheduling. Kim and Tanchoco[3] consider a manufacturing job shop with the automated guided vehicles. Such a system is characterized by several complex features such as the random pattern of material flows between machines, the limited capacity of machine buffers, shop deadlock phenomena, floating bottlenecks, etc. Van der Heijden and M.C., Ebben[4] apply a serial scheduling method to solve the vehicle scheduling problem in an automated transportation network. In their method, they neglect finite resource capacities at the terminals. In their cases, there are no parking and storage restrictions. M.J.R. Ebben and M.C. van der Heijden[5] present a dynamic vehicle scheduling problem with multiple resource capacity constraints. The transport resources include vehicles, docks, parking places, and storage. David Naso and Michele Surico[6] consider the supply chain for the production and distribution of ready-mixed concrete. The limited capacities of trucks, loading dock and unloading dock are taken into account. Loading, transport, unloading, and return activities are scheduled simultaneously. In these papers, the problem of multiple objectives optimized simultaneously does not be covered.

The multi-objective RCTTSP can be characterized as an activity scheduling problem with the precedence and resource constraints. Resource constraints and activity scheduling are common for the resource-constrained project scheduling problem (RCPSP). The RCPSP is a generalization of the static job shop, flow shop, assembly line balancing, related scheduling problem and hence belongs to the class of NP-hard problems[7,8]. Two well-known heuristics for the resource-constrained project scheduling problem are the serial and parallel scheduling methods[9,10]. The classical RCPSP considers a determinate set of activities to be scheduled. But, the RCTTSP needs to determine some activities such as empty vehicle travel when the vehicle is rented from the other entity temporarily during the process of scheduling, not before scheduling. So the serial or parallel scheduling method for RCPSP can not be applied directly, so that it is needed to extend the serial or parallel scheduling method with the specific heuristic rules for transport activity scheduling. In the research area on RCPSP, some authors consider the hybrid genetic algorithm based on the serial scheduling method, which has been shown can improve serial scheduling method efficiently and is superior to the large-scale RCPSP problems[11,12,13]. Furthermore, the adaptive genetic algorithm has been developed to improve the convergence performance and the global search ability of the traditional GA which sets the GA parameter values fixed[14,15,16].

Recent years, the GA has been applied to resolve the multi-objective scheduling problem. G.Celano[17] proposes a multi-objective genetic algorithm for the scheduling of a mixed model assembly line. The simple sum and the weighted sum methods are used to calculate the fitness of individual. P.C. Chang[14] discusses a

scheduling problem for drilling operation in a real-world printed circuit board factory is considered. Two derivatives of multi objectives genetic algorithms are proposed under two objectives with the weighted sum method which the weights consist of random numbers. Loo Hay Lee[18] proposes a multi-objective genetic algorithm (MOGA) to solve the flight scheduling problem. The rank-based fitness assignment method and the SIMAIR 2.0 simulation model are used.

In this paper, the scheduling model for the activity scheduling of the multi-objective RCTTSP is proposed, which needs to obey the multiple resource constraints and the precedence constraints in a transportation environment with multiple transport entities. Then an adaptive MO-HGA based on the serial scheduling method and the fuzzy logic controller are developed to solve the multi-objective RCTTSP. Next section describes the multi-objective RCTTSP and presents the mathematical model of the multi-objective RCTTSP. Section 3 proposes a multi-objective hybrid GA which extends the standard GA with the serial scheduling method based on the priority of task and the heuristic rules of the transport activity scheduling, and the weighted sum based fitness assignment method and the rank-based fitness assignment method are used. Moreover, an adaptive GA parameters tuning method based on the fuzzy logic controller is implemented to improve the performance of the MO-HGA. Section 4, the computational experiments of various problem sizes are presented, and the effectiveness and efficiency of the MO-HGA are compared. The conclusion follows in section 5.

2 Multi-objective RCTTSP

The multi-objective resource-constrained transport task scheduling problem is described as follows: (1) Multiple transport entities are considered, each entity has multiple tasks to be scheduled, each task has a series of activities which needs various transport resource, and the start time of each activity is dependent upon the completion of some other activities (precedence constraints). (2) A feasible schedule is to determine the earliest release and due time of each activity, and the precedence and resource constraints are obeyed. (3) The scheduling objective is to minimize the total operational cost and the total delay. The mathematical model of the RCTTSP can be formulated as follows:

$$\begin{aligned} &\min C = \sum_{i=1}^{I}\sum_{l=1}^{L_i}\sum_{m=1}^{M_{il}}\sum_{r=1}^{R_i} q_{mr} * c_{mr} * d_m; \\ &\min TT = \sum_{i=1}^{I}\sum_{l=1}^{L_i} FT_l - DT_l; \\ &r = 1,2,...,R_i;\ m = 1,2,...,M_{il};\ l = 1,2,...,L_i;\ i = 1,2,...,I. \end{aligned} \tag{1}$$

Subject to:

$$FT_n \le FT_m - d_m \qquad n \in P_m; \tag{2}$$

$$\sum_{m=1}^{M_{il}} d_m = FT_l - ST_l \qquad l \in L_i; \tag{3}$$

$$\sum_{m \in M_t} q_{mr} \le Q_r \qquad t = 1,2,...,T; \tag{4}$$

$$0 \le FT_m \le T,\ 0 \le ST_l \le T,\ 0 \le FT_l \le T,\ 0 \le DT_l \le T. \tag{5}$$

i: entity index, i = 1, 2, … ,I. l: task index in entity i, l = 1, 2, … , L_i. m: activity index in task l of entity i, m = 1, 2, … , M_{il}. r: resource index in entity i, r = 1, 2, … , R_i. t: scheduling time, t = 1, 2, … , T.

C: the total operational cost of all tasks. TT: the total delay of all tasks. q_{mr}: the quantity of resource r consumed by activity m of each task of entity i, q_{mr} = 0, 1, 2, …. c_{mr}: the unit cost of the resource r which is utilized for activity m of each task of entity i, c_{mr} = 0, 1, 2, …. d_m: the duration of activity m, d_m = 0, 1, 2, …. FT_m: the finish time of activity m, FT_m = 0, 1, 2, …. ST_l: the start time (the release time) of task l, ST_l = 0, 1, 2, …. FT_l: the finish time (the due time or the ready time) of task l, FT_l = 0, 1, 2, …. DT_l: the expected finish time of task l, DT_l = 0, 1, 2, …. P_m: the set of immediate predecessors of activity m. Q_r: the maximum of resource r at entity i, Q_r = 0, 1, 2, …. M_t: the set of activities being in process period t (t-1, t). T: the maximum of the finish time of all activities.

The objective function (1) minimizes the total cost of all tasks and the total delay. Constraint (2) considers the precedence relations between each pair of activities (n, m), where n immediately precedes m. Constraint (3) denotes that the interval between the release time and the ready time of each task equals the sum of the duration of all activities executed for this task. Constraint (4) states the limitation of the total resource usage within each period. Finally, constraint (5) defines the upper limit of the scheduling time. The executive process of one transport task is shown in Fig. 1.

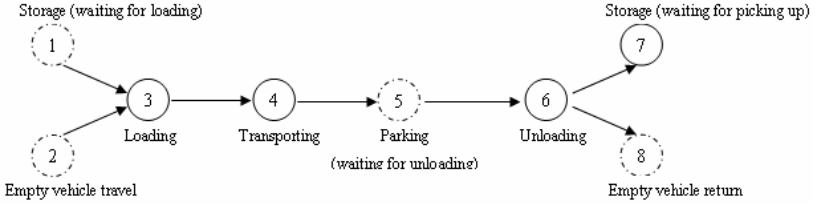


Fig. 1. The executive process of one transport task

A task of the original entity is available to be executed at the release time. In case the cargo cannot be transported immediately, the cargo is stored in the storage. When the vehicle and dock resources are available, the cargo can be loaded, and the vehicle can start driving to the destination after loading. When the vehicle arrives at the destination, it is possible that the vehicle has to wait at the parking place until the unloading and storage capacities are available. After unloading, the cargo is stored in the storage until the due time, and the empty vehicle needs to return or is assigned to transport for a new task immediately. During the process of one task to be executed, the activities 1, 2, 5, 8 are optional. If there are sufficient vehicle and loading resources, the activities 1 and 2 need not to be scheduled. If there is a lack of unloading resource, the activity 5 needs to be considered. The activity 8 could not be executed if the transport destination is just the origin of the vehicle.

3 Adaptive MO-HGA

Because the activities have to be determined during the scheduling process, it is considered to encode the priority sequence of the transport tasks. The position-based crossover (PBC), the partially mapped crossover (PMC), the swap mutation, and the elitist selection are used.

The activities of each task should be scheduled by using a special local search method based on the heuristic rules, which is used to schedule the target value in the individual evaluation phase and to determine the individual fitness. The weighted sum method and the rank-based fitness assignment method are used to assign the individual fitness. The heuristic rules consider the cost and the delay factors, and the serial scheduling process produces two feasible schemes simultaneously.

For the fine-tuning of GA parameters, the fuzzy logic controller (FLC) has been proved very useful. Kwan Woo Kim and Mitsuo Gen[12,13] use the fuzzy logic technique to regulate the increasing and decreasing range of the crossover and mutation probabilities. The scheduling process is shown in Fig. 2.

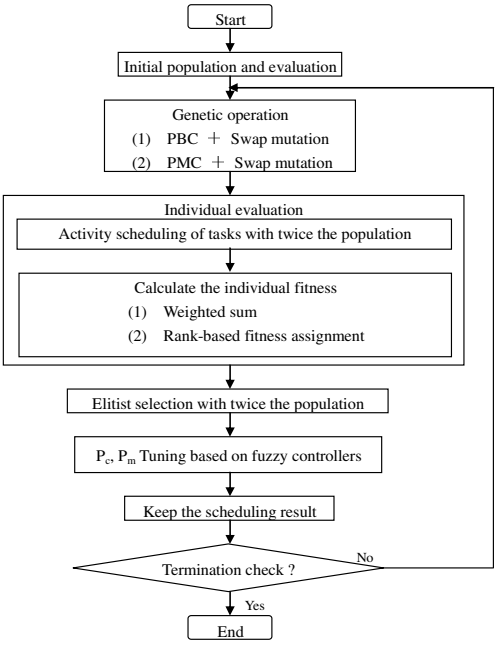


Fig. 2. The scheduling process of the adaptive MO-HGA

3.1 Chromosome Encoding and Genetic Operator

For encoding the sequence of the transport tasks, the position of a gene denotes a task and the value of a gene denotes the priority associated with the task. The value of a gene is an integer exclusively within [1, n] (n: the number of total tasks). During the process of evolution, the position-based crossover operator is to take some genes from

one parent at random and to fill vacuum position with genes from the other parent by a left-to-right scan. The partially mapped crossover operator is to select two random crossover positions and to produce the offspring according to the mapping relation between two parents firstly, and then to perform a repairing procedure to resolve precedence conflict. The swap mutation operator is used, which simply selects two positions at random and swaps their contents. The elitist selection is based on the proportional selection.

3.2 Population Initialization

The initial population to be evolved is produced by using the serial scheduling method based on the priority rules of tasks. Firstly, the entity to be scheduled preferentially is selected. Secondly, a kind of priority rule for the task scheduling is chosen, and the priority of all tasks is determined. The priority rules include the earliest due time, earliest release time, maximum task cost, and maximum duration. Lastly, the other individuals are initialized.

3.3 Individual Evaluation

In individual evaluation phase, the feasible scheduling of each activity to accomplish a task can be obtained by a heuristic local search method. It is to determine the scheduling target value and individual fitness simultaneously. In order to minimize the total cost and the total delay of all tasks, two kinds of heuristic rule for how to obtain the vehicle resources are used. The serial scheduling process with twice the population includes 4 steps:

Step 1 is to convert the priority sequence to task scheduling sequence.

Step 2 is to produce the activity scheduling by using the serial scheduling method based on the special heuristic rules, and to determine the start and finish time of each activity. The heuristic procedure is described as follows:

(1) To check the resource constraints at the original entity i. The vehicle capacity is checked firstly. If there is a shortage of vehicle at entity i, vehicle needs to be rented from the other entity k according to the minimum cost rule, so that the earliest time while a vehicle can be available is determined. Then the loading capacity is checked to determine the earliest time while a loading resource is ready, and the earliest time of departure can be calculated at the same time.

(2) To check the resource constraints at the destination entity j. After the cargo has been transported to entity j, the capacity of parking, unloading, and storage are checked in turn. The earliest time of parking (arriving), unloading, empty vehicle return, and storage can be determined. If the parking resource is not sufficient, the earliest time of vehicle needs to be postponed.

(3) To combine the task l being scheduled with the unscheduled task of the destination entity j. If there is any task of entity j which is suitable for scheduling, and the destination entity of this new task is just the provider of the vehicle.

Step 3 is to calculate the target value (the total operational cost), to convert the target value to the individual fitness, and to produce twice the population.

Step 4 is to assign the fitness of individuals with the weighted sum method and the rank-based fitness assignment method.

3.4 Fitness Assignment

The principle of weighted sum method[14] is to transform multiple objectives into single objective through the weighted sum of objectives. Once the weights are determined, the searching direction is fixed. To search all optimal solutions, the searching directions are changed to cover the whole solution space. The changes rely on the weights. The weights consist of random numbers. This rank-based fitness assignment method[18] explicitly emphasizes non-dominated solutions while maintaining diversity in the non-dominated solutions; it ranks each individual solution according to the number of solutions in the current population by which it is dominated. All non-dominated individuals are thus assigned Rank 1, and any other individual is assigned a rank equal to the number of solutions that dominate solution plus one.

3.5 Adaptive GA Parameters Tuning

The tuning principle for the crossover and mutation probabilities (P_c and P_m) is to consider the changes in the average fitness of the population. For the minimization problem, the changes of the average fitness at generation t and t-1 are $\Delta Fit_{avg}(V:t)$ and $\Delta Fit_{avg}(V:t-1)$, and the changes of the P_c and P_m are ΔP_c and ΔP_m. These values can be considered to regulate the P_c and P_m as follows:

(1) The inputs of the crossover and mutation FLCs are $\lambda\Delta Fit_{avg}(V:t)$ and $\lambda\Delta Fit_{avg}(V:t-1)$, and the outputs are ΔP_c and ΔP_m. λ is a scaling factor that regulates the average fitness. (2) The membership functions of fuzzy input and output variables are shown in table 1 and table 2, which are the same fuzzy decision tables quoted from the study by KwanWoo Kim[12]. (3) To calculate the ΔP_c and ΔP_m, $\Delta P_c = \alpha Z(i,j)$, $\Delta P_m = \beta Z(i,j)$, where the $Z(i,j)$ is the corresponding values of the $\Delta Fit_{avg}(V:t)$ and $\Delta Fit_{avg}(V:t-1)$ for defuzzification. The $Z(i,j)$ is the same for the P_c and P_m. The α and β are the given values to regulate the increasing and decreasing range for the P_c and P_m. (4) To update the P_c and P_m by the following equations: $P_c(t) = P_c(t-1) + \Delta P_c$, $P_m(t) = P_m(t-1) + \Delta P_m$, where the $P_c(t)$ and $P_m(t)$ are the crossover and mutation probabilities at generation t. (5) To reset the P_c and P_m. If the best target value of population has not changed in N successive generations and the P_c and P_m equal 1 or 0, then $P_c(t) = P_c(0)$ and $P_m(t) = P_m(0)$. (6) In this paper,

Table 1. Input and output results of discretization

Inputs	Outputs	Inputs	Outputs
$x \le -0.7$	-4	$0.1 < x \le 0.3$	1
$-0.7 < x \le -0.5$	-3	$0.3 < x \le 0.5$	2
$-0.5 < x \le -0.3$	-2	$0.5 < x \le 0.7$	3
$-0.3 < x \le -0.1$	-1	$x > 0.7$	4
$-0.1 < x \le 0.1$	0		

Table 2. Defuzzification table for control action of $Z(i, j)$

$Z(i, j)$	i								
	−4	−3	−2	−1	0	1	2	3	4
−4	−4	−3	−3	−2	−2	−1	−1	0	0
−3	−3	−3	−2	−2	−1	−1	0	0	1
−2	−3	−2	−2	−1	−1	0	0	1	1
−1	−2	−2	−1	−1	0	0	1	1	2
j 0	−2	−1	−1	0	2	1	1	2	2
1	−1	−1	0	0	1	1	2	2	3
2	−1	0	0	1	1	2	2	3	3
3	0	0	1	1	2	2	3	3	4
4	0	1	1	2	2	3	3	4	4

$\lambda = 4$, $\alpha = 0.125$, $\beta = 0.125$, and N = 5. Where x is the input result of the $\Delta Fit_{avg}(V:t)$ and $\Delta Fit_{avg}(V:t-1)$.

4 Numerical Experiments

In order to verify the effectiveness and efficiency of the adaptive MO-HGA, three test problems with different amounts of transport entities and tasks are designed. The amounts of entities are 3, 5, and 10, and the amounts of tasks are 12, 40, and 120. For each test problem with the same size of tasks, the no-overlapping execution time (T1) and the overlapping execution time (T2) are set, which is to analyze how the time complexity of tasks to influence the effectiveness and efficiency of the algorithm. The transport tasks data of the first test problem, the transport resource, unit cost of transport resource, time to transport, loading and unloading time parameters are shown in Table 3 to Table 6. The GA parameters of these problems are shown in Table 7. The transport tasks data of the other two test problem is similar to the first one, except the amount of entities and the amount of tasks.

According to the problems with different sizes, the genetic operators, the fitness assignment methods, and the GA parameters, the four criteria are used to evaluate the effectiveness and efficiency of the algorithm. First one is the number of Pareto optimal solutions searched by the algorithm with different genetic operators and fitness assignment methods (PO1). The second one is the number of non-dominated solutions of the algorithm with different genetic operators and fitness assignment methods after the results are compared each other (PO2). The third one is the percent of non-dominated solutions (PO2 / PO1 (%)).The last one is the elapsed computation time (CPU Times). The proposed algorithm is developed in visual c++ language on PC with Pentium 1.8G CPU and 1G RAM.

The scheduling results of the adaptive MO-HGA with the first set of GA parameters, different genetic operators and fitness assignment methods for the three test problems are shown in Table 8. For the third test problem, the scheduling results of the algorithm with the first three set of GA parameters are shown in Table 9, and the results with the first and last two sets of GA parameters are shown in Table 10.

Table 3. Transport task data

No.	Original entity No.	Task No.	Terminal entity No.	T1		T2	
				Start time	Finish time	Start time	Finish time
1	1	1	2	1	11	1	11
2	1	2	2	2	12	7	17
3	1	3	3	1	11	4	14
4	1	4	3	2	12	9	19
5	2	1	1	1	11	1	11
6	2	2	1	2	12	7	17
7	2	3	3	1	11	4	14
8	2	4	3	2	12	9	19
9	3	1	1	1	11	1	11
10	3	2	1	2	12	7	17
11	3	3	2	1	11	4	14
12	3	4	2	2	12	9	19

Table 4. Transport resource parameters

Entity No.	Loading/unloading		Vehicle		Storage		Parking	
	Max	Unit cost	Max	Unit cost	Max	Unit cost	Max	Unit cost
1	4	8	4	8	4	8	4	8
2	3	10	3	10	3	10	3	10
3	4	8	4	8	4	8	4	8

Table 5. Transport time parameters

Entity No.	1	2	3
1	0	1	2
2	1	0	1
3	2	1	0

Table 6. Loading/unloading time parameters

Entity No.	Transport time
1	3
2	2
3	3

Table 7. GA parameters

No.	Population size	Initial P_c	Initial P_m	Max Generation	Tuning
1	100	0.8	0.6	1000	no
2	50	0.8	0.6	1000	no
3	200	0.8	0.6	1000	no
4	100	0.8	0.6	1000	yes
5	100	0.1	0.1	1000	yes

Firstly, for each test problem, Table 8 shows that the effectiveness and efficiency of the algorithm with the rank-based fitness assignment method is better than the weighted sum method, and there is not significant distinction for different genetic operators. For the same genetic operator and fitness assignment method, the elapsed computation time of T2 is longer than T1, and the scheduling effectiveness of T2 is worse than T1, which means that the difficulty and the time of search increase with

Table 8. Results of three test problems with the first set of GA parameters

Test Problem	Genetic Operator + Fitness Assignment Method	PO1		PO2		PO2/PO1(%)		CPU Times(s)	
		T1	T2	T1	T2	T1	T2	T1	T2
1	PMC + WS	23	25	11	11	48	44	30	27
1	PBC + WS	31	36	9	12	29	33	31	29
1	PMC + RBFA	33	39	11	12	33	31	36	34
1	PBC + RBFA	35	41	12	13	34	32	35	32
2	PMC + WS	21	23	9	11	43	48	101	97
2	PBC + WS	25	28	10	12	40	43	105	99
2	PMC + RBFA	30	32	10	11	33	34	113	109
2	PBC + RBFA	31	35	9	10	29	29	112	110
3	PMC + WS	19	20	7	9	37	45	215	210
3	PBC + WS	17	18	6	8	35	44	211	209
3	PMC + RBFA	19	19	6	7	32	37	227	219
3	PBC + RBFA	21	23	7	8	33	35	229	223

Table 9. Results of the third test problem with the first three sets of GA parameters

GA Parameters	Genetic Operator + Fitness Assignment Method	PO1		PO2		PO2/PO1(%)		CPU Times(s)	
		T1	T2	T1	T2	T1	T2	T1	T2
1	PMC + WS	19	20	7	9	37	45	215	210
1	PBC + WS	17	18	6	8	35	44	211	209
1	PMC + RBFA	19	19	6	7	32	37	227	219
1	PBC + RBFA	21	23	7	8	33	35	229	223
2	PMC + WS	17	19	7	8	41	42	111	107
2	PBC + WS	16	18	8	9	50	50	109	111
2	PMC + RBFA	18	19	7	8	39	42	125	117
2	PBC + RBFA	20	22	7	7	35	32	120	113
3	PMC + WS	21	21	8	9	38	43	375	357
3	PBC + WS	20	21	8	10	40	48	379	209
3	PMC + RBFA	23	25	9	7	39	28	397	389
3	PBC + RBFA	22	23	8	8	36	35	391	381

Table 10. Results of the third test problem with the first and last two sets of GA parameters

GA Parameters	Genetic Operator + Fitness Assignment Method	PO1		PO2		PO2/PO1(%)		CPU Times(s)	
		T1	T2	T1	T2	T1	T2	T1	T2
1	PMC + WS	19	20	7	9	37	45	215	210
1	PBC + WS	17	18	6	8	35	44	211	209
1	PMC + RBFA	19	19	6	7	32	37	227	219
1	PBC + RBFA	21	23	7	8	33	35	229	223
4	PMC + WS	19	21	8	10	42	48	217	210
4	PBC + WS	17	18	7	9	41	50	210	207
4	PMC + RBFA	19	20	7	8	37	40	228	218
4	PBC + RBFA	22	23	8	9	36	39	228	225
5	PMC + WS	20	20	8	10	40	50	216	213
5	PBC + WS	18	18	7	8	39	44	212	207
5	PMC + RBFA	20	19	7	9	35	37	229	221
5	PBC + RBFA	23	22	9	8	35	36	227	223

the increase of the time complexity of task execution. Furthermore, the difficulty and the time of search increase with the increase of the size of problems. Secondly, for the third test problem and the first three sets of GA parameters, Table 9 shows that the increase of the population size can improve the scheduling effectiveness, but the scheduling times increase at the same time. Lastly, for the third test problem and the first and last two sets of GA parameters, Table 10 shows that the effectiveness of the algorithm with adaptive

tuning the P_c and P_m is better than without tuning the P_c and P_m, but there is not significant distinction for the scheduling times. For different initial P_c and P_m, the effectiveness of the algorithm change. In other words, the effectiveness of the algorithm can be improved by setting proper initial P_c and P_m.

5 Conclusions

In this paper, an adaptive multi-objective hybrid genetic algorithm is proposed to solve the multi-objective resource-constrained transport task scheduling problem, which is to initialize the population and evaluate the individual by using the special serial scheduling method. The weighted sum method and the rank-based fitness assignment method are used for individual fitness assignment. An adaptive GA parameters tuning method based on the fuzzy logic controller is implemented to improve the performance of the MO-HGA. The computational experiments with different amounts of transport entities and tasks are presented, and the effectiveness and efficiency of the MO-HGA with different genetic operators and fitness assignment methods are compared. The results of numerical experiments show that the adaptive MO-HGA can schedule the tasks of multiple transport entities with multiple resource constraints and multiple optimal objectives effectively, and the adaptive GA parameters tuning method can improve the scheduling effectiveness of the MO-HGA further.

References

1. Fisher, M.: Vehicle Routing. In: Ball, M.O., Magnanti, T.L., Monma, C.L., Nemhauser, G.L. (eds.) Network Routing, pp. 1–33. North-Holland, Amsterdam (1995)
2. Desrosiers, J., Dumas, Y., Solomon, M.M., Soumis, F.: Time Constrained Routing and Scheduling. In: Ball, M.O., Magnanti, T.L., Monma, C.L., Nemhauser, G.L. (eds.) Network Routing, pp. 35–139. North-Holland, Amsterdam (1995)
3. Kim, C.W., Tanchoco, J.M.A., Koo, P.H.: AGV Dispatching Based on Workload Balancing. International Journal of Production Research 37, 4053–4066 (1999)
4. Ebben, M.J.R., Gademann, A.J.R.M., Van Harten, A.: Scheduling Vehicles in Automated Transportation Systems. OR Spektrum 24, 31–58 (2002)
5. Ebben, M.J.R., Van der Heijden, M.C.: Dynamic Transport Scheduling under Multiple Resource Constraints. European Journal of Operational Research 167, 320–335 (2005)
6. David, N., Michele, S., Biagio, T., Uzay, K.: Genetic Algorithms for Supply-chain Scheduling: A Case Study in the Distribution of Ready-mixed Concrete. European Journal of Operational Research 177, 2069–2099 (2007)
7. Jozefowska, J., Mika, M., Rozycki, R., Waligora, G., Weglarz, J.: Solving the Discrete-Continuous Project Scheduling Problem via Its Discretization. Math. Methods Operat. Res. 52, 489–499 (2000)
8. Yun, Y.S., Gen, M.: Advanced Scheduling Problem Using Constraint Programming Techniques in SCM Environment. Comput. Ind. Eng. 43, 213–229 (2002)
9. Kolisch, R.: Serial and Parallel Resource-constrained Project Scheduling Methods Revisited -Theory and Computation. European Journal of Operational Research 90, 320–333 (1996)

10. Peter, B., Andreas, D., Rolf, M.: Resource-constrained Project Scheduling: Notation, Classification, Models, and Methods. European Journal of Operational Research 112, 3–41 (1999)
11. Kim, K.W., Gen, M., Yamazaki, G.: Hybrid Genetic Algorithm with Fuzzy Logic for Resource-constrained Project Scheduling. Applied Soft Computing 2, 174–188 (2003)
12. Kim, K.W., Yun, Y.S., Yoon, J.M., Gen, M., Yamazaki, G.: Hybrid Genetic Algorithm with Adaptive Abilities for Resource-constrained Multiple Project Scheduling. Computers in Industry 56, 143–160 (2005)
13. Valls, V., Ballestín, F., Quintanilla, S.: A Hybrid Genetic Algorithm for the Resource-constrained Project Scheduling Problem. European Journal of Operational Research (in Press, 2007)
14. Jih-Chang Hsieh, P.C., Wang, C.Y.: Adaptive Multi-objective Genetic Algorithms for Scheduling of Drilling Operation in Printed Circuit Board Industry. Applied Soft Computing 7, 800–806 (2007)
15. Bingul, Z.: Adaptive Genetic Algorithms Applied to Dynamic Multiobjective Problems. Applied Soft Computing 7, 791–799 (2007)
16. San José-Revuelta, L.M.: A New Adaptive Genetic Algorithm for Fixed Channel Assignment. Information Sciences 177, 2655–2678 (2007)
17. Celano, G., Fichera, S., Grasso, V., La Commare, U., Perrone, G.: An Evolutionary Approach to Multi-objective Scheduling of Mixed Model Assembly Lines. Computers & Industrial Engineering 37, 69–73 (1999)
18. Lee, L.H., Lee, C.U., Tan, Y.P.: A Multi-objective Genetic Algorithm for Robust Flight Scheduling Using Simulation. European Journal of Operational Research 177, 1948–1968 (2007)

Evolutional Aspects of the Construction of Adaptive Knowledge Base

Istvan Elek

Eotvos Lorand University, Faculty of Computer Science
Pazmany Peter setany 1/A, 1117 Budapest, Hungary
elek@map.elte.hu
http://lazarus.elte.hu/~elek

Abstract. This paper introduces a theoretical approach of the construction of a self developing and adaptive artificial digital organism with huge remembrance and the ability of the interpretation of the surrounding world. The paper describes the self development of intelligence of digital organisms from small fragments of digital knowledge.

Keywords: digital evolution, knowledge graph.

1 Introduction

There is no right definition of intelligence. Instead of making useless definitions, it is more promising to set up some essential principles that adjust the process of collecting and interpreting data from the surrounding world. There are many research papers that describe machines which collect data from the surroundings and they have some kind of remembrance [1]. Their ability of interpretation of the environment is restricted. The limits of these constructions are obvious: their intelligence never becomes similar to that of mammals or octopuses.

Many papers [2,3,4] dealing with artificial organisms emphasized the importance of complexity. Not only life shows serious complexity. There are human made constructions as well, such as the topology of Internet, that have increasing complex structures with fractal properties [5,6,7]. This paper establishes some principles that are really simple but like bricks they can be combined into arbitrarily complex buildings. The purpose is to construct a self developing and adaptive artificial digital organism, a so called digital evolution machine (DEM) that collects and arranges, sometimes even restructures its database.

1.1 Evolution and Knowledge

Paleontology and geology serve many exciting examples of the one way evolution. The time flows in one direction, forward. Life always tries to adapt itself to the circumstances, mainly to the weather (temperature, seasons, climate). If the climate has changed, the adaptive organisms also change their right properties, skills and manners. If one million years later the climate changed back, the evolution did not remember the previous stage of the environment. The adaptivity

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 1053–1061, 2009.

also produces new properties that help to survive the changes. The evolution seems to be recursive. It means the regulation is a kind of recursion, like a feedback. The current stage is the basis of the next step. Evolution never gets back. There are no general physical laws in the background where the processes can be computed from.

If an organism did not recognize the enemy, the food and the environment, it had died soon or immediately. Consequently, organisms had to collect data from their surroundings (perception) and interpret them (understanding). The remembrance is probably one of the most important components of understanding the world. The unreversable time, i.e. the serial recording of events and data of the environment produces a huge database. It has to contain everything that happened to an organism in its lifetime. The key of surviving is the interpretation of the surrounding world and the action. The speed of the reaction is also important among animals.

1.2 The Early Evolution

Look at the following event that took place in the ancient ocean many million years ago. A very primitive organism was swimming in the water. The temperature of the sea became somewhat colder for him. He detected the low temperature and realized it was bad for him. He decided to swim toward warmer waters (Fig 1).

The question is whether this reaction involves intelligence or not. A quick look says, yes, it does. A thermostat does the same: detects, compares, makes a dicision and acts something, but it is not yet considered intelligent, because a kind of remembrance requires a huge knowledge-base and a fast graph algorithm.

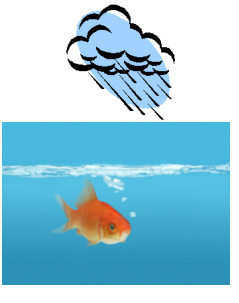


Fig. 1. The weather and climate were the most effective factors of the evolution of organisms. Every organism needs to interpret the measured and stored data of the environment if they wanted to survive. This ability required a huge database containing everything that ever happened to the organism. This is experience. The interpretation required a fast graph algorithm that could find the right answer to the current challenge.

1.3 The Construction of the Knowledge Base

Biologists say there is a kind of self organisation among protein molecules in vitro. Even if we assume it was the first step toward a real organism, it is

evident there is no huge knowledge database and complicated interpretation logic in it. Protein molecules seem to be inclined to form combinations. Their data collection logic also has to be simple. Consequently a knowledge database can be constructed from simple steps of data collection. The algorithm of the knowledge graph search has to be simple.

The hierarchy of nature sciences is a traditional knowledge database. This construction is hierarchical like a tree constructed by scientists. For example the physics consists of mechanics, electrodinamics, thermodinamics, quantum-mechanics, relativity theory and so on. How to construct a knowledge base for a simple organism that has no scientists? The following sections try to answer this question.

2 Principles of an Organic Knowledge-Base

Let us see some essential principles that regulate the data collection (perception) for an organism that is named DEMentity in the followings.

2.1 General Principles

1. Data collection process is a lifelong task that builds up the knowledge base.
2. Let knowledge elements be named atoms.
3. Atoms are situated in a tree structure that is a graph.
4. There is no lonely atom, every atom connects to another one at least.
5. Any atom can be a member of any group. A group of complex knowledge elements is named a context.
6. Identical operations are valid for atoms and contexts.
7. Let us allow contradictory atoms in the knowledge base.
8. Let us name a path the trajectory between two or more atoms.
9. If a search of the graph produces succes, the path becomes stronger (imprinting). If a search produces fail, the path becomes weaker (oblivion).

2.2 Quantification of the Knowledge and Atomic Distances

Let us construct the knowledge-graph (Fig 2). It consists of atoms and their connections. Atoms are the nodes and connections are the edges of knowledge-graph. Let us define a context that includes arbitrary atoms of the tree; consequently, the structure of the graph is not predefined. It depends on the atomic connections that depend on a time series, when events took place in time order one after the other. The general principles declared the equality of atoms and contexts. In other worlds a context can contain simple atoms or other contexts as well.

Let a_{ij} denote the i-th atom in the j-th context which contains N atoms. Let the quantity of the knowledge of a context (k^j) be the sum of the quantity of the knowledge of every atom in it:

$$k^j = \sum_{i=1}^{N} a_i^j \tag{1}$$

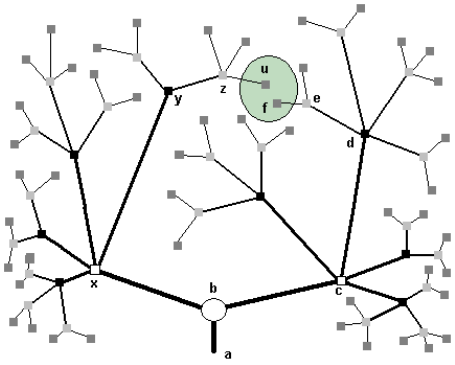


Fig. 2. The Knowledge-graph

The knowledge-base has two basic functions: receiving a question and answering. The key of the problem is to find the path from the questioning node to the answering node in the knowledge-base.

Let l_{ij} denote the strength of the connection between a_i and a_j. Let $l_{ij} = l_{ij}+1$ if the tour produces good result and $l_{ij} = l_{ij} - 1$ if the result is bad. This logic makes good paths stronger and bad ones weaker.

Look at the u and f nodes in the knowledge-graph (Fig 2). What is the right distance definition for them? In the case of Euclidean distance, u and f nodes are near. Since there is no direct connection between them, a better distance definition has to depend on the length of the path along branches.

Let a_i and a_j be two nodes of the knowledge-graph where the path includes $m = j - i$ atoms between them. Let d_{ij} be the distance of these two nodes, let the strength of their connection be denoted by l_{ij}, which is the reciprocal of the sum of the strength of the connections between a_i and a_j. The stronger the connection between two nodes, the closer they are.

The goal of the knowledge-graph is to answer a question arrising from the circumstance. How to find the rigth path from the questioning node to the answering one? The fastest is the right path, probably. This logic produces very fast reaction in well known problems and may result fail in unknown cases. The fail means unsuccessful escape, capture or something important for the organism. If it survived the situation, i.e. the reaction was successful, and the path that produced the success became stronger. If it did not survive the situation or the result of the action was failed, i.e. the result of the action was unsuccessful, the organism was knocked out or a path in the knowledge-graph became weaker.

2.3 Non Hierarchical Knowledge-Base

Let $\mathbf{K}$ denote the knowledge-base which consists of n atoms a_i, $a_j \in \mathbf{K}$. Let us name it the knowledge-matrix.

$$\mathbf{K} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \tag{2}$$

Some of the atoms are in touch with other atoms in **K**. Let us describe the links of a_i and a_j atoms with l_{ij}, where

$$l_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \text{ and no link between them} \\ u - v & \text{else where } u \text{ succeeful and } v \text{ unsuccessful} \end{cases}$$

Let us organize the atomic links into a matrix form, and name it a link matrix and denote it by **L**. The elements of the link matrix are l_{ij} that describe the link of the atomic pointpairs. **L** is diagonal ($l_{ii} = 1$) and describes the relationships of the atoms in the knowledge-matrix:

$$\mathbf{L} = \begin{pmatrix} 1 & l_{12} & \dots & l_{1n} \\ l_{21} & 1 & \dots & l_{2n} \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \dots & 1 \end{pmatrix} \tag{3}$$

2.4 The Properties of the K and L Matrices

1. Every atom has one link at least.
2. Let **C** be a context. $\mathbf{C} \subseteq \mathbf{K}$, i.e. **C** consits of any atoms of **K**.
3. Any atoms can be the member of any context.
4. Any contexts can be a member of any contexts. In this case, the knowledge-matrix (**K**) is a hyper matrix where matrix elements can be matrices.
5. In summary an atom can be a
   (a) Simple atom that is really elementary and belongs to one context.
   (b) Multi-member atom that is also elementary but belongs to more than one context.
   (c) Aggregated atom that is a kind of simple atom, but its value is a representation of a context. In other words, its value is a determinant or a spur of the knowledge-matrix of a certain context.
   (d) A complex atom that is a context that includes any kinds of atoms above (Fig 3).

2.5 The Evolutional Snippets of the Knowledge-Base

1. The knowledge-base is a continuously increasing database which stores everything that happened to it. This is a one way process. The knowledge-base is different from entity to entity.

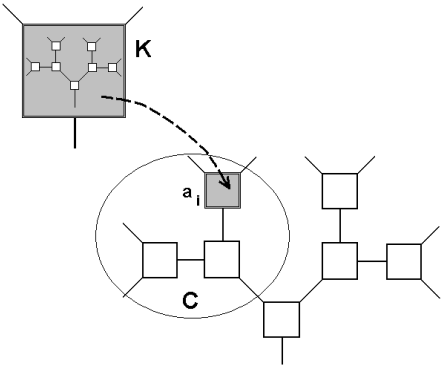


Fig. 3. A context is a complex atom in the knowledge-base. Any arbitrarily complex knowledge-base can be constructed based on this logic. Since the knowledge-base is a continuously increasing dataset, links can be arbitrarily complex too. In this case, the context **K** is a complex atom (a_i) of the context **C**.

2. There is no changing or erasing function in the knowledge-base.
3. Let the data collection be extensive if a DEM-entity perceives the circumstance and stores data. There are some important consequences of the extensive mode:
   (a) Every individual knowledge-base is different. It depends on the career of a certain DEM-entity. There are as many DEM-entities as many kinds of knowledge-base exist.
   (b) If some changes happen that certainly produces the same circumstance in the past, the previously recorded atoms has not been changed, simply a new atom has appended the end of the knowledge-base.
4. Let us have another mode that was named intensive, when there is no perception. It is a ,,meditative" stage when the knowledge-base acquires the data came from the extensive stage. The result of the data acquisition in intensive mode may produce new contexts, faster path, more reliable work. This stage is extremely important in learning the circumstance.
5. The feedback is a process when a DEM-entity is informed about the result of its reaction. The result of this process is success or fail. As mentioned previously, the success/fail makes stronger/weaker a certain path in the knowledge-graph of the knowledge-base. The DEM-entity's knowledge-base becomes much stronger/ weaker if the feedback such as rewarding or punishment comes from an external intelligent entity, because its knowledge-base can be considered as an included context.
6. In the history of the Earth there was never only one organism. There were always ensembles. Ensembles make organisms competitive. Competition results in different skills i.e. different knowledge-bases.
7. Different circumstances cause different experiences for the organisms.

2.6 Searching in the Knowledge-Base

It is an elementary function to find the right answer quickly if a question appears. The knowledge-graph contains the data that are the source of answering. How to find the right answer?

1. Regarding the early era of the evolution where the combinations of protein molecules were the most complex structure in the world, we can not suppose the existence of fascinating graph algorithms in this structure. Only simple algorithm can be like the preference of the fastest path, the most stronger path, or something like that. It is named preconception in the everyday life. The preconception accelerates getting answer.
2. So there is no more complicated algorithm that answer a certain question. While the knowledge-base is increasing, the graph structure becomes more and more complicated. The same simple algorithms have to serve the right answer.
3. Questions and answers have to be stored together in the knowledge-base. Questions may identify the start up context where the answer can be found. This is also a kind of preconception.
4. The judgement of an answer of the organism (or DEM-entity) comes from the environment. If the answer is right, the organism confirms the right path in the knowledge-graph, if not the path has weakened. The result is a more developed knowledge-base.

3 Some Constructional Aspects of the Knowledge-Base

The evolution is a process in time and in space dimensions. Regarding the results of paleontology it is known that evolution is recursive, i.e. an evolution step depends on the previous step only and it is a one way process, it can not be turned back. This is known as Dollo's law in paleontology [8]. This law was first stated by Dollo in this way: ,,An organism is unable to return, even partially, to a previous stage already realized in the ranks of its ancestors". According to this hypothesis, a structure or organ that has been lost or discarded through the process of evolution will not reappear in that line of organisms.

The development means not only physical but mental changes as well in an organism. Mental means intelligence in this context. The adaptivity comes from its knowledgebase. The quality of the knowledge-base has to influence the physical properties also. Consequently the physical and mental evolution work collateraly.

3.1 Synthetic Worlds, Artificial Environment

If we are willing to create experienced DEM-entities there is no wizard unfortunately. There is no a recipe to install them from little pieces. How to construct a DEM-entity? Probably, it is impossible to construct only one of it. First we should create an ensemble from many initial DEM-entites and leave them to live in their own world.

We have two access points to this problem. The first one is to construct an artificial circumstance where DEM-entities live in. The second one is to construct many initial DEM entities with simple perception and interpetation functions. Let us look at some of the details:

1. The artificial world (AW) has some essential properties that define the frame of this world.
   (a) Let AW be huge where circumstances have spatial dependencies. Regarding the size of this world, environmental parameters are obviously different. If we leave DEM-entities alone in this world they will have different experiences and different knowledge-bases because of climatic differences.
   (b) If the AW is huge enough, the survival strategies will be different. One of the DEM-entities escapes from the unfriendly circumstances but others try to adapt. Different strategies result different knowledge-bases.
2. If there are many DEM-entities on the same territories, what is the consequence?
   (a) There are many DEM-entities who try to get better strategy in order to be more successful than others. Someone gets advantages but someone gets disadvantages since it fails to answer a certain question.
   (b) Regarding the different DEM-entities and unique knowledge-bases, many different strategies can coexist in the AW. Consequently, many different strategies can be successful at the same time. Someone preferes the escape, but someone the competition.
3. Many DEM-entities will have many different knowledge-bases.

3.2 The Digital Evolution Machine

The question is how to construct the prototype of a DEMentity? Before the construction of the prototype, let us create the artificial world that will be the space for DEM-entities. If AW has been created already many DEM-entities should be available to start up in it. Properties and abilities of DEM-entities were introduced previously, so the task is to make their software representation. Regarding the quantity of DEMs and the huge sized world with different spatial properties may result many formed DEMs, and these entities have different knowledge-bases. An intelligent DEM-entity can exist in ensemble only. There is no lonely intelligency because it is a collective product that is realized in some successful entities? knowledgebases.

3.3 Pending Questions

There are many pending questions unfortunately. Let us see some of them:

1. The existence of the extensive stage is obvious, but not the intensive one. The existence of the intensive stage does not result from the principles mentioned above.

2. The next unresolved problem is related to the intensive stage as well. The instensive stage is a process when the perceived data coming from the extensive stage have been evaluated and restructured into existing or new contexts. Why and how do the evaluation produce a new context? Which principle made the knowledge-base restructured?
3. It would be very useful if there were a lifetime for a DEM-entity. DEMs collect data every day, every minute and the knowledge-base is building up. If the starting date of data collection is too far from now the knowledge-base has many unimportant and useless information even contexts. For example is there any value for you if you know how to conquer a knight in tournament? No, it is absolutely indifferent to you because you don not have any sword, horse and there is no knights ever. Unfortunately the necessity of the limited lifetime does not come from the principles.

4 Conclusions

This paper tried to introduce some principles and suppositions of how to construct adaptive and self-developing digital evolution machines having their own knowledge-bases that help them to understand and survive the challenge of the environment. This is only a theoretical approach until we do not develop the huge artificial world and millions of DEM-entities. The monitoring of DEMs' life will show the way of the evolution of intelligence.

References

1. Russel, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice-Hall, Englewood Cliffs (2002)
2. Adami, C.: Ab Initio of Ecosystems with Artificial Life, arXiv:physics/0209081 v1 22 (2002)
3. Adami, C.: Sequence Complexity in Darwinian Evolution, Complexity, vol. 8(2). Wiley Periodicals, Chichester (2003)
4. Adami, C., Ofria, C., Collier, T.: Evolution of Biological Complexity, arXiv:physics/0005074v1 26 (2000)
5. Barabasi, A.L., Albert, R., Jeong, H.: Mean-field Theory for Scale-Free Random Networks. Elsevier Preprint 5 (submitted to, 2002)
6. Albert, R., Barabasi, A.L.: Statistical Mechanics of Complex Networks. Reviews of Modern Physics 74 (2002)
7. Newman, M., Barabasi, A.L., Watts, D.J.: The Structure and Dynamics of Networks. Princeton University Press, Princeton (2006)
8. Louis Dollo, http://en.wikipedia.org/wiki/Dollo's_law

Evacuation Route Planning Algorithm: Longer Route Preferential

Maimai Zeng and Cheng Wang

Hubei Key Laboratory of Digital Valley Science and Technology,
Huazhong University of Science & Technology, Wuhan 430074, China
zengmaimai@gmail.com

Abstract. Given a transportation network with capacity constraints, the initial occupancies and the destination nodes, evacuation route planning generates a set of evacuation routes and a schedule for the movement of people and vehicles along these routes, such that the evacuation is completed in the shortest possible time. This is a critical step in disaster emergency management and homeland defense preparation. In order to avoid the large storage and calculation costs brought by the time-expended-graph for transforming dynamic network flow problem to static network flow problem, the heuristic algorithm of the Capacity Constrained Route Planner (CCRP) is researched. One defect of the original CCRP algorithm is studied carefully and the new algorithm with longer route preferential is proposed to improve it. The result of the experiment shows the feasibility of the algorithm.

Keywords: Evacuation planning, Routing and scheduling, Transportation network, Capacity constraint.

1 Introduction

Given a transportation network, a population distribution that vulnerable, and a set of destinations, evacuation route planning identifies routes to minimize the time to evacuate the people in the vulnerable area. Evacuation route planning is a vital component of efforts by civil authorities to prepare for both natural and man-made disasters, which include: hurricanes, terrorist acts, risky chemicals leakage, nuclear leakage, war, etc [1-4].

Various approaches to solve the evacuation route planning problem have been proposed in domains such as transportation science, mathematics and computer science. The Evacuation Route Planning researches mainly fall into three categories: (1) Linear Programming methods that generate optimal evacuation plans which minimize the total evacuation time [5-8], (2) Simulation methods that models traffic flow at single vehicle level [9-10], and (3) Heuristic methods do not always generate optimal evacuation routes but able to reduce the computational cost of the process dramatically, such as the Capacity Constrained Route Planner algorithm [11].

The network flow algorithms are used in the linear programming based approaches to evaluate the routes. The transportation network is transformed into a time-expanded network [11] by duplicating the original evacuation network G for each

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 1062 – 1071, 2009.

discrete time unit $t = 0,1,...,T$. The multiple copies are connected by an edge cost (e.g., travel time). Then, the evacuation problem can be defined as a minimum cost network flow problem [12-13] on the time-expanded network G_T . Finally, the expanded network G_T can be fed to the minimum cost network flow solvers, such as NETFLO, RelaxIV, CS2 [14-15], to calculate the optimal solution. Although these evacuation planning algorithms can generate optimal plans, they are expensive with respect to memory and take a long time to solve problems of the sizes encountered in urban evacuation scenarios. For example, if we solve a problem with 1,000 nodes and a maximum 300 minute evacuation time, the time expanded network will consist of 300,000 nodes. Otherwise, this method requires a prior knowledge of an upper bound on the evacuation time T to generate the time-expanded network, which might be hard to estimate precisely. An underestimated bound T would result in a failure to reach a solution, whereas, an overestimated value for T would result in an over-expanded network, leading to unnecessary storage and run time. In practice, LP based approaches have been mainly used in scenarios that involve small sized networks such as, in the building evacuations.

Simulation methods assume that the behavior of individual drivers is under the influence of environment like vehicles in their proximity, usually modeling the interaction between cars with car-following models. Theodoulou and Wolshon [16] used CORSIM microscopic traffic simulation to model the freeway evacuation around New Orleans. With the help of a micro scale traffic simulator, they were able to suggest alternative evacuation routes in a detailed manner. However, these simulation models are often accompanied with labor intensive network coding and significant running time, making it difficult to compare alternative configurations and validate the validity. Evacuation route planning with other microscopic traffic simulation (e.g., MITSIMLab [17]) have shown similar limitations. Thus, they are inappropriate for large evacuation scenarios.

Though the methods in the third category based on heuristics do not always generate optimal evacuation routes, they have been able to reduce the computational cost of the process dramatically and to overcome the limitations of LP methods as well as simulation methods. A well-known approach that falls in this category is the Capacity Constrained Route Planner (CCRP) [11]. This method makes use of shortest path algorithms and extends them by incorporating capacity constraints. It models capacity as a time series to account for the time dependent nature of the networks. It uses only the original evacuation network instead of the time-expanded network used by the LP based approach and thus requires less memory. The performance evaluation of the three experiments has proved the advantages compared with one LP method of the NETFLO [11]. The main researchers of the heuristic algorithm for evacuation route planning are in the Department of Computer Science in the University of Minnesota. In 2003, Qingsong Lu first mentioned the heuristic algorithm for calculating the evacuation route planning, and proposed the Single-Route Capacity Constrained Planner (SRCCP) and the Multiple-Route Capacity Constrained Planner (MRCCP) algorithm [18]. In 2005, Qingsong Lu improved the former algorithm and proposed the Capacity Constrained Route Planner (CCRP) algorithm [11]. In 2007, Sangho Kim proposed some policies to improve the CCRP algorithm [19].

In the CCRP algorithm, capacity is modeled as a time series because available capacity of each node and edge may vary during the evacuation. And a generalized shortest path search algorithm is used to account for route capacity constraints. This algorithm can divide evacuees from each source into multiple groups and assign a route and time schedule to each group of evacuees based on an order that is prioritized by each group's destination arrival time. It then reserves route capacities for each group subject to the route capacity constraints. The quickest route available for one group is recalculated in each iteration based on the available capacity of the network. Performance evaluation on various network configurations shows that the CCRP algorithm produces high quality solutions, and significantly reduces the computational cost compared to linear programming approach. CCRP is also scalable to the number of evacuees and the size of the network [11].

One defect of the original CCRP algorithm is that the evacuees with the quicker path usually have earlier stating time and occupied the time and capacity resource prior. Then the remainder evacuees with longer paths to the destinations have to go through longer route and start later. As a result, the total evacuation time may delayed by the evacuees with later started longer route. The core idea of the algorithm in this paper is distributing the capacity on the routes to the evacuees which should go through the longer route preferentially. The method to implement which is choosing the longest route in the shortest paths of all the source-destination nodes pairs in each iteration, instead of choosing the shortest route every time in the original CCRP algorithm.

The rest of the paper is organized as follows. In section 2, the evacuation problem is described detailed and a sample is given. Section 3 explains the new algorithm. In section 4, an experiment is devised to checkout the performance the new algorithm. Section 5 gives the conclusion and the future work.

2 Problem Description

In transportation science, there are various ways (e.g., microscopic, mid-scope and macroscopic) to interpret and formulate an evacuation situation. In this paper, the macroscopic model using mathematical graphs (i.e., flow network) is used to describe the evacuation situation due to its increased public attention, improved techniques and computational capacity [20]. A precise formulation of the research problem on evacuation route planning is as follows.

Given: A transportation network with
(1) integer capacity constraints on nodes and edges,
(2) integer travel time on edges,
(3) number of evacuees and their initial locations,
(4) locations of evacuation destinations.

To Find: An evacuation plan consisting of a set of origin destination routes and a scheduling of evacuees on each route.

Object: Minimize the evacuation time.

Constraint: (1) The scheduling of evacuees on each route should observe the capacity constraints.
(2) Edge travel time preserves First-In First-Out.
(3) Limited amount of computer memory and the calculate ability.

The problem definition described above is illustrated in Fig.1[11]. Each node is shown by a circle, the source nodes are with two attributes: maximum node capacity and initial node occupancy, and the other nodes have just the attribute of maximum node capacity. For example, at node N1, the maximum capacity is 50, which indicates that this node can hold at most 50 evacuees at any time instant. The initial occupancy is shown to be 10, which means there are 10 evacuees at this node when the evacuation starts and this is a source node. If the node capacity is not meaningful (e.g., node as a city or an intersection), it can be set to infinity. Each edge, shown as an arrow, represents a link between two nodes. Each edge also has two attributes: maximum edge capacity and travel time. For example, at edge N1-N3, the maximum edge capacity is 7, which means at each time point, at most 7 evacuees can start to travel from node N1 to N3 through this link. The travel time of this edge is 1, which means it takes 1 time unit(s) to travel from node N1 to N3. The time unit is defined according to the requirement of precision in evacuation time (e.g., minute or hour). The example in Fig.1 has 10 evacuees at node N1, 5 at node N2, and 15 at node N8. The goal is to compute an evacuation plan that evacuates the 30 evacuees to the two destinations (thick lined, nodes N13 and N14) using the least amount of time.

Table 1 shows an example evacuation plan for the evacuation network in Fig.1. Each row in the table describes the schedule of a group of evacuees moving together with a series of node IDs and arrival time of each node. Take source node N8 for example; initially there are 15 evacuees at N8. They are divided into 3 groups: 6, 6 and 3 people. The first group starts from node N8 at time 0 and moves to node N10,

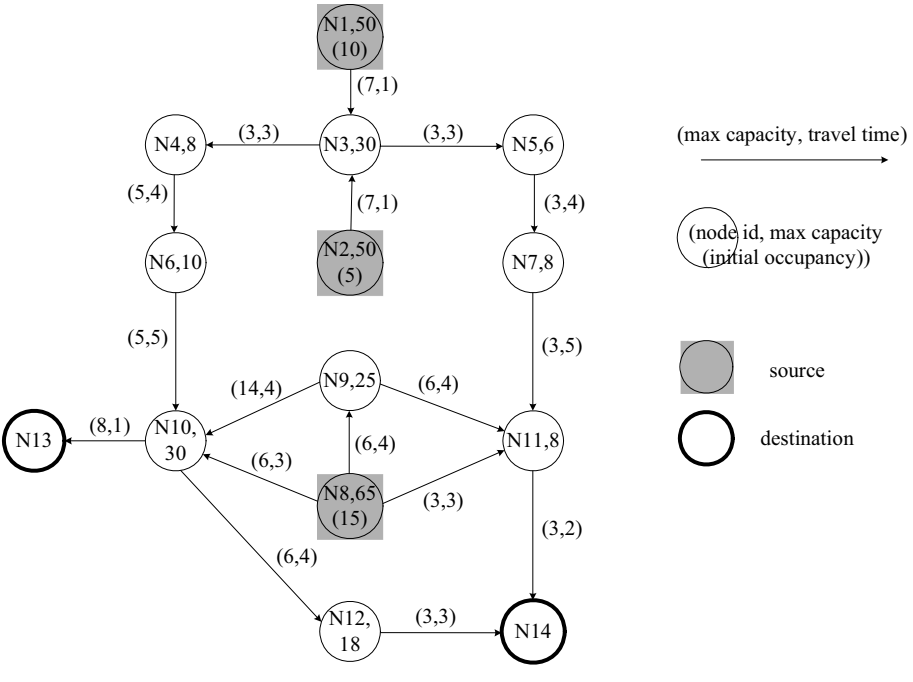


Fig. 1. An example of evacuation problem modeled using graph

Table 1. A feasible rout planning solution of the evacuation problem example

Source	# of Evacuees	Route with Schedule nodeID(arrivalTime)-nodeID(arrivalTim)-…	Destination arrival time
N8	6	N8(0)-N10(3)-N13(4)	4
N8	6	N8(1)-N10(4)-N13(5)	5
N8	3	N8(0)-N11(3)-N14(5)	5
N1	3	N1(0)-N3(1)-N4(4)-N6(8)-N10(13)-N13(14)	14
N1	3	N1(0)-N3(2)-N4(5)-N6(9)-N10(14)-N13(15)	15
N1	1	N1(0)-N3(1)-N5(4)-N7(8)-N11(13)-N14(15)	15
N2	2	N2(0)-N3(1)-N5(4)-N7(8)-N11(13)-N14(15)	15
N2	3	N2(0)-N3(3)-N4(6)-N6(10)-N10(15)-N13(16)	16
N1	3	N1(1)-N3(2)-N5(5)-N7(9)-N11(14)-N14(16)	16

then moves from node N10 at time 3 to node N13, and reaches destination N13 at time 4. The second group follows the same route of the first group, but has a different schedule due to the capacity constraints of this route. The second group moves from N8 at time 1 to N10, then moves from N10 at time 4 to N13, and reaches destination N13 at time 5. The third group takes a different route. It starts from N8 at time 0 and moves to N11, then moves from N11 at time 3 to N14, and reaches destination N14 at time 5. The procedure is similar for other groups of evacuees from source node N1 and N2. The whole evacuation egress time is 16 time units since the last groups of people reach their destination at time 16. This evacuation plan is an optimal plan for the evacuation scenario shown in Fig.1 [11].

3 The Algorithm for the Route Planning

The CCRP algorithm that Qingsong Lu proposed is a heuristic algorithm of greedy. In each iteration it choose the shortest path, that means to evacuate the evacuees in the source node which has the earliest arrive time to a destination first, then arrange time and space resources to the evacuees with the longer arrive time to a destination. The thought of which is effective. But when we consider the final object again, it can be found that the total evacuation time is always lying on the last evacuee that arrive the destination. If we consider the evacuation time as the route passing time adding the start time, one defect of the original CCRP algorithm would lie in: the evacuees with the shorter path usually have earlier stating time and occupied the time and capacity resource prior. Then the remainder evacuees with longer paths to the destinations have to go through longer route and start later. As a result, the total evacuation time may delayed by the evacuees with longer routes.

A thought to change this status is to distribute the capacity and time resources to the evacuees that should go through the longer route preferentially. Every time, we let the evacuees longer from the destinations start first, to shorten the total evacuation time of them. The evacuees closer with the destinations would start later, the arrival time would longer than in the original algorithm. But the total evacuation time

depends on the evacuees through longer paths, so it can be shortened in the new algorithm.

The core idea of the algorithm in this paper is distributing the capacity on the routes to the evacuees that should go through the longer route first. The method is choosing the longest route in the shortest paths of all the source-destination nodes pairs in each iteration, instead of choosing the shortest route every time in the original CCRP algorithm. The overall algorithm structure is described below:

Input:

(1) $G(N,E)$: a graph G with a set of nodes N and a set of edges E;

Each node $n \in N$ has tow properties:

$Maximum\_Node\_Capacity(n)$: non-negative integer

$Initial\_Node\_Occupancy(n)$: non-negative integer

Each edge $e \in E$ has two properties:

$Maximum\_Edge\_Capacity(e)$: non-negative integer

$Travel\_time(e)$: non-negative integer

(2) S : set of source nodes, $S \subseteq N$;

(3) D : set of destination nodes, $D \subseteq N$.

Output:

Evacuation plan: Routes with schedules of evacuees on each route

Pre-process network: add virtual source node s_0 to network, in the s_0 node, the

$Maximum\_Node\_Capacity = TotalEvacuatorNum$;

link s_0 to each source nodes with an edge which

$Maximum\_Edge\_Capacity() = EvacuatorNumInSourceNode$

and $Travel\_time() = 0$; (0)

Main algorithm: while the super source node s_0 has evacuee do {

Find route $R < n_0, n_1, ..., n_k >$ with time schedule $< t_0, t_1, ..., t_k >$ using one generalized shortest path search from virtual source s_0 to all destinations (where $s \in S$, $d \in D$, $n_0 = s$, $n_k = d$), such that R has the latest destination arrival time among routes between all (s,d) pairs, and

$Available\_Edge\_Capacity(e_{n_i n_{i+1}}, t_i) > 0$

and $Available\_Node\_Capacity(n_{i+1}, t_i + Travel\_time(e_{n_i n_{i+1}})) > 0$,

$\forall i \in \{0,1,...,k-1\}$; (1)

$flow$ = Min (number of evacuees still at the node s_0 ,

$Available\_Edge\_Capacity(e_{n_i n_{i+1}}, t_i) > 0$,

$Available\_Node\_Capacity(n_{i+1}, t_i + Travel\_time(e_{n_i n_{i+1}})) > 0$,
$\forall i \in \{0,1,...,k-1\}$); (2)

$for(i = 0; i < k; i++)$ {
$Available\_Edge\_Capacity(e_{n_i n_{i+1}}, t_i)$ reduced by $flow$;
$Available\_Node\_Capacity(n_{i+1}, t_i + Travel\_time(e_{n_i n_{i+1}}))$ reduced by flow; } (3)
}

In each iteration of while loop, the algorithm first searches for the route with the latest arrival time in the shortest paths from any source to any destination, taking available edge capacity into consideration. That is the main different from the algorithm of the CCRP. The next step finds the minimum flow along the path found which is equivalent to the size of a group of evacuees who travel through the path. Then, the last step reserves the minimum flow. It is a task of updating available capacity.

4 Experiment

4.1 The Input Data

The graph data for test is generated by the random graph generator: NETGEN [21]. The input parameters of the NETGEN are listed in Table 2.

Table 2. The input parameter of the NETGEN generator for the experiment data. The '#' means the number, the 'trans' is the shortening of the 'transshipment' and the 'caped' is the shortening of the 'capacitated'.

Name	Net8_1	Net8_2	Net8_3	Net8_4	Net8_5	Net10_1
seed	101027544	101027544	101027544	101027544	101027544	120429482
problem	1	1	1	1	1	1
# of nodes	256	256	256	256	256	1024
# of sources	26	26	26	26	26	102
# of sinks	26	26	26	26	26	102
# of arcs	768	768	768	768	768	3072
min arc cost	1	1	1	1	1	1
max arc cost	50	50	50	50	50	50
total supply	7500	10000	12500	15000	17500	10000
trans. sources	0	0	0	0	0	0
trans. sinks	0	0	0	0	0	0
max cost arc %	0	0	0	0	0	0
caped arc %	100	100	100	100	100	100
min arc cap	1	1	1	1	1	1
max arc cap	200	200	200	200	200	200

To represent the typical traffic network features, the edges are 3 times of the nodes in every network listed in table 2 [11]. The sizes of the networks are the same in the first five networks, but the evacuee number increase gradually. The nodes of the sixth network increased to 2^{10} from 2^8 that the former networks have. The capacities of the nodes are set random and are different between every two networks.

NETGEN doesn't generate random capacities for the intermediate nodes, so in the experiment, which are generated use the pseudo random number generating function of $rand()$ in the C++. The seed is set to the nodeID, and the upper limit of the capacity is set to $rand()\%(\alpha * totalSupply / sourcNodesNum) + \beta$, that means, the minimum capacity is β and the maximum capacity is $rand()\%(\alpha * totalSupply / sourcNodesNum)$, in the experiment, the parameters are set to: $\alpha = 3, \beta = 20$, which can be reset according to the actual situation.

4.2 The HarDware for the Experiment

The experiment is calculated on a compatible desktop computer. The main resource needed for the calculation is the CPU and memory, so just these two items listed in the Table 3.

Table 3. The hardware configure of the computer for experiment

CPU	Intel Core 2 Duo E7200 @ 2.53GHz, L1cache :2*32KB, L2 cache :3MB
memory	DDR2 2G, PC2-6400 (400 MHz)

4.3 The Experiment Result

The result include the evacuation route planning like the lists is the table1, the total evacuating time, the iteration times and the execution time. We just list the evacuation time, the planning number (equal to the iteration times) and the execution time in the Table 4.

Table 4. The result data of the experiment

Name	Net8_1	Net8_2	Net8_3	Net8_4	Net8_5	Net10_1
evacuation time	1655	1706	2517	2833	2760	1015
# of plan	1124	1487	1987	2291	2900	1216
execute time (s)	192	301	738	902	1553	308

To verify the validity of the algorithm in this paper, the result of the comparison of the evacuation time with the original CCRP algorithm is listed in the Table 5

Table 5. The comparison of the evacuation time with the original CCRP algorithm

Name	Net8_1	Net8_2	Net8_3	Net8_4	Net8_5	Net10_1
CCRP	1601	2160	3153	2632	2439	1126
Longer Route Preferential	1655	1706	2517	2833	2760	1015

4.4 The Analysis of the Result

Compare with the original CCRP algorithm, the evacuation time of the Net8_2, Net8_3 and Net10_1 network has shortened distinctly, which proved the validity of the new algorithm. Meanwhile, it can be seen that the new algorithm doesn't suit for all the cases; for example, the evacuation time is degenerative in the network Net8_1, Net8_4 and Net8_5. Which is in anticipation, no method can be the most suitable for all the situations because of the complexity of the network characteristics and the time process. And specifically for the new algorithm, in which exist risk that giving up some shorter routes but adopt some longer path. In practice, the two algorithms can be combined to take the better solution.

5 Conclusion and the Future Work

To solve the evacuation route planning problem, the heuristic algorithm of CCRP is researched, and found that the method of choosing the shortest route in every iteration may cause a defect that prolonging the total evacuation time. To improve it, a new thought of the longer route preferential method is put forward and the new algorithm is proposed to try to reduce the total evacuation time. To verify the feasibility of the algorithm, an experiment is designed and both of the CCRP algorithm and the longer route preferential algorithm calculated the six instances, and the result shows the validity of the new algorithm.

In the future, more work should be done to improve the performance of the evacuation route planning algorithms. And some standard data for experiment could be researched to compare the performance of the different algorithms and check the distance of them to the optimal evacuation time.

References

1. ESRI: GIS for Homeland Security. An ESRI white paper (2001)
2. The Homeland Security Council.: Planning Scenarios, Executive Summaries, Created for Use in National, Federal, State, and Local Homeland Security Preparedness Activities (2004)
3. Brown, S.: Building America's Anti-Terror Machine: How Infotech Can Combat Homeland Insecurity. J. Fortune, 99–104 (2002)
4. The Volpe National Transportation Systems Center: Improving Regional Transportation Planning for Catastrophic Events (FHWA). Volpe Center Highlights, 1–3 (2002)
5. Hamacher, H.W., Tjandra, A.: Mathematical modeling of evacuation problems: State of the Art. J. Pedestrian and Evacuation Dynamics, 227–266 (2001)

6. Chalmet, L., Franis, R., Saunders, P.: Network Model for Building Evacuation. J. Management Science. 28, 86–105 (1982)
7. Francis, R., Chalmet, L.: A Negative Exponential Solution To An Evacuation Problem. Research Report. National Bureau of Standards. Center for Fire Research (1984)
8. Hoppe, B., Tardos, E.: Polynomial Time Algorithms for Some Evacuation Problems. In: Proceedings for the 5th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 433–441 (1994)
9. Akiva, M.B., et al.: Development of a Deployable Real-Time Dynamic Traffic Assignment System: DynaMIT and DynaMIT-P User's Guide. Massachusetts Institute of Technology (2002)
10. Mahmassani, H., Sbayti, H., Zhou, X.: DYNASMART-P Version 1.0 User's Guide. Maryland Transportation Initiative, University of Maryland (2004)
11. Lu, Q., George, B., Shekhar, S.: Capacity Constrained Routing Algorithms for Evacuation Planning: A Summary of Results. In: Bauzer Medeiros, C., Egenhofer, M.J., Bertino, E. (eds.) SSTD 2005. LNCS, vol. 3633, pp. 291–307. Springer, Heidelberg (2005)
12. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows: Theory, Algorithms, and Applications. Prentice-Hall, Englewood Cliffs (1993)
13. Ford, L.R., Fulkerson, D.R.: Flows in Network. Princeton University Press, Princeton (1962)
14. Kennington, J., Helgason, R.: Algorithm for Network Programming. John Wiley & Sons, Inc., New York (1980)
15. Frangioni, A., Manca, A.: A Computational Study of Cost Reoptimization for Min-Cost Flow Problems. J. Informs journal on computing, 61–70 (2006)
16. Theodoulou, G., Wolshon, B.: Alternative Methods to Increase the Effectiveness of Freeway Contraflow Evacuation. J. Technical Report Transportation Research Record. The Journal of Transportation Research Board 1865, 48–56 (2004)
17. Jha, M., Moore, K., Pashaie, B.: Emergency Evacuation Planning with Microscopic Traffic Simulation. J. The Journal of Transportation Research Board 1886, 40–48 (2007)
18. Lu, Q., Huang, Y., Shekhar, S.: Evacuation Planning: A Capacity Constrained Routing Approach. In: Chen, H., Miranda, R., Zeng, D.D., Demchak, C.C., Schroeder, J., Madhusudan, T. (eds.) ISI 2003. LNCS, vol. 2665, pp. 111–125. Springer, Heidelberg (2003)
19. Kim, S., George, B., Shekhar, S.: Evacuation Route Planning: Scalable Heuristics. In: Proceedings of the 15th International Symposium on Advances in Geographic Information Systems. ACM GIS (2007)
20. Hoogendoom, S., Bovy, P.: State of the Art of Vehicular Traffic Flow Modeling. J. Proceedings of the Institution of Mechanical Engineers, Part 1: Journal of Systems and Control Engineering 215, 283–303 (2001)
21. Klingman, D., Napier, A., Stutz, J.: NETGEN: A Program for Generating Large Scale Capacitated Assignment, Transportation, and Minimum Cost Flow Network Problems. J. Management Science 20, 814–821 (1974)

Using Neural Networks for the Foreign Investment Management Decision Support System

Sihai Guo[1,2], Shan Feng[1], Yong Zhao[1], and Kaibo Zhou[1]

[1] Institute of Systems Engineering, Huazhong University of Science and Technology, Wuhan 430074, China
[2] School of Automation, Wuhan University of Technology, Wuhan 430070, China

Abstract. How to effectively anticipate the correct future trends of the foreign investment are the most important job and critical issue for the economic development. With data warehouse and neural network, the development states of foreign investment are predicted in foreign investment management decision support system. The paper illustrates the architecture and implementation of decision support system oriented to foreign capital, moreover, the method which multidimensional data-mining based on neural network is applied in data warehouse is presented. The experimental result shows that the method can be used to analysis and decision of foreign investment information.

Keywords: Neural network, Data warehouse, Decision support system, Data mining, Foreign investment.

1 Introduction

With the continuous development of the global economic integration, it has a significant impact on China's economic development trend. There are various analysis demands of foreign investment, such as trend analysis, real-time evaluation, important factor analysis etc [1]. Those results of analysis will contribute to formulate China's foreign policy and deal with international economic issues [2]. Therefore, it is necessary to build a decision support system which uses intelligent technology to obtain the hidden knowledge of foreign investment.

The data of foreign investment are usually stored in database. The data has some characteristics, such as the rapid growth in data volume, the dynamic update of history data. It is hard to analyze the mass and wide reference date with the anciently state method. Therefore, foreign investment management decision support system (FIMDSS) is firstly built according to data warehouse technology. In this paper, the data warehouse architecture of FIMDSS is introduced, including the analysis topic, fact table, and dimension etc. On the basis of data warehouse, the neural network is used for multi-dimensional data mining of foreign investment data [3]. According to the results of data mining, the model of foreign investment development is established in order to achieve high-level information analysis and forecasting functions of foreign investment.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 1072–1079, 2009.

2 Data Warehouse Architecture of FIMDSS

FIMDSS will provide unified management platform by integrating familiar technologies, such as knowledge base, model base, and data warehouse methods. In FIMDSS, data warehouse is used for the relational data analysis, multidimensional analysis, data mining and online analytical processing. The model base can provide guidance for all analytical tools. The knowledge base is used to sum up history experience and discovery new knowledge [4]. The data warehouse is the core in FIMDSS. By building a hierarchical architecture of data warehouse, the decision support system can not only make up for the lack of information management system and multi-dimensional analysis, but also coordinate the foreign investment data, analytical models, decision methods and knowledge to make better decision results [5]. According to the decision demands, the architecture of FIMDSS can be composed by four main levels: data preparation, data process, rules options and result assessment. The details are shown in Fig. 1.

The data warehouse of FIMDSS has two main analytical subjects: performance analysis of foreign-funded enterprises and investment data analysis of country around the world. The former brings foreign-funded enterprises as the core objects. The analysis contents include total investment analysis, the registered capital analysis, the contracted foreign capital analysis, the actually utilized foreign capital analysis and actual investment analysis [6]. The related analysis dimensions include time, region, trade, enterprise type, etc. The latter centralizes the investment data of countries around the world. The analysis contents include investment and loan. The related analysis dimensions include

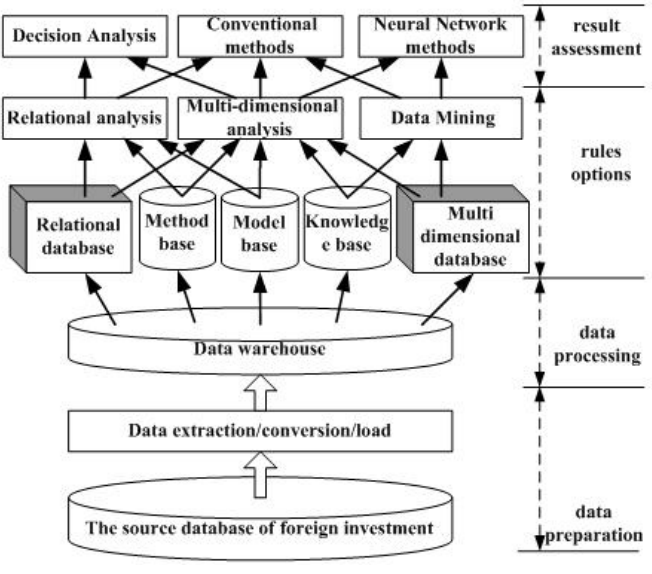


Fig. 1. The architecture of FIMDSS based on data warehouse

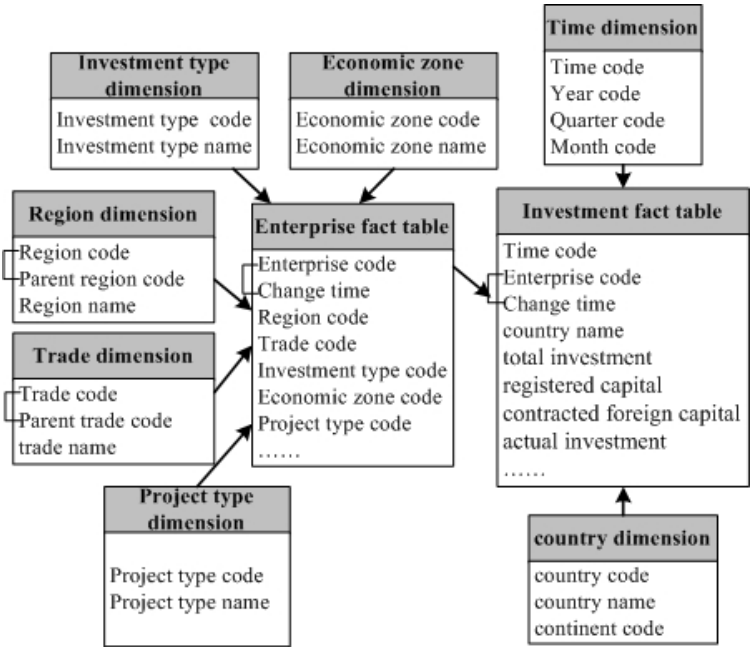


Fig. 2. The extensible star schema model of FIMDSS

investment country, investment region, investment trade, investment type and so on.

The dimensional data store of foreign investment can be implemented physically in the form of several different schemas model. Based on the demand of efficiency and maintenance, the multi-dimension data store schema of FIMDSS is determined as an extensible star schema model. In a star schema, the fact table which is located in the center of schema wills storages the fact data [7]. The dimension tables will storage dimension data. The fact table links to related dimension table with the dimension key in the fact table. The benefit of having a star schema is that it is simpler than other dimensional data store schemas, making it easier for the data preparation processes to load the data into FIMDDS. The multi-dimension data store schema of FIMDSS is shown in Fig. 2. The center of the schema is two fact tables: basic information of foreign-funded enterprises (BIFFE Fact table) and detail content of foreign investment (DCFI Fact table). The BIFFE fact table contains unique identifying data for foreign-funded enterprises, as well as data unique to the enterprise itself. The five keys in the BIFFE fact table link the fact table with the four dimension tables: Investment type, economic zone, region, trade, project type. On the one hand the DCFI fact table links to the BIFFE fact table with the two keys which are enterprise code and change time; on the other hand the DCFI fact table has two foreign key relationships to the time dimension table and country dimension table. The benefit of extensible star schema of FIMDSS is that the number of dimension tables can reduce even if the fact data are relatively vast.

3 The Multi-dimensional Data Mining Model

In FIMDSS, the problem which the decision-makers are focus on is that the investment factors are impacted by the dimensions in the specific subject. For example, which dimension members are key factors for the contractual foreign investment in the performance analysis subject of foreign-funded enterprises? An approach to solving the problem is that the key dimensions and members are discovered in related multi-dimensional model. The issue can be briefly described as follows: In subject S, there is a mining task T involving factor X. The object of task T is to discovery which dimensions and members are most affected by factor X. The task T is involved in n dimension. The member number of Each dimension is D_n. Some members of dimension are hierarchical. How to select members of dimensions is the key of task T.

The dimension is characterized by hierarchical structure, discrete data and non-numerical attributes. There is difficult how to calculate the correlation between quantity (Q) of task T and n dimension. Because the direct members of dimension reflect the property value of dimension, the property values are closely related to Q according to statistically analysis method. Therefore, the function model can be build to solve the problem. However, the function can not be derived from general analysis method because of complex non-linear features. The artificial neural network can solve those problems effectively [8].

In the hierarchical feed-forward neural network, the hidden layer is the feature extraction layer and represents the characteristics of the input mode [9]. The dimension characteristics of the data warehouse can be reflected on the hidden layer units. The related weight can determine the dimension value of the particular output level [10]. Aim at the features of foreign investment data warehouse,

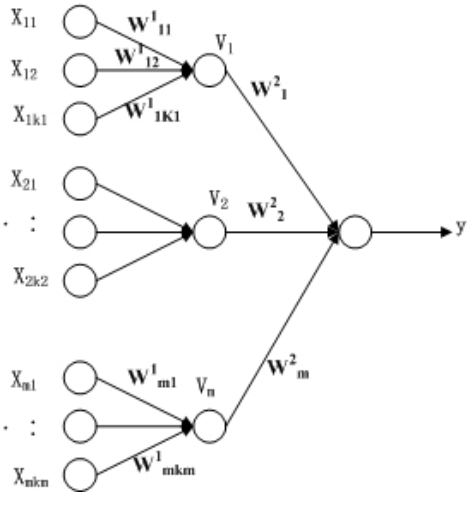


Fig. 3. The neural network model of multi-dimensional data mining

an extensible hierarchical feed-forward neural network is built to complete the task of data mining [11]. The details are shown in Fig. 3.

Fig. 3 shows the $X - V - Y$ (X denotes input neurons, V denotes hidden neurons, and Y denotes output neurons) architecture of a neural network model. The input layer can be considered the dimension data of foreign investment and be divided into m groups according to the dimensions number such as investment type, trade, region etc. The *ith* group is also divided into k_i components according to the dimension members. So the input mode is $X = ((x_{11}, x_{12}, \cdots, x_{1k_1}), (x_{m1}, x_{m2}, \cdots, x_{mk_m}))$. The outputs layer represents the key factor of foreign investment. The hidden layer determines the mapping relationships between input and output layers, whereas the relationships between neurons are stored as weights of the connecting links. The input signals are modified by the weight $W^1 = ((w_{11}, w_{12}, \cdots, w_{1k_1}), (w_{m1}, w_{m2}, \cdots, w_{mk_m}))$, which represents the interconnection of the *ith* node of the input layer to the *mth* node of the hidden layer. The sum of the modified signals is then modified by sigmoid transfer (f) function and the output is collected at the output layer. The sum of the modified signals (total activation) is then modified by a sigmoid transfer function (f). Similarly, the output signals of the hidden layer are modified by interconnection weight W^2 of the node of the output layer to the *mth* node of the hidden layer. The weight will identify the correlation between dimensions and key factors. The weight is greater; the related dimension is more importance to key factors.

Let $I_p = ((I_{p1}, I_{p2}, \cdots, I_{px}), p = 1, 2, \cdots, N)$ be the *pth* pattern among N input patterns. Where $w^1_{ik_j}$ and w^2_j are connection weights between the input neuron to the *mth* hidden neuron, and the *mth* hidden neuron to the output neuron, respectively output from a neuron in the input layer is $O_{pi} = I_{pi}(i = 1, 2, \cdots, x)$.

Output from a neuron in the hidden layer is

$$O_{pi} = f(Net_{pj}) = f(\sum_{i=1}^{x} w^1_{ik_j} O_{pi})(j = 1, 2, \cdots, m) \tag{1}$$

Output from a neuron in the output layer is

$$O_p = f(Net_p) = f(\sum_{j=1}^{m} w^2_j O_{pj}) \tag{2}$$

The network activation function can choose linear function or S-type function in accordance with the actual situation; network learning algorithm is still using the following algorithm:

$$w^l_{ji}(k+1) = w^l_{ji}(k) + \alpha \Delta w^l_{ji}(k) + \eta(w^l_{ji}(k-1) - w^l_{ji}(k-1)\Delta w^l_{ji}(k)) = \frac{\partial E}{\partial w^l_{ji}(k)} \tag{3}$$

α is learning step; η is momentum factor.

4 The Process of Multi-dimensional Data Mining of Foreign Investment

The above model can be used for the date mining of FIMDSS application involving specific subject. The following sample is multi-dimensional data mining of contracted foreign investment in the performance analysis subject of foreign-funded enterprises. The contracted foreign investment is one of the three major indexes in foreign investment statistics and can accurately reflect actual situation of foreign investment. The decision makers are interest in the in-depth analysis result of contracted foreign investment. For example, whose dimension member (trade, region, investment type) are important for the contracted foreign investment? The answer can provide the trend of foreign investment and policy-making knowledge. Based on the standardized description of data mining tasks, the sample task is: whose dimension member (trade, region, investment type) are more important for the contracted foreign investment in the performance analysis subject of foreign-funded enterprises from February 2002 to September 2004.

There are eleven dimensions related to performance analysis subject of foreign-funded enterprises in FIMDSS. Some dimensions are not necessary to analysis according to the management demand of actual foreign investment. Those dimensions will be removed from the dimensions space of mining model in order to reduce the complexity and simplify the processing. By the important analysis of all dimensions, three dimensions are identified to participate in the mining model, including trade, region and investment dimension. Because the source data of FIMDSS is update monthly, time dimension of problem model should be located in month layer and related statistical mining data will be grouped by month. The sample data is selected from February 2002 to September 2004. The portion data are shown in Table 1.

CFI is the contracted foreign investment. Q is the CFI rate of monthly change and the measure of mining task. X_{11} is the CFI rate of monthly change of Sino-foreign joint ventures. X_{12} is CFI rate of monthly change of Sino-foreign

Table 1. The portion data of mining sample

Year	*Month*	Q	X_{11}	X_{12}	X_{13}	X_{14}	X_{21}	X_{22}	X_{23}	X_{31}	X_{32}	X_{33}
2002	2	-0.412	-0.345	-0.353	-0.439	-0.250	-0.418	-0.309	-0.418	-0.341	-0.457	-0.253
2002	3	0.723	0.671	1.709	0.635	-0.816	0.748	0.544	0.468	0.407	0.720	0.761
2002	4	0.238	0.278	-0.236	0.314	3.526	0.225	0.351	0.371	-0.055	0.373	-0.089
2002	8	-0.260	-0.157	-0.480	-0.260	-0.513	-0.218	-0.412	-0.712	-0.585	-0.253	-0.264
2002	9	0.391	0.401	0.146	0.411	0.566	0.333	0.739	1.868	1.335	0.314	0.622
2003	1	-0.086	-0.127	0.158	-0.088	-0.607	-0.078	-0.438	0.492	-0.170	-0.056	-0.171
2003	2	-0.403	-0.402	-0.527	-0.394	0.542	-0.357	-0.614	-0.713	-0.417	-0.434	-0.298
2003	3	0.491	0.447	0.015	0.578	-0.794	0.434	1.905	0.329	1.429	0.461	0.511
2003	12	0.518	1.034	0.382	0.395	0.814	0.453	1.250	0.676	1.156	0.416	0.801
2004	1	-0.464	-0.551	-0.286	-0.442	-0.808	-0.440	-0.738	-0.203	-0.620	-0.428	-0.543
2004	8	0.218	0.767	0.444	0.132	-0.038	0.082	-5.678	0.659	0.423	0.223	0.188
2004	9	0.233	0.170	0.264	0.246	-0.052	0.227	0.587	-0.084	0.044	0.285	0.094

cooperative enterprises. X_{13} is CFI rate of monthly change of wholly foreign-owned enterprises. X_{14} is CFI rate of monthly change of foreign-invested joint-stock enterprises. X_{21} is CFI rate of monthly change of eastern region. X_{22} is CFI rate of monthly change of central region. X_{23} is CFI rate of monthly change of western region. $X_{31}1$ is CFI rate of monthly change of the primary industry. X_{32} is CFI rate of monthly change of the secondary industry. X_{33} is CFI rate of monthly change of the tertiary industry.

The mining model is the hierarchical feed-forward neural network. The training data set is chosen from February 2002 to December 2003. The test data set is chosen from January 2004 to September 2004. The neural network training and verification are done in the neural network toolbox of Matlab. The trainbpx is used for training function because it provides momentum method and adaptive learning adjustment strategy. The training results confirm that the trainbpx can improve the learning speed and increase the reliability of the algorithm. The initial weights of neural network use initff. The learning parameters are chosen by domain experts, including $\eta = 0.01, \alpha = 0.01, error = 0.01$. The sum squared error of training is shown in Fig. 4. After network training process is completed, the weights of hidden layer to output layer are shown in Table 2.

According to the mining result, The conclusion is that the foreign joint ventures of secondary industry in the eastern region are the most important dimension member for the contracted foreign investment. The CFI rate of monthly change of these enterprises directly affects the trend of national foreign investment. The training result is 90.3 percent accuracy. The test result is 85.2 percent accuracy. The mining model is proved feasibly and the analysis result can be provided to decision making.

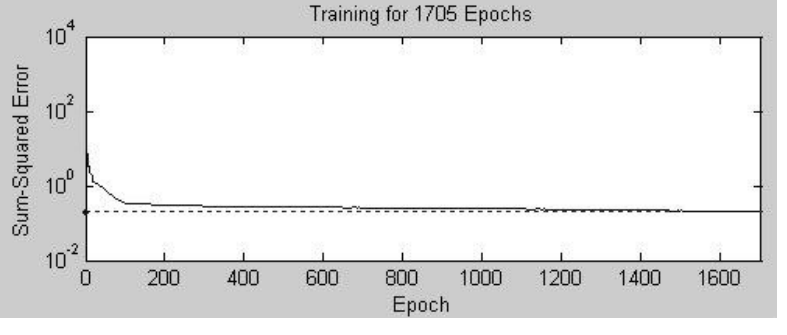


Fig. 4. The sum squared error of training process

Table 2. The weights of hidden layer to output layer

Weight name	*Weight value*	*Related dimension*
W_1^2	0.278546950	Investment type
W_2^2	0.5581839069	Region
W_3^2	0.3117403824	trade

5 Conclusion

This research aimed at decision analysis of the foreign investment in the financial statement, and used the hierarchical feed-forward neural network and data warehouse technology to mining the in-depth knowledge of foreign investment, in order to find a reliable basis for policy-making. This research adopted the necessary dataset from the source database of foreign investment. This data was then used to carry out a dimension member analysis. The sample is given to clarify the feasibility and effectiveness of the evaluation model.

At present, data mining is a new and important area of research, and neural network itself is very suitable for solving the problems of data mining because its characteristics of good robustness, self-organizing adaptive, parallel processing, distributed storage and high degree of fault tolerance. The combination of data mining method and neural network model can greatly improve the efficiency of data mining methods, and it has been widely used. It also will receive more and more attention.

References

1. Yu, L., Wang, S.Y., Lai, K.K.: A Hybrid Econometric-AI Ensemble Learning Model for Chinese Foreign Trade Prediction. IEEE Trans. Computational Science 43, 106–113 (2007)
2. National Bureau of Statistics of China, http://www.stats.gov.cn/
3. Toeho, H., Ingoo, H.: Knowledge-based Data Mining of News Information on the Internet Using Cognitive Maps and Neural Networks. Expert Systems with Applications 23, 231–243 (2002)
4. Guan, L., Liang, H.J.: Data Warehouse and Data Mining. Microcomputer Applications 15, 17–23 (1999)
5. Drummond, S., Joshi, A., Sudduth, K.A.: Application of Neural Networks: Precision Farming. IEEE Trans. Neural Netw. 9, 211–215 (1998)
6. Fausett, L.V.: Fundamentals of Neural Networks. Prentice Hall, Englewood Cliffs (1998)
7. Heimlich, R.: Precision Agriculture Information Technology for Improved Resource Use. Agricultural Outlook 52, 19–23 (1998)
8. Hu, X., Wang, J.: Solving Generally Constrained Generalized Linear Variational Inequalities Using the General Projection Neural Networks. IEEE Trans. Neural Netw. 18, 1697–1708 (2007)
9. MacKay, D.J.: Bayesian interpolation. Neural Computation 4, 415–447 (1992)
10. Serele, C.Z., Gwyn, Q.H., Boisvert, J.B.: Corn Yield Prediction with Artificial Neural Network Trained Using Airborne Remote Sensing and Topographic Data. IEEE Trans. Geoscience and Remote Sensing 25, 384–386 (2000)
11. Wagner, P., Schneider, M.: Economic Benefits of Neural Network-Generated Site-specific Decision Rules for Nitrogen Fertilization. IEEE Trans. Precision Agriculture 53, 775–782 (2007)

Allocation Method of Total Permitted Pollution Discharge Capacity Based on Uniform Price Auction

Congjun Rao[1,2], Zhongcheng Zhang[1], and June Liu[1]

[1] College of Mathematics and Information Science, Huanggang Normal University, Huanggang 438000, China
[2] Institute of Systems Engineering, Huazhong University of Science and Technology, Wuhan 430074, China
raocjun79@163.com

Abstract. To study the allocating problem of total permitted pollution discharge capacity(TPPDC), an allocation method with variable supply based on the uniform price auction of divisible goods is proposed in this paper firstly. Then, a linear equilibrium bidding strategy of this new method is given. Lastly, the incentive compatibility and validity of this method are proved. Therefore, this method will provide valuable theoretical basis and guidance for building the pollution emission permits trade system.

Keywords: Allocating of TPPDC, Uniform price auction, Bidding strategies, Incentive compatibility.

1 Introduction

As a limited resource, the values of total permitted pollution discharge capacity (TPPDC) depend on location, enterprise and pollutant discharge period. So the TPPDC must be allocated to polluters reasonably. The fairness and effectiveness of allocating the TPPDC is the basis of total capacity control, pollution control and environmental sustainable development, and is the key to maximize the utility of pollution control district[1].

TPPDC may be classed as divisible goods, whose allocation usually involves complicated private information. Recently, considerable attention has been given to the use of auction by divisible goods with uniform price. Many scholars have studied the divisible goods auction with uniform price. Ortega Reichert[2] analyzed the properties of sequential English auction, discriminatory price auction, and uniform price auction, and presented revenue equivalence theorem of multi-object auctions for the first time. Harris and Raviv[3] first gave an optimal conclusion, i.e., if the bidders' valuations are independent, and follow uniform distribution, then the auction mechanism in Ref.[2] is optimal. Maskin[4] gave a complete characterization for the multi-object auctions and generalized the conclusions in Ref.[3] to any valuation distribution. Especially, for the auctions of divisible goods, Back and Zender[5, 6] compared the single object auction with divisible multi-object auctions, and designed a special uniform price auction mechanism of divisible goods. This is a new idea of

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 1080 – 1089, 2009.

studying the auction of divisible goods. Thenceforward, Wang and Zender[7] derived equilibrium bidding strategies in divisible good auctions for asymmetrically informed risk neutral and risk averse bidders when there is random noncompetitive demand. Kremer and Nyborg[8] studied the impact of different allocation rules in divisible good, uniform price auction. Damianov[9] concluded that low-price equilibria in the uniform price auction with endogenous supply do not exist if the seller employs the proportional rationing rule and is consistent when selecting among profit-maximizing quantities. Indranil Chakraborty[10] studied the asymptotic price in the uniform price auction, the results showed that the expected price becomes large depend only on the aggregate of the marginal distributions of each bidder's marginal values, and not on the correlation between the marginal values. However, most of these research results are obtained based on some simple and especial conditions, for example, unitary demand for every bidder's valuation follows uniform distribution, the bid price of bidders are discrete, the bidders are symmetrical and so on. When these conditions are changed, the corresponding conclusions need to be reconsidered.

In this paper, an allocation method with variable supply based on the uniform price auction is proposed. It aimed at improving the social creditability and validity of TPPDC allocation, and provides a universally applicable method for auctioning and allocating the emission rights, stocks, treasury bills, network bandwidth and son on.

2 The Auction Model

Let Q_0 be the quantity of TPPDC, and n is the total number of polluters. Let $f_i(x)$ and $g_i(x)$ denote the actual marginal cost function and declared marginal cost function of pollution treatment respectively of the *i*th polluter, and G_i and $q_i \in [0,\infty)$ ($i = 1,2,\cdots,n$) denote the actual pollutant discharge capacity and the permitted pollution discharge capacity respectively of the *i*th polluter, where $x = G_i - q_i$ denotes the surplus pollution treatment.

Supposed that the government allocates TPPDC, Q_0 , under a uniform price $p = g_i(G_i - q_i)$ ($\forall i = 1,2,\cdots\cdots,n$) in which the marginal cost $g_i(x)$ is declared by the *i*th polluter. This polluter must pay pq_i to obtain the allocated permitted pollution discharge capacity q_i . The government's decision goal is $\underset{p}{Max}\ U = pQ_0$. Then, a specific allocation can be described as follows.

$$(D_1)\qquad \begin{aligned} &Max \quad U = pQ_0 \\ &s.t. \quad \begin{cases} \sum_{i=1}^{n} q_i = Q_0 \\ g_i(G_i - q_i) = p, \quad i = 1,2,\cdots,n \\ 0 \le q_i \le G_i, \qquad i = 1,2,\cdots,n \end{cases} \end{aligned}$$

Proposition 1. The bidders' collusion and low-price equilibria will occur if the total permitted pollution discharge capacity are allocated by the method of model D_1 .

Proof: The proof process can be seen in [3-6, 11].

Proposition 1 shows that the government would be unwise to use Model D_1, because it does not guarantee sufficient fairness and validity of TPPDC allocation. The main reason is that the equality $\sum_{i=1}^{n} q_i = Q_0$ in model D_1 restricts any decision by the government. Therefore, we must improve the Model D_1 to improve the government decisions, and to enhance the creditability and validity of TPPDC allocation.

Actually, $\underset{p}{Max}\, pQ_0$ of Model D_1 is completely determined by polluters. Thus, in order to improve government decisions and limit the scale of false declarations by polluters, we improve the Model D_1. The assumptions are as follows.

Assumption 1. Suppose that the equality restriction $\sum_{i=1}^{n} q_i = Q_0$ be modified such that $\sum_{i=1}^{n} q_i = Q \le Q_0$, where $Q \in [0, Q_0]$ is a random variable. This information is common knowledge.

Assumption 1 shows that the government should announce an upper limit on Q_0, and determine the actual capacity Q according to the principle of maximizing government income based on all polluters' specific declared marginal costs or prices.

Assumption 2. The marginal treatment cost $g_i(x)$ declared by the *i*th polluter is invariably less than the actual treatment cost $f_i(x)$, i.e., $f_i(x) \ge g_i(x)$.

Assumption 3. $\forall i$, $\frac{df_i(x)}{dx} \ge 0$ and $\frac{dg_i(x)}{dx} \ge 0$. In other words, the marginal costs increase with pollution treatment capacity x, such that $f_i(x) = a_i^{'} x + b_i^{'}$ (where $a_i^{'} \ge 0$ and $b_i^{'} \ge 0$ represent the variable and the fixed cost coefficient, respectively). For every polluter, $a_i^{'}$ and $b_i^{'}$ are all constants. Specially, we set $f_1(x) = f_2(x) = \cdots = f_n(x)$.

Assumption 4. A strategy for bidder *i* is a nonincreasing continuous differentiable function $q_i(p)$.

Assumption 5. There is no co-operation among the polluters.

Based on the above assumptions, we establish a new auction model.

The government's goal is to maximize its income $\sum_{i=1}^{n} pq_i = pQ$, raised from TPPDC allocation by choosing a specific total capacity $Q \le Q_0$ and a uniform price $p > 0$. Hence,

$$(D_2)\quad \begin{matrix} Max & pQ \\ S.T. & \begin{cases} \sum_i q_i = Q \le Q_0 \\ g_i(G_i - q_i) = p, i = 1,2,\cdots,n \\ 0 \le q_i \le G_i, \quad i = 1,2,\cdots,n \end{cases} \end{matrix}$$

The goal of the *i*th polluter is to maximize income

$$Max\int_{G_i-q_i}^{G_i}(f_i(x)-p)dx = Max\int_{G_i-q_i}^{G_i} f_i(x)dx - pq_i(p)$$

by choose the optimal bidding strategy $q_i(p)$.

D_2 may be thought to describe an auction method of completely divisible goods under a uniform price. Next we give the linear equilibrium bidding strategy of this new method, and discuss the incentive compatibility and validity of this method in the next section.

3 Bidding Strategy Analysis

Proposition 2. Suppose that all polluters are risk neutral, then a linear equilibrium bidding strategy $q(p)$ exists in the auction model D_2, i.e.,

$$q(p) = \bar{q} - \frac{n(n-1)\bar{q} - (n-2)Q}{b'Q - na' - nb'G_i}p$$

where $\bar{q}$ is the demand when auction base price is zero, and $\bar{q} \in [\frac{Q}{n}, \infty)$.

Proof. We give two steps to prove this conclusion. Firstly, we suppose the linear equilibrium bidding strategy exists in the auction model D_2, and solve this strategy.

Let the equilibrium bidding strategies of all polluters are $q(p) = a - bp$, $(a > 0, b \ge 0)$. When $p = 0$, we have $q(0) = \bar{q}$, thus

$$\bar{q} = a \tag{1}$$

In a symmetric equilibrium, $nq(p^*) = Q$, where p^* denotes the equilibrium price, so we have $n(a - bp^*) = Q$, and

$$p^* = \frac{a}{b} - \frac{Q}{nb} \tag{2}$$

Suppose that the polluters $1,2,\cdots,i-1,i+1,\cdots,n$ submit the same bidding strategies $q(p)$, and the polluter i's best response function is $y(p)$. When in an equilibrium, we have $(n-1)q(p_e) + y(p_e) = Q$, thus

$$y(p_e) = Q - (n-1)q(p_e) \tag{3}$$

So the polluter i 's income will be

$$R = \int_{G_i - q(p_e)}^{G_i} f_i(x)dx - p_e y(p_e) \tag{4}$$

By Assumption 3, we set $f_i(x) = a^{'}x + b^{'}$, then we obtain

$$R = a^{'}G_i + \frac{G_i^2}{2}b^{'} - a^{'}(G_i - a + bp_e) - \frac{(G_i - a + bp_e)^2}{2}b^{'} - p_e[Q - (n-1)(a - bp_e)]$$

According to the optimization condition $\frac{\partial R}{\partial p_e} = 0$, namely,

$$-a^{'}b - (G_i - a + bp_e)bb^{'} - Q + (n-1)(a - 2bp_e) = 0$$

We have

$$p_e = \frac{abb^{'} - a^{'}b - bb^{'}G_i - Q + a(n-1)}{2(n-1)b + b^2b^{'}} \tag{5}$$

In addition, the maximize second-order condition $\frac{\partial^2 R}{\partial (p_e)^2} = -b^2b^{'} - 2(n-1)b < 0$ is also satisfied.

Because we assume that the linear strategy $q(p)$ is a equilibrium strategy, so polluter i will not deviate this linear strategy, namely, $y(p) = q(p) = a - bp$, and $p_e = p^*$. Then we have

$$p_e = \frac{abb^{'} - a^{'}b - bb^{'}G_i - Q + a(n-1)}{2(n-1)b + b^2b^{'}} = p^* = \frac{a}{b} - \frac{Q}{nb}$$

Thus

$$b = \frac{n(n-1)}{b^{'}Q - a^{'}n - b^{'}nG_i}a - \frac{(n-2)Q}{b^{'}Q - a^{'}n - b^{'}nG_i} \tag{6}$$

From (1) and (6), we obtain

$$a = \bar{q}, \quad b = \frac{n(n-1)\bar{q} - (n-2)Q}{b^{'}Q - na^{'} - nb^{'}G_i} \tag{7}$$

Substituting the values of a, b into $q(p) = a - bp$, we have

$$q(p) = \bar{q} - \frac{n(n-1)\bar{q} - (n-2)Q}{b^{'}Q - na^{'} - nb^{'}G_i}p \tag{8}$$

Secondly, we prove the above linear strategy (8) is a real equilibrium strategy. On the premise that polluters $1,2,\cdots,i-1,i+1,\cdots,n$ all submit the same bidding strategies

(8), we need prove that the polluter i will not deviate this linear strategy. The detailed method is as follows: We are not consider the assumption of equilibrium, and substituting (7) into (2) and (5) respectively. We are easy to obtain $p_e = p^*$, which means linear strategy (8) is a real equilibrium strategy.

Proposition 3. Suppose that all polluters are risk neutral, then the expected Uniform price in auction model can be expressed as

$$E[p_0(Q)] = E\left[\frac{b'Q - na' - nb'G_i}{n(n-1)\bar{q} - (n-2)Q}[\bar{q} - \frac{Q}{n}]\right].$$

Proof. By Proposition 2, we have

$$q(p) = \bar{q} - \frac{n(n-1)\bar{q} - (n-2)Q}{b'Q - na' - nb'G_i}p$$

Thus

$$p(Q) = \frac{b'Q - na' - nb'G_i}{n(n-1)\bar{q} - (n-2)Q}[\bar{q} - q(Q)]$$

In a symmetric equilibrium, we have $q(Q) = \frac{Q}{n}$. So if all polluters are risk neutral, then the equilibrium auction price is

$$p_0(Q) = \frac{b'Q - na' - nb'G_i}{n(n-1)\bar{q} - (n-2)Q}[\bar{q} - \frac{Q}{n}] \tag{9}$$

In addition, the auction price is a random variable before the auction is over. So all polluters can't know the accurate value of auction price, and they only know a expected value based on the prior distribution of Q. Thus

$$E[p_0(Q)] = E\left[\frac{b'Q - na' - nb'G_i}{n(n-1)\bar{q} - (n-2)Q}[\bar{q} - \frac{Q}{n}]\right] \tag{10}$$

4 Stimulant and Validity Analysis

Every polluter possesses the private information which unknown by the auctioneer and the other polluters. If don't give them proper stimulant, then they will not announce their private information honestly. In general, an auction mechanism is an effective and feasible mechanism, the condition of incentive compatibility must be satisfied. In this section, we will discuss the incentive compatibility of Model D_2.

Proposition 4. In Model D_2, the government's optimal decision will be $Q^* = Q_0$.

Proof. Suppose that the equilibrium price is p^*, the equilibrium allocation quantity of polluter i is q_i^*, and an equilibrium total capacity chosen by the government is Q^* where $Q^* < Q_0$. If there exists a polluter i who can profit from the deviation of the equilibrium while the deviation can also make a profit for the government, then the players (comprising the government and the polluters) will reach their equilibrium only while $Q^* = Q_0$.

We set $Q_1 \in [0, Q_0]$, $Q_2 \in [0, Q_0]$, $Q_1 < Q_2$. On the one hand, when the government's quantity supplied increases from Q_1 to Q_2, we consider the changes in government's income.

By (9), the government's income can be denoted as

$$B = pQ = \frac{b'Q - na' - nb'G_i}{n(n-1)\bar{q} - (n-2)Q}[\bar{q} - \frac{Q}{n}]Q$$

$$= \frac{-\frac{b'}{n}Q^2 + (a' + 2b'\bar{q})Q - na'\bar{q} - nb'\bar{q}^2}{\frac{n(n-1)\bar{q}}{Q} - (n-2)} \triangleq \frac{k_2}{k_1}$$

Obviously, for the denominator k_1, the greater the value of Q is, the smaller the value of k_1 is. For k_2, when the government's quantity supplied increases from Q_1 to Q_2, the variable quantity of k_2 is

$$\Delta k_2 = -\frac{b'}{n}Q_2^2 + (a' + 2b'\bar{q})Q_2 - [-\frac{b'}{n}Q_2^2 + (a' + 2b'\bar{q})Q_2]$$

$$= (Q_2 - Q_1)[a' + b'(\bar{q} - \frac{Q_2}{n}) + b'(\bar{q} - \frac{Q_1}{n})]$$

Because $a' > 0, b' > 0, \bar{q} - \frac{Q_2}{n} > 0, \bar{q} - \frac{Q_1}{n} > 0$, $Q_1 < Q_2$, so we have $\Delta k_2 > 0$, in other words, when the value of Q increase, the value of k_2 will increase. Therefore, when the government's quantity supplied increases from Q_1 to Q_2, the government's income $B = \frac{k_2}{k_1}$ will increase, which means the government will make a profit.

On the other hand, we consider the changes in polluter *i*'s income. When the government's quantity supplied increases, the allocated capacity q_i of the *i*th polluter satisfies $q_i \geq q_i^*$, and the auction clearing price $p < p^*$. So its income will be

$$\int_{G_i - q_i}^{G_i} v_i(x)dx - pq_i = [\int_{G_i - q_i}^{G_i - q_i^*} v_i(x)dx - p(q_i - q_i^*)] + [\int_{G_i - q_i^*}^{G_i} v_i(x)dx - pq_i^*]$$

$$\geq \int_{G_i - q_i^*}^{G_i} v_i(x)dx - pq_i^* \geq \int_{G_i - q_i^*}^{G_i} v_i(x)dx - p^* q_i^*$$

which means the *i*th polluter will also profit by its smart deviation. Thus by accumulating the players' profits, the former equilibrium capacity Q^* chosen by the government must increase until $Q^* = Q_0$.

Proposition 5. The auction mechanism given by model D_2 is an incentive compatibility mechanism.

Proof. Suppose that the equilibrium price is p^*, the equilibrium allocation quantity of polluter i is q_i^*, and an equilibrium total capacity chosen by the government is $Q^* = Q_0$. The allocated capacities $q_1, q_2, ..., q_n$ may be regarded as functions of a uniform price p, where $q_i = Q - \sum_{\substack{j=1 \\ j\neq i}}^{n} q_j$ and $\frac{dq_i}{dp} \le 0$ ($i = 1,2,\cdots,n$).

First, the government's goal is $\underset{p}{Max}\, pQ$. According to the optimization condition and Proposition 4, we have $\left.\frac{dU}{dp}\right|_{p=p^*} = 0$. Thus $\left.\frac{dU}{dp}\right|_{p=p^*} = Q(p^*) + p^* \left.\frac{dQ(p)}{dp}\right|_{p=p^*} = 0$. So we have

$$\left.\frac{dQ}{dp}\right|_{p=p^*} = -\frac{Q_0}{p^*} \tag{11}$$

Then the *i*th polluter's goal is $Max\ R = \int_{G_i - Q + \sum_{\substack{j=1 \\ j\neq i}}^{n} q_j}^{G_i} f_i(x)dx - pq_i(p)$. According to the optimization condition, we have

$$\left.\frac{dR}{dp}\right|_{p=p^*} = \frac{d}{dp}\left[\int_{G_i - Q + \sum_{\substack{j=1 \\ j\neq i}}^{n} q_j}^{G_i} f_i(x)dx - p(Q - \sum_{\substack{j=1 \\ j\neq i}}^{n} q_j)\right]\Bigg|_{p=p^*} = 0$$

Namely,

$$-f_i(x_i^*)\left.\frac{d(G_i - Q + \sum_{\substack{j=1 \\ j\neq i}}^{n} q_j)}{dp}\right|_{p=p^*} - q_i^* - p^*\left(-\sum_{\substack{j=1 \\ j\neq i}}^{n} \frac{dq_j}{dp}\right)\Bigg|_{p=p^*} = 0$$

Thus

$$-f_i(x^*)\left(\sum_{\substack{j=1 \\ j\neq i}}^{n} \frac{dq_j}{dp}\right)\Bigg|_{p=p^*} - q_i^* + p^* \sum_{\substack{j=1 \\ j\neq i}}^{n} \left.\frac{dq_j}{dp}\right|_{p=p^*} = 0$$

So we obtain

$$q_i^* = -\left(f_i(x^*) - p^*\right)\left(\sum_{\substack{j=1 \\ j\neq i}}^{n} \frac{dq_j}{dp}\bigg|_{p=p^*} + \frac{dq_i}{dp}\bigg|_{p=p^*}\right) + \left(f_i(x^*) - p^*\right)\frac{dq_i}{dp}\bigg|_{p=p^*}$$

$$= -\left(f_i(x^*) - p^*\right)\frac{dQ}{dp}\bigg|_{p=p^*} + \left(f_i(x^*) - p^*\right)\frac{dq_i}{dp}\bigg|_{p=p^*} \tag{12}$$

Taking the sum of both sides of Equations (12) where *i* is from 1 to n

$$Q_0 = -\sum_{i=1}^{n}\left(f_i(x^*) - p^*\right)\frac{dQ}{dp}\bigg|_{p=p^*} + \left(f_i(x^*) - p^*\right)\frac{dQ}{dp}\bigg|_{p=p^*}$$

From (11) and (12), we have $Q_0 = [-\sum_{i=1}^{n-1} f_i(x^*) + (n-1)p^*]\left(-\frac{Q_0}{p^*}\right)$, Thus

$$p^* = \frac{\sum_{i=1}^{n-1} f_i(x^*)}{n}$$

By assumption 2, $f_i(x) = f(x)$, $i = 1,2,\cdots,n$, then $p^* = \frac{n-1}{n} f_i(x_i^*)$.

Proposition 5 shows that Model D_2 can guarantee a base price of allocation or auction for a government, and the government can almost gain the actual average marginal treatment cost of the polluters, provided n is sufficiently large. It is shown that the auction mechanism given by model D_2 is an incentive compatibility mechanism.

5 Conclusions

This paper studies the allocating problem of total permitted pollution discharge capacity (TPPDC), and presents an allocation method with variable supply based on the uniform price auction. If the polluters are symmetric, such that their marginal treatment cost functions $f(x)$ and pollutant discharge capacities G are each the same, it has been shown that the government obtains an equilibrium price, expressed as $p^* = \frac{n-1}{n} v(x^*)$. This price is in accordance with, but more universal in applicability than an equilibrium price derived by Back[6] and Damianov[9]. It is shown that our new allocation method is feasible and effective.

Acknowledgements

This work was supported by the Postgraduate Science & Technology Innovation Fund of HUST (No. HF-06-007-08-184), the Excellent Youth Project of Hubei Provincial

Department of Education (No. Q20082703), and the Significant Project Grant of Hubei Provincial Department of Education (No. 20082704).

References

1. Ma, Z., Dudek, D.: Pollutant Gross Control and Pollutant Discharge Right Trade. Chinese Environment Science Publishing Company, Beijing (1999)
2. Ortega Reichert, A.: A Sequential Game with Information Flow. Chapter 8 in Models for Competitive Bidding under Uncertainty, Stanford University PhD Thesis, 232–254 (1981)
3. Harris, M., Raviv, A.: A Theory of Monopoly Pricing Schemes with Demand Uncertainty. American Economic Review 71, 347–365 (1981)
4. Maskin, E., Riley, J.: Optimal Multi-Unit Auction. In: Fed, H. (ed.) The Economics of Missing Markets. Information and Games. Oxford University Press, New York (1989)
5. Back, K., Zender, J.F.: Auctions of Divisible Goods: on the Rationale for the Treasury Experiment. Review of Financial Studies 6, 733–764 (1993)
6. Back, K., Zender, J.F.: Auctions of Divisible Goods with Endogenous Supply. Economics Letters 73, 29–34 (2001)
7. Wang, J.J.D., Zender, J.F.: Auctioning Divisible Goods. Economic Theory 19, 673–705 (2002)
8. Kremer, I., Nyborg, K.: Divisible-Good Auctions: the Role of Allocation Rules. Rand Journal of Economics 35, 147–159 (2004)
9. Damianov, D.S.: The Uniform Price Auction with Endogenous Supply. Economics Letters 77, 101–112 (2005)
10. Indranil, C., Richard, E.W.: Asymptotic Prices in Uniform-Price Multi-Unit Auctions. Economic Theory 4, 983–987 (2005)
11. Zhao, Y., Borthwick, A., Wang, Q.: Competition Allocation of Total Permitted Pollution Discharge Capacity Based on Divisible Goods Auction. European Journal of Operational Research (in press, 2008)

Fuzzy Group Decision Making Method and Its Application

Cheng Wang[1,2], Zhongcheng Zhang[1], and Congjun Rao[1,2]

[1] College of Mathematics and Information Science, Huanggang Normal University, Huanggang 438000, China
[2] Institute of Systems Engineering, Huazhong University of Science and Technology, Wuhan 430074, China
Wangc80@163.com

Abstract. This paper studies the problems of fuzzy multi-attribute group decision making in which the attribute values are given in the form of linguistic fuzzy numbers, and presents a new fuzzy group decision making method. Firstly, some definitions and conclusions about trapezoidal fuzzy numbers are given. Secondly, an operator named TOWA is presented to aggregate the trapezoidal fuzzy numbers. Thirdly, a new method is presented for the problems of fuzzy multi-attribute group decision making via the TOWA operator and grey relative degree. Finally, this new method is applied in the problem of official evaluation and selection. Therefore, a feasible and effective way is obtained to solve the problems of fuzzy multi-attribute group decision making under the uncertain environments.

Keywords: Multi-attribute group decision making, Trapezoid fuzzy numbers, TOWA (Trapezoidal Ordered Weighted Averaging) operator, Grey relative degree.

1 Introduction

In practical decision making, decision makers usually express their preference information over alternatives with the type of linguistic fuzzy variables[1-3]. For example, when they make evaluation on the problems of human's makings, automobile's capability, pollution extent and so on, linguistic labels like worst, worse, bad, common, good, better, best" or "lowest, lower, low, common, high, higher, highest "[2] are usually used. Nowadays, it is very important and popular for us to study on the problems of multi-attribute group decision making with the attribute values in the forms of linguistic fuzzy numbers. Some authors have paid attention to this research domain, and proposed some approaches to solving the multi-attribute decision making problem with linguistic information [4-9]. However, these methods are easy to lose decision information, so the precise decision results can't be obtained. In addition, the structures of these models are very complicated, the maneuverability and practicability of these models are not very good. Based on these disadvantages, this paper studies the problems of fuzzy multi-attribute group decision making in

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 1090 – 1097, 2009.

which the attribute values are given in the form of linguistic fuzzy numbers, and presents a new decision making method. It tries to find a new and effective way to deal with the problems of fuzzy multi-attribute group decision making under the uncertain environments.

2 Decision Principle and Method

First of all, we describe the problems of fuzzy multi-attribute group decision making as follows.

Let $X=\{X_1,X_2,\cdots,X_m\}$ be the set of alternatives, and let $P=\{p_1,p_2,\cdots,p_n\}$ the set of attributes. $W=(w_1,w_2,\cdots,w_n)$ is the vector of attribute weights, where w_j satisfies $0\le w_j\le 1,\ \sum_{j=1}^{n} w_j=1$. Let $D=\{d_1,d_2,\cdots,d_l\}$ be the set consisting of decision makers. The weight vector of decision makers is denoted as $V=(v_1,v_2,\cdots,v_l)$, where $0\le v_k\le 1,\ \sum_{k=1}^{l} v_k=1$.

Suppose that the evaluation value of alternative x_i $(i=1,2,\cdots,m)$ on attribute p_j $(j=1,2,\cdots,n)$, which given by decision maker $d_k\in D$, is in the form of linguistic fuzzy number, it is denoted as $\tilde{t}_{ij}^{(k)}$, $i=1,2,\cdots,m$, $j=1,2,\cdots,n$, $k=1,2,\cdots,l$. So there exists one decision making matrix given by every decision maker d_k , it is denoted as T_k , where $T_k=(\tilde{t}_{ij}^{(k)})_{m\times n}$, $k=1,2,\cdots,l$.

Now if the information of all matrixes $T_1,T_2,\cdots,T_l$ are given, then we need to make evaluation and rank order for all alternatives, and choose the best one among all alternatives. The decision principle and method are given as follows.

2.1 Processing Data for Attributes

In practical decision making, if the attribute values are given in the form of linguistic fuzzy numbers by the decision makers, such as "worst, worse, bad, common, good, better, best" or "lowest, lower, low, common, high, higher, highest ". We can transform all linguistic fuzzy numbers into trapezoidal fuzzy numbers. Firstly, we give some definitions and conclusions about trapezoidal fuzzy numbers.

Definition 1. Let $\tilde{a}_1=(b_1,c_1,d_1,e_1)$ and $\tilde{a}_2=(b_2,c_2,d_2,e_2)$ be two trapezoidal fuzzy numbers. Then

(1) $\tilde{a}_1+\tilde{a}_2=(b_1,c_1,d_1,e_1)+(b_2,c_2,d_2,e_2)=(b_1+b_2,c_1+c_2,d_1+d_2,e_1+e_2)$

(2) $\lambda\tilde{a}_1=\lambda(b_1,c_1,d_1,e_1)=(\lambda b_1,\lambda c_1,\lambda d_1,\lambda e_1)$, $\lambda>0$

Based on the above operations of trapezoidal fuzzy numbers, we will define the distance between two trapezoidal fuzzy numbers. Distance may be defined in many ways. In this paper we give the following definition based on absolute value.

Definition 2. Let $\tilde{a}_1 = (b_1, c_1, d_1, e_1)$ and $\tilde{a}_2 = (b_2, c_2, d_2, e_2)$ be two trapezoidal fuzzy numbers. The distance from $\tilde{a}_1$ to $\tilde{a}_2$ is defined by

$$D(\tilde{a}_1, \tilde{a}_2) = \frac{1}{4}(|b_1 - b_2| + |c_1 - c_2| + |d_1 - d_2| + |e_1 - e_2|)$$

Obviously, the above distance $D(\tilde{a}_1, \tilde{a}_2)$ satisfies the following conditions:

(i) $D(\tilde{a}_1, \tilde{a}_2) \geq 0$, $D(\tilde{a}_1, \tilde{a}_2) = 0 \Leftrightarrow \tilde{a}_1 = \tilde{a}_2$;
(ii) $D(\tilde{a}_1, \tilde{a}_2) = D(\tilde{a}_2, \tilde{a}_1)$;
(iii) $D(\tilde{a}_1, \tilde{a}_3) \leq D(\tilde{a}_1, \tilde{a}_2) + D(\tilde{a}_2, \tilde{a}_3)$.

For any two trapezoidal fuzzy numbers $\tilde{a}_1 = (b_1, c_1, d_1, e_1)$ and $\tilde{a}_2 = (b_2, c_2, d_2, e_2)$, if $b_1 = c_1 = d_1 = e_1$, $b_2 = c_2 = d_2 = e_2$, then both $\tilde{a}_1$ and $\tilde{a}_2$ become real numbers. So we have

$$D(\tilde{a}_1, \tilde{a}_2) = \frac{1}{4}(|b_1 - b_2| + |b_1 - b_2| + |b_1 - b_2| + |b_1 - b_2|) = |b_1 - b_2| = d(b_1, b_2)$$

Definition 3 [10]. Let $\tilde{a}_1 = (b_1, c_1, d_1, e_1)$ be a trapezoidal fuzzy numbers, the expected value of $\tilde{a}_1$ is defined by $E[\tilde{a}_1] = \dfrac{b_1 + c_1 + d_1 + e_1}{4}$.

Theorem 1. Let $\tilde{a}_1 = (b_1, c_1, d_1, e_1)$ and $\tilde{a}_2 = (b_2, c_2, d_2, e_2)$ be two trapezoidal fuzzy numbers. Then $\tilde{a}_1 > \tilde{a}_2$ iff $E[\tilde{a}_1] > E[\tilde{a}_2]$.

Proof. The Theorem 1 may be proved by a similar process of Ref. [10].

The Theorem 1 gives a method for ranking trapezoidal fuzzy variables based on expected value operator.

Definition 4. If the elements of vector $\overline{R}$ are all trapezoidal fuzzy numbers, then $\overline{R}$ is called a trapezoidal fuzzy vector. If the elements of two trapezoidal fuzzy vector are all just the same, then we call these two vectors are equal vectors.

Secondly, We can transform all linguistic fuzzy numbers into trapezoidal fuzzy numbers by the following Definition 5.

Definition 5. Suppose that S_1 ={worst, worse, bad, common, good, better, best}[2] or S_2 ={lowest, lower, low, common, high, higher, highest}[11], then S_1 and S_2 are called the set of linguistic fuzzy numbers, and their corresponding trapezoidal fuzzy numbers are defined as follows:

(i) For the type of revenue attributes, we have

best=(0.8,1,1,1) ; better=(0.7,0.9,1,1) ; good=(0.6,0.8,0.8,0.9) ;
common=(0.3,0.5,0.5,0.7) ; bad=(0,0.2,0.2,0.4) ; worse=(0,0,0.1,0.3) ;
worst=(0,0,0,0.2)

(ii) For the type of cost attributes, we have

best=(0,0,0,0.2) ; better=(0,0,0.1,0.3) ; good=(0,0.2,0.2,0.4) ;
common=(0.3,0.5,0.5,0.7) ; bad=(0.6,0.8,0.8,0.9) ; worse=(0.7,0.9,1,1) ;
worst=(0.8,1,1,1)

the corresponding trapezoidal fuzzy numbers for S_2 are the same as S_1.

When the attribute values are all transformed into trapezoidal fuzzy numbers, they are denoted as $\overline{R_k}$, where $\overline{R_k} = (\tilde{s}_{ij}^{(k)})_{m\times n}$, $\tilde{s}_{ij}^{(k)} = (b_{ij}^{(k)}, c_{ij}^{(k)}, d_{ij}^{(k)}, e_{ij}^{(k)})$, $k = 1,2,\cdots,l$. We need to normalize the matrix $\overline{R_k}$. The following algorithms[11] are given to normalize the matrix $\overline{R_k}$. Suppose that the matrix $\overline{R_k}$ is transformed into the matrix Y_k, where

$$Y_k = (\tilde{y}_{ij}^{(k)})_{m\times n}, \quad \tilde{y}_{ij}^{(k)} = (b_{ij}^{'(k)}, c_{ij}^{'(k)}, d_{ij}^{'(k)}, e_{ij}^{'(k)}),$$

$$i = 1,2,\cdots,m, \ j = 1,2,\cdots,n, \ k = 1,2,\cdots,l$$

For the type of revenue attributes, we have

$$\tilde{y}_{ij}^{(k)} = (b_{ij}^{'(k)}, c_{ij}^{'(k)}, d_{ij}^{'(k)}, e_{ij}^{'(k)})$$

$$= \left(\frac{b_{ij}^{(k)}}{\max\limits_i (e_{ij}^{(k)})}, \frac{c_{ij}^{(k)}}{\max\limits_i (d_{ij}^{(k)})}, \frac{d_{ij}^{(k)}}{\max\limits_i (c_{ij}^{(k)})} \wedge 1, \frac{e_{ij}^{(k)}}{\max\limits_i (b_{ij}^{(k)})} \wedge 1 \right) \tag{1}$$

For the type of cost attributes, we have

$$\tilde{y}_{ij}^{(k)} = (b_{ij}^{'(k)}, c_{ij}^{'(k)}, d_{ij}^{'(k)}, e_{ij}^{'(k)})$$

$$= \left(\frac{\min\limits_i (b_{ij}^{(k)})}{e_{ij}^{(k)}}, \frac{\min\limits_i (c_{ij}^{(k)})}{d_{ij}^{(k)}}, \frac{\min\limits_i (d_{ij}^{(k)})}{c_{ij}^{(k)}} \wedge 1, \frac{\min\limits_i (e_{ij}^{(k)})}{b_{ij}^{(k)}} \wedge 1 \right) \tag{2}$$

where operator "$\wedge$"is the floor operator.

In the process of decision making, we need to synthetize the decision results of all decision makers to make evaluation for all alternatives. For solving this problem, an operator named TOWA is presented as follows.

Definition 6. Let $\tilde{a}_1, \tilde{a}_2, \cdots, \tilde{a}_n$ be the trapezoidal fuzzy numbers which need to be aggregated, where $\tilde{a}_i = (b_i, c_i, d_i, e_i)$, $i = 1,2,\cdots,n$. The $TOWA$ operator is defined by

$$TOWA_W(\tilde{a}_1, \tilde{a}_2, \cdots, \tilde{a}_n) = W \cdot \tilde{G} = \sum_{i=1}^{n} w_i \cdot \tilde{g}_i = (\sum_{i=1}^{n} w_i \cdot b_i, \sum_{i=1}^{n} w_i \cdot c_i, \sum_{i=1}^{n} w_i \cdot d_i, \sum_{i=1}^{n} w_i \cdot e_i)$$

where $W = (w_1, w_2, \cdots, w_n)$ is a vector of weights, and it satisfies $0 \le w_i \le 1$, $\sum_{i=1}^{n} w_i = 1$. $\tilde{G} = (\tilde{g}_1, \tilde{g}_2, \cdots, \tilde{g}_n)$ is a ranking vector of $\tilde{a}_1, \tilde{a}_2, \cdots, \tilde{a}_n$, and $\tilde{g}_j$ is the j-th number which ranked by the ranking approaches given in Theorem 1.

By using the TOWA operator given by Definition 6, the matrixes $Y_1, Y_2, \cdots, Y_l$ are aggregated as the group decision making matrix $Z = (\tilde{z}_{ij})_{m\times n}$, where

$$\tilde{z}_{ij} = (b_{ij}^{"}, c_{ij}^{"}, d_{ij}^{"}, e_{ij}^{"}) = TOWA_W(\tilde{y}_{ij}^{(1)}, \tilde{y}_{ij}^{(2)}, \cdots, \tilde{y}_{ij}^{(l)}) = V \cdot \tilde{G}$$

$$= \sum_{k=1}^{l} v_k \cdot \tilde{g}_k = (\sum_{k=1}^{l} v_k \cdot b_{ij}^{'(k)}, \sum_{k=1}^{l} v_k \cdot c_{ij}^{'(k)}, \sum_{k=1}^{l} v_k \cdot d_{ij}^{'(k)}, \sum_{k=1}^{l} v_k \cdot e_{ij}^{'(k)}) \tag{3}$$

and $V = (v_1, v_2, \cdots, v_n)$ is the weight vector of decision makers, $\tilde{G} = (\tilde{g}_1, \tilde{g}_2, \cdots, \tilde{g}_l)$ is a ranking vector of $\tilde{a}_1, \tilde{a}_2, \cdots, \tilde{a}_n$, and $\tilde{g}_k$ is the k-th number which is ranked by the ranking approaches given in Theorem 1.

2.2 Algorithm of Grey Relative Degree

Grey relational space is the measure space of difference information which satisfies grey relational axioms, and difference information is the numerical value which represents the difference between reference sequence and compared sequence [12]. In this subsection, from the idea of difference information, we will present an algorithm of grey relative degree for ranking all alternatives in the problems of fuzzy group decision making.

Firstly, the definitions of fuzzy positive ideal and grey relative degree of the trapezoidal fuzzy vector are given.

Definition 7. Using the ranking approaches given by Theorem 1 to choose the maximum number among all trapezoidal fuzzy numbers of every column in the group decision making matrix $Z = (\tilde{z}_{ij})_{m\times n}$, and these n new trapezoidal fuzzy numbers form a new fuzzy vector, it is denoted as $T = (\tilde{t}_1, \tilde{t}_2, \cdots, \tilde{t}_n)$. Then T is called a fuzzy positive ideal.

Definition 8. Let fuzzy positive ideal $T = (\tilde{t}_1, \tilde{t}_2, \cdots, \tilde{t}_n)$ be the reference sequence, and alternative points $X_1, X_2, \cdots, X_m$ be the compared sequences, where $X_i = (\tilde{z}_{i1}, \tilde{z}_{i2}, \cdots, \tilde{z}_{in})$, $i = 1,2,\cdots,m$. Let

$$r(\tilde{t}_j, \tilde{z}_{ij}) = \frac{0.5 \max\limits_i \max\limits_j D(\tilde{t}_j, \tilde{z}_{ij})}{D(\tilde{t}_j, \tilde{z}_{ij}) + 0.5 \max\limits_i \max\limits_j D(\tilde{t}_j, \tilde{z}_{ij})}, \tag{4}$$

$$r(T, X_i) = \sum_{j=1}^{n} w_j r(\tilde{t}_j, \tilde{z}_{ij}), \tag{5}$$

where $D(\tilde{t}_j, \tilde{z}_{ij})$ is the distance from $\tilde{t}_j$ to $\tilde{z}_{ij}$ (The definition of distance between two trapezoidal fuzzy numbers is given by Definition 2), and w_j is the weight of the j-th attribute p_j, $j = 1,2,\cdots,n$. Then $r(T, X_i)$ is called the grey relational degree

between trapezoidal fuzzy vectors T and X_i, its shortened form is grey relational degree of trapezoidal fuzzy vectors.

Theorem 2. The grey relational degree $r(T, X_i)$ defined above satisfies the grey relational analysis (GRA) four axioms, i.e., normality, symmetry, wholeness, approachility.

Proof. The Theorem 6 may be proved by a similar process of Ref. [13, 14].

Applying the grey relational degree defined in Definition 15, we can calculate the grey relational degree $r(T, X_i)$ between alternative point X_i and fuzzy positive ideal T. The greater the value of $r(T, X_i)$ is, the better the alternative X_i is.

Secondly, we give the steps for the algorithm of grey relative degree.

Step 1: For m decision alternatives, determining the original decision making matrixes $T_1, T_2, \cdots, T_l$ by all decision makers.

Step 2: Using Definition 5 to transform all linguistic fuzzy numbers into trapezoidal fuzzy numbers, we get matrixes $\overline{R}_1, \overline{R}_2, \cdots, \overline{R}_l$.

Step 3: Using (1), (2), (3) to process data of $\overline{R}_1, \overline{R}_2, \cdots, \overline{R}_l$, then the group decision making matrix $Z = (\tilde{z}_{ij})_{m\times n}$ is obtained.

Step 4: Determining the positive ideal solution T and alternative points $X_1, X_2, \cdots, X_m$ from matrix Z respectively.

Step 5: Calculating the grey interval relational degree $r(T, X_i)$, $i = 1,2,\cdots,m$.

Step 6: Ranking all alternatives in accordance with $r(T, X_i)$.

3 An Application Example

Official evaluation and selection is a problem of fuzzy multi-attribute group decision making. Now one department will select an optimal candidate among four alternatives x_i $(i = 1,2,3,4)$, and five attributes[15] are given to evaluate these four alternatives, i.e. moral (p_1), work style (p_2), the level and the structure of knowledge (p_3), leadership ability (p_4), deploitation ability (p_5).Three decision makers (denoted as d_1, d_2, d_3) are invited to evaluate these four alternatives according to the five attributes above, and their evaluation information are listed in Table 1 to Table 3.

Table 1. The decision making matrix given by decision maker d_1

	P_1	P_2	P_3	P_4	P_5
x_1	better	best	best	common	better
x_2	best	good	common	good	best
x_3	good	good	best	better	best
x_4	good	good	worse	better	best

Table 2. The decision making matrix given by decision maker d_2

	P_1	P_2	P_3	P_4	P_5
x_1	better	good	best	common	good
x_2	best	better	common	better	best
x_3	best	better	good	good	best
x_4	better	good	common	better	best

Table 3. The decision making matrix given by decision maker d_3

	P_1	P_2	P_3	P_4	P_5
x_1	common	good	good	better	best
x_2	good	better	better	best	good
x_3	good	good	best	good	best
x_4	common	better	worse	better	best

Suppose that the vector of attribute weights is $W = (0.2,0.3,0.2,0.2,0.1)$, and the weight vector of decision makers is $V = (0.3,0.4,0.3)$. Our aim is to rank all alternatives and to choose the optimal candidate.

1) By Step 1 , Step 2 and Step 3, the group decision making matrix $Z = (\tilde{z}_{ij})_{m\times n}$ is obtained as follows

$$Z = \begin{pmatrix} (0.51,0.71,0.73,1) & (0.73,0.93,1,1) & (0.77,0.97,1,1) & (0.39,0.59,0.63,1) & (0.7,0.9,0.94,1) \\ (0.77,0.97,1,1) & (0.63,0.83,0.92,1) & (0.39,0.59,0.59,0.91) & (0.7,0.9,0.97,1) & (0.77,0.97,1,1) \\ (0.73,0.93,1,1) & (0.67,0.87,0.97,1) & (0.77,0.97,1,1) & (0.67,0.87,0.97,1) & (0.8,1,1,1) \\ (0.54,0.74,0.79,1) & (0.63,0.83,0.92,1) & (0.09,0.29,0.29,0.61) & (0.6,0.8,0.86,1) & (0.8,1,1,1) \end{pmatrix}$$

2) Determine the fuzzy positive ideal T from matrix Z

$$T = ((0.77,0.97,1,1),(0.73,0.93,1,1),(0.77,0.97,1,1),(0.7,0.9,0.97,1),(0.8,1,1,1)$$

3) Calculate the grey interval relational degree $r(T, X_i)$, $i = 1,2,3,4$, respectively.

$$r(T, X_1) = 0.816 \ ; \ r(T, X_2) = 0.839 \ ; \ r(T, X_3) = 0.945 \ ; \ r(T, X_4) = 0.701$$

4) Rank all the alternatives in accordance with $r(T, X_i)$, $i = 1,2,3,4$.
Since

$$r(T, X_3) > r(T, X_2) > r(T, X_1) > r(T, X_4),$$

so the alternatives are ranked as follows:

$$x_3 \succ x_2 \succ x_1 \succ x_4$$

Therefore, x_3 is the optimal candidate.

4 Conclusions

In this paper, a new model of multi-attribute group decision making is proposed based on the TOWA operator and grey relative degree. According to the results of the application example, we can conclude that this new model has some advantages, for example, the decision principle of this new model is scientific and feasible, and the computation process is simple. Therefore, it has a great theoretical value and application value in practice.

Acknowledgements. This project was supported by the Excellent Youth Project of Hubei Provincial Department of Education (No. Q20082703, Q20082705), and the Significant Project Grant of Hubei Provincial Department of Education (No. 20082704).

References

1. Hwang, C., Lin, M.: Group Decision Making under Multiple Criteria: Methods and Applications. Springer, Berlin (1987)
2. Xu, Z.S.: Uncertain Multiple Attribute Decision Making: Methods and Applications. Tsinghua University Press, Beijing (2004)
3. Wang, J., Ren, J.: Approach to Group Decision-making with Different Forms of Preference Information. Systems Engineering and Electronics 12, 2057–2059 (2005)
4. Herrera, F., Herrera-Viedma, E., Verdegay, J.L.: A model of Consensus in Group Decision Making Under Linguistic Assessments. Fuzzy sets and Systems 1, 73–87 (1996)
5. Bordogna, G., Fedrizzi, M., Pasi, G.: A Linguistic Modelling of Consensus in Group Decision Making Based on OWA Operators. IEEE Trans on Systems, Man, and Cybernetics 27, 126–132 (1997)
6. Herrera, F., Martínez, L.: A Fusion Method for Managing Multi-granularity Linguistic Terms Sets in Decision Making. Fuzzy Sets and Systems 114, 43–58 (2000)
7. Herrera, F., Martinez, L.: A Model based on Linguistic Two-tuples for Dealing with Multigranualrity Hierarchical Linguistic Contexts in Multiexpert Decision-making. IEEE Trans on Systems, Man, and Cybernetics 31, 227–233 (2001)
8. Cordón, O.: Linguistic Modeling by Hierarchical Systems of Linguistic Rules. IEEE Trans on Fuzzy Systems 10, 1–19 (2002)
9. Fan, Z., Xiao, S.: An Optimization Method for Integrating Two Kinds of Preference Information in Group Decision Making. Computer & Industrial engineering 46, 329–335 (2004)
10. Liu, B.D.: Uncertainty Theory: An Introduction to its Axiomatic Foundations. Springer, Berlin (2004)
11. Li, R.J.: The Foundation and Application of Fuzzy Multiple Attribute Decision Making. Science Press, Beijing (2002)
12. Deng, J.L.: Grey Forecasting and Grey Decision-making. HUST Press, Wuhan (2002)
13. Rao, C.J., Xiao, X.P.: Grey Matrix Relative Degree Method for Dynamic Hybrid Multi-attribute Decision Making under Rsk. Systems Engineering and Electronics 28, 557–561 (2006)
14. Rao, C.J., Xiao, X.P., Peng, J.: Novel Combinatorial Algorithm for the Problems of Fuzzy Grey Multi-attribute Group Decision Making. Journal of Systems Engineering and Electronics 18, 774–780 (2007)
15. Xu, Z.S.: Method Based on Fuzzy Linguistic Assessments and GIOWA Operator in Multi-attribute Group Decision-making. Journal of System Science and Mathematics 24, 218–224 (2004)

Research on Multi-time Period Production Plan of Supply Chain under Demands Uncertainty

Hali Pang, Yongjun Wei, Shan Wang, and Xiaobin Liu

College of Information Science & Engineering,
Northeastern University, Shenyang 110004, China
panghali@ise.neu.edu.cn

Abstract. Considering capacity constraints of manufacturer, a problem of multi-time period production planning of supply chain under customers' demands uncertainty was studied in this paper. According to characteristics of the problem, an optimal stochastic chance constraint programming model with objective of minimum costs was presented, and a real number matrix encoding and stochastic sampling constraints verifying based PSO algorithm was proposed. A numeric example is performed, and the results illustrate the feasibility and the effectiveness of proposed model and algorithm.

Keywords: Supply chain, Multi-time period production plan, Demands uncertainty, Stochastic chance-constraint programming, PSO algorithm.

1 Introduction

Supply chain is a functional network that connects suppliers, manufacturers, distributors, retailers and end customers into a whole. Production planning is one of key problems of supply chain management, which plays an important role in maximizing profits or minimizing cost of the supply chain. So the problem of production planning of supply chain has been studied extensively.

Shapiro et al. analysed the mode of supply chain management from strategic, tactical and operational level respectively [1]. Garavelli and Okogbaa presented a single time period model of production - distribution planning for single product with objective of maximizing profits, and the genetic algorithm was used to solve the model [2]. Hou Younghang and Chang Yinghua adopted the model proposed in ref. [2], and used the evolutionary programming algorithm to solve the problem [3]. Lanzenauer and Glombik proposed a mix integer programming model for tactical planning problem of supply chain [4]. Vidai and Goetschaalcks proposed a multi-products single time period tactical production planning model of supply chain with objective of maximizing profits [5]. Considering situation of multi-products and multi-time periods, a production- distribution planning model with goal of minimizing costs was presented by Zubair [6]. Jang Y J *et al*. divided entire supply chain network into three two-class sub-network for modelling and solving respectively [7]. Aiming at inventory and distribution management, a resource allocation model is built by Han [8]. In view of

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 1098 – 1106, 2009.

that the operations of supply chain will be influenced by many uncertain factors, recently, more and more researchers have paid their attention to supply chain optimizing under uncertain environment. Escudero and Galindo studied the planning decision problem of manufacturing - distribution system under uncertainty environment [9]. Gupta and Maranas studied medium-term planning problem of multi-echelon supply chain under uncertainty demand, dividing the problem into production stage and logistics stage, a two-stage stochastic programming mathematical model was used to describe the problem and a solving algorithm was presented [10].

The researches existing mostly focus on the problem of single time period production planning. There are only a few literatures to deal with the problem of multi-time period production planing of supply chain. At the same time, almost there is no literature considering the problem of multi-time period production planing of supply chain under uncertainty environment.

Considering a planning horizon that includes multi time periods, this paper focuses on making decision of production and distribution plan of multiple products of supply chain under customers' demand uncertainty with objective of minimizing costs. According to characteristics of the problem, we build stochastic programming mathematical model and proposed a solving approach of real number genome matrix based particle swarm optimization combining with stochastic sampling constraints verifying. A numeric example is performed and the results illustrate that proposed model and algorithm are feasible and effective.

2 The Problem of Multi-time Periods Production Planning of Supply Chain under Demand Uncertainty

2.1 Problem Description and Denotation

Considering a general supply chain network include a manufacturing plant and some customers shown as Fig.1.

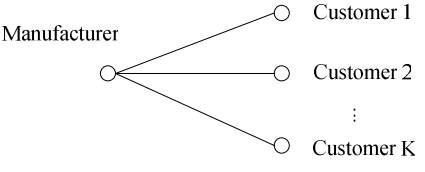


Fig. 1. Supply chain structure

The plant produces N products for K customers. The production of Manufacturer is driven by consumers' orders. The demands of products from customers are uncertainty that will cause products surplus or insufficient and reduce the manufacturer's profit and customers' satisfaction degree. The multi- time periods production planning is to determine the amount of each product produced and delivered according to stochastic demands of customers at each time period under capacity constraints of production

with the objective of minimum costs. So, in order to ensure that optimum of decision in whole planning horizon, it is necessary to combine all time periods to a whole to make production and delivery decisions. For convenient modelling, we definite notations and variables as follow:

Notation: k: the index of customer (k=1,2,…,K); n: the index of products (n=1,2,…,N); t: the index of planning time period (t=1,2,…,T); Cap_t production capacity of manufacturer at t time period; P_n :capacity consumption of producing unit product n; $C_{n,t}$: production cost of producing unit product n at the t time period; θ_n : setup cost of producing product n; $I_{n,0}$: initial storage of product n; $I_{n,t}$: product n storages at t time period; $H_{n,t}$: storage cost of unit product n; $\gamma_{n,k,t}$:unit transportation cost conveying product n to customer k at t time period; $\alpha_{k,n,t}$: confidence coefficient at which stochastic chance constraint that customer k needs product n is contented at time period t; $\tilde{D}_{k,n,t}$: amounts of product n demanded by customer l at time period t (random).

Variables:

$Qm_{n,t}$: Quantity of product n produced at t time period (decision variable);

$Q_{k,n,t}$: Quantity of product n distributed to customer k at t time period (decision variable);

2.2 Stochastic Chance Constrain Programming Model

Considering affection of stochastic factors, suppose that customers' demands are stochastic variables and described as $\tilde{D}_{k,n,t} \sim N_i(\mu_i, \sigma_i^2)$, i = 1, 2, ...,I. Planning horizon is known, and can be divided into some equal time periods; the costs in each time period are known and stable, and without regard to the safe inventory and lead time of ordering; the stochastic chance constrain programming mode is built as follows:

$$\min C = \sum_t \sum_n (C_{n,t} \cdot Qm_{n,t} + \theta_n) + \sum_t \sum_k \sum_n Q_{k,n,t} \cdot \gamma_{k,n,t} + \sum_t \sum_n I_{n,t} \cdot H_{n,t} + \sum_t \sum_k \sum_n E\left[\tilde{D}_{k,n,t} - Q_{k,n,t}\right]^+ \cdot \alpha_{k,n,t} \quad (1)$$

s.t.

$$\sum_n p_n \cdot \sum_k Qm_{n,t} \le Cap_t \qquad t=1,2,\cdots,T \quad (2)$$

$$I_{n,t-1} + Qm_{n,t} - \sum_k Q_{k,n,t} = I_{n,t} \quad t=1,2,\cdots,T$$

$$n=1,2,\cdots\cdots,N \quad (3)$$

$$\Pr(\tilde{D}_{k,n,t} - Q_{k,n,t} \le 0) \ge \alpha_{k,n,t} \qquad \text{n=1,2,……,N ;}$$

$$\text{l=1,2,……,L} \quad (4)$$

$$[x]^{+} = \begin{cases} x, & if\ x > 0, \\ 0, & otherwise, \end{cases} \tag{5}$$

$$Qm_{n,t} \geq 0 \ ; \ Q_{k,n,t} \geq 0 \tag{6}$$

The objective of the model is minimum total cost of supply chain, and the first part is the variable cost and set up cost of manufacturing; the second part is storage cost of products; the third part is punishment when products cannot satisfy customers' demands. Constraint (2) describe that the amount of products produced should be within the manufacturer's capacity; constraint (3) is the quantity balance of manufacturing, delivering and storage; constraint (4) indicates that the probability of customer's demand for each product satisfied may keep above confidence coefficient $\alpha_{k,n,t}$ in time period *t*; In the constraint (3), if $\tilde{D}_{k,n,t} \geq I_{n,t-1} + Qm_{n,t}$, then $\tilde{D}_{k,n,t} = Q_{k,n,t}$ and $I_{n,t} \geq 0$, manufacturer will hold the storage of products; if $\tilde{D}_{k,n,t} \geq I_{n,t-1} + Qm_{n,t}$, then $I_{n,t} = 0$, namely manufacturer can not satisfy the demands of customer.

3 Solving Method

It is very difficult to solve stochastic chance constraint programming model by using traditional mathematic method. Particle Swarm Optimization is an intelligent optimization algorithm which emerges in recent years, and used in solving many optimization problems. In view of intractable characteristic of the model presented above, we adopt Particle Swarm Optimization algorithm combining with stochastic sampling constraints verifying to solve the model. The overall solving process is as follows:

(1) Setting the algorithm parameters: swarm size (swarm-size), learning factor, maximum speed, inertia weight, maximum number of evolution generations (Generations);
(2) Creating particle individuals randomly for initializing the swarm and checking feasibility;
(3) Adopting stochastic sampling technology to calculate the fitness value of all particles in the swarm;
(4) Update speed and position of particles;
(5) Checking the termination condition is satisfied or not, if not satisfied, goes to step (3); else, finish calculation, output global historical optimal solution.

In above solving process, we should design corresponding procedures according to characteristics of the problem.

3.1 Individual Presentation and Swarm Initializing

Considering the relation between time periods, the amount of products produced and delivered in different time periods must be determined according to the probability distribution of customers demand. We use a real number encode based matrix that

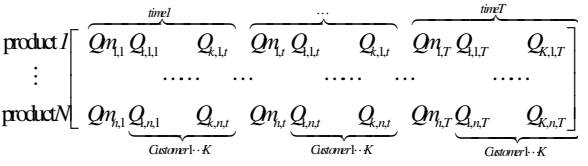


Fig. 2. Solution structure

denotes decision variable among multi-time periods production planning. The feasible solution structure can be represented as Fig 2.

The rows of matrix denote the different products; the columns of matrix denote the amount of products produced and delivered in each time period.

The initial swarm is generated stochastically according to the probability distribution of the customers' demands within the capacity of production.

3.2 Stochastic Chance Constraints Verifying and Objective Function Calculating

The key of using PSO algorithm to solve constraint optimal problem is to find suitable encoding way to represent solution of the problem, to deal with constraints effectively, and to guarantee the rationality and validity of swarm in evolution process.

Regarding the presented model, constraint (4) is a stochastic chance constraint concerning satisfaction degree of customer. It is necessary to verify in solving process to guarantee to satisfy customers' demands. We adopt stochastic sampling technology to examine the constraint. For stochastic chance constraint $\Pr(\tilde{D}_{k,n,t} - Q_{k,n,t} \le 0) \ge \alpha_{k,n,t}$ here, $\tilde{D}_{k,n,t}$ is a stochastic vector, whose cumulative probability distribution is $\Phi(\tilde{D}_{k,n,t})$. For given decision variable $Q_{k,n,t}$, M independent stochastic variables $\tilde{D}_{k,n,t}$, can be generated according to probability distribution $\Phi(\tilde{D}_{k,n,t})$, supposing that M' is the times that inequality $\tilde{D}_{k,n,t} - Q_{k,n,t} \le 0$ is tenable in M samples, the frequency M'/M is used to estimate the probability. Therefore, only when $M'/M \ge \alpha_{k,n,t}$, stochastic chance constraint is tenable.

Because there is a stochastic variable $\tilde{D}_{k,n,t}$ in objective function, the value of objective function can also be calculated through computing the mathematic expectation of stochastic variables $\tilde{D}_{k,n,t}$ generated according to the probability distribution $\Phi(\tilde{D}_{k,n,t})$.

As the problem studied is to find minimizing cost, the objective function must be translated to fitness function. The fitness function of individual particles and the swarm are as follow:

$$F_i(x) = \begin{cases} C_{\max} - f_i(x) & f_i(s) < C_{\max} \\ 0 & f_i(x) \ge C_{\max} \end{cases} \tag{6}$$

$$F(x) = \max\{f_i(x)\} \tag{7}$$

Here, $F_i(x)$ and $f_i(x)$ are fitness and objective value of individual particle respectively, C_{max} is the maximum value of the objective among current all particles; $F(x)$ is the fitness of the swarm.

4 Computation Example

Considering a supply chain including three customers and one manufacturing plant, the plant may produce 3 kinds of products. Demands of customers are stochastic, and the probability distributions are known in each time period. The problem is to determine the quantity of products produced and delivered that can satisfy the demands of customers, and make overall cost minimum. Table 1-2 give all data known.

Table 1. Production capacity and products demand distribution

Time period		1	2	3	4
Production capacity		500	450	460	430
Products demand distribution					
Customer1	Product1	*N*(40,10)	*N*(30,9)	*N*(35,9)	*N*(45,10)
	Product2	*N*(50,10)	*N*(40,9)	*N*(45,10)	*N*(47,8)
	Product3	*N*(30,5)	*N*(25,8)	*N*(35,10)	*N*(30,10)
Customer2	Product1	*N*(25,5)	*N*(20,9)	*N*(30,10)	*N*(20,10)
	Product2	*N*(40,8)	*N*(50,15)	*N*(40,10)	*N*(55,8)
	Product3	*N*(30,5)	*N*(20,9)	*N*(35,8)	*N*(25,5)
Customer3	Product1	*N*(35,9)	*N*(30,5)	*N*(38,8)	*N*(40,10)
	Product2	*N*(40,10)	*N*(35,8)	*N*(45,9)	*N*(50,20)
	Product3	*N*(30,10)	*N*(30,5)	*N*(35,10)	*N*(30,9)

Table 2. Cost of production, inventory and transportation

Product	1	2	3
Setup cost	300	400	200
Unit cost of production	7.22	5.24	8.32
Inventory cost	2.89	2.10	3.33
Unit capacity consuming	1	2	1
Transportation cost			
Customer1	1.89	2.53	2.14
Customer2	1.10	1.38	1.47
Customer3	2.33	1.31	2.06

Because customers' demands are uncertain, the quantity of products produced and delivered may not satisfy the customers' demands some time, and will result in lacking loss of products. The satisfaction of customer k can be defined as $Q_{n,k,t} \big/ E\left[\tilde{D}_{n,k,t}\right]$, the penalty of lacking products can be measured by the satisfaction degree of the customers. Supposing the penalties are the same in each time period, the penalty function can be given as Fig 3.

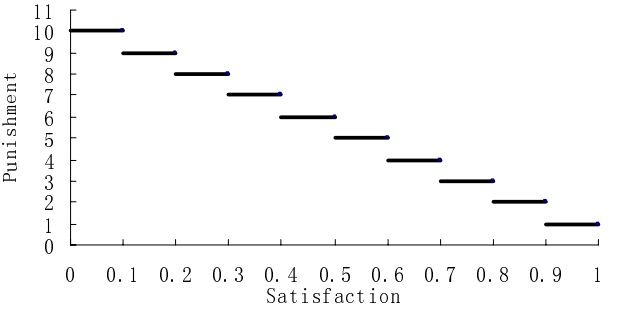


Fig. 3. Penalty function

The algorithm is programmed with VC++6.0 and appropriate algorithm parameters could be determined through computing experiments of setting different parameters and comparing the computation results. Fig. 4 gives the convergence trends of the algorithm under three kinds of different parameters when confident level $\alpha_{k,n,t}$ is 0.9. The suitable algorithm parameters are swarm size=100, earning factors=2.0, $V_{max} = 2.0$. The evolution of the maximum generation is 500, and whole process of the running time is approximately 15seconds, the results are given in Table 3 and Table 4.

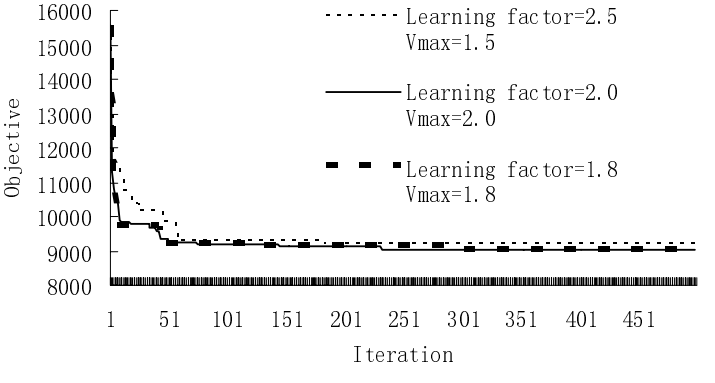


Fig. 4. Convergent trends of the algorithm

Table 3. Throughput of each time period

Product	Period 1	Period 2	Period 3	Period 4
1	109.7	93.3	103.4	98.0
2	138.2	123.6	134.1	126.9
3	96.0	79.8	92.3	83.4

Table 4. Distribute quantity of each customer

	Product	Period 1	Period 2	Period 3	Period 4
Customer1	1	43.32	37.65	40.27	36.77
	2	50.83	46.89	51.33	50.58
	3	32.25	24.31	31.66	28.41
Customer 2	1	27.88	22.66	26.35	25.68
	2	40.99	36.51	41.78	37.77
	3	29.59	25.29	30.35	29.34
Customer 3	1	36.70	30.24	35.59	32.90
	2	36.76	37.57	41.33	38.23
	3	30.17	26.93	29.22	27.02

Considering the uncertain demands of customers, we use the results (products produced and delivered in each time period) to verifying the satisfaction degree of customer, and give the results in Table 5. It is obviously that the satisfaction degrees of customers are more than 94% in each time period.

Table 5. Customer satisfaction

	Product	Period 1	Period 2	Period 3	Period 4
Customer 1	1	98%	95%	98%	94.5%
	2	99%	94%	98%	98%
	3	98%	97.5%	99%	94%
Customer 2	1	99%	92%	98%	98%
	2	98%	92.5%	99%	95%
	3	99%	94%	99%	98%
Customer 3	1	98%	92%	97%	95.5%
	2	99%	95%	98%	94%
	3	98%	93%	96%	96%

5 Conclusion

This paper studied multi-time periods production planning problem of supply chain under demands uncertainty, a stochastic chance constraint programming model with objective of minimizing costs was built, according to the characteristics of the model, a

solving approach combining Particle Swarm Optimization algorithm with stochastic sampling constraint verifying was presented. A numeric example was performed and the computation results illustrate the feasibility and the effectiveness of proposed model and algorithm.

References

1. Shapiro, J.: Bottom-up Versus Top-down Approaches to Supply Chain Modelling in: Quantitative Models for Supply Chain Management, pp. 739–759. Kluwer Academic Publishers, Dordrecht (1998)
2. Garavelli, A.C., Okogbaa, O.G., Violante, N.: Global Manufacturing System: a Model Supported by Genetic Algorithms to Optimize Production Planning. Computers & Industrial Engineering 31, 193–196 (1996)
3. Hou, Y.H., Chang, Y.H.: The New Efficient Hierarchy Combination Encoding Method of Evolution Strategies for Production Allocation Problems. Computers & Industrial Engineering 43, 577–589 (2002)
4. Lanzenauer, C.H., Glombik, P.K.: Coordinating Supply Chain Decisions: an Optimization Mode. OR Spectrum 24, 59–78 (2002)
5. Vidai, C.J., Goetschalcks, M.A.: Global Supply Chain Model with Transfer Pricing and Transportation Cost Allocation. European Journal of Operational Research 129, 134–158 (2001)
6. Zubair, M.M.: An Integrated Production-distribution Model for a Multi-national Company Operating under Varying Exchange Rates. International Journal of Production Economies 57, 81–92 (1999)
7. Jang, Y.J., Jang, S.Y., Chang, B.M.: A Combined Model of Network Design and Production /Distribution Planning for a Supply Network. Computers & Industrial Engineering 43, 263–289 (2002)
8. Han, C.: Stochastic Modeling of a Two-echelon Multiple Sourcing Supply Chain System with Genetic Aorithm. Journal of Manufacturing Technology Management 16, 87–108 (2005)
9. Escudero, L.F., Schumann, E.G.: A Modeling Framework for Supply Chain Management under Uncertainty. European Journal of Operations 119, 14–34 (1999)
10. Gupta, A., Maranas, C.D.: Management Demand Uncertainty in Supply Chain Planning. Computers & Chemical Engineering 27, 1219–1227 (2003)

Research on Coordination Model of Multiplexed System Based on DEA

Zhengping He, Hongtao Zhou, Wei Zeng, Yushuo Chen, and Qi Fei

Institute of Systems Engineering, Huazhong University of Science & Technology, Wuhan 430074, China
{Hongtao Zhou,zht730}@yahoo.com.cn

Abstract. First, based on the viewpoint of system theory, this paper explained the coordination and its degree's concept and connotation. Then it introduced the performance of Data Envelopment Analysis (DEA) method and the membership concept of fuzzy mathematics degree, which established the coordination degree evaluation model of multiplexed system, appraised the running status and the tendency of the coordination among the multiplexed system's essential factors and subsystems. Finally we took an empirical analysis with this model into Chinese Guangxi's multiplexed system which concludes economy, education and Science Technical between 1994 and 2006. The result indicated: this multiplexed system's static coordination degree changed in a big way, and the dynamic coordination degree swings nearby 0.92, the multiplexed system development is at a basic coordinated condition.

Keywords: Multiplexed System, Coordination Degree, Data Envelopment Analysis（DEA).

1 Introduction

Coordination is used in a very widespread concept in the scholar research and the application, but actually does not have a very explicit definition. The system coordination's basic philosophy is, by organizing and regulating the system through some methods, to seek for the solution of contradictious or conflicted plan, and finally causes the system to transform from the disorder into the order, achieves a better coordination degree or a harmonious condition. The system's coordinated goal is to reduce its negative effect, enhances its whole output function and overall effect [1].

The multiplexed system's coordinated essence is to make a full use of and to promote the positive relations of it. Coordination could manifest the condition of harmony, coordination, and the optimized relations of the various subsystems or the essential factors among the subsystems. The development performance could be regarded as a complicated system's process of movement [2]. To realize multiplexed system's coordination, it must regulate relations between various subsystems, including many kinds of mechanisms such as system structure's coordination, the interior, exterior coordination as well as the organization management coordination and so on [3]. In multiplexed system, each subsystem is independent, but also

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 1107 – 1114, 2009.

interacted, multiplexed system's coordinated operating condition is the coordinated development among various subsystems. Therefore, coordination is not only one adjustment method, is also one kind of management and control function, sometimes also presents one condition to indicate the harmonious relations that between various subsystems, essential elements, system functions, the structures or the goals, using for describing system's overall effect.

2 Coordination Model Research of Multiplexed System

2.1 Statement of the Appraisal System Method

Because the coordinated developing system is a huge and complex system, and composed of many production subsystems, therefore, when appraised coordinated developing condition, the coordination coefficient that should be calculated is not one, but more. They connect and restrict mutually, constituting a system, these coordinated coefficients could be called coordination coefficient system. We give the following description regarding this coordinated coefficient system:

$w(i/j)$: the coordinated coefficient from system i to system j, which means the close degree of what the output system i requests ,when takes system i as the input, and system j as the output;

$w(i/j,k)$: the coordinated coefficient from system i to system j and system k, which means the close degree of what the output system i requests ,when takes system i as the input, and system j,k as the output;

$w(i/j,k,l)$: the coordinated coefficient from system i to system j,k,l, which means the close degree of what the output system i requests ,when takes system i as the input, and system j,k,l as the output.

Although $w(i/j)$ and $w(j/i)$ could be calculated, but they only describe the coordinated degree from one system to another system, can not be able to reflect the coordinated degrees between the two systems. Coordinated degrees between two systems should be reflected by the formula $w(i,j)$:

$$w(i,j)=\frac{Min\{w(i/j),w(j/i)\}}{Max\{w(i/j),w(j/i)\}} \tag{1}$$

Likewise, there needs to calculate the coordinated degree among three system. Its formula is:

$$W_{(i,j,k)}=\frac{W_{(i/j,k)}*W_{(j,k)}+W_{(j/i,k)}*W_{(i,k)}+W_{(k/i,j)}*W_{(i,j)}}{W_{(i,k)}+W_{(j,k)}+W_{(i,j)}} \tag{2}$$

Inference in turn, we may describe coordinated degree formulas among four, five, six systems as $w(i,j,k,l)$, $w(i,j,k,l,m)$, $w(i,j,k,l,m,n)$.

By above, the coordinated degree function $w(i, j, k...)$ reflected the coordinated developing system's synthesis coordinated coefficient that among many systems. its expression is the highest level. It reflects the overall system's coordinated developing degree comprehensively. The coordinated coefficient architecture relations, not can only describe the coordinated development degree between various levels' system, but may also carry on the factor analysis, discover the reason of uncoordinated, and provide the basis for the formulation coordinated development strategy.

2.2 Coordination Model of Multiplexed System Based on DEA

Take long-term balanced relational various subsystems of a system as the input-output system mutually. With the method DEA from the macroscopic angle, it narrates and comments the coordinated development among the various subsystems, and through the definition to judge coordinated degree between subsystems's, drawing support in the concept of membership of the fuzzy mathematics [4].

Refers to massive methods and predecessor's work, there may establish a set of scientific coordinated coefficient computation system. First there need to calculate the coordination degree between each two subsystems, then calculate coordination degrees between an element and the other two elements in turn, finally carries on the synthesis according to the formula, then there could obtain the coordination degree of three subsystems, and coordination degree computation among the multi-system's also depend on the above analogy.

According to DEA and the concept of membership of the fuzzy mathematic, the coordination degree's computation step between two subsystems is:

The 1st step: Takes a subsystem's (marked s_1) index as the input, and another subsystem's (marked s_2) index as the output. There can obtain the input surplus of s_1, the output debt of s_2, the scale benefit value $\sum \lambda$ and the valid value (w_1) of the DEA, using the CCR model of the DEA method.

The 2nd step: Takes a subsystem's (marked as s_2) index as the input, and another subsystem's (marked as s_2) index as the output. There can obtain the input surplus of s_2, the output debt of s_1, the scale benefit value $\sum \lambda$ and the valid value (w_2) of the DEA, using the CCR model of the DEA method.

The 3rd step: Applying the ideology of the concept of membership of the fuzzy mathematics, which could establish coordination function of the development condition between two subsystems, to express the subordination degree of system to the coordination which is a fuzzy concept under certain value? To definite the function of membership as $w(\theta) = \theta$, θ represents the effective value mutually between input and output of the subsystems. Mark the static coordination from s_1 to s_2 with w_1, which means the close degree of the development from the subsystem s_2 to the requested harmonious condition that subsystem s_1 asks s_2 for; In addition,

we mark the static coordination from s_2 to s_1 with w_1, which means the close degree of the development from the subsystem s_1 to the requested harmonious condition that subsystem s_2 asks s_1 for. Then we could define the static coordination degree between the two subsystems as $w_{1,2}$ or $w_{2,1}$:

$$w_{1,2} = \frac{\min\{w_1, w_2\}}{\max\{w_1, w_2\}} \tag{3}$$

The 4th step: How to obtain the dynamic coordination degree. Suppose the $w(\theta_1, t_1)$, $w(\theta_2, t_2)$,..., $w(\theta_n, t_n)$ are the coordination degrees each time between the time section of the multiplexed system from t_1 to t_n, the dynamic coordination degree's function:

$$w(\theta_1 ... \theta_n; t_1 ... t_n) = \frac{1}{t_n - t_1 + 1} \sum_{i=1}^{n} w(\theta_i, t_i) \tag{4}$$

$w(\theta_1 ... \theta_n; t_1 ... t_n)$ represents the dynamic coordination degree of the multiplexed system from $t_{0,}$ to t_n ,and it could simplified as $w(t_{0,} t_n)$ (t_0 is the base time).

According to the synthesis formula computation, we could get the coordination degree function among three subsystems:

$$W_{(i,j,k)} = \frac{W_{(i/j,k)} * W_{(j,k)} + W_{(j/i,k)} * W_{(i,k)} + W_{(k/i,j)} * W_{(i,j)}}{W_{(i,k)} + W_{(j,k)} + W_{(i,j)}} \tag{5}$$

Among them,

$W_{(i,j,k)}$ represents the coordination degree among three subsystems;

$W_{(i/j,k)}$ represents the coordination degree between one subsystems and the other two;

$W_{(j,k)}$ represents the coordination degree between two subsystems.

When there are more than three subsystems, the coordination degree which marked with $w(i, j, k...)$ among them also could be reasoned out by the above.

3 Computation of Coordination Degree of Multiplexed System

According to the requisition of the appraisal of coordination degree, the evaluation index system has been designed as follows [5, 6].

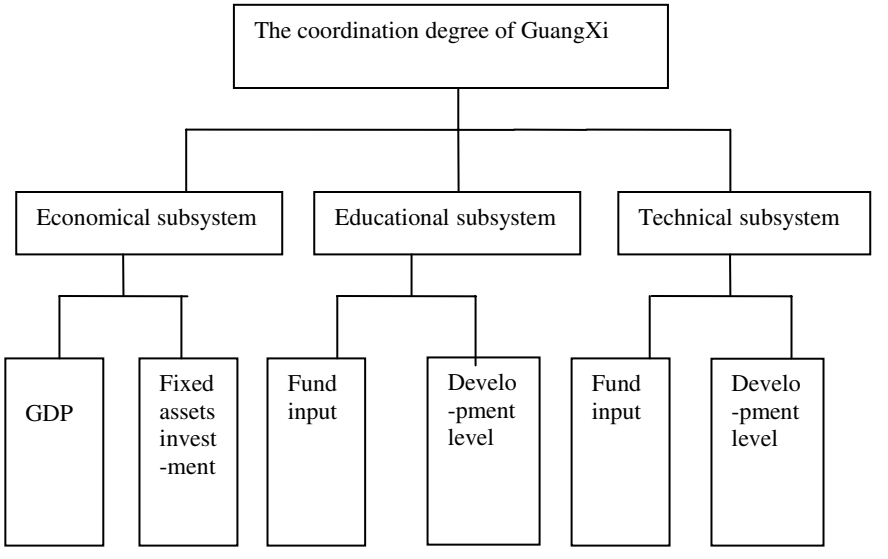


Fig. 1. Index structure for the coordination degree model of GuangXi

The following index data are from the almanac of GuangXi which are showed in the following table 1. The index of development level of education and technique had been converted more intuitive, and we passed the expatiator detail over.

Table 1. Index data from the almanac of GuangXi

years	Economical subsystem		Educational subsystem		Technical subsystem	
	GDP (hundred-million)	Fixedassets investment (hundred-million)	Fund input (ten-million)	Develop-ment level	Fund input (ten-million)	Develop-ment level
1994	1198	382	104	1205	27.37	643
1995	1498	423	127	1253	30.28	738
1996	1697	476	137	1391	31.58	722
1997	1817	479	96	1533	32.47	719
1998	1911	571	114	1411	36.07	801
1999	1971	620	111	1565	40.87	805
2000	2080	660	166	1801	48.79	784
2001	2279	731	325	1647	67.18	767
2002	2523	835	264	1664	79.35	953

Table 1. (*continued*)

years	Economical subsystem		Educational subsystem		Technical subsystem	
	GDP (hundred-million)	Fixedassets investment (hundred-million)	Fund input (ten-million)	Develop-ment level	Fund input (ten-million)	Develop-ment level
2003	2821	987	374	1962	88.04	1119
2004	3433	1263	418	1979	98.79	1157
2005	4075	1769	527	2149	114.68	1434
2006	4828	2246	502	2529	146.00	1624

We chose the index which could represent the output and input of each subsystem best. And because there is no need to consider the units when use the method DEA, so we took the original data ,except for the development level of Technic and Education,in order to reflect the condition of each year in GuangXi more direct. Take the advantage of our model; we obtained our result as following table show.

Table 2. Static coordination degree of multiplexed system of GuangXi

year	1994	1995	1996	1997	1998	1999	2000	2001	2002
DEA	0.93	0.96	0.93	1	0.9	0.92	0.92	0.88	0.88
year	2003	2004	2005	2006					
DEA	0.9	0.96	0.97	0.88					

Table 3. Dynamic coordination degree of multiplexed system of GuangXi

year	1994	1995	1996	1997	1998	1999	2000	2001	2002
DEA	0.93	0.945	0.94	0.955	0.944	0.94	0.937	0.93	0.924
year	2003	2004	2005	2006					
DEA	0.922	0.925	0.929	0.925					

Through the curves of the static and dynamic coordination degree among three subsystems, the multiplexed system of GuangXi was basically coordinated from 1994 to 2006, which could be reflected by the dynamic coordination values that were all above 0.92.And except for the year 2001, 2002 and 2006, the static coordination values were all above 0.9. First, we get our conclusion by analyzing the three years whose static coordination values are below 0.9(2001, 2002, 2006) from the coordination values between each two subsystems or between one and the other two. The biggest transformation of 2001 is the increase which reached almost 100% of the

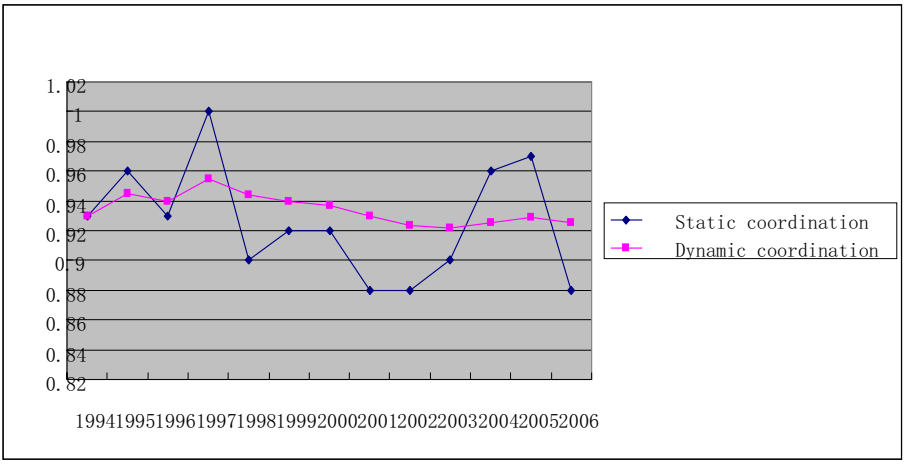


Fig. 2. Change tendency of the coordination degree of GuangXi

index of technical fund input, at the same time, the population of Science and Technique also get a fast increase ,and the development of the economic and education are comparatively smooth, that is the reason which make 2001's coordination value a smaller value. And in 2002, while the education and economic is at a fast development, there has been a significant negative growth of the index of science and technique, which may be caused by the government's Macro-control. In 2006, the increase of the index of the S&T slow down again while the other two index are at a sustained growth, which makes the coordination value smaller, in particular to reduce the fund for S&T .

From the coordination values of 1997, which is1, we found there is an obvious increase compared to the 0.93 of 1996.In 1997, the economic index growing speed, although compared the years to be slow, is quick compared to the education and S&T. So this indicated that the speed of development of S&T and education is in advance compared to economy before 1996.

In 2004, 2005 two years, the coordination values have amounted to the mean value which are the continuously highest two years. Three subsystems' rates of rise are all very quick, which showed that these rates of rise for each of them are quite ideal, we may take these three sub-system's growth rates of these two years to be the ideal reference value.

We could observe from the dynamic coordination degree curve that the values are better in the years before 2000 than after 2000, the main causes were the higher education reform around 2000 and the funds fluctuation of science and technology each year. But the impact of the higher education reform, which reduced the coordination value, was temporary. We could foresee that the education will cause the other two subsystems to promote to make the multiplexed system of GuangXi change for the better. The hysteretic quality could be observed by the growth of coordination values of 2004 and 2005.

In brief, it could be discovered that from 1994 to 2006, the main cause for the years which have relatively small values of coordination degree is because of the fluctuation of index of S&T, and the rates of growth were quite ideal in 2004 and 2005.

4 Summaries

From the viewpoint of interaction and mutual connection of multiplexed system, we have established coordination degree evaluation model using the DEA method and the concept of membership of the fuzzy mathematics .The model used statistical data from time series and was based on the multiplexed system whose subsystems accord with cointegration, in order to appraise the running status and the tendency of the multiplexed system among the subsystems or among the essential factors. Through the empirical analysis to the multiplexed system including Economy, Education and S&T of Guangxi from 1994 to 2006, static coordination degree of the multiplexed system changed in a big way, but the dynamic coordination degree swinged nearby 0.92, so the multiplexed system is at a quite coordinated condition. Multiplexed system's coordinated development is the general objective which caused the social economy development sustainable; we have to take the scientific development concept as the instruction to plan the development of economy, education, and S&T in order to promote the ability for sustainable fast development.

References

1. Engle, R.F., Granger, W.J.: Cointegration and Error Correction: Representation, Estimation and Testing. Econometrica 55, 251–276 (1987)
2. Johansen, S., Juselies, K.: Maximum Likelihood Estimation and Inference on Cointegration with Application to the Demand for Money. Oxford of Bulletin Economics and Statistics 52, 169–210 (1990)
3. Johansen, S.: Estimation and Hypothesis Testing of Cointegration Vectors in Gaussian Vector Autoregressive Models. Econometrica 59, 1551–1580 (1991)
4. Wang, Z.P., Han, L.Y.: Application of Fuzzy Mathematics. Beijing Economic College Press (1989)
5. Lau, L., Jamison, D., Liu, S.: Education and Economic Growth: Some Cross-sectional Evidence from Brazil. Journal of Development Economics 41, 45–70 (1993)
6. Lin, T.: Education, Technical Progress, and Economic Growth: the Case of Taiwan. Economics of Education Review 22, 213–220 (2003)

Co-integration Analysis of Multiplexed System Based on VAR

Zhengping He, Hongtao Zhou, Wei Zeng, Jiang Jiang, and Qi Fei

Institute of Systems Engineering, Huazhong University of Science & Technology, Wuhan 430074, China
{Hongtao Zhou,zht730}@yahoo.com.cn

Abstract. The variation of elements in a compound system is mostly non-stationary. Whether these element variables have long-term mutual connection or balanced relations, is essential to determine the interaction relationships between the subsystems. Co-integration is usually used to describing the long-term balanced relations between these non-stationary variables in a statistical method. In view of the fact that elements of a compound system is mostly non-stationary, this paper gave a method using co-integration analysis to determine the long-term inter-relationship between elements of the compound system or its subsystems, and carried this method on the co-integration analysis of the Guangxi Province compound system, composed of education, economy science and technology system, the result of which indicated that economy growth was the reason of the development of education and technological achievements in Guangxi from 1994 to 2006.

Keywords: Vector Auto-Regressive, Multiplexed System, Co-integration Analysis.

1 Introduction

After having been reforming and opening up to the outside world for near 30 years since1978, China's comprehensive national strength has been being greatly enhanced. However, China's development is mainly depending on the massive resources consumption and the cheap labor force. This kind of extensive growth style makes China face lots of serious problems, such as the depletion of resources, Environmental Pollution, no remarkable improvement of people's living standard and so on. Therefore The Party Central Committee puts forward a series guidelines, for instance, “the economy development must depend upon the science and technology, the technical work must face to the economy development” as well as “the education must serve the socialist construction, the socialist construction must depend upon the education”, to transform our economy developing style and form a virtuous circle formation that science and technology innovation promotes the economy development, the economy development improves the educational level enhances, education lays the foundation for the science and technology innovation. So it is very important to study the internal relations between the education - economy – science &technology compound system and find out their combinative point in order to promote their coordinated development.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 1115 – 1122, 2009.

The multivariate co-integration method was applied in this paper to find out the long-term relationship between the education - economy - science and technology compound system. So we chose the development indexes of the GDP (g), the number of graduates (e), scientific and technological level (t) as the parameters to measure the level of education, economy and science and technology.

2 Evaluation to Interrelation of Sub-systems

The variation of elements in a compound system is mostly non-stationary. Whether these element variables have the long-term mutual connection or balanced relations, is essential to determine the interaction and relationship between the sub-systems. Co-integration is usually used to describing the long-term balanced relations between these non-stationary variables in a statistical method [1-3]. As the multi-variable VAR (Vector Auto-Regressive, VAR) model has the higher reliability than a single equation, and the elements in a compound system is mostly non-stationary, so we chose to determine the relationship between these elements in a multi-variable VAR system, in which a k-order VAR model can be expressed as:

$$\begin{aligned} Y_t &= \sum_{i=1}^{k} \Pi_i Y_{t-i} + U_t \\ &= \Pi_1 Y_{t-1} + \Pi_2 Y_{t-2} + ... + \Pi_k Y_{t-k} + U_t \end{aligned} \tag{1}$$

$$U_t \sim \Pi D(0, \Omega)$$

In this formula, $Y_t = (Y_{1,t}, Y_{2,t}, ..., Y_{N,t})$ and $\Pi_1, \Pi_2, ..., \Pi_k$ are parameter matrixes, U_t is a random error column vector and Ω is a covariance matrix.

If there are some co-integration relationships between the non-stationary variables in the VAR model, we can establish a vector error correction mode (VECM) based on the VAR model by co-integration.

$$\Delta Y_t = \Gamma_1 \Delta Y_{t-1} + \Gamma_2 \Delta Y_{t-2} + ... + \Gamma_{k-1} \Delta Y_{t-k+1} + \Pi Y_{t-k} + U_t \tag{2}$$

In which Π is influence matrix (or compression matrix), equaling to the sum of all the parameter matrixes subtracted a unit matrix or being interpreted as follow formula:

$$\Pi = \Pi_1 + \Pi_2 + \Pi_3 + ... + \Pi_k - I \tag{3}$$

And Π can also be decomposed into $\Pi = \alpha\beta^T$, in which β is co-integration coefficient matrix and all of its columns are co-integration vectors, and α is adjusting coefficient matrix. We can choose the optimum lag period "k" by evaluating the joint significance of variables' largest lag period.

3 Co-integration Analysis Theory of Multiplexed System

3.1 Unit Root Test

An application of stationary test to the variables is required before the multivariate co-integration analysis. Only all the variables in the t-order I(t) are stationary, can we carry on the co-integration analysis.

Assumed that the time serials { y_t } can be expressed as:

$$y_t = \alpha + \beta t + \rho y_{t-1} + \sum_{i=1}^{P} \lambda_i \Delta y_{t-i} + \xi_t \tag{4}$$

In which the lag length "p" should be large enough generally to make sure that there is no serial correlation between these error term or could be determined by the minimum AIC or Schwartz(SC) values, thus we can firstly estimate the following unrestricted regression equation based on OLS estimator:

$$\Delta y_t = \alpha + \beta t + (\rho - 1) y_{t-1} + \sum_{i=1}^{P} \lambda_i \Delta y_{t-i} + \xi_t \tag{5}$$

And then obtain the OLS estimator of the following restricted regression equation, which constrained by the hypothesis of unit root $H_0 : \beta = \rho - 1 = 0$:

$$\Delta y_t = \alpha + \sum_{i=1}^{P} \lambda_i \Delta y_{t-i} + \xi_t \tag{6}$$

Finally, according to the results of regression analysis, we can calculate the F-value to test whether y_t is statistically significant tenable or not.

3.2 Co-integration Test

Co-integration was firstly built by Granger in 1981 just as a conceptive tentative idea, then extended by Engle and Granger, who gave rigorous theorem proving and concrete operational framework in 1987. And in the following 10 years, due to the remarkable works of Engle-Yoo, Johansen and Phillips et al., the co-integration models have become an effective and indispensable instrument for the analysis of non-stationary time-series and have been widely applied in practice.

Co-integration theory, simply speaking, involves a group of variables, which are non-stationary separately and can drift when being combined together. These kind of drift shows that there is a linear relationship between these variables and can be regarded as a way to research the balanced relations of the variables in economy. Its definition could be interpreted as: a set of time-series vector has such property that all of the time series in this vector are integrations and a certain linear combination of

these time series is a stationary sequence. Besides, this concept can also be expressed in mathematical language as: if time series $X_{1t}, X_{2,t},..., X_{kt}$ are all d-order integrations and there exists a vector $\alpha = (\alpha_1, \alpha_2,..., \alpha_k)$ such that $Z_t = \alpha X_t \sim I(d-b)$ where b>0, $X_t = (X_{1t}, X_{2t},..., X_{kt})$, then this time series $X_{1t}, X_{2,t},..., X_{kt}$ could be considered as (d,b) co-integration, or denoted as $X_t \sim CI(d,b)$, and α is co-integration vector.

3.3 Granger Causality Test

It is a common problem to determine whether one time-series is the cause to another one in social economics. So Granger first put forward a set of tests to reveal something about causality, which is known as Granger Causality Test.

A time series X is said to Granger-cause Y if it can be shown, usually through a series of F-tests on lag length of X (and with lag length of Y also known), that those X values provide statistically significant information about future values of Y.

This work begins by estimating the following two regression models for testing the hypotheses that whether X Granger-cause Y or not:

$$Y = \sum_{i=1}^{p} \alpha_i Y_{t-i} + \sum_{j=1}^{p} \beta_j X_{t-i} + \xi_t \tag{7}$$

The model without restrictions:

$$Y = \sum_{i=1}^{p} \alpha_i Y_{t-i} + \xi_t \tag{8}$$

The model with restrictions:

In which p is the optimum lag length, and then make OLS regression toward these two models for unrestricted sum of squares of deviations RSS_1 and restricted sum of squares of deviations RSS_2. So we can calculate the F-statistics, which obeys a $F(Q, N-K)$ distribution, by using the following formula:

$$F = (N-K)\frac{(RSS_1 - RSS_2)}{QRSS_1} \tag{9}$$

Where N is the sample capacity, K is the number of the estimated parameters in the unrestricted model, and Q is the number of the parameters to be estimated in the unrestricted model. Followed, we should examine whether this set of coefficients $(\beta_1, \beta_2,..., \beta_p)$ significantly equals to zero simultaneity. If really so, it should reject the null hypothesis that X does not Granger causes Y.

Then the null hypothesis－X does not Granger cause Y－can be tested in the some way, but it has to exchange the position of X and Y and test if the lag item of Y significantly equals to zero. For obtaining the conclusion that X brings about the variation of Y, the null hypothesis－X does not Granger cause Y should be rejected, whereas accepting the opposite hypothesis.

4 Co-integration Analysis on the Economy-Education-Technology Multiplexed System in Guangxi Province

4.1 Data Source

There are differences in range between the development of education, the growth of economic and the advance in technology, but as a whole, they have close relationship [4-6]. In order to investigate whether there are the long-term mutual connection or balanced relations in the compound system of Guangxi, which composed of education, economy, science and technology system, this paper chose the development indexes of the GDP (g), the number of graduates (e), scientific and technological level (t) from 1994 to 2006 as the parameters to measure the level of education, economy and science and technology. What was shown in Table 1 is the 14 years' standardized development index data of GDP, the number of graduates, the scientific and technological achievements of Guangxi province, and their trend were shown in Figure 1.

Table 1. Standardized development index data of Guangxi province

Year (y)	GDP (g)	Graduates (e)	Technology (t)
1994	0.000	0.000	0.000
1995	0.107	0.137	0.041
1996	0.187	0.116	0.146
1997	0.264	0.111	0.237
1998	0.359	0.220	0.158
1999	0.436	0.225	0.263
2000	0.512	0.199	0.403
2001	0.592	0.176	0.308
2002	0.693	0.393	0.312
2003	0.790	0.554	0.478
2004	0.902	0.587	0.492
2005	1.026	0.802	0.575
2006	1.153	0.927	0.738

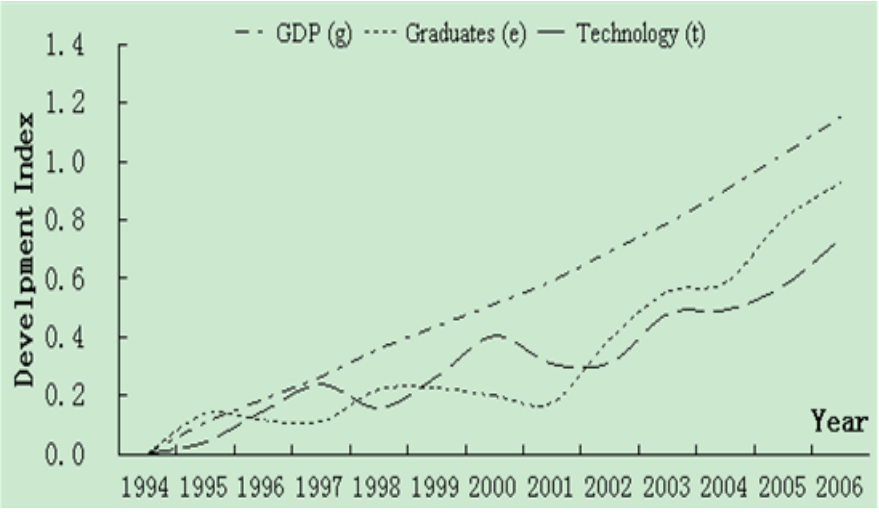


Fig. 1. Trend of development index data of Guangxi province

4.2 Unit Root Test

An application of stationary test to the three variables mentioned above was required firstly before we carry on the co-integration analysis. Only all the variables in t-order I(t) are stationary, can we carry on the co-integration analysis. This paper made use of the ADF unit root testing method to test the stationary of these variables, the results of which were shown in Table 2.

Table 2. The results of unit root test

Variable	ADF test Value	Testing type (c, t, d)	critical value	AIC value	Stationarity
e	-0.716084	(c, t, 0)	-3.388330**	-1.694055	Non -tationary
Δe	-2.788455	(c, 0, 0)	-2.728985**	-1.574121	stationary
t	-0.28475	(c, t, 1)	-3.933364*	-2.191312	Non -tationary
Δt	-5.071917	(c, 0, 1)	-3.212696*	-2.203401	stationary
g	-1.471024	(c, t, 2)	-4.008157*	-4.189669	Non -tationary
Δg	-4.990870	(c, t, 2)	-4.107833*	-4.900132	stationary

As it can be seen from Table 2, all of the variables are non-stationary series, and we should reject their null hypothesis of unit root. So the time-series e, g, t can be considered as non-stationary. And for all variables calculated with the first order difference, their hypothesis of unit root cannot be rejected, it is to say, their first

difference is stationary or integrated of order one, i. e. I(1). Obviously, we can test the relevance of these non-stationary variables by using co-integration method but not traditional linear regression analysis.

4.3 Co-integration Test

Using the method firstly proposed by Johansen, we can make co-integration tests on these variables directly by constructing the likelihood ratio statistics and the trace statistics based on the maximum Eigen value. The test results were shown in Table 3 and Table 4 respectively. From these tables we can see that there are 3 co-integration relationships between these time series: e, g, t.

Table 3. Judgment with the trace statistic

Null hypothesis	Latent Root	Trace Statistic	critical value	P-value
None cointegration	0.922	46.930	29.797	0.0002
At most 1 cointegration	0.723	18.810	15.494	0.0152
At most 2 cointegration	0.344	4.654	3.841	0.0309

Table 4. Judgment with λ-max statistic

Null hypothesis	Latent Root	λ-max Statistic	critical value	P-value
None cointegration	0.9224	28.1196	21.1316	0.0044
At most 1 cointegration	0.7238	14.1562	14.2646	0.0519
At most 2 cointegration	0.3449	4.6541	3.8414	0.0309

4.4 Granger Causality Test

By using the Granger test on these three time series of GDP (g), the number of graduates and the level of technology (t), its results was shown in Table 5.

Table 5. Granger causality test

Null hypothesis	Lag length	F statistic	P-value
g does not Granger Cause e	2	3.6501	0.0918
e does not Granger Cause g	2	0.4976	0.6310
T does not Granger Cause e	2	3.6297	0.0926
e does not Granger Cause t	2	6.1362	0.0354
T does not Granger Cause g	2	0.6521	0.5542
G does not Granger Cause t	2	10.2555	0.0115

Conclusions: There are two-way Granger causality between t and e; g Granger cause e and the probability that e does not Granger because g is high up to 63.10%; g Granger cause t and the probability that t does not Granger because g is 55.42%.

5 Summary

According to the above testes, there are long-run co-integration relationships between the economies, education, and technology multiplexed system of Guangxi province. And also we can learn from the results of these testes that economy growth is the reason of the education development and technological achievements from 1994 to 2006. Besides, Guangxi's high-speed economic growth has enhanced the financial capacity, greatly enlarging the government investment in education and scientific research. It is just in this period that the quality of education in Guangxi had a substantive leap. With the completely popularization of nine-year free education and the increase of the quantity of the senior high schools and variety of vocational technical schools, a large number of high-quality labor are cultivated to serve society. The higher education would be soon accessing to a popular state, and the rapid growth of the quantity of specialized personnel brings to the enrichment of the scientific and technological achievements. So the development of technology and education promote and influence each other, establishing the benign circulation mechanism of the two systems. But what we can not ignore is that the technology and education of Guangxi province are still in a low level, playing a limited role in the development of economy.

Therefore Guangxi should continue to develop education and scientific research vigorously in future. Focusing on the development of vocational education combining with practices, efforts should be paid to improve the quality of education and increase the average education years as an important means to guarantee the construction of the talents who can comply with the social demand; And Guangxi province should make use of any means to draw on the talents, strengthen the transformation of scientific and technological achievements and optimize industrial structure. Only thus economic development can be promoted in a better and faster way.

References

1. Engle, R.F., Granger, W.J.: Cointegration and Error Correction: Representation, Estimation and Testing. Econometrica 55, 251–276 (1987)
2. Johansen, S., Juselies, K.: Maximum Likelihood Estimation and Inference on Cointegration with Application to the Demand for Money. Oxford of Bulletin Economics and Statistics 52, 169–210 (1990)
3. Johansen, S.: Estimation and Hypothesis Testing of Cointegration Vectors in Gaussian Vector Autoregressive Models. Econometrica 59, 1551–1580 (1991)
4. Lau, L., Jamison, D., Liu, S.: Education and Economic Growth: Some Cross-sectional Evidence from Brazil. Journal of Development Economics 41, 45–70 (1993)
5. Lin, T.: Education, Technical Progress, and Economic Growth: the Case of Taiwan. Economics of Education Review 22, 213–220 (2003)
6. Barro, R.J., Jong-Wha, L.: International Comparisons of Educational Attainment. Journal of Monetary Economy 32, 363–394 (1993)

Research on Stability Region for a Type of Production Inventory Control System

Yongchang Wei and Hongwei Wang

Institute of System Engineering and
the State Key Laboratory of Image Processing and Intelligent Control,
Huazhong University of Science and Technology, Wuhan 430074, China
hwwang@mail.hust.edu.cn

Abstract. This paper focus on the stability analysis of the Automatic pipeline, variable inventory and order based production control system. Traditionally, dynamics of production inventory control system has been studied with transfer functions in frequency domain and the stability analysis is usually based on a fixed production delay. In this paper, state space representations are derived directly by a group of difference equations. The linear system theory is used to determine the stability region. The most significant point is that the region is independent on the production delay. Furthermore, two commonly used production ordering strategies are analyzed by the given region.

Keywords: Inventory control, Production delay, APVIOBPCS, Difference equation.

1 Introduction

A well designed production planning and inventory control system is essential for competitive dynamic behavior, and choosing the properly decision parameters will be a difficult problem. This paper focuses on how the decision parameters affect the stability of a type of production inventory control system especially for the situation when the production delay is any long.

The dynamics of production inventory control system has been extensively studied over years since the first work by Simon [1], who applied servomechanism continuous-time theory to manipulate the production rate in a simple system involving just a single product. The idea was extended to discrete-time models by Vassion [2]. A breakthrough, however, was experienced in the late 1950s by the so-called "industrial dynamics" methodology, which was introduced by Forreste [3]. The methodology, later referred as "system dynamics" used a feedback perspective to model, analyze and improve dynamic systems, including the production inventory system. However, the "industrial dynamics" methodology was criticized for not containing sufficient analytical support and for not providing guidelines to the system engineers on how to improve performance. Motivated by the need to develop a new framework that could be used as a base for seeking new novel control laws, Towill [4] presented the inventory and order based production control system (IOBPCS) in a

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 1123 – 1130, 2009.

block diagram form. Some later work was studied on the framework by changing the strategy of production planning or the demand forecasting methods, which can be convenient to measure and eliminate the bullwhip effect [5,6,7,8]. However, the systems studied were mostly based on small delay or fixed delay. Additionally, with a large production lag, it is hard to analyze the stability. However, the long production delay exists in real world; for example, a chip can be made only after hundreds of production procedure in the semiconductor industry.

This paper focuses on the Automatic pipeline, variable inventory and order based production control system (APVIOBPCS). State space representations are derived by a group of difference equations. The linear system theory is used to determine the stability region with any production delay. Furthermore, the given region is tested by simulation.

The structure of the paper is as follows: In Section 2, the studied production inventory control system is described by a group of difference equations, based on which state space representations are derived. In Section 3, a stability region is derived for any production delay. In section 4, two commonly used production planning strategies are analyzed by the given region. Finally, some concluding remarks are given in Section 5.

2 Difference Equations and State Space Representations

The production planning problem of a manufacturer in each period is considered. We focus on how the decision parameters affect the system stability. The timing of the events is as follows: (a) At the beginning of a period, to receive the goods ordered before and to review the inventory including the inventory working in progress (WIP) and the net stock. (b) Using the predictive demand and the actual demand in the previous period to predict the possible demand in the current period. Exponential smoothing is the chosen method because it is easy to implement and relatively accurate for short-term forecasts [6]. After the prediction, a production order is created. (c) Demand is fulfilled and the excess demand will be backlogged.

A group of difference equations can be derived according to the above events order. Suppose that the current period is dented by t; the lead time denoted by T_p, which is the time from the issue of orders until the receipt of goods from production line. It is assumed that the receiving goods in the current period, $Comrate(t)$, equal to the order quantity T_p periods before, $Orate(t-T_p)$. It should be known that the T_p is the multiple of integers.

$$Comrate(t) = Orate(t-T_p)\,. \tag{1}$$

System inventory levels are the accumulated sum of the difference between the production completion rate and the actual consumption rate. The difference equation required to capture inventory levels is shown in (2)

$$Ainv(t) = Ainv(t-1) + Comrate(t) - Const(t-1)\,, \tag{2}$$

where $Ainv(t)$ and $Const(t-1)$ denotes the inventory level and actual demand.

The acual work-in-progress levels (WIP) are calculated as the accumulated sum of the difference between production order rate and the production completion rate. The difference required to monitor WIP is shown in (3).

$$Wip(t) = Wip(t-1) + Order(t-1) - Comrate(t), \tag{3}$$

where $Wip(t)$ denotes the WIP at time t.

The following event is the demand forecasting. The method chosen is exponential smoothing. Our forecasting mechanism is according to the actual demand happened in the last period, $Const(t-1)$, and the value of prediction in last period, $Avcon(t-1)$, to predict the demand in the current period, which is expressed by (4).

$$Avcon(t) = (1-\theta)Avcon(t-1) + \theta Const(t-1), \tag{4}$$

where θ is a forecasting parameter.

The strategy of production planning plays an essential role in a production control system. With different decision parameters, the system might exhibit completely different dynamics and performances. In order to make the planning activities more flexible, we consider the Automatic pipeline, variable inventory and order based production control system(APVIOBPCS). The APVIOBPCS can be expressed in words as "Let the production targets, $Order(t)$, be equal to the sum of an exponentially smoothed representation of demand, $Avcon(t)$,plus a fraction $(1/T_i)$ of the inventory error in net inventory, plus a fraction $(1/T_w)$ of the WIP error [6]. The target of the inventory replenishment term in the production ordering policy is to bring the actual inventory towards the desired inventory. Because the demand may be fluctuate from time to time, the desired inventory of the APVIOBPCS is $a \times Avon(t)$. The WIP contribution to the ordering policy allows the ordering rate to depend on the WIP, and the WIP will also change as the demand fluctuating. The goal of WIP is often set to cover the demand during the production lead time, which is $T_p \times Avon(t)$. As a result, the production order is

$$Order(t) = Avcon(t) + \frac{1}{T_i}(a \times Avcon(t) - Ainv(t)) + \frac{1}{T_w}(T_p \times Avcon(t) - Wip(t)). \tag{5}$$

The relationship between APVIOBPCS and other production strategies is given in Table 1 [8].

Substitute (2), (3), (4) into (5). Let $\mu = [T_p, a, \theta, T_i, T_w]^T$ be the parameter vector and $[Ainv(t), Wip(t), Avcon(t), Order(t)]^T$ be the state vector. Integrate the above difference equations, the state space representations can be described as (6)

$$x(k) = A(\mu)x(k-1) + A_d(\mu)(x - T_p) + B(\mu)u(k-1), \tag{6}$$

Where $u(k-1)$ denotes the demand in the $(k-1)$ period and

$$B(\mu) = \left[-1 \quad 0 \quad \theta \quad \theta(1 + \frac{a}{T_i} + \frac{T_p}{T_w}) + \frac{1}{T_i} \right]^T, \tag{7}$$

$$A(\mu)=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1-\theta & 0 \\ -\frac{1}{T_i} & -\frac{1}{T_w} & (1-\theta)(1+\frac{a}{T_i}+\frac{T_p}{T_w}) & -\frac{1}{T_w} \end{bmatrix}, \tag{8}$$

$$A_d(\mu)=\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{T_i}+\frac{1}{T_w} \end{bmatrix}. \tag{9}$$

From the view of modern control theory, the system is a discrete time system with a fixed delay and the stability is determined by the element of vector μ .

Table 1. The brief relationship between production strategy in APVIOBPCS and other strategies in IOBPCS family

Model	Description	Inventory feedback gain	WIP feedback gain	Inventory feed-forward gain
OBPCS	Order based production control system	$T_i=\infty$	$T_w=\infty$	$a=0$
IBPCS	Inventory based production control system	T_i	$T_w=\infty$	$a=0$
VIBPCS	Variable inventory based production control system	T_i	$T_w=\infty$	a
APIOBPCS	Automatic pipeline, inventory and order based production control system	T_i	T_w	$a=0$
APVIOBPCS	Automatic pipeline, variable inventory and order based production control system	T_i	T_w	a

3 Stability Analysis on APVIOBPCS

The focus of this paper is to give a parameter region, in which the system might be stable. Before analyzing the system stability, three theorems are given as follows:

Theorem 1. [9] With real or complex coefficient, if the polynomial expression $P(z)=a_s z^s+a_{s-1}z^s+....a_1 z+a_0$ such that $|a_s|>|a_{s-1}|+|a_{s-2}|+...|a_0|$, then all the roots of $P(z)=0$ will be in the unit circle.

Theorem 2. Let $\varsigma = T_p - 1, \varsigma \geq 0$, the system denoted by (6)-(9) is stable if and only if for $0 < \theta < 1$ and the roots of the equation

$z^{\varsigma+1} + (1/T_w - 1)z^{\varsigma} + (1/T_i - 1/T_w) = 0$ are in the unit circle.

Proof. For a linear constant system, the BIBO stability can be satisfied by the structure stability.

$x(k) = A(\mu)x(k-1) + A_d(\mu)(x-\tau)$ can be converted into

$x(z) = A(\mu)z^{-1}x(z) + A_d(\mu)z^{-\tau}x(z)$ by using the z transformation.

$f(z) = \left| zI - A(\mu) - A_d(\mu)z^{-\varsigma} \right|$ is the corresponding eigenfunction. Suppose that the eigenvalues of $A(\mu) + A_d(\mu)z^{-\varsigma}$ are $\lambda_i(\mu, \varsigma), i = 1,2,3,4$, where μ is the parameter vector. The eigenfunction is given by $f(z) = \prod_{i=1}^{4}(z - \lambda_i(\mu, \varsigma))$.

By computation, the eigenvalues of $A(\mu) + A_d(\mu)z^{-\varsigma}$ are $\lambda_1 = 0$, $\lambda_2 = 1$, $\lambda_3 = 1 - \theta$ and $\lambda_4 = -(1/T_i)z^{-\varsigma} + (1/T_w)z^{-\varsigma} - (1/T_w) + 1$.

Because the demand don't affect WIP directly, the condition keeps the system stable occurs only when $0 < \theta < 1$ and the roots of

$z^{\varsigma+1} + (1/T_w - 1)z^{\varsigma} + (1/T_i - 1/T_w) = 0$ are in the unit circle.

Theorem 3. the system denoted by (6)-(9) is stable if θ, T_i and T_w satisfy:

$$(T_i, T_w, \theta) \in \left\{ (T_i, T_w, \theta) \mid \left| T_w - T_i \right| + \left| T_i - T_i T_w \right| < T_i T_w, 0 < \theta < 1, T_i > 0, T_w > 0 \right\}. \quad (10)$$

Theorem 3 can be immediately obtained according theorems 1 and 2, which gives a sufficient condition that keeps the system of (6)-(9) stable. The most significant point is that the stability region is independent on the production delay. The parameter region is shown as Fig.1. Furthermore, the conservation of the region is tested by simulation to be very low.

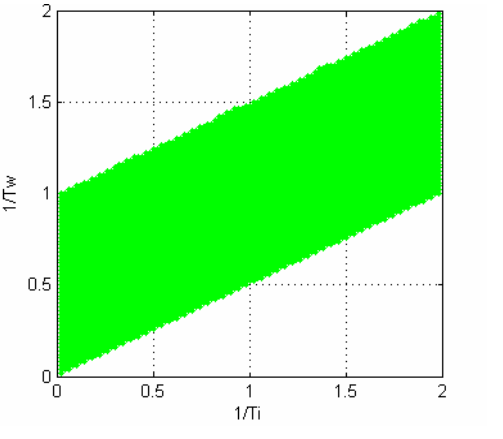


Fig. 1. Illustration of the stability region

The coefficient of exponential smooth method, θ, satisfies $0<\theta<1$. Without losing the generality, the value of θ is set to 0.8 throughout the paper. Assume that the distribution of demand follows normal distribution with a mean equal to 40 and a variance equal to 4. For the case $T_i=1/1.5, T_w=1/0.8, T_p=10$, the system will be stable as the case shown by Fig.2(a). Fig.2(b) shows the situation when $T_i=1/1.5, T_w=1/0.7, T_p=10$.

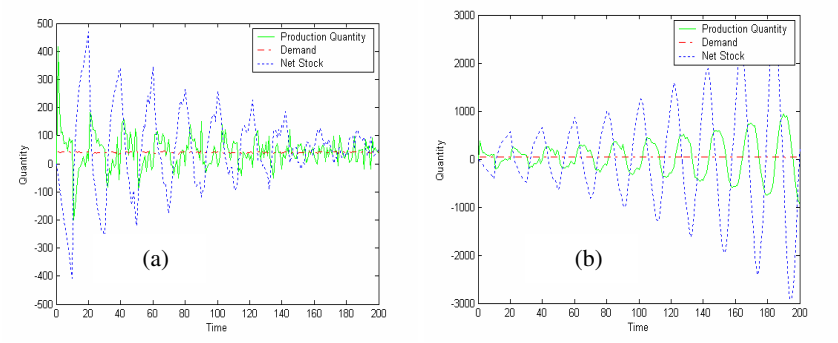


Fig. 2. (a, b) Impact of T_i, T_w, T_p on the stability

4 Extensions

With the stabilty result obtained in section 3, we can extend it to two commonly used ordering policies.

4.1 Stability Analysis on Order-Up-to Policy

In any order-up-to policy, ordering decisions are as follows:

$$O_t = S_t - inventory\ position\ , \tag{11}$$

where O_t is the ordering quantity, S_t is the order-up-to level in the current period. The inventory position equals to the net stock plus on WIP, where the net stock equals inventory on hand minus backlog. The order up to level is updated every period according to

$$S_t = \hat{D}_t^L - k\hat{\sigma}_t^L\ , \tag{12}$$

where L denotes the order lead time and $\hat{D}_t^L$ is an estimate of mean demand over the L periods. $\hat{\sigma}_t^L$ is an estimation of standard deviation of the demand over L periods, and k is a chosen constant to meet a desired service level. The order-up-to level will be constant when the average and the standard deviation of the demand during the

production lead time are known with certainty. When the demand is unknown, we must forecast the demand, and this forecasting creates variability in the order-up-to level. To simplify the analysis, we set k equal to zero. The value of L is inflated and the extra inventory denotes the safety stock, then as for the order-up-to level $(T_p+1)Avocn(t)$, the order rule is as follows:

$$Order(t) = \left(T_p+1\right)Avcon(t) - (Ainv(t) + Awip(t)), \tag{13}$$

while the order policy in our model can be converted into

$$Order(t) = (1+\frac{a}{T_i}+\frac{T_p}{T_w})Avcon(t) - \frac{1}{T_i}Ainv(t) - \frac{1}{T_w}Wip(t). \tag{14}$$

By contrast, (13) and (14) are the same when a is zero and both T_i and T_w equal to 1. T_i and T_w equal to 1 fall into the region given by theorem 3, which means the order up to level policy can keep the system in stability.

4.2 Stability Analysis on DE-APIOBPCS Policy

With a special case of $T_i = T_w$, the production inventory systems will be called as DE-APIOBPCS [10]. As being pointed out, $T_i = T_w$ guarantees the system to be stable and robust to changes in the distribution of the production delay. $z^{\varsigma+1} + (1/T_w - 1)z^{\varsigma} + (1/T_i - 1/T_w) = 0$ can be changed to $z^{\varsigma}(z + (1/T_w - 1)) = 0$ according to the theorem 2, which means that the system is stable when $0 < 1/T_i = 1/T_w < 2$. This is a sufficient and necessary condition. Meanwhile, the parameters satisfying this condition are in the region that theorem 3 has given. It is noticeable that the system will be unstable when $1/T_i$ and $1/T_w$ are more than two.

5 Conclusion

This paper gives a sufficient stability parameter region for a type of general production inventory control system. The conservation is very low, which is tested by simulations. The most important point is that the stability region is independent on the production delay. Furthermore, we analyzed the stability of two commonly used ordering policy by the given stability region.

In the future research, we can consider the performance in the given region, such as the bullwhip effect, cost and customer service level. The study can also be extended to two echelon supply chains. Another direction of research includes modeling the nonlinear and random factors, which might exhibit much more complex behaviors such as chaos, bifurcations and so on.

Acknowledgements

The research was supported by National Natural Science Foundation of China under the grant No. 60674085 and 70572033.

References

1. Simon, H.A.: On the Application of Servomechanism Theory in the Study of Production Control. Econometrica 20, 247–268 (1952)
2. Vassian, J.H.: Application of Discrete Variable Servo Theory to Inventory Control. Operational Research 3, 272–282 (1955)
3. Forrester, J.W.: Industrial Dynamics. MIT Press, Cambridge (1961)
4. Towill, D.R.: Dynamic Analysis of an Inventory and Order Based Production Control System. International Journal of Production Research 20, 671–687 (1982)
5. John, S., Naim, M.M., Towill, D.R.: Dynamic Analysis of a WIP Compensated Decision Support System. International Journal Manufacturing Systems Design 1(4), 283–297 (1994)
6. Disney, S.M., Towill, D.R.: A Discrete Linear Control Theory Model to Determine the Dynamic Stability of Vendor Managed Inventory. International Journal of Production Research 40(1), 179–204 (2002)
7. Disney, S.M., Towill, D.R.: Eliminating Inventory Drift in Supply Chains. International Journal of Production Economics 93–94, 331–344 (2005)
8. Dejonckheere, J., Disney, S.M., Lambrecht, M.R., Towill, D.R.: Measuring and Avoiding the Bullwhip Effect: A Control Theoretic Approach. European Journal of Operational Research 147, 567–590 (2003)
9. Lalwani, C.S., Disney, S.M., Towill, D.R.: Controllable, Observable and Stable State Space Representations of a Generalized Order-up-to Policy. International Journal of Production Economics 101, 172–184 (2006)
10. Ronald, E.M.: Difference Equations. Von Nonstand Reinhold Company, Inc., New York (1987)
11. Deziel, D.P., Eilon, S.: A Linear Production-Inventory Control Rule. The Production Engineer 43, 93–104 (1967)

Analyses and Improvement of Case-Based Decision Model of Product Conceptual Design

Qing Wang, Yong Zhao, and Congjun Rao

Institute of Systems Engineering, Huazhong University of Science and Technology, Wuhan 430074, China
puppy@vip.qq.com

Abstract. The present article describes a case-based decision model at the background of product conceptual design. The notion of the bounded rationality in the process of estimating similarity is used for proposing an improved similarity algorithm based on a distance threshold. The general algorithm and the improved algorithm are used to analyze the similarity between new target and finished products for conceptual design of circular gear reducers. An example validates the improved algorithm can effectively exclude useless cases.

Keywords: Case-based decision, Product conceptual design, Bounded rationality, Distance threshold.

1 Introduction

The product design process is a complex creative activity based on knowledge and experience. And the conceptual design determines the basic features and the main frame [1], which is not only the key link but also the bottleneck problem of the product design process. In traditional artificial design, product conceptual design always uses analogical design methods or empirical design methods to design the new product by altering finished products. Using artificial intelligence to design the product conceptual model which can improve the efficiency of product design has been the hotspot in the research of product conceptual design [2]. The Case-Based Reasoning (CBR) method which uses past successful experience to solve new problems is the most important reasoning technique of artificial intelligence. The method can reduce a lot of workload of obtaining knowledge, and have low requirements for knowledge complement. The most important point is that the case-based reasoning is closer to the thinking model of the designer. Currently the technique has been used in the area of industry design [3].

The Cased-Based Decision Theory (CBDT) [4] which is established on the base of case-based reasoning is proposed by Gilboa and Schmeidler. Many ideas of cognitive psychology and artificial intelligence are applied in CBDT where decision-makers solve a new decision problem depending on the experience of similar problems in memory. The dependence of experience is based on the assumption that similar problems have similar solutions. Along with experience accumulating, referable cases increase rapidly. Searching all the case is too costly, so decision-makers don't always

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 1131–1137, 2009.

rely on the whole experience when the time is limited. Eyke Hullermeier proposed a nearest neighbor classification method [5] which collects all the cases whose nearest neighbor is the targeted problem. This method also assumes that all cases have similarity with the target. However, during the process of similarity estimation, people cannot perceive the similarity between any couple of things. People always estimate with bounded rationality which makes them can only perceive the similarity within the perceivable range.

This paper proposes a case-based decision model for product conceptual design. We explore the bounded rationality during similarity estimation. Our considerations give rise to improve the similarity algorithm of case-based decision model. The improved similarity algorithm is validated by an example of conceptual design.

We start with an overview of CBDT in Section II. In section III, we discuss the frame of product conceptual design case-based decision model from the standpoint of decision-making. The Euclidian distance is analyzed as the independent variable of the similarity function and an improved similarity algorithm is proposed in section IV. Section V validates the algorithm with an example of conceptual design of circular gear reducers. The paper ends with some remarks on the improved algorithm and discusses the further research direction.

2 Overview of CBDT

The method of CBDT connects the solved problems and the problem to solve. We call the problem demanding solution as *targeted problem* and call the solved problems as *memory problem*. The decision process depends on the similarity between the targeted problem and the memory problem.

CBDT views cases as instances of decision making. A case is formally expressed as $c(p,a,r)$ which splits each case to three components: the decision problem (p), the solution that was chosen by the decision maker (a), and the outcome he has experienced (r). The set of cases is the *case-base* denoted as $C \doteq P \times A \times R$, in which P is the problem-set, A is the solution-set, and R is the result-set. Utility function $u(r)$ is the utility expression of the decision results. Subset $M \subseteq C$ is the memory of the decision maker. Cases in M express solved problems and their results. The similarity is formally expressed as σ_p : $P \times P \rightarrow$ [0, 1]. Given a targeted problem p_0 and a memory problem p_i, the similarity between p_0 and p_i is $\sigma_p(p_0, p_i)$.

The valuation of a case depends on the similarity between the targeted problem and all the solved problems and the utilities of their results. That is, given a memory $M \subseteq C$ and the targeted problem p_0, every solution a_0 is evaluated by the function

$$U(a_0) = U_{p_0,M}(a_0) \doteq \sum_{(p,a_0,r)\in M} \sigma_p(p, p_0) \cdot u(r) \tag{1}$$

where a maximizer of this functional is to be chose.

3 Case-Based Decision Model of Product Conceptual Design

According to the general process of product conceptual design, we can consider the product design as a decision problem. The product design target describes a decision problem. The design technique is the solution. Actual performance attributes of the product is the outcome. So the performance attributes of finished products are not only the description of the problem, but also the evaluation of the decision result. Then we establish the case-based decision model of product conceptual design as below:

The case-base C is the set of products. The memory M contains all the finished products. The case can be expressed as $c_i(p_i, a_i)$. If a type of products has n performance attributes $(x_{*1}, x_{*2}, \ldots, x_{*n})$ and the attributes' weights can be expressed by $w = (w_1, w_2, \ldots, w_n)$. Then $p_i = (x_{i1}, x_{i2}, \ldots, x_{in})$ denotes actual performance attributes of the product, and a_i denotes the technique and structure used to design the product, solution of the problem namely. The targeted problem $p_0 = (x_{01}, x_{02}, \ldots, x_{0n})$ characterizes the target of the new product. Here use σ_p : $P \times P \rightarrow$ [0, 1] to denote the similarity of the targeted problem and memory problems as well.

In the product conceptual design, the more similar with the target attributes, the more valuable for design the finished product is. The objective function

$$C(A) = \{(p_i, a_i) \mid \sigma_p(p_0, p_i) = \max_{(p,a) \in M} \sigma_p(p_0, p)\} \tag{2}$$

defines the choice set $C(A)$ of solution-set A. The cases in the choice set have the highest similarity with the targeted problem. The decision maker can obtain the design solution of the new product by altering the solutions of the cases in choice set.

4 Bounded Rationality Analyses and Improved Similarity Algorithm

To calculate the similarity of multi-attribute problems, the general way is to get the discrepancy of the targeted problem p_0 and the memory problem p_i first, and then calculate the similarity according to the function of discrepancy and similarity. Here use Euclidian distance

$$d(p_0, p_i) = \sqrt{\sum_{j=1}^{n} {w_j}^2 \cdot (x_{0j} - x_{ij})^2} \tag{3}$$

as the discrepancy. Most literatures use

$$sim(p_0, p_i) = \frac{1}{1 + d(p_0, p_i)} \text{ [6]} \tag{4}$$

to express the relationship of discrepancy and similarity.

The discrepancy is noted as d and the similarity is noted as s for short. Then the relation between them can be described as

$$s = \frac{1}{1+d}. \tag{5}$$

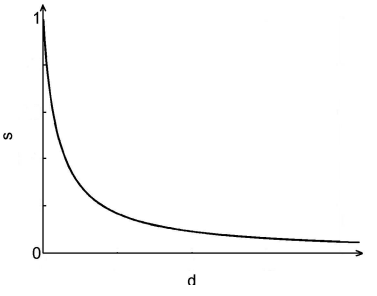


Fig. 1. If d and s satisfies function (5), the curves describes the change of d with s

Fig.1 shows that between the targeted problem and memory problems there always exists a similarity except that the discrepancy tends to infinite. The slope of the curve increases when the discrepancy decreases. It implies that the discrepancy is smaller the similarity changes more quickly.

Practically, people's perception capability of discrepancy and similarity isn't consistent with Fig.1 completely. During the similarity estimations people always characterize some kind of bounded rationality. From the point of view of discrepancy perception, there always exists a perception threshold which defines the perceptible range. If and only if the discrepancy of two things is within the perceptible range, people think the two things are comparable and the similarity is more than 0. If the discrepancy is beyond the perceptible range, the two things are incompatible and the similarity is 0. On the other hand, in the beginning (the end) of the perceptible range people are short-sighted (far-sighted). Namely, in the perceptible range, people are not sensitivity to the change of the discrepancy which is close to 0 or the threshold.

We define λ as the *distance threshold* which is the maximal Euclidian distance decision makers can realize. According to the analyses to the bounded rationality above, we use a piecewise function to describe the relation of d and s:

$$s = \begin{cases} 1 - 2(\frac{d}{\lambda})^2, 0 \le d < \frac{\lambda}{2} \\ 2(1 - \frac{d}{\lambda})^2, \frac{\lambda}{2} \le d < \lambda \\ 0, d \ge \lambda \end{cases} \tag{6}$$

where

$$d = d(p_0, p_i) = \sqrt{\sum_{j=1}^{n} w_j{}^2 \cdot (x_{0j} - x_{ij})^2} \,. \tag{7}$$

The relation curve of *d* and *s* is showed in Fig.2.

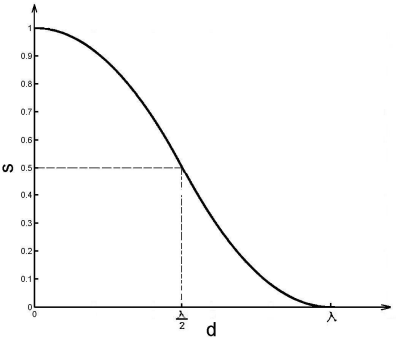


Fig. 2. If *d* and *s* satisfies function (6), the curve describes the change of *d* with *s*

In Fig.2, in the area of $d \in [0, \lambda)$ the curve is over the *d* axis, and in the area of $d \in [\lambda, \infty)$ the curve is overlapping with *d* axis. The line of $d = \frac{\lambda}{2}$ is the boundary of the curve. The curve on its left is convex where the slope grows with *d*'s growth. And on its right side, the curve is concave where the slope declines with *d*'s growth. Therefore the piecewise function (6) correctly presents the bounded rationality during the similarity estimations. Definition 1 defines the similarity algorithm formally.

Definition 1. Suppose the targeted problem is $p_0 = (x_{01}, x_{02}, \ldots, x_{0n})$ and the memory problem is $p_i = (x_{i1}, x_{i2}, \ldots, x_{in})$. $x_{*1}, x_{*2}, \ldots, x_{*n}$ are *n* attributes of the problems and their importance can be presented by a weight vector $w = (w_1, w_2, \ldots, w_n)$ which is of the form $\sum_{j=1}^{n} w_i = 1$. Euclidian distance between p_0 and p_i is denoted as $d(p_0, p_i)$. If a distance threshold λ is given, the similarity of p_0 and p_i is defined as

$$sim(p_0, p_i) = \begin{cases} 1 - 2(\frac{d(p_0, p_i)}{\lambda})^2, 0 \le d(p_0, p_i) < \frac{\lambda}{2} \\ 2(1 - \frac{d(p_0, p_i)}{\lambda})^2, \frac{\lambda}{2} \le d(p_0, p_i) < \lambda \\ 0, d(p_0, p_i) \ge \lambda \end{cases}. \tag{8}$$

5 Example

This section uses (4) and (8) to calculate the example of circular gear reducer conceptual design in reference [7]. Compute the similarity of finished gears and the design target to validate the algorithm of (8).

The key feature attributes of 10 finished gears are listed in table 1. The weights of the key feature attributes are in table 2. Table 3 lists the target for the new product.

Table 1. Key feature attributes of 10 finished circular gear reducers

Product Number	Transmission Power	Service Life	Transmission Ratio
C1	95	10	4.5
C2	59	10	3.55
C3	142	10	20
C4	68	15	16
C5	40	15	12.5
C6	161	10	18
C7	36	15	14
C8	105	10	14
C9	73	15	14
C10	68	15	2.5

Table 2. Weights of the key feature attribute

	Transmission Power	Service Life	Transmission Ratio
Weight	$\frac{3}{10}$	$\frac{9}{20}$	$\frac{1}{4}$

Table 3. Design target of the new product

Product Number	Transmission Power	Service Life	Transmission Ratio
C0	70	14	10

The distance of the target and every finished product is calculated as below.

Table 4. Distance of the target and every finished product

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
C0	7.867	4.209	21.652	2.382	9.285	27.318	10.445	10.500	2.423	3.700

With the distances in table 4, we can use (4) and (8) to calculate the similarity separately with the distance threshold $\lambda = 10$ given by decision maker. The results are listed in table 5.

Table 5. Similarity of the target and every finished product

Algorithm	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
(4)	0.113	0.192	0.044	0.296	0.097	0.035	0.087	0.087	0.292	0.213
(8)	0.091	0.646	0	0.887	0.010	0	0	0	0.883	0.726

We can order 10 finished products according to their similarity with the target. With the algorithm in (4), we have C4>C9>C10>C2>C1>C5>C8>C7>C3>C6. With the algorithm in (8), we have C4>C9>C10>C2>C1 whose similarity is positive.

Some products which have slight similarity with the target are filtered by the improved algorithm. The improved algorithm narrows the distance between the similarities of the most similar finished products to the target. This improvement makes the decision maker evaluate the most valuable products for the new one. The orders show that improvement does not influence the result.

6 Conclusion

This paper established a case-based decision model of product conceptual design which provided a way of using decision methods to solve the problems in machine design. The improved similarity algorithm which can effectively choose most useful cases for decision and make noise filtered described a kind of bounded rationality during the similarity estimations. The example proved that if only the decision maker has enough general knowledge and can choose reasonable distance threshold, he could make the correct decision even with some kind of bounded rationality.

The choice of attributes weight is not mentioned in this paper although it's the key point for rational decision. This part will be the next work target.

Acknowledgements. This work was supported by National Nature Science Foundation of Hubei Province (No. 2007ABA283) , the Project Sponsored by the State Education Ministry Scientific Research Foundation for the Returned Overseas Chinese Scholars, and the National "863" Program of China.

References

1. Shi, Q., Du, J., Li, Y., Yang, H.: A CBR (Case-Based Reasoning) Conceptual Design Model for Product Development. Journal of Northwestern Polytechnical University 20, 203–207 (2002) (in Chinese)
2. Giretti, A., Spalazzi, L.: A Conceptual Design-Support System. Engng. Applic. Artif. Intell. 10, 99–111 (1997)
3. Chen, H., Chen, J.: A Rapid Product Design System of Case-Based Reasoning Technology. Machine Tool & Hydraulics 3, 130–184 (2004) (in Chinese)
4. Gilboa, I., Schmeidler, D.: A Theory of Case-Based Decision. Cambridge University Press, United Kingdom (2001)
5. Eyke, H.: Exploiting Similarity and Experience in Decision Making. In: IEEE International Conference of Fuzzy Systems, pp. 729–734. IEEE Press, New York (2002)
6. Jie, C.: Study on the Case-Based Reasoning Retrieval Strategy of Data Mining Algorithm. Taiyuan Polytechnics Master Thesis (2007) (in Chinese)
7. Wang, F.: Research on Case-Based Reasoning for Product Conceptual Design. Wuhan University Of Technology Master Thesis (2004) (in Chinese)

Information Integration Approach to Vendor Selection Group Decision Making under Multiple Criteria

Wu Li[1], Xiaomei Zhang[2], and Yan Chen[3]

[1] Department of Mechanical and Electrical Engineering, Hunan Institute of Science and Technology, Yueyang 414006, China
[2] School of Information Engineering, Zhongzhou University, Zhengzhou 450044, China
[3] Department of Mathematics, Hunan Institute of Science and Technology, Yueyang 414006, China
liwu0817@163.com

Abstract. Vendor selection is a vital issue for compgaoanies in supply chains. In practice it is usually made by a group of decision makers according to multiple criteria. However, almost all the vendor selection models published were based on single-person decision-making and laid less emphasis on ordinal data. An information integration approach based on multi-criteria group decision-making (MCGDM) is developed for vendor selection with ordinal preferences of alternatives given by multiple decision makers in respect to each criteria considered. A 0-1 programming model considering the weights of the criteria and the decision makers under every criteria was proposed to obtain the integrated rankings of the alternatives for the group. An illustrative case shows the proposed method is effective and simple in computation. This study extends Bernardo's method for multi-criteria decision-making to MCGDM.

Keywords: Supply chain management, Vendor selection, Multi-criteria decision-making, Group decision-making, Ordinal preference.

1 Introduction

Because of the emphasis on outsourcing, strategic partnering, strategic alliances, and relationship marketing, many organizations purchase not only raw materials and basic supplies but also complex fabricated components with very high value-added content and services over the last two decades. The need to gain a global competitive edge on the supply side has increased substantially. Particularly for companies who spend a high percentage of their sales revenue on parts and material supplies, and whose material costs represent a larger portion of total costs, savings from supplies are of particular importance. Moreover, the emphasis on quality and timely delivery in today's globally competitive marketplace adds a new level of complexity to outsourcing and vendor selection decisions. These, strongly urge for a more systematic and transparent method to purchasing decision making, especially regarding the area of

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 1138–1143, 2009.

vendor (supplier) selection. Selecting the right suppliers significantly reduces the purchasing cost and improves corporate competitiveness, on which the success of a supply chain is highly dependent.

The overall objective of vendor selection process is to reduce purchase risk, maximize overall value to the purchaser, and build the closeness and long-term relationships between buyers and vendors. Selecting vendors from a large number of possible alternatives with various levels of capabilities and potential is a difficult task. Nowadays, on the one hand, simply looking for vendors offering the lowest prices is not "efficient sourcing" any more. Multiple criteria including quality, delivery, performance history, warranties, price, technical capability and financial position need to be taken into account simultaneously in the vendor selection decision-making process. Dickson [1] emphasized that cost, quality, and delivery performance were the three most important criteria in vendor evaluation. In a comprehensive review of vendor selection methods, Weber et al. [2] reported that quality was perceived to be the most important vendor selection criterion followed by delivery performance and cost. With the increasing significance of vendor selection and competition of global environment, the approach to traditional criteria has been changed to reflect the new requirements according to the role of vendors in the supply chain. For instance, instead of price, total cost of ownership is considered, instead of quality, total quality and certification issues become the major concern, etc. [3]. In modern management, one needs to consider many other factors with the aim of developing a long-term vendor relationship. So vendor selection is a multi-criteria decision-making problem. Due to the complexity and importance, on the other hand, vendor selection is usually made by multiple decision makers such as experts in quality control, financial management or supply chain management, and related administrative officials. So vendor selection is inherently a typical multi-criteria group decision-making (MCGDM) problem [4-6].

According to De Boer et al. [7], a vendor selection problem typically consists of four phases, namely (1) problem definition, (2) formulation of criteria, (3) qualification of suitable vendor (or pre-qualification), and (4) final selection of the ultimate vendor(s). The vast majority of the decision models for vendor selection apply to the final choise phase. Among these models multi-criteria decision-making approaches are frequently used [8]. To the best of our knowledge, however, almost all literatures published on vendor selection only have regarded it as a single-person decision-making problem except [5,6,9] and laid less emphasis on ordinal data. Thus we propose a MCGDM approach based on 0-1 programming to vendor selection in which every decision maker gives his rankings of alternatives with respect to each criteria considered. This method can rank the alternatives for the decision maker group and generalize Bernardo's method [4] for multi-criteria decision-making to MCGDM considering the weights of criteria and decision makers.

The remainder of the paper is organized as follows. The following section formulates the vendor selection problem and details the proposed method that ranks the alternative vendors for the group. In section 3, we present a numerical example and some discussions on the results. Finally, some concluding remarks are provided in section 4.

2 The Proposed Method

Now we shall develop a new MCGDM model based on 0-1 programming for vendor selection with ordinal preferences of the alternatives given by the decision makers.

Firstly, we define and formulate the vendor selection problem as follows. Consider a vendor selection problem with I decision makers evaluating J vendors according to K criteria. D_i $(i=1,2,\cdots,I)$ is decision maker, S_j $(j=1,2,\cdots,J)$ is vendor, and C_k $(k=1,2,\cdots,K)$ is criteria. λ_k $(k=1,2,\cdots,K)$ is the weight of criteria C_k which satisfies $\sum_{k=1}^{K}\lambda_k=1$ and $\lambda_k>0$. ω_k^i $(k=1,2,\cdots,K;i=1,2,\cdots,I)$ is the weight of decision maker D_i under C_k which satisfied $\sum_{i=1}^{I}\omega_k^i=1$ and $\omega_k^i>0$. r_{kj}^i is the ranking of alternative S_j with respect to criteria C_k given by decision maker D_i and $r_{kj}^i\in\{1,2,\cdots,J\}$. For certain i $(i=1,2,\cdots,I)$, all r_{kj}^i constitute individual decision matrix $R^i=\left[r_{kj}^i\right]_{J\times K}$ as follows.

$$R^i=\begin{bmatrix} r_{11}^i & r_{12}^i & \cdots & r_{1K}^i \\ r_{21}^i & r_{22}^i & \cdots & r_{2K}^i \\ \vdots & \vdots & & \vdots \\ r_{J1}^i & r_{J2}^i & \cdots & r_{JK}^i \end{bmatrix}$$

To obtain the final rankings of all vendors for the decision maker group, we develop the solution approach based on 0-1 programming. The algorithm proceeds as follows.

Step 1. Define the consistency matrix $E_k^i=\left[e_{kmn}^i\right]_{J\times J}$ for D_i $(i=1,2,\cdots,I)$ under C_k $(k=1,2,\cdots,K)$ where

$$e_{kmn}^i=\begin{cases}1, \ when\ r_{km}^i=n;\\ 0, \ otherwise.\end{cases}(m,n=1,2,\cdots,J)$$

Step 2. Calculate the decision maker-weighted consistency matrix E_k for the group under C_k $(k=1,2,\cdots,K)$.

$$E_k=\left[e_{kmn}\right]_{J\times J}=\sum_{i=1}^{I}\omega_k^i\times E_k^i$$

Step 3. Calculate the integrated consistency matrix E for the group after the weights of criteria are considered.

$$E = [e_{mn}]_{J\times J} = \sum_{k=1}^{K} \lambda_k \times E_k$$

Step 4. Obtain the ranking r_j of S_j $(j = 1,2,\cdots,J)$ for the group by solving the following 0-1 programming problem:

$$\begin{aligned} (P) \qquad & \max \sum_{m=1}^{J}\sum_{n=1}^{J} e_{mn} x_{mn} \\ & s.t. \ \sum_{n=1}^{J} x_{mn} = 1,\ m = 1,2,...,J \\ & \sum_{m=1}^{J} x_{mn} = 1,\ n = 1,2,...,J \\ & x_{mn} \in \{0,1\},\ m,n = 1,2,...,J \end{aligned}$$

$x_{mn} = 1\,(m,n = 1,2,\cdots,J)$, the solution of (P), indicates $r_m = n$, that is, the integrated ranking of S_m for the group is n.

3 Illustrative Case and Discussion

Consider a vendor selection committee consisting of $I = 3$ decision makers evaluates $J = 5$ feasible vendors according to $K = 3$ criteria: quality, price and delivery. The weights of criteria and decision makers and ordinal preferences of the suppliers are listed in table 1. $R^i\,(i = 1,2,3)$ are as follows.

Table 1. Weights of criteria and decision makers under each criteria

k	λ_k	w_k^1	w_k^2	w_k^3
1	0.3	0.3	0.4	0.3
2	0.3	0.3	0.3	0.4
3	0.4	0.4	0.3	0.3

$$R^1 = \begin{bmatrix} 5 & 3 & 3 \\ 2 & 1 & 2 \\ 3 & 4 & 4 \\ 4 & 5 & 5 \\ 1 & 2 & 1 \end{bmatrix}, R^2 = \begin{bmatrix} 4 & 1 & 3 \\ 2 & 3 & 1 \\ 5 & 4 & 5 \\ 3 & 2 & 4 \\ 1 & 5 & 2 \end{bmatrix}, R^3 = \begin{bmatrix} 4 & 4 & 4 \\ 1 & 2 & 1 \\ 5 & 5 & 5 \\ 3 & 3 & 2 \\ 2 & 1 & 3 \end{bmatrix}$$

Now we first detail the procedure of obtaining the final rankings of the vendors for the committee by the proposed method.

Step 1. Define $E_k^i = \left[e_{kmn}^i \right]_{J\times J}$ $(i,k=1,2,3)$. Below we just list E_1^1, E_1^2, E_1^3.

$$E_1^1 = \begin{bmatrix} 0&0&0&0&1 \\ 0&1&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&1&0 \\ 1&0&0&0&0 \end{bmatrix}, E_1^2 = \begin{bmatrix} 0&0&0&1&0 \\ 0&1&0&0&0 \\ 0&0&0&0&1 \\ 0&0&1&0&0 \\ 1&0&0&0&0 \end{bmatrix}, E_1^3 = \begin{bmatrix} 0&0&0&1&0 \\ 1&0&0&0&0 \\ 0&0&0&0&1 \\ 0&0&1&0&0 \\ 0&1&0&0&0 \end{bmatrix}$$

Step 2. Calculate E_k $(k=1,2,3)$ as follows. For example, $E_1 = 0.3E_1^1 + 0.4E_1^2 + 0.3E_1^3$.

$$E_1 = \begin{bmatrix} 0&0&0&0.7&0.3 \\ 0.3&0.7&0&0&0 \\ 0&0&0.3&0&0.7 \\ 0&0&0.7&0.3&0 \\ 0.7&0.3&0&0&0 \end{bmatrix}, E_2 = \begin{bmatrix} 0.3&0&0.3&0.4&0 \\ 0.3&0.4&0.3&0&0 \\ 0&0&0&0.6&0.4 \\ 0&0.3&0.4&0&0.3 \\ 0.4&0.3&0&0&0.3 \end{bmatrix}, E_3 = \begin{bmatrix} 0&0&0.7&0.3&0 \\ 0.6&0.4&0&0&0 \\ 0&0&0&0.4&0.6 \\ 0&0.3&0&0.3&0.4 \\ 0.4&0.3&0.3&0&0 \end{bmatrix}$$

Step 3. Calculate E as follows $E = 0.3E_1 + 0.3E_2 + 0.4E_3$.

$$E = \begin{bmatrix} 0.09&0&0.37&0.45&0.09 \\ 0.42&0.49&0.09&0&0 \\ 0&0&0.09&0.34&0.57 \\ 0&0.21&0.33&0.21&0.25 \\ 0.49&0.30&0.12&0&0.09 \end{bmatrix}$$

Step 4. Obtain the final ranking r_j $(j=1,2,\cdots,5)$ by solving the following 0-1 programming problem:

$$\begin{aligned} (P)\ \max\ & (0.09x_{11} + 0.37x_{13} + 0.45x_{14} + 0.09x_{15} \\ & + 0.42x_{21} + 0.49x_{22} + 0.09x_{23} \\ & + 0.09x_{33} + 0.34x_{34} + 0.57x_{35} \\ & + 0.21x_{42} + 0.33x_{43} + 0.21x_{44} + 0.25x_{45} \\ & + 0.49x_{51} + 0.3x_{52} + 0.12x_{53} + 0.09x_{55}) \\ s.t.\ & \sum_{n=1}^{5} x_{mn} = 1,\ m = 1,2,...,5 \\ & \sum_{m=1}^{5} x_{mn} = 1,\ n = 1,2,...,5 \\ & x_{mn} \in \{0,1\},\ m,n = 1,2,...,5 \end{aligned}$$

The result is $x_{14} = x_{22} = x_{35} = x_{43} = x_{51} = 1$ which indicates $r_1 = 4, r_2 = 2, r_3 = 5$, $r_4 = 3, r_5 = 1$. So the final ranking for the committee is $S_5 \succ S_2 \succ S_4 \succ S_1 \succ S_3$.

4 Conclusions

Vendor evaluation and selection is one of the most critical activities of companies in supply chains. Because of the complication and importance of the problem, vendor selection is a typical MCGDM and the decision makers tend to give their ordinal preferences of the alternatives.

We develop a MCGDM model based on 0-1 programming for vendor selection with ordinal data. The numerical case illustrates the proposed method is effective and simple in computation. More importantly, this study generalizes Bernardo's method [4] for multi-criteria decision-making to MCGDM.

We did not involve the topics of weight elicitation, consensus and other group interactions in the study. How to solve vendor selection problems with ordinal preferences in other form is also should be investigated further.

Acknowledgments. This work is partially supported by National Natural Science Foundation of China (No. 60774084), Scientific Research Fund of Hunan Provincial Education Department (No. 08B030) and Scientific Research Fund of Hunan Institute of Science and Technology (No. 2008Y17).

References

1. Dickson, G.W.: An Analysis of Vendor Selection Systems and Decisions. Journal of Purchasing 2(1), 5–17 (1966)
2. Weber, C.A., Current, J.R., Benton, W.C.: Vendor Sselection Criteria and Methods. European Journal of Operational Research 50, 2–18 (1991)
3. Choy, K.L., Lee, W.B., Lau, H.C.W., Choy, L.C.: A Knowledge-based Supplier Intelligence Retrieval System for Outsource Manufacturing. Knowledge-Based Systems 18, 1–17 (2005)
4. Hwang, C.L., Lin, M.J.: Group Decision Making under Multiple Criteria: Methods and Applications. Springer, Berlin (1987)
5. Li, W., Chen, Y.G., Fu, Y.Z.: Combination of TOPSIS and 0-1 Programming for Supplier Selection in Supply Chain Management. In: 2008 IEEE International Conference on Networking, Sensing and Control, pp. 1531–1535. IEEE Press, New York (2008)
6. Li, W., Chen, Y.G., Chen, Y.: Generalizing TOPSIS for Multi-criteria Group Decision-Making with Weighted Ordinal Preferences. In: The 7th World Congress on Intelligent Control and Automation, pp. 7505–7508. IEEE Press, New York (2008)
7. De Boer, L., Labro, E., Morlacchi, P.: A Review of Methods Supporting Supplier Selection. European Journal of Purchasing & Supply Management 7, 75–89 (2001)
8. Ng, W.L.: An Efficient and Simple Model for Multiple Criteria Supplier Selection Problem. European Journal of Operational Research 186, 1059–1067 (2008)
9. Chen, C.T., Lin, C.T., Huang, S.F.: A Fuzzy Approach for Supplier Evaluation and Selection in Supply Chain Management. International Journal of Production Economics 102, 289–301 (2006)

Models Choice of Grid-Based Decision Support System

Zhiwu Wang[1,2], Yanhui Zhang[1,⋆], Haigang Song[3], and Xueguang Chen[2]

[1] Henan Institute of Engineering, Zhengzhou, 451191, P. R. China
[2] Institute of System Engineering, Huazhong University of Science and Technology, Wuhan, 430074, P. R. China
[3] Basic Research Service of the Ministry of Science and Technology of the P. R. China, Beijing, 100862, P. R. China
wangzhiwu1974@gmail.com, huihui2001922@163.com, xgchen9@mail.hust.edu.cn

Abstract. Grid-based Decision Support System created a new research area. Decision-making model offers assistant support for decision-makers in model service just like the idea of grid service. How to make a better choice among many model services is the new problem which decision-makers are facing. In this paper, through the definition of model service quality, we transfer the problem of model service choice into multi-attributive decision of model service quality evaluation. TOPSIS method is used to find the solution of this problem, and we also discuss the principle of three kinds of weight mode. At last, we prove the scientific rationality of subjective-objective weight mode by exact samples.

Keywords: Models choice, DSS, Gird, Model service, QoS.

1 Introduction

The emerging concept of Grid-based Decision Support System (GBDSS) expands and enriches the theory and application of Decision Support System (DSS) [1]. Model is the important resource of DSS, and the essential feature of DSS is to assistant decision through model or combined models. Model management has long been the core of research in DSS, which directly influences the development of DSS, and thus in GBDSS.

GBDSS weakens the component of local model base and model management system (MMS) in traditional DSS. It treats the Internet as its limitless storeroom of decision resources and uses it intelligently to meet users' needs [2] and [3]. In grid environment, model offers assistant decision for decision-makers in model service style. Decision-makers face the first of all the problems to choose among the many models. Scientific and reasonable comment on model services can offer effective help for decision-makers' choice.

In this paper, we propose a new method of model service choice in GBDSS, through the definition of model service about quality attribute, multi-attributive

⋆ Corresponding author.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 1144–1153, 2009.

decision of changing model service choice into comment on model service quality, TOPSIS method is used to find the solution of this problem. Section 2 defines the quality attribute of model service; Section 3 discuss three different kinds of weight mode; Section 4 proves its reasonability by some exact cases; Section 5 is our conclusion.

2 Quality Attribute of Model Service

Quality attribute of model service can be described in many ways in current evaluation model of Web service and grid service, such as response time, price, performance, reliability, availability, security, reputation, Throughput, etc [4] and [5]. But careful analysis shows some targets are crossed and some are not easy to measure. So this paper chooses four targets (service price, response time, reliability and reputation), which can indicate decision-makers' requests of model service quality to describe model service's quality attribute. These targets are independent, easy to get and have clear definition, which are based by mentioned QoS (Quality of Service).

Definition: QoS of model service can be divided into four groups:

$$QoS(MS) = \{QoS_{pr}(MS), QoS_{RT}(MS), QoS_{re}(MS),\ QoS_{RD}(MS)\}. \quad (1)$$

$QoS(MS)$ represents model service quality and consists of four attached attribute:

1. Price of model service ($QoS_{pr}(MS)$): refers to related cost caused by the use of model service, which is announced by supplier of model service.
2. Response time of model service ($QoS_{RT}(MS)$): refers to the interval between the applying for service and getting response of service by service applier, including the transportation time and process time, which can be calculated in the following list:

$$QoS_{RT}(MS) = T_{trans}(MS) + T_{process}(MS). \quad (2)$$

$T_{trans}(MS)$ can be defined by the service historical record, and take the average transit time of several time. The formula is.

$$T_{trans}(MS) = \sum_{i=1}^{n} T_i(MS) \Big/ n. \quad (3)$$

Among this, $T_i(MS)$ is the recorded transmission time, n is the number.

Service processing time ($T_{process}(MS)$): released by provider of model service.

3. Reliability of model service ($QoS_{re}(MS)$): refers to the probability that keeps model service normally work, is the rate of successful implementation number and the total call number. Model service is selected by call numbers, the reliability can be calculated by statistical methods as the following formula:

$$QoS_{re}(MS) = N_c(MS)/k. \quad (4)$$

Among this, $N_c(MS)$ refers to successful operation number, k refers to total call number.

4. Reputation degree of model service($QoS_{RD}(MS)$): refers to the measure of the credibility of service reputation, which is influenced by experience of the end-users. Different users have different opinions on the same service. The formula is the following:

$$QoS_{RD}(MS) = \sum_{i=1}^{n} R_i \Big/ n. \tag{5}$$

Among this, R_i refers to the judge on the model service by users, which often chooses the decimal between 0 and 1, n is the number that model service is judged.

3 Evaluation on Model Service Quality

3.1 Selection of Methods to Evaluate *QoS*

It is multi-attribute decision problem to evaluate model service quality that can satisfy function request. TOPSIS(Technique for Order Preference by Similarity to Ideal Solution) is stated as the evaluation method for model service quality [6].

The Solution Idea of TOPSIS. The positive ideal solution x^* is a suppositional value, it's composed of all best values attainable of attribute, whereas the negative ideal solution x^0 is made up of all worst values attainable of attribute. In n-dimensional space, the comparison of alternatives x^i and the distance of x^* and x^0, based on the concept that the chosen alternative should have the shortest distance from the positive ideal solution (PIS) and the farthest from the negative-ideal solution (NIS) for solving a multi-attribute decision making problem.

The Solution Steps of TOPSIS

- Step 1: Normalize the evaluation matrix. If $A = \{a_{ij}\}$ is a decision matrix of QoS, $Z = \{z_{ij}\}$ is a normalized one, then:

$$z_{ij} = \mathrm{a_{ij}} \Big/ \sqrt{\sum_{i=1}^{m} a_{ij}^2}, i = 1, ..., m; j = 1, \cdots, n. \tag{6}$$

- Step 2: Construct the weighted normalized evaluation matrix: $X = \{x_{ij}\}$. If the weight of each service quality attribute is set(which will be discussed in the following 3.2), $w = (w_1, w_2, \cdots, w_n)^T$, then $x_{ij} = w_j \cdot z_{ij}, j = 1, \cdots, n$.
- Step 3: Determine positive ideal Solution x^* and negative ideal solutions x^0. No. j's attribute value of positive Ideal solution x^* is x_j^*, No. j's attribute value of the negative ideal solutions x^0 is x_j^0, then:

$$\text{Positive ideal solution } x_j^* = \begin{cases} \max\limits_i x_{ij} \;\; j(\text{beneficial attribute}) \\ \min\limits_i x_{ij} \;\; j(\text{cost attribute}) \end{cases} \quad j = 1, \cdots, n. \tag{7}$$

$$\text{Negative ideal solution } x_j^0 = \begin{cases} \max\limits_i x_{ij} \;\; j(\text{cost attribute}) \\ \min\limits_i x_{ij} \;\; j(\text{beneficial attribute}) \end{cases} \quad j = 1, \cdots, n. \tag{8}$$

– Step 4: To determine the distance from each alternatives x^i to x^* and x^0. The distance of x^i to x^* and x^0 is:

$$\begin{aligned} d_i^* &= \sqrt{\sum_{j=1}^{n} (x_{ij} - x_j^*)^2}, i = 1, \cdots, m; \\ d_i^0 &= \sqrt{\sum_{j=1}^{n} (x_{ij} - x_j^0)^2}, j = 1, \cdots, m. \end{aligned} \tag{9}$$

– Step 5: To determine the composite evaluation index of each alternatives.

$$C_i^* = d_i^0 / (d_i^0 + d_i^*), i = 1, \cdots, m. \tag{10}$$

– Step 6: The order of S_i alternative is ranked according to inverted sequence of C_i*.

3.2 Discussion on Weight Mode

No matter what kind of analysis method is used, attributive weight mode should be determined first to multi-attribute decision problem. Three kinds of weight modes, Subjective Weight Mode, Objective Weight Mode and Subjective-objective Weight Mode will be discussed in the following.

Subjective Weight Mode. To imagine weight index is decided by user's hobby, service discovering system will choose service by the way of over-best right [6].

To imagine w_j is q_j's weight index, w_j^* is the determined weight index,

$$\begin{cases} \min f_1 == \sum\limits_{k=1}^{n} \sum\limits_{j=1}^{n} (d_{kj} w_j - w_k)^2 \\ subject \quad to \quad \sum\limits_{j=1}^{n} w_j = 1 \\ w_j \geq 0, \quad j = 1, 2, \cdots, n. \end{cases} \tag{11}$$

$$d_{kj} > 0, d_{jk} = 1/d_{kj}, d_{kk} = 1, \sum_{k=1}^{n} d_{kj} = \sum_{k=1}^{n} w_k \Big/ w_j, \quad i, k = 1, 2, \cdots, n. \tag{12}$$

Objective Weight Mode. To imagine weight index is decided by objective data without considering user's hobby [7] [8], service discovering system will choose service by

$$w_j^* \in w = \{\sum_j^n w_j = 1, w_j \geq 0, j = 1, 2, \cdots, n\}. \tag{13}$$

To imagine w_j is service attribute, q_j is weight mode index, w_j^* is the index decided by average difference method.

$$\begin{cases} \min f_2 = \sum\limits_{i=1}^{m} \sum\limits_{j=1}^{n} (b_j^* - b_{ij})^2 w_j^2 \\ subject \quad to \quad \sum\limits_{j=1}^{n} w_j = 1 \\ w_j \geq 0, \quad j = 1, 2, \cdots, n. \end{cases} \tag{14}$$

$b_j^* = max\{b_{1j}, b_{2j}, \cdots, b_{mj}\}$ represents q_j's ideal index, target f_2 requests the difference between ideal index and other prepared service q_j to be the least.

Subjective-objective Weight Mode. To imagine weight mode is decided by the objectivity of service quality and user's hobby [9] [10], service discovering system will choose service by $w_j^* \in w = \{\sum\limits_{j=1}^{n} w_j = 1, w_j \geq 0, j = 1, 2, \cdots, n\}$ to imagine w_j is q_j' weight mode, $w_j^* \in w = \{\sum\limits_{j=1}^{n} w_j = 1, w_j \geq 0, j = 1, 2, \cdots, n\}$ is the index determined by the following:

$$\begin{cases} \min f_1 = \sum\limits_{k=1}^{n} \sum\limits_{j=1}^{n} (d_{kj} w_j - w_k)^2 \\ \min f_2 = \sum\limits_{i=1}^{m} \sum\limits_{j=1}^{n} (b_j^* - b_{ij})^2 w_j^2 \\ subject \quad to \quad \sum\limits_{j=1}^{n} w_j = 1 \\ w_j \geq 0, \quad j = 1, 2, \cdots, n. \end{cases} \tag{15}$$

To solve the pattern above, following list is arranged:

$$\begin{cases} \min f_3 = \alpha \sum\limits_{k=1}^{n} \sum\limits_{j=1}^{n} (d_{kj} w_j - w_k)^2 + \beta \sum\limits_{i=1}^{m} \sum\limits_{j=1}^{n} (b_j^* - b_{ij})^2 w_j^2 \\ subject \quad to \quad \sum\limits_{j=i}^{n} w_j = 1 \\ w_j \geq 0, \quad j = 1, 2, \cdots, n. \end{cases} \tag{16}$$

Among this α and β represents the relative importance degree of subjective and objective weight mode and satisfied the following list: $\alpha + \beta = 1, 0 < \alpha < \beta$.

4 Case Analysis

4.1 Case Introduction

This paper supposes that 10 models services satisfy user's function demand and offer Four-dimensional $(QoS_{pr}(MS), QoS_{re}(MS), QoS_{RT}(MS), QoS_{RD}(MS))$ to satisfy model service user's similarity threshold value constrictthe following Table 1 is easy to get.

Table 1. QoS attributes of 10 similarity model services

Order No	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}
Price(Pr)	0.19	0.25	0.15	0.20	0.23	0.25	0.20	0.21	0.23	0.20
Response time(RT)	226	201	209	203	216	226	241	206	201	196
Reliability(re)	0.72	0.77	0.74	0.67	0.74	0.72	0.76	0.70	0.74	0.74
Reputation degree(RD)	0.83	0.89	0.82	0.86	0.89	0.79	0.83	0.87	0.83	0.79

To imagine the attribute comparison matrix of D_1 and D_2 given by model service user as the following:

$$D_1 = \begin{bmatrix} 1 & 1/3 & 1/4 & 1/4 \\ 3 & 1 & 2 & 3 \\ 4 & 1/2 & 1 & 4/5 \\ 4 & 1/3 & 5/4 & 1 \end{bmatrix}, \; D_2 = \begin{bmatrix} 1 & 2 & 1/3 & 1/3 \\ 2 & 1 & 1/2 & 1/2 \\ 3 & 2 & 1 & 3/4 \\ 3 & 2 & 4/3 & 1 \end{bmatrix}.$$

4.2 Case Solution

Solving this case with TOPSIS.

Standardization of Attributes Vectors. The following is resulted from standardization of attributes vectors in the Table 1.

Table 2. Standardization of attributes vectors matrix

Order No	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}
Pr	0.2821	0.3712	0.2227	0.2969	0.3415	0.3712	0.2969	0.3118	0.3415	0.2970
RT	0.3356	0.2985	0.3104	0.3015	0.3208	0.3356	0.3579	0.3059	0.2985	0.2911
re	0.3117	0.3333	0.3203	0.2900	0.3203	0.3117	0.3290	0.3030	0.3203	0.3203
RD	0.3122	0.3348	0.3084	0.3235	0.3348	0.2972	0.3122	0.3272	0.3122	0.2972

Calculate Standardization of Vectors Weighted Vector Matrix

1. Calculate weight vector according to D1 and D2, separately by Subjective Weight Mode, Objective Weight Mode and Subjective-objective Weight Mode, show as the Table 3.
2. Standardization of Weighted Vector Matrix can be achieved according to Table 3 and Table 2, show as the Table 4.

Table 3. Weighted Vector

D W	D_1				D_2			
	Pr	RT	re	RD	Pr	RT	re	RD
subjective	0.08	0.48	0.24	0.2	0.13	0.15	0.33	0.39
objective	0.15	0.35	0.31	0.19	0.15	0.35	0.31	0.19
subjective-objective	0.10	0.42	0.27	0.21	0.14	0.29	0.32	0.25

Table 4. Standardization of Weighted Vector Matrix

		S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}
sub	Pr	0.0226	0.0297	0.0178	0.0238	0.0273	0.0297	0.0238	0.0250	0.0273	0.0238
	Rt	0.1611	0.1433	0.1490	0.1447	0.1540	0.1611	0.1718	0.1468	0.1433	0.1397
	re	0.0748	0.0800	0.0769	0.0696	0.0769	0.0748	0.0790	0.0727	0.0769	0.0769
	Rd	0.0624	0.0670	0.0617	0.0647	0.0670	0.0594	0.0624	0.0655	0.0624	0.0594
obj	Pr	0.0423	0.0557	0.0334	0.0446	0.0512	0.0557	0.0446	0.0468	0.0512	0.0446
	Rt	0.1175	0.1045	0.1086	0.1055	0.1123	0.1175	0.1253	0.1071	0.1045	0.1019
	re	0.0966	0.1033	0.0993	0.0899	0.0993	0.0966	0.0102	0.0939	0.0993	0.0993
	Rd	0.0593	0.0636	0.0586	0.0615	0.0636	0.0565	0.0593	0.0622	0.0593	0.0565
Sub-obj	Pr	0.0282	0.0371	0.0223	0.0297	0.0342	0.0371	0.0297	0.0312	0.0342	0.0297
	Rt	0.1410	0.1254	0.1304	0.1266	0.1347	0.1410	0.1503	0.1285	0.1254	0.1223
	re	0.0842	0.0900	0.0865	0.0783	0.0865	0.0842	0.0888	0.0818	0.0865	0.0865
	Rd	0.0656	0.0703	0.0648	0.0679	0.0703	0.0624	0.0656	0.0687	0.0656	0.0624

Table 5. The results of d_i^*, d_i^0 and C_i^* according to D_1

	Subjective Weight			Objective Weight			Subjective-objective Weight		
	d_i^*	d_i^0	C_i^*	d_i^*	d_i^0	C_i^*	d_i^*	d_i^0	C_i^*
S_1	0.0230	0.0142	0.3819	0.0332	0.0176	0.3462	0.0210	0.0145	0.4086
S_2	0.0124	0.0313	0.7160	0.0380	0.0407	0.5167	0.0152	0.0287	0.6538
S_3	0.0111	0.0268	0.7072	0.0191	0.0243	0.5597	0.0104	0.0263	0.7162
S_4	0.0132	0.0282	0.6820	0.0292	0.0313	0.5177	0.0147	0.0254	0.6337
S_5	0.0174	0.0208	0.5443	0.0365	0.0298	0.4501	0.0176	0.0195	0.5262
S_6	0.0261	0.0119	0.3128	0.0444	0.0286	0.3914	0.0258	0.0110	0.2993
S_7	0.0330	0.0115	0.2584	0.0420	0.0178	0.2977	0.0294	0.0133	0.3105
S_8	0.0125	0.0263	0.6774	0.0307	0.0310	0.5020	0.0137	0.0238	0.6342
S_9	0.0115	0.0297	0.7200	0.0341	0.0366	0.5180	0.0136	0.0266	0.6614
S_{10}	0.0101	0.0334	0.7683	0.0279	0.0361	0.5636	0.0114	0.0302	0.7259

Confirm the Ideal Solution and Negative Ideal Solution. X^* and X^0 may be calculated according to the Table 4 and Formula (7), (8).

1. by subjective weight mode, the following can be got:
   $X_1^* : (0.0178, 0.1397, 0.0800, 0.0670); X_1^0 : (0.0297, 0.1718, 0.0696, 0.0594)$
2. by objective weight mode, the following can be got:
   $X_2^* : (0.0334, 0.1019, 0.1033, 0.0636); X_2^0 : (0.0557, 0.1253, 0.0899, 0.0565)$

Table 6. The results of d_i^*, d_i^0 and C_i^* according to D_2

	Subjective Weight			Objective Weight			Subjective-objective Weight		
	d_i^*	d_i^0	C_i^*	d_i^*	d_i^0	C_i^*	d_i^*	d_i^0	C_i^*
S_1	0.0153	0.0152	0.4989	0.0332	0.0176	0.3462	0.0178	0.0161	0.4754
S_2	0.0193	0.0223	0.5359	0.0380	0.0407	0.5167	0.0209	0.0240	0.5348
S_3	0.0115	0.0233	0.6696	0.0191	0.0243	0.5597	0.0096	0.0269	0.7373
S_4	0.0179	0.0164	0.4793	0.0292	0.0313	0.5177	0.0178	0.0205	0.5349
S_5	0.0166	0.0190	0.5332	0.0365	0.0298	0.4501	0.0192	0.0178	0.4808
S_6	0.0262	0.0079	0.2317	0.0444	0.0286	0.3914	0.0271	0.0095	0.2588
S_7	0.0165	0.0171	0.5087	0.0420	0.0178	0.2977	0.0228	0.0167	0.4228
S_8	0.0157	0.0166	0.5137	0.0307	0.0310	0.5020	0.0165	0.0192	0.5386
S_9	0.0183	0.0151	0.4522	0.0341	0.0366	0.5180	0.0182	0.0206	0.5306
S_{10}	0.0181	0.0171	0.4866	0.0279	0.0361	0.5636	0.0146	0.0240	0.6218

3. by subjective-objective weight mode, the following can be got:
   $X_3^* : (0.0223, 0.1223, 0.0900, 0.0703); X_3^0 : (0.0371, 0.1503, 0.0783, 0.0624)$

Calculate Distance. This part will calculate distance from every scheme to ideal and negative ideal solutions. d_i^* and d_i^0 may be calculated according to the Formula (9) and Table 4, the results are as shown in Table 5.

Calculate Distance

1. C_i^* can be used to calculate every service case's on-line index, C_i^* may be calculated according to the Formula (10) and Table 5, the results are as shown in Table 5.
2. Based on the method above, d_i^*, d_i^0 and C_i^* can be achieved as the following when the attribute comparison pattern is D_2, the results are as shown in Table 6.

4.3 Result Analysis

According to calculation result, the order of model services with three kinds of weight mode are compared and analyzed.

Adopting Subjective Weight Mode. Only user's hobby information is considered calculating weight when subjective mode is adopted. Decision-makers' evaluation will be stated by the comparison of matrix D_1 and D_2. When D_1 is adopted, the list order for achieved every service attribute case is $S_{10} \succ S_9 \succ S_2 \succ S_3 \succ S_4 \succ S_8 \succ S_5 \succ S_1 \succ S_6 \succ S_7$. When D_2 is adopted, the list order for achieved every service attribute case is $S_3 \succ S_2 \succ S_5 \succ S_8 \succ S_7 \succ S_1 \succ S_{10} \succ S_4 \succ S_9 \succ S_6$. Comparison of two list order shows great changes exist. Evaluation index of every service attribute case C_i^* has big difference while D_1 and D_2 are separately adopted, evaluation of some special attribute case exists jerky nature., which resulted from the followings: weight mode result relying on

the index of attribute comparison pattern; attribute comparison pattern having hobby information subjectively decided by decision-makers, which directly influences evaluation result. This indicates when subjective weight mode is adopted, evaluation of service attribute is decided by users' hobby and has great free rein.

Adopting Objective Weight Mode. Only real index of every attribute target is considered for weight mode calculation when objective weight mode is adopted, and has no relation with users' hobby information. Pattern D_1 and D_2 will have the same list order, that is $S_{10} \succ S_3 \succ S_9 \succ S_4 \succ S_2 \succ S_8 \succ S_5 \succ S_6 \succ S_1 \succ S_7$. Even his method is instructed by strict mathematics theory, it cannot reflect users' hobby, which shows it cannot care for different user' individual request.

Adopting Subjective-objective Weight Mode. Weight mode calculation will consider objective information of every attribute target index and users' hobby when subjective-objective weight mode is adopted. Users' hobby is stated by attribute comparison pattern D_1 and D_2. When D_1 is adopted, the list order for achieved every service attribute case is $S_{10} \succ S_3 \succ S_9 \succ S_2 \succ S_8 \succ S_4 \succ S_5 \succ S_1 \succ S_7 \succ S_6$. When D_2 is adopted, the list order for achieved every service attribute case is $S_3 \succ S_{10} \succ S_8 \succ S_4 \succ S_2 \succ S_9 \succ S_5 \succ S_1 \succ S_7 \succ S_6$.The comparison of the two result shows small change exist. The change indicates users' hobby plays its function, while small change indicates users' hobby information does not change the result of service order list. Attribute target index of service objectively stabilizes order list result of service. This method is more scientific as it shows users' hobby, also promises its objectivity.

5 Conclusion

How to select model service to satisfy the request is a big trouble for the users of model service. This article offers a detailed analysis on model service research, then by a further step, Abstract model service into multi-attribute decision problem. TOPSIS is offered to evaluate quality attribute of model service. How to calculate weight mode is also analyzed by offering three kinds of methods: subjective weight mode, objective weight mode and subjective-objective weight mode. The reasonability and availability of this model service is proved by exact samples and analysis.

Acknowledgments. This work was supported by the NSFC (Grant. 60773188) and Natural Science Foundation of the Education Department of Henan Province (Grant 2008B120002) and Doctoral Research Foundation of Henan Institute of Engineering.

References

1. Chi, J.Y., Chen, X.G.: A Model of Grid Based Decision Support System. Computer Science 33, 121–124 (2004)

2. Wang, Z.W., Hu, M.S., Chen, X.G.: A Model of Agent and Grid-based Open DSS. Wseas Transactions On Computer Research 2, 73–78 (2007)
3. Chen, X.G., Chi, J.Y., Zhou, D.Q., Sun, L.: A Study on the Model of Open Decision Support System Based on Agent Grid. In: Proceedings of International Conference on Grid and Cooperative Computing (GCC 2004), Wuhan, pp. 769–776 (2004)
4. Li, Y.H., Chen, Y.K., Lu, Z.D.: The User Centered Data Mining Ontology Development on Universal Knowledge Grid. Computer Science 34, 159–164 (2007)
5. Yang, L., Dai, Y., Zhang, B., et al.: A dynamic Web Service Composite Platform based on QoS of Services. In: Shen, H.T., Li, J., Li, M., Ni, J., Wang, W. (eds.) AP-Web Workshops 2006. LNCS, vol. 3842, pp. 709–716. Springer, Heidelberg (2006)
6. Yue, C.Y.: Theory and Methods for Decision. Science Press, Beijing (2003)
7. Hwang, C.L., Yoon, K.: Multiple Attribute Decision Making: Methods and Applications. Springer, Berlin (1981)
8. Guo, Y.J.: Theory and Method of Synthetic Evaluation. Science Press, Beijing (2006)
9. Fan, Z.P., Zhang, Q., Ma, J.: An Integrated Approach to Determining Weights in Multiple Attribute Decision Making. Journal of Management Sciences 1.1, 50–53 (1998)
10. Ma, J., Fan, Z.P., Huang, L.H.: A Subjective and Objective Integrated Approach to Determine Attribute Weights. European Journal of Operational Research 112, 397–404 (1999)

Implementing Power System Management via Semantic Web Services Composition

Qing Liu[1], Jinyu Wen[2], and Haishun Sun[2]

[1] Information School, Renmin University of China
qliu@ruc.edu.cn
[2] College of Electrical and Electronic Engineering, Huazhong University of Sci.& Tech.
jinyu.wen@hust.edu.cn, haishunsun@mail.hust.edu.cn

Abstract. The increasing amount of application software in power system demands for an effectual software integration pattern. Web services satisfy the requirement. But WSDL and UDDI standards lack the capability of semantic representation. In this paper, we present a semantic description for service based on OWL-S, and design an exactor which gets IPOE information from WSDL/UDDI, propose an architecture SM4PS based on OWL-S and power system domain ontology for service discovery and composition. A prototype of composing service for a provincial power system operation management is shown.

Keywords: Semantic, Web services, OWL-S, Power system.

1 Introduction

In recent years, China has undergone a rapid development in power system. To ensure the secure and stable operation of the power system, it is required to develop and apply more advanced software. Within power companies, software systems had evolved over the decades and have become more and more complex, the systems are numerous having lots of applications like CRM, GIS, and SCADA from different vendors like ABB, SIEMENS and other IT companies. The software architecture had changed from central to distributed, and now the web-based architecture. These different applications have to integrate and exchange data. Therefore, the research on the problem of application software integration and interaction in electric power system is increasing importance and urgency.

In past years, some well-known systems such as CORBA, COM/DCOM (only for Microsoft systems), and Java RMI achieve communication of heterogeneous systems but not coordination between components. Their communication does not guarantee the exploitation of the full functionality in some products, especially between heterogeneous systems. So such systems are in fact successful only with platforms of the same vendor.

Web services are self-contained, Web-enabled applications capable not only of performing business activities on their own, but also possessing the ability to engage other Web services in order to complete higher-order business transactions [1]. Web services Protocol Stack including SOAP, WSDL, UDDI, and BPEL enable the

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 1154–1160, 2009.

interoperation and analysis of existing software component or complex services which is possibly offered by different service providers. All these standards are XML format document. Simple Object Access Protocol (SOAP) is a web service communication and invocation protocol. The Web Services Definition Language (WSDL) outlines what input the service expects and what output it returns. Universal Description, Discovery and Integration (UDDI) is a registry standard for discovering web services. UDDI describes businesses and classifies services, but it doesn't provide information on what a service does. The Business Process Execution Language (BPEL) is used to formally specify business processes and interactions for web services. Overall, Web services technology is basically a syntactical solution and lacking the semantic part. So finding correct web service is difficult despite the existence of UDDI registries because descriptions are in text and can only be searched by keyword, and composing web services is still considered challenging and requires an expert. Semantic description of web services is an attractive solution. Semantic Web services are Web services which have been marked or annotated with machine-interpretable semantic markup, in the form of ontologies [2]. Semantic interface description languages include OWL-S [3] or WSMO [4]. At present, there are many research works on ontology building for power system [5, 6].

In this paper we focus on the composition of power system management applications based on web services. We propose two steps to compositing power system services, first, the services semantic information are described, and domain-specific ontologies are created based on OWL. Then services composition is made more automatically and correctly.

2 Semantic Description for Power System Management Services

Two main issues are involved in power system management services composition. First of all Web service discovery aims at reaching the user goal and selecting the suitable service. The second issues is to plan a workflow which describes how these services interact and how the functionality they offer could be orchestrated and monitored at runtime. For the WSDL and UDDI have not explicit representation of the whole semantic description of Web services, some extended semantic definitions are given in the subsections below.

2.1 Definition for Semantic Service of Power System

Web Ontology Language - Services (OWL-S) [3] is ontology, based on OWL, to semantically describe web services. It enables Web service discovering, selecting, composing and interoperating automatically. OWL-S is characterized by three modules: Service Profile, Process Model and Grounding. Service Profile describes the capabilities of web services, it includes four elements IPOE:

Input: Set of necessary inputs that the requester should provide to invoke the service.

Output: Results that the requester should expect after interaction with the service provider is completed.

Preconditions: Set of conditions that should hold prior to service invocation.

Effects: Set of statements that should hold true if the service is invoked successfully
Process Model describes how a service works; it defines three types of process:

Atomic processes: directly invoked by an agent, have no subprocesses, executed in a single step.
Composite processes: consist of other non-composite or composite processes. They have a composedOf property, by which the control structure of the process is indicated, using a ControlConstruct subclasses.
Simple processes: abstract concepts, used to provide a view of some atomic process, or a simplified representation of some composite process.

Definition 1. Domain ontology DO, a set of definitions of elements in power system.

Definition 2. Service is defined as s= (I_s, P_s, O_s, E_s), where I_s is the input list, P_s is the preconditions, O_s is the output list, E_s is the service affect.

Definition 3. Services container SC, a set of Web services. SC= {s1, s2,……, sn}

Definition 4. Services query is defined as Q=(I_Q, P_Q, O_Q, E_Q), where I_Q is the expected service input list, P_Q is the expected service preconditions, O_Q is the expected service output list, E_Q is the expected service affect.

Definition 5. Services discovery: Given a services container SC and a query Q, automatically finding a set of services S from SC, S={s| s=(I_s, P_s, O_s, E_s), s∈SC, $I_Q \subseteq I_s$, $P_Q \subseteq P_s$, $O_Q \subseteq O_s$, $E_Q \subseteq E_s$}.

2.2 Description of services composition

In this paper, directed graph is used as the structure of the services composition.

Definition 6. Let G= (V, E, C,vs,vg) as a connected, directed graph with given weight, where V={ v1,……,vn } is the vertices collection of the graph G, every vertex vi is related to a service si; E is the collection of directed edges of the graph G, if (vi, vj)∈E,then vi ∈ V, vj ∈ V;C = (c1, ……, ck), is the k global constraints which is brought out by users for the composite service; vs is the start node that users bring up; vg is the end node that users bring up.

Definition 7. Multi-constrained path: given the directed graph G= (V, E, C,vs,vg), a path P from start node to the end node is called multi-constrained path.

3 Composition Architecture for Power System Service

The SM4PS (Services Matcher for Power System) architecture (Fig 1) is proposed to tackle the Web service discovery challenge. The SM4PS includes four modules: services publish, query, composition module, services container, services-query matcher and OWL-S extractor. Once a service provider publishes his service in services container, the SM4PS will produce the semantic description and select the services automatically.

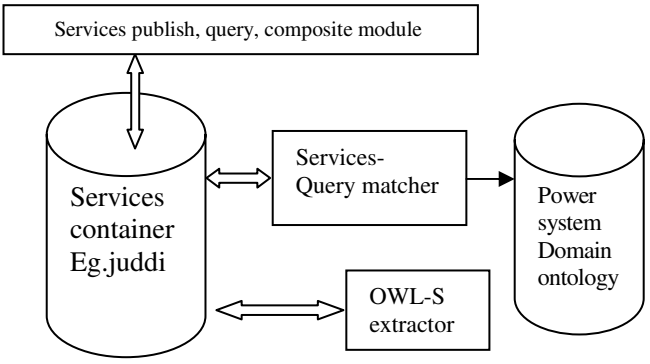


Fig. 1. The architecture of services matcher SM4PS

The services container can be UDDI registry or not. In SM4PS we use JUDDI registry and augment the JUDDI with semantic definition. When a service provider publishes his service in our JUDDI, the OWL-S extractor will get information from tModel. If the services container is not a UDDI registry, the OWL-S extractor will fetch service profile information form the WSDL file. The OWL-S/WSDL/UDDI mapping is given in table 1 below.

Table 1. OWL-S/WSDL/UDDI Mapping

OWL-S Profile	WSDL Document	UDDI
input	wsdlInputMessage	Input_TModel
inputtype	wsdlInputMessagepart	
Output	wsdlOutputMessage	Output_TModel
Outputtype	wsdlOutputMessagepart	
Preconditions		Preconditions_TModel
Effects		Effect_TModel

Some domain ontologies for power system descriptions exist [5, 6, and 7] and building them is a challenging task. A major impediment is the lack of guidelines on how to build such ontologies, what knowledge they should contain and what design principles they should fellow. The process of building power system domain ontology is out if scope of the paper. According to A. Bernaras's suggestion [7], power system domain ontologies are complex and may be a combination of small-scale ontologies. For example, the transport ontology describes component of the network, including electrical components and compound structures. Breaker is a kind of electrical components; Breakers are further specialized into central breaker, lateral breaker, bus-bar coupling breaker, and normal breaker (Fig.2).

When users query some services, the services-Query matcher will match the inputs and the outputs of the request against the inputs and the outputs of correlating advertisements in the services container based on domain ontology. The matching process is efficient.

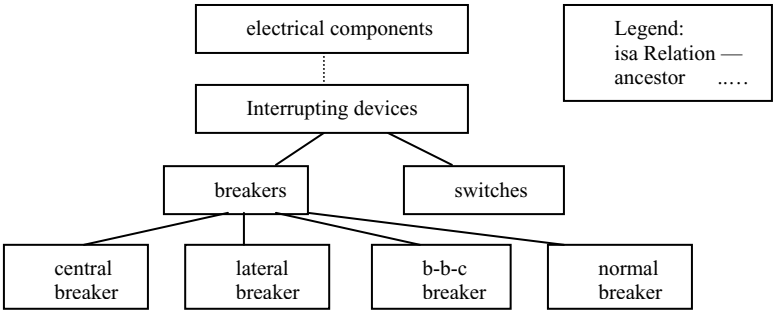


Fig. 2. A fragment of the power system ontology

4 The Design and Implement of a Prototype Composition System

This section presents an example to show how the composing services works in practice. In a dispatch center of a provincial power system in China, when a breaker Kn of a transmission line in a substation Si is going to be placed under repair, changes to the system configuration and load rearrangement will be done by different applications software (services). These services have the function of getting real time information of the power system, power system generation planning and equipment overhauling arrangement, protective relaying operation and management, operation sheet generating for operator.

Consider the above scenario with three different Web services (table 2): Daily schedule service receives a device repairing requirement from a substation Si, it returns a plan P1. In accordance with the arrangement of Daily schedule service, Protective relay service returns the necessary arrangement R1 of the protective relay devices used, Operation sheet service returns the operation detail O1.

Table 2. Semantic Web services for power system management

Web Services	Daily schedule	Protective relay	Operation sheet
Input	Kn, Si	Kn, Si	Kn, Si
Output	P1	R1	O1
Preconditions	PS	PS,P1	PS,P1,R1
Effects	E_{Ds}	E_{Pr}	E_{Os}

The workflow of the three services is shown in Fig 3. Based on the semantic services matching method described in section 3, a composite service for power system operation management has been created; the Process Model in OWL-S is following:

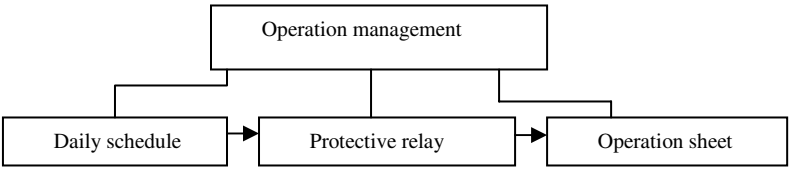


Fig. 3. The example of operation management process

```
<rdfs:Class rdf:ID="OperationManagement">
  <rdfs:subClassOf rdf:resource="#CompositeProcess" />
  <rdfs:subClassOf rdf:resource="http://www.daml.org/Process#Sequence" />
  <daml:subClassOf>
    <daml:Restriction>
      <daml:onProperty rdf:resource="http://www.daml.org/Process#components"/>
       <daml:toClass>
          <daml:subClassOf>
            <daml:unionOf rdf:parseType="daml:collection">
              <rdfs:Class rdfs:about="#Daily schedule" />
              <rdfs:Class rdfs:about="#Protective Relay " />
              <rdfs:Class rdfs:about="#Operation sheet" />
            </daml:unionOf>
          </daml:subClassOf>
       </daml:toClass>
    </daml:Restriction>
  </daml:subClassOf>
</rdfs:Class>
```

5 Conclusion and Future Work

In this paper we outlined the main challenges faced by power system application software. Then we showed the advantage of Web services for software integration in power system. We analyzed the shortcoming of Web services existing standards WSDL, UDDI which lack the capability of semantic representation, and presented a semantic description for service based on OWL-S. The SM4PS architecture was designed to discover and compose service based on OWL-S and power system domain ontology. An exactor which gets IPOE information from WSDL/UDDI was proposed. A prototype of composing service for a provincial power system operation management shows the availability and flexibility of our implementation.

For further studies we will extend the current work in service composition with semantic context of power system, and do more research on semantic matching method.

Acknowledgments. The work is funded by the National Natural Science Foundation of China (Foundation No. 60773217).

References

1. Jian, Y.: Web Services Componentization. Comunications of the ACM 46, 35–40 (2003)
2. Sycara, K.P., Paolucci, M., et al.: Automated Discovery, Interaction and Composition of Semantic Web Services. J. Web Sem 1, 27–46 (2003)
3. Ankolekar, A., Burstein, M., et al.: OWL-S 1.1 Release, OWL-based Web Service Ontology. Web-Ontology Working Group at the World Wide Web Consortium (2004)
4. Roman, D., Keller, U., Lausen, H.: WSMO – Web Service Modeling Ontology. Digital Enterprise Research Institute (DERI) (2004)
5. Ma, Q., Guo, J., Yang, Y.: An Ontology for Power System Operation Analysis. In: 2004 IEEE International Conference on Electric Utility Deregulation, Restructuring and Power Technologies, pp. 597–601. IEEE Press, New York (2004)
6. Mathias, U.: Semantic Interoperability within the Power Systems Domain. In: 1st International Workshop on Interoperability of Heterogeneous Information Systems, pp. 39–44. ACM, New York (2005)
7. Bernaras, A., Laresgoiti, I., Bartolomc, N., Corera, J.: An Ontology for Fault Diagnosis in Electrical Networks. In: The International Conference on Intelligent Systems Applications to Power Systems, pp. 199–203. IEEE Press, New York (1996)
8. Paolucci, M., Kawamura, T., et al.: Semantic Matching of Web Services Capabilities. In: Horrocks, I., Hendler, J. (eds.) ISWC 2002. LNCS, vol. 2342, pp. 333–347. Springer, Heidelberg (2002)

Optimal Auction Model Analysis and Mechanism Design of Indivisible Goods

Congjun Rao[1,2], Yong Zhao[1], Huiling Bao[1], and Qing Wang[1]

[1] Institute of Systems Engineering, Huazhong University of Science and Technology, Wuhan 430074, China
[2] College of Mathematics and Information Science, Huanggang Normal University, Huanggang 438000, China
raocjun79@163.com

Abstract. In this paper, an optimal auction model that maximizes the seller's expected utility is proposed for a kind of indivisible goods. Firstly, the correlation of bidders' valuations and the asymmetry of the private information are analyzed. Then, the properties of the optimal auction are given. Thirdly, the feasibility of optimal auction is discussed. Finally, an example of the optimal auction is given to show how to apply the discriminatory auction to realize the optimal auction mechanism. Therefore, this paper effectively generalizes the auction models with single-unit.

Keywords: Indivisible goods, optimal auction, mechanism design.

1 Introduction

With the development of economy, the multi-unit auctions are widely applied in our life. People have more concern about the theory research on multi-unit auctions, and the mechanism design of multi-unit auctions is becoming one of the most active research fields in auction theory[1].

Ortega Reichert[2] analyzed the properties of sequential English auction, discriminatory price auction, and uniform price auction, and presented revenue equivalence theorem of multi-unit auctions for the first time. Harris and Raviv[3] first gave an optimal conclusion, i.e., if the bidders' valuations are independent, and follow uniform distribution, then the auction mechanism in Ref.[2] is optimal. Maskin[4] gave a complete characterization for the multi-unit auctions and generalized the conclusions in Ref.[3] to any valuation distribution. Especially, for the auctions of divisible goods, Back and Zender[5, 6] compared the single object auction with divisible multi-unit auctions, and designed a special uniform price auction mechanism of divisible goods. This is a new idea of studying the auction of divisible goods. Thenceforward, Wang and Zender[7] derived equilibrium bidding strategies in divisible good auctions for asymmetrically informed risk neutral and risk averse bidders when there is random noncompetitive demand. Recently, Kremer and Nyborg[8] used a model of fixed supply divisible-good auctions, to study the effect of different rationing rules on the set of equilibrium prices. Damianov[9] then showed that a low-price equilibrium cannot exist in a uniform price auction with endogenous supply, if the seller employs

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 1161–1170, 2009.

a proportional rationing rule and is consistent when selecting among profit-maximizing quantities. Indranil Chakraborty[10] studied the asymptotic price in the uniform price auction, the results showed that the expected price becomes large depend only on the aggregate of the marginal distributions of each bidder's marginal values, and not on the correlation between the marginal values. These are all the important research results in auction theory in the past few years. However, most of these research results are obtained based on some simple and especial conditions, such as unitary demand for every bidder's valuation follows uniform distribution, the bid price of bidders are discrete, and so on. When these conditions are changed, the corresponding conclusions need to be reconsidered.

Based on these existing studies, an optimal auction mechanism and model that maximize the seller's expected utility are studied in this paper. In this model, the correlation of bidders' valuations are considered, and the bidders are unsymmetrical, and the probability is used to describe the optimal auction mechanism.

2 The Assumptions

Without loss of generality, we give the following assumptions.

Assumption 1. There is only one seller, and he is risk neutral.

Assumption 2. There are m units of indivisible goods to be sold.

Assumptions 1 and 2 describe the supply side of the model. The following assumptions describe the demand side and the information structure.

Assumption 3. There are $n(n>1)$ bidders, and they are all risk neutral. They all want to maximize their expected profit.

Assumption 4. Each potential bidder has his own private information about the value of the good, T_i, and any two variables T_i and T_j with $i \neq j$ are independent.

Assumption 5. The value of T_i is only known by bidder i, other bidders do not observe the realization of T_i and treat it as a draw from a cumulative distribution $F_{T_i}(\cdot)$, with support $\mathbf{H}_i = [\underline{t}_i, \bar{t}_i]$. The density function of $F_{T_i}(\cdot)$ is denoted as $f_{T_i}(\cdot)$.

Assumption 6. The value of the k-th unit of the good to potential bidder i, $V_{ik}(\boldsymbol{T})$, satisfies $E\{V_{ik}(\boldsymbol{T}) \mid \boldsymbol{T} = \boldsymbol{t}\} = \sum_{j=1}^{n} \gamma_{ikj} \overline{V}_{ikj}(t_j)$, where the function $\overline{V}_{ikj}(t_j)$ is the expected effect of bidder j's information on bidder i's marginal valuation of the k-th unit, and the functions $\overline{V}_{ikj}(\cdot)$ are differentiable and increasing. The parameter γ_{ikj} ($j \neq i$) is a influence coefficient that the bidder j's information will affect bidder i, and they are nonnegative. γ_{iki} is positive.

In a multi-unit auction model, the relationship between the valuations for the several units needs to specify. The following assumption 7 gives the downward sloped demand function condition.

Assumption 7. The expected valuation for additional units is nonincreasing, i.e., for all t_j, $\sum_{j=1}^{n}\gamma_{ikj}\overline{V}_{ikj}(t_j) \geq \sum_{j=1}^{n}\gamma_{i,k+1,j}\overline{V}_{i,k+1,j}(t_j)$, $k \in \{1,\cdots,m-1\}$.

The last two assumptions are given to guarantee that in the optimal auction the seller will not want to use stochastic allocation rules.

3 Optimal Auction Mechanism

Based on above Assumptions and conditions, we discussed the optimal auction of indivisible goods.

Definition 1. An allocation is a vector $(k_1,\cdots,k_n)$, such that each element k_i, the number of units allocated to bidder i, is a nonnegative integer and satisfies $\sum_{i=1}^{n}k_i \leq m$. The set of all allocations is denoted by $A(m,n)$.

A multi-unit auction is defined as follows.

Definition 2. An auction is any pair of functions (p,c) with $c_i: T_i \to R$, and such that there exists a function $P: \boldsymbol{A}(m,n)\times\boldsymbol{T} \to [0,1]$. For all i, k, $(k_1,\cdots,k_n)$ and $\boldsymbol{t}$, satisfy the following conditions[11].

$$p_{ik}(t) = \sum_{\substack{(k_1,\cdots,k_n)\in A(m,n)\\ k_i \geq k}} p(k_1,\cdots,k_n;\boldsymbol{t}); \tag{1}$$

$$\sum_{(k_1,\cdots,k_n)\in A(m,n)} p(k_1,\cdots,k_n;\boldsymbol{t}) \leq 1; \tag{2}$$

$$p(k_1,\cdots,k_n;\boldsymbol{t}) \geq 0. \tag{3}$$

In an auction (p,c), given all announcements $\boldsymbol{t}$, $p_{ik}(\boldsymbol{t})$ denotes the probability that bidder i will receive at least k units, and $c_i(\boldsymbol{t})$ is bidder i's expected payment; $p(k_1,\cdots,k_n;\boldsymbol{t})$ is the probability that the allocation $(k_1,\cdots,k_n)$ is implemented.

The seller's utility from any auction (p,c) is determined by the expected payments of the bidders, and can be denoted as $U_0(p,c) = E\left\{\sum_{i=1}^{n}c_i(T_i)\right\}$.

In addition, the bidders pay attention to not only about their expected payments but also about the number of units they will receive as well, and bidder i's utility (Weber, 1983) [12] can be written as $U_i(p,c,t_i) = E\{\sum_{k=1}^{m}V_{ik}(\boldsymbol{T})p_{ik}(\boldsymbol{T}) \mid T_i = t_i\} - c_i(t_i)$, where $p_{ik}(t)$ is the probability that bidder i receives at least k units. To simplify the characterization of feasible auctions, we denote bidder i's expected probability of getting at least k units by $Q_{ik}(p,t_i) = E\{p_{ik}(\boldsymbol{T}) \mid T_i = t_i\}$.

Next we discuss the incentive compatible of auction (p,c).

Definition 3. If for all i, t_i and $\hat{t}_i$, the following condition satisfies, i.e.,

$$U_i(p,c,t_i) \geq E\left\{\sum_{k=1}^{m} V_{ik}(\boldsymbol{T}) p_{ik}(\hat{t}_i, T_{-i}) \mid T_i = t_i\right\} - c_i(\hat{t}_i), \tag{4}$$

then we say the auction (p,c) is incentive compatible.

The right hand side of condition (4) is the expected utility of a bidder with type t_i who announces type $\hat{t}_i$. Therefore, condition (4) states that a truthful report maximizes the utility of any bidder. Condition (4) is equivalent to the following Proposition 1.

Proposition 1. An auction (p,c) is incentive compatibility if and only if, for all i, t_i and $\hat{t}_i$, the following condition satisfied:

$$U_i(p,c,t_i) \geq U_i(p,c,\hat{t}_i) + \sum_{k=1}^{m} \gamma_{iki}(\overline{V}_{iki}(t_i) - \overline{V}_{iki}(\hat{t}_i)) Q_{ik}(p,\hat{t}_i). \tag{5}$$

Proof. By Assumption 6, the right hand side of condition (4) can be written as

$$\begin{aligned}
&E\{\sum_{k=1}^{m} V_{ik}(\boldsymbol{T}) p_{ik}(\hat{t}_i, T_{-i}) \mid T_i = t_i\} - c_i(\hat{t}_i) \\
&= E\{\sum_{k=1}^{m} V_{ik}(T_i) p_{ik}(T_i) \mid T_i = \hat{t}_i\} - c_i(\hat{t}_i) + E\{\sum_{k=1}^{m} \gamma_{iki}(\overline{V}_{iki}(t_i) - \overline{V}_{iki}(\hat{t}_i)) p_{ik}(\hat{t}_i, T_{-i})\} \\
&= U_i(p,c,\hat{t}_i) + \sum_{k=1}^{m} \gamma_{iki}(\overline{V}_{iki}(t_i) - \overline{V}_{iki}(\hat{t}_i)) Q_{ik}(p,\hat{t}_i).
\end{aligned}$$

So the right hand side of condition (4) is equal to the right hand side of condition (5). Therefore, condition (4) is equivalent to condition (5).

In addition, the auction mechanism must satisfy the second constraint, i.e., any bidder may participate the auction freely. When he participate the auction, his expected utility is equal or greater than the expected utility when he didn't participate the auction.

Definition 4. If for all i and t_i, $U_i(p,c,t_i) \geq 0$. Then we call auction (p,c) is individually rational.

Definition 5. The auction (p,c) is a feasible mechanism if and only if (p,c) satisfies the conditions of incentive compatibility and individual rationality.

Next we discuss the properties of feasible auction.

Lemma 1. Suppose (p,c) is a auction, then for all i, k and $\boldsymbol{t}$, the following conditions must be satisfied:

(i) $\sum_{i=1}^{n}\sum_{k=1}^{m} p_{ik}(\boldsymbol{t}) \leq m$; (ii) $p_{ik}(\boldsymbol{t}) \geq p_{i,k+1}(\boldsymbol{t})$; (iii) $0 \leq p_{ik}(\boldsymbol{t}) \leq 1$.

The proof of Lemma 1 can be seen in [11].

In order to find the optimal auction, we must pay some attention to feasible auction. Based on Lemma 1, the following Proposition 2 provides a set of necessary conditions for feasible auctions.

Proposition 2. If an auction (p,c) is feasible, then, for all i, k, t_i and $\hat{t}_i$, we have

$$\sum_{k=1}^{m}\gamma_{iki}(\overline{V}_{iki}(t_i)-\overline{V}_{iki}(\hat{t}_i))Q_{ik}(p,t_i)\geq\sum_{k=1}^{m}\gamma_{iki}(\overline{V}_{iki}(t_i)-\overline{V}_{iki}(\hat{t}_i))Q_{ik}(p,\hat{t}_i) \tag{6}$$

$$\partial U_i(p,c,t_i)\Big/\partial t_i=\sum_{k=1}^{m}\gamma_{iki}\overline{V}'_{iki}Q_{ik}(p,t_i) \tag{7}$$

$$U_i(p,c,t_i)\geq 0 \tag{8}$$

$$\sum_{i=1}^{n}\sum_{k=1}^{m}p_{ik}(\boldsymbol{t})\leq m \tag{9}$$

$$p_{ik}(\boldsymbol{t})\geq p_{i,k+1}(\boldsymbol{t}) \tag{10}$$

$$0\leq p_{ik}(\boldsymbol{t})\leq 1\ . \tag{11}$$

Proof. First of all, we prove (6). By (5), we have

$$\sum_{k=1}^{m}\gamma_{iki}(\overline{V}_{iki}(t_i)-\overline{V}_{iki}(\hat{t}_i))Q_{ik}(p,\hat{t}_i)\leq U_i(p,c,t_i)-U_i(p,c,\hat{t}_i)$$

and

$$U_i(p,c,\hat{t}_i)\geq U_i(p,c,t_i)+\sum_{k=1}^{m}\gamma_{iki}(\overline{V}_{iki}(\hat{t}_i)-\overline{V}_{iki}(t_i))Q_{ik}(p,t_i)\,,$$

Therefore

$$\sum_{k=1}^{m}\gamma_{iki}(\overline{V}_{iki}(t_i)-\overline{V}_{iki}(\hat{t}_i))Q_{ik}(p,\hat{t}_i)\leq U_i(p,c,t_i)-U_i(p,c,\hat{t}_i)$$

$$\leq\sum_{k=1}^{m}\gamma_{iki}(\overline{V}_{iki}(t_i)-\overline{V}_{iki}(\hat{t}_i))Q_{ik}(p,t_i)\,.$$

Divide the terms in this inequality by $(t_i-\hat{t}_i)$, and take limits as $\hat{t}_i\to t_i$. The result in the center is $\partial U_i(p,c,t_i)\Big/\partial t_i$, while both bounds converge to $\sum_{k=1}^{m}\gamma_{iki}\overline{V}'_{iki}Q_{ik}(p,t_i)$, and (7) follows. Expression (8) follows directly from individual rationality, and (9)-(11) from Lemma 1.

Based on Proposition 2, we analyze the optimal auction.

Proposition 3. Suppose (p^*,c^*) is an optimal auction. Then a bidder reporting the lowest possible private signal has zero utility, i.e., $U_i(p^*,c^*,\underline{t}_i)=0$, where $\underline{t}_i$ denotes the lowest possible private signal, and, for all i and t_i, the expected payment satisfies

$$c_i^*(t_i)=E\left\{\sum_{k=1}^{m}(V_{ik}(\boldsymbol{T})p_{ik}^*(\boldsymbol{T})-\gamma_{iki}\int_{\underline{t}_i}^{T_i}\overline{V}'_{iki}(x)p_{ik}^*(x,\boldsymbol{T}_{-i})dx)\,|\,T_i=t_i\right\}\ . \tag{12}$$

Proof. The proof process can be seen in [11].

The characterization of optimal auctions is completed with a description of the optimal allocation probabilities, $p(\cdot)$. If the optimal $p(\cdot)$ corresponds to a deterministic auction, it must solve the problem specified in the following proposition 4.

Proposition 4. Consider an auction (p^*, c^*) satisfying (12). Let p^* solve

$$(D1): \max_{p(\cdot)} E\left\{\sum_{i=1}^{n}\sum_{k=1}^{m}\left(V_{ik}(\boldsymbol{T}) - \gamma_{iki}\overline{V}'_{iki}(T_i)\frac{1-F_{T_i}(T_i)}{f_{T_i}(T_i)}\right)p_{ik}(\boldsymbol{T})\right\}$$

$$s.t\begin{cases}\sum_{k=1}^{m}\gamma_{iki}(\overline{V}_{iki}(t_i) - \overline{V}_{iki}(\hat{t}_i))Q_{ik}(p,t_i) \geq \sum_{k=1}^{m}\gamma_{iki}(\overline{V}_{iki}(t_i) - \overline{V}_{iki}(\hat{t}_i))Q_{ik}(p,\hat{t}_i) \\ \sum_{i=1}^{n}\sum_{k=1}^{m}p_{ik}(\boldsymbol{t}) \leq m \\ p_{ik}(\boldsymbol{t}) \geq p_{i,k+1}(\boldsymbol{t}) \\ 0 \leq p_{ik}(\boldsymbol{t}) \leq 1\end{cases} \tag{13}$$

then (p^*, c^*) is an optimal auction.

Proof. Because (p^*, c^*) satisfies (12), by the derivation process of (12), we can obtain (7) and (8) in Proposition 2 are satisfied. In addition, combine the constraint conditions of problem $(D1)$ and equation (13), we can conclude (p^*, c^*) is an optimal auction.

In the objective function of problem $(D1)$, we set

$$E\{V_{ik}(\boldsymbol{T})\big|\boldsymbol{T}=\boldsymbol{t}\} - \gamma_{iki}\overline{V}'_{iki}(t_i)\frac{1-F_{T_i}(t_i)}{f_{T_i}(t_i)} \triangleq G_{ik}(t_i) \ . \tag{14}$$

Obviously, when the value of $p_{ik}(T)$ is determined, the greater the value of $G_{ik}(\cdot)$ is, the greater the value of $U_0(p,c)$ is. In the optimal auction, the seller will want to allocate the units that are associated with the highest contributions to his utility[11]. Thus, the ordering of the $G_{ik}(\cdot)$ will be important for to characterize the optimal auction.

Therefore, in an optimal auction, the seller can compute the values of $G_{ik}(\cdot)$ and rank them by gathering the bidder's information. The seller allocates m units to m bidders who have the highest positive value of $G_{ik}(\cdot)$ according to the order. If the numbers of positive $G_{ik}(\cdot)$ is less than m, then the seller will keep the remaining units.

4 A Mechanism Design for Auction of Indivisible Goods

4.1 Problem Analysis

Consider a private values model with two units and two bidders. Conditional on the individual signals, the expected valuations are given by

$$E\{V_{i1}(\boldsymbol{T})|\boldsymbol{T}=\boldsymbol{t}\} = \sqrt{t_i} \ , \tag{15}$$

$$E\{V_{i2}(\boldsymbol{T})|\boldsymbol{T}=\boldsymbol{t}\} = t_i \ . \tag{16}$$

Obviously, (15) and (16) satisfy Assumption 4. Suppose the private signals are independently drawn from a uniform distribution on [0,1], so (15) and (16) satisfy the Assumptions 5, 6 and 7.

By the assumption of private signals are independently drawn from an uniform distribution on [0,1], and from (14), (15) and (16), we can calculate the values of $G_{ik}(t_i)$: $G_{i1}(t_i)=\sqrt{t_i}-\frac{1-t_i}{2\sqrt{t_i}}$, $G_{i2}(t_i)=2t_i-1$.

For bidder i to receive at least one unit his private signal must be such that $G_{i1}(t_i)>\max\{0,G_{j2}(t_j)\}$; otherwise, the good is not allocated or it is allocated to the other bidder. But, for $G_{i1}(t_i)>0$ it must be satisfied $t_i>1/3$; similarly, for $G_{i1}(t_i)>G_{j2}(t_j)$, it must be satisfied $t_i>\left(\frac{2t_j-1+2\sqrt{t_j^2-t_j+1}}{3}\right)^2$ $(i\neq j)$. Thus, when bidder i receives at least one unit, his private signal must be such that

$$t_i>\max\left\{\frac{1}{3},\left(\frac{2t_j-1+2\sqrt{t_j^2-t_j+1}}{3}\right)^2\right\}.$$

Similarly, If the bidder i receives two units, his private signal must be such that $G_{i2}(t_i)>\max\{0,G_{j1}(t_j)\}$, then we can obtain $t_i>\max\left\{\frac{1}{2},\frac{3t_j+2\sqrt{t_j}-1}{4\sqrt{t_j}}\right\}$. Therefore, we define two functions, $\beta_1(\cdot)$ and $\beta_2(\cdot)$, which will be interpreted as the minimum winning announcement given the other bidder's announcement:

$$\beta_1(x)=\max\left\{\frac{1}{3},\left(\frac{2x-1+2\sqrt{x^2-x+1}}{3}\right)^2\right\},\quad \beta_2(x)=\max\left\{\frac{1}{2},\frac{3x+2\sqrt{x}-1}{4\sqrt{x}}\right\}.$$

Let (p^*,c^*) be an optimal auction. When $t_i>\beta_1(t_j)$, the optimal auction will allocate one unit to bidder i if $G_{i1}(t_i)>0$ and greater than $G_{j2}(t_j)$, so we have

$$p_{i1}^*(t)=\begin{cases}1 & t_i>\beta_1(t_j)\\ 0 & otherwise\end{cases}. \tag{17}$$

Similarly, when the bidder i receives two units, it satisfies

$$p_{i2}^*(t)=\begin{cases}1 & t_i>\beta_2(t_j)\\ 0 & otherwise\end{cases}. \tag{18}$$

Substitute (17) and (18) in (12), we have

$$c_i^*(t_i)=\begin{cases}0 & t_i\le 1/3\\ \int_0^{\beta_2(t_i)}\sqrt{\beta_1(x)}dx & 1/3<t_i\le 1/2\\ \int_0^{\beta_2(t_i)}\sqrt{\beta_1(x)}dx+\int_0^{\beta_1(t_i)}\beta_2(x)dx & t_i>1/2\end{cases}. \tag{19}$$

In Figure 1, we can get the optimal allocations for the possible vectors of private signals. The ordered pairs in each subset of the signals' space corresponds to the number of units allocated to the bidders: the first number is the number of units allocated to bidder 1, and the second number is the number of units allocated to bidder 2. If bidders have very low private signals, then they get no unit; with intermediate values they get one unit, and with high values they get two units. However, this is a private values model, the boundaries among low, intermediate and high signals, depend on the other bidder's signals. So these factors will bring some uncertainty to the allocated results. Therefore, the optimal mechanism cannot be implemented through a generalized standard auction. However, we can construct a modified discriminatory price auction that implements the optimal mechanism.

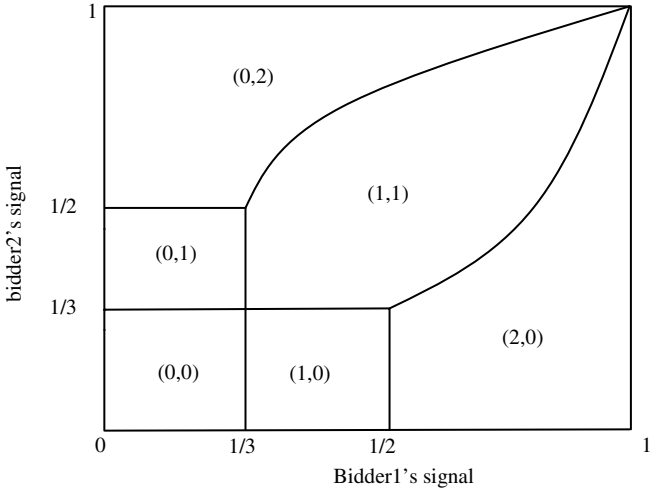


Fig. 1. Optimal allocations in an example of a two unit auction

4.2 Feasible Mechanism Design

In order to describe the modified discriminatory price auction, we suppose the bidders' expected valuations are as follows:

$$V_1(t_i) = E\left\{V_{i1}(\beta_1(T_j), T_j) \mid \beta_1(T_j) < t_i\right\},\ \ V_2(t_i) = E\left\{V_{i2}(\beta_2(T_j), T_j) \mid \beta_2(T_j) < t_i\right\}.$$

Each bidder i submits bids for the two units, d_{i1} and d_{i2}. From the submitted bids the seller calculates the modified bids $\tilde{d}_{ik}$ as : $\tilde{d}_{ik} = G_{ik}(V_k^{-1}(d_{ik}))$, $k = 1,2$, where $V_k^{-1}(\cdot)$ is the inverse function of $V_k(\cdot)$. For each unit received a bidder pays the value of the winning original bid; a bidder that receives no unit pays nothing.

To show that this modified discriminatory price auction implements the optimal mechanism, we show that if each bidder bids

$$\overline{d}_k(t_i) = V_k(t_i) \quad . \tag{20}$$

Which means the bids are $\overline{d}(t_i) = (\overline{d}_1(t_i), \overline{d}_2(t_i)) = (V_1(t_i), V_2(t_i))$, then $(\overline{d}(t_1), \overline{d}(t_2))$ is an equilibrium of the auction and generates expected payments as in (19). Next we prove this conclusion.

Suppose that bidder *j* uses bids $\overline{d}(t_j)$. Bidder *i* chooses bids d_{i1} and d_{i2} to maximize the objective

$$E\left\{(V_{i1}(\boldsymbol{T}) - d_{i1})P_{G_{j2}(T_j)\le G_{i1}(V_1(d_{i1}))} \mid T_i = t_i\right\} + E\left\{(V_{i2}(\boldsymbol{T}) - d_{i2})P_{G_{j1}(T_j)\le G_{i2}(V_2(d_{i2}))} \mid T_i = t_i\right\} \quad (21)$$

The objective function is separable in (21), so the optimization can be performed for each bid independently. The optimal bid for the first unit will solve the following problem $\underset{d_{i1}}{Max}\int_0^{\beta_2(V_1^{-1}(d_{i1}))}(\sqrt{t_i} - d_{i1})dx$.

Thus, the first order necessary condition for an optimal bid is

$$\frac{\beta_2'(V_1^{-1}(d_{i1}))}{d_{i1}'(V_1^{-1}(d_{i1}))}(\sqrt{t_i} - b_{i1}) - \beta_2(V_1^{-1}(d_{i1})) = 0. \quad (22)$$

If $d_{i1} = \overline{d}_1(t_i)$, then the condition (22) becomes

$$\overline{d}_1'(t_i) = \frac{\beta_2'(t_i)}{\beta_2(t_i)}(\sqrt{t_i} - d_{i1}). \quad (23)$$

Compare the Equation (23) to $\overline{d}_1'(t_i) = (\sqrt{t_i} - \overline{d}_1(t_i))\dfrac{\beta_2'(t_i)}{\beta_2(t_i)}$, which follows from (20). So in (23), we have $d_{i1} = \overline{d}_1(t_i)$. Similarly, it can be shown that $\overline{d}_2(t_i)$ is a bidder i's optimal bid for the second unit. By the symmetry, we conclude that $(\overline{d}(t_1), \overline{d}(t_2))$ is an equilibrium of the modified discriminatory price auction.

Furthermore, to check that the modified discriminatory price auction implements the optimal mechanism, we must prove that it allocates the goods optimally and that it yields the optimal expected payments.

By above proof, we know $(\overline{d}(t_1), \overline{d}(t_2))$ is an equilibrium of the modified discriminatory price auction, then the modified bids are $\tilde{d}_{ik} = G_{ik}(V_k^{-1}(d_{ik}))$ $= G_{ik}(V_k^{-1}(\overline{d}_k(t_i))) = G_{ik}(V_k^{-1}(V_k(t_i))) = G_{ik}(t_i)$. Hence, the units are optimally allocated. The expected payments are

$$\begin{aligned} c_i(t_i) &= E\{\overline{d}_1(T_i)P_{\beta_1(T_j)<T_i} + \overline{d}_2(T_i)P_{\beta_2(T_j)<T_i} \mid T_i = t_i\} \\ &= E\{V_{i1}(\beta_1(T_j), T_j)P_{T_j<\beta_2(T_1)} \mid T_i = t_i\} + E\{V_{i2}(\beta_2(T_j), T_j)P_{T_j<\beta_1(T_1)} \mid T_i = t_i\} \\ &= \int_0^{\beta_2(t_i)} \sqrt{\beta_1(x)}dx + \int_0^{\beta_1(t_i)} \beta_2(x)dx \end{aligned}$$

which is equal to the optimal expected payments given in (19).

5 Conclusions

This paper proposed an optimal auction mechanism for a kind of indivisible goods. Under the private signal and regularity conditions, the feasibility of optimal auction mechanism, i.e., incentive compatibility and individual rationality are analyzed, and the properties of the optimal auction are given. Finally, a private values model with two units and two bidders is considered to show how to apply the discriminatory auction to realize the optimal auction mechanism. Therefore, this paper generalizes the auction models with single-unit well, and it is significant in both theory and application.

Acknowledgements. This work was supported by the Postgraduate Science & Technology Innovation Fund of HUST (No. HF-06-007-08-184), the National Natural Science Foundation of China Grant (No.70771041), and the Project Sponsored by the State Education Ministry Scientific Research Foundation for the Returned Overseas Chinese Scholars.

References

1. Milgrom, P.: Putting Auction Theory to Work. Tsinghua University Press, Beijing (2006)
2. Ortega Reichert, A.: A Sequential Game with Information Flow. Chapter 8 in Models for Competitive Bidding under Uncertainty, Stanford University PhD Thesis, 232–254 (1981)
3. Harris, M., Raviv, A.: A Theory of Monopoly Pricing Schemes with Demand Uncertainty. American Economic Review 71, 347–365 (1981)
4. Maskin, E., Riley, J.: Optimal Multi-Unit Auction. In: The Economics of Missing Markets, Information and Games. Oxford University Press, New York (1989)
5. Back, K., Zender, J.F.: Auctions of Divisible Goods: on the Rationale for the Treasury Experiment. Review of Financial Studies 6, 733–764 (1993)
6. Back, K., Zender, J.F.: Auctions of Divisible Goods with Endogenous Supply. Economics Letters 73, 29–34 (2001)
7. Wang, J.J.D., Zender, J.F.: Auctioning Divisible Goods. Economic Theory 19, 673–705 (2002)
8. Kremer, I., Nyborg, K.: Divisible-Good Auctions: the Role of Allocation Rules. Rand Journal of Economics 35, 147–159 (2004)
9. Damianov, D.S.: The Uniform Price Auction with Endogenous Supply. Economics Letters 77, 101–112 (2005)
10. Indranil, C., Richard, E.W.: Asymptotic Prices in Uniform-Price Multi-Unit Auctions. Economic Theory 4, 983–987 (2005)
11. Fernando, B.: Multiple unit auctions of an indivisible good. Economic Theory 8, 77–101 (1996)
12. Weber, R.J.: Multi-Object Auction. New York University Press, New York (1983)

GUPTDSS: Grid Based Urban Public Transport Decision Support System

Yu Wang[1], Haigang Song [2], Liu Hong[1], and Xueguang Chen [1,*]

[1] Institute of Systems Engineering, Huazhong University of Science and Technology, Wuhan 430074, China
[2] Basic Research Service of the Ministry of Science and Technology of the P. R. China, Beijing 100862, China
xgchen9@mail.hust.edu.cn

Abstract. The Grid-based Urban Public Transport Decision Support System (GUPTDSS) is developed to establish an open high-level information environment to support transport information management and decision-making for solving traffic problems. GUPTDSS is characterized with stronger computing and resource sharing abilities to cope with the problems that the traditional Advanced Public Transport Systems could hardly resolve, such as large-scale transport data storage, high-performance transport simulation, and cross-domain cooperation etc. This paper presents an overview of services provided by the developed GUPTDSS and the architecture in which they fit. A set of system methods are discussed as well, including semantic representation of metadata model, distributed data management, and safety strategy etc. A case study is conducted to validate this architecture.

Keywords: GUPTDSS, Grid service, Traffic metadata, Data access, Security.

1 Introduction

The conflict between the continuous growth of urban economy and the effective utilization of limited urban space makes urban public transport system becoming more and more significant. It is reported that China lost 31.25 billion US dollars due to traffic congestion in 2003, which is 2 percent of the year's GDP and enough to build 500 kilometers of subway.

Currently new technologies have been utilized to improve the information gathering abilities of advanced public transport system (APTS) [1], including Geographic Information Service (GIS), Global Position System (GPS), monitors, and sensors etc. However, it is hard for them to provide effective problem-solving ability for high-level traffic decision support, since their architectures and standards can hardly solve some important APTS problems, such as how to store and share massive traffic data among different systems and organizations, integrate heterogeneous data to offer high-level information retrieval on the semantic level, and cooperate workflows among different domains [2].

* Corresponding author.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 1171–1180, 2009.

The emergence of Grid may effectively solve the problems mentioned above. The Grid [3] is an aggregation of geographically dispersed computing, storage and network resources, coordinated to deliver improved performance, higher quality of service, better utilization and easier access to data. It seems to be a suitable platform of GUPTDSS for the following reasons:

1. Large-scope data storage: The Grid provides a powerful infrastructure to store and manage massive traffic data through distributed storage resources.

2. Heterogeneous data integration: The Grid provides a series standard protocol groups to represent, access, transport and integrate data resources, which makes it possible for users to utilize heterogeneous, distributed and non-integrated transport data through a direct unique standard interface.

3. Large-scale computing: The Grid uses hundreds of computers distributed over wide regions as a single unified virtual computing environment. It allows users belonging to different traffic organizations to acquire right amount of computational power for computing-intensive traffic simulation and analysis.

4. Effective Cooperation: The Grid encapsulates all resources into services. It breaks the barrier amongst traditional close APTSs and makes these APTSs become a continuum to solve cross-domain traffic problems in cooperation.

The GUPTDSS proposed here aims to crash the bottlenecks of traditional APTS by exploiting Grid, GIS, GPS and General Packet Radio Service (GPRS) technologies to establish an open standard infrastructure for upper-layer transport decision-making support. The rest of this paper is organized as follows. Section 2 gives an overview of the infrastructure and architecture of GUPTDSS. Section 3 discusses some key methods used in GUPTDSS. Section 4 describes a typical service implemented on GUPTDSS. And finally, section 5 presents conclusions and outlines the future work.

2 The Infrastructure and Architecture of GUPTDSS

2.1 Infrastructure

The infrastructure of GUPTDSS is composed of four parts including data acquisition, data transfer, Grid service system, and information presentation. Fig.1 illustrates the GUPTDSS infrastructure.

Data acquisition system includes: GPS, in-vehicle terminals, on-board cameras, traffic webcam monitors, and sensors. To obtain real-time bus running information, in-vehicle terminals in the running bus get the GPS data through GPS satellite signal once in every 10 seconds, and transmit it simultaneously to the Grid storage nodes through GPRS/CDMA wireless communication network. Due to its large amounts and low real-time demands, video data acquired from on-board cameras is transmitted to the Grid storage nodes through wireless Access Point (AP) while bus arriving at final stations. Traffic data acquired from webcam monitors and sensors is transported to Grid storage nodes through cable networks for real-time analysis and control.

Grid service system includes four parts: computing cluster, storage cluster, global information server, and global schedule server. Both clusters are composed of a Dawn3000 supercomputer, a mini server and plenty of PCs. As a domain information

center, the mini server is responsible for domain information registry and communication between domain and global information server. PCs are responsible for the real implementation of requested services. Meanwhile, all services and resources in clusters are scheduled by Dawn3000 supercomputer. Global information server and global schedule server are mini servers that the front is in charge of global information request and consistency mapping, and the later is responsible for global job scheduling, monitoring and management.

Information presentation equipments includes:PDA, in-vehicle terminals, PC, Notebook PC, E-Stop. These equipments access a unique standard Grid portal to exchange messages through GPRS/CDMA wireless networks and Internet/Intranet.

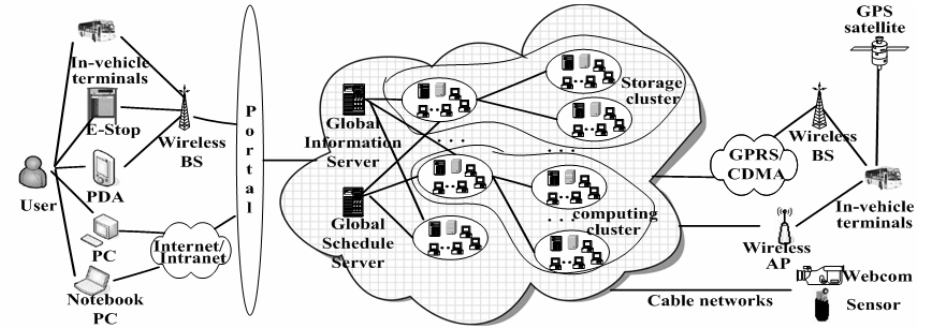


Fig. 1. The infrastructure of GUPTDSS

2.2 Architecture and Services

The architecture of GUPTDSS is a service oriented model based on Open Grid Service Architecture (OGSA) [4] which makes it a generic infrastructure for system integration. Its software components are classified into several hierarchical layers as shown in Fig.2:

1) User Layer falls into five categories, including urban transport managers, customers, decision-makers, researchers and other users;

2) Grid Portal is the access gateway to GUPTDSS Grid environment. There are two types of Grid Portal: application-based portal and user-based portal. The former one provides a workbench for users to submit and execute their own application programs; and the late one provides a uniform single access for users to seamlessly call the pre-developed transport Grid services. In the perspective of GUPTDSS users, user space is created as a virtual operation system. One can execute operations on user space as he/she can do on local systems.

3) Transport Services Library is the development layer of actual transport applications which provides outstanding transport Grid services to support transport decision-making. Its basic services include Dynamic Route Guidance (DRG) Service, Electronic Map Information (EMI) Service, Auto Vehicle Tracking (AVT) Service, and Real-time Road Status (RTRS) Service. These basic services respectively utilize some single transport data to help users find out the real-time public transport status. Meanwhile, by integrating the basic services, Transport Services Library offers some more advanced transport services including Operation Planning (OP) Service, Bus

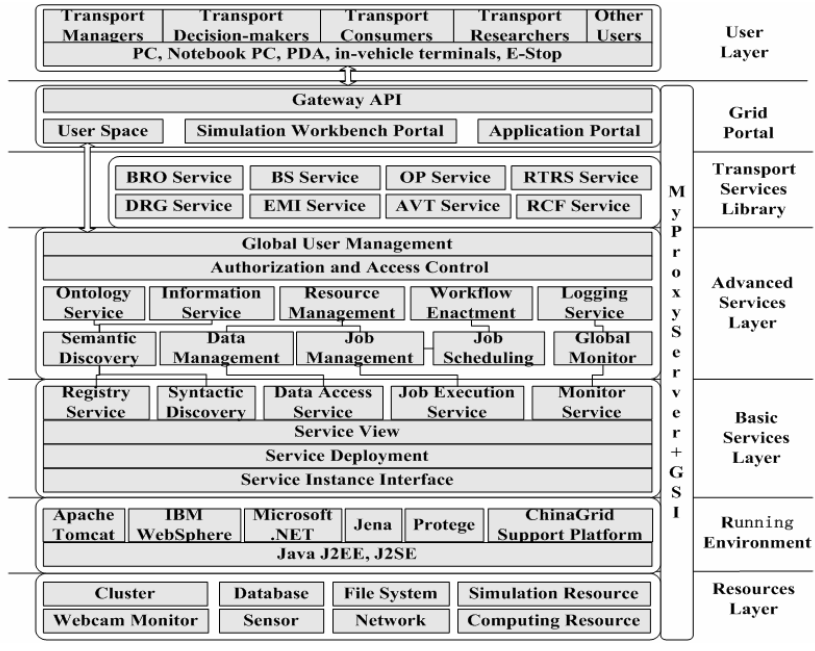


Fig. 2. The architecture and services of GUPTDSS

Route Optimization (BRO) Service, and Bus Schedule (BS) Service. These advanced services help users dynamically adjust the plan and enforcement of public transport operations through real-time Grid computing, resource sharing and cross-domain co-operation.

4) The security framework of GUPTDSS is based on Grid Security Infrastructure (GSI) [5] and MyProxy [6]. Based on Public Key Infrastructure (PKI), GSI provides a set of protocols, information bases and tools which enables users to access grid resources smoothly. MyProxy allows credential transferring among users, resource management and services so that it is no need to store the certificate and private key of users in the same machine.

5) Advanced Services Layer and Basic Services Layer are the key compositions of GUPTDSS. They provides a set of Grid services to support open, transparent, secure, and semantic data access, information sharing and job management. These services fall into four categories: services that enhance the semantic service registry and discovery, which are ontology service, information service, semantic discovery, syntactic discovery and registry service; services for resource access and management including resource management, data management, and data access service; services that manage the establishment and execution of application, which are workflow enactment, job scheduling, job management, and job execution service; services for Grid monitor and security, such as authorization and access control, monitor service, global monitor and logging service.

6) The two bottom layers of GUPTDSS can be considered as the Grid infrastructure that provides running environment and resources.

3 Methods

3.1 Metadata Model and Semantic Representation

A transport application requires plentiful background knowledge to establish workflows or query distributed resource efficiently and effectively from appropriate transport data and services. Unfortunately, many GUPTDSS users are devoid of enough background knowledge of both transport and Grid. One way to improve this kind of situation is to describe data and services in a formal manner that is interpretable both by humans and computer. In GUPTDSS we propose the transport metadata models to describe the structure of GUPTDSS resources. Besides, a hierarchical transportation ontology model is introduced to represent the semantic concepts and relationships of metadata.

At present the metadata models in GUPTDSS are divided into seven categories that are metadata of Grid node, network, video, database, replica, service and user. These metadata models are defined by RDF/XML and store in a Schema Repository Service (SRS) [7]. The resources registered to information server should follow a schema in SRS according to its category. And each information server in GUPTDSS manages a LADP-based XML document to store the metadata information of its own resources. Users can make an XPath query to information server according to these schemas to request the required resources through a single unified query interface.

Metadata crashes the bottleneck of low-level syntactic description and discovery of GUPTDSS resources. However, the discovery progress requires a much more high-level conceptual description of data and services to across the following gaps such as ambiguous semantics, insufficient relationships, and inconsistent descriptions of information. Therefore, for enabling more sophisticated semantic representation and discovery of transport data and services, and enhancing the collaboration, openness, and intelligence of GUPTDSS, we propose a hierarchical ontology model described by OWL [8] as shown in Fig.3.

Syntactic Layer is composed of Wordnet [9] and Sinica BOW [10] which prepare the English and Chinese lexical vocabulary for the ontology. As a large open formal ontology stated in first-order logic, SUMO [11] is used here to mapping the domain ontologies so as to improve search, communication and interoperation of different organizations.

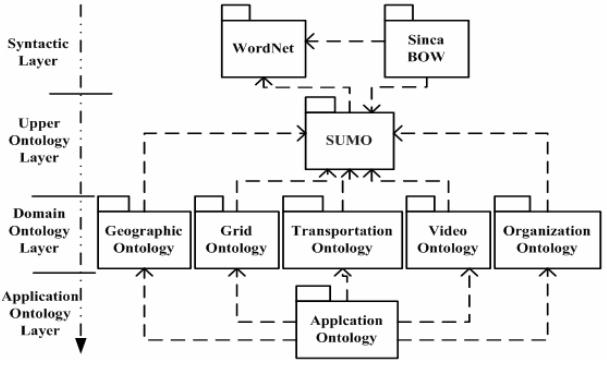


Fig. 3. The ontology model of GUPTDSS

We have defined lots of domain ontologies as shown in Fig3 for data, service and knowledge representation and reasoning. Since video is a particular and pivotal resource in GUPTDSS, we adopt a video ontology based upon MPEG-7 [12] to represent its complicated semantic relations at different levels of abstraction. According to the proposed ontology, a video can be represented as a set of objects, relations among object, relevant events occurring among these objects, and some temporal and spatial properties to describe these objects and events.

By utilizing ontologies, GUPTDSS services (e.g. ontology service, registry service, information service, workflow enactment, etc.) are semantic-aware and beneficial for resource description, information integration, knowledge sharing, and workflow orchestration.

3.2 Distributed Heterogeneous Data Access

All decision support services of GUPTDSS are based on the access, retrieval, processing, integration and analysis of large, heterogeneous and distributed traffic data. Therefore, CGSP Heterogeneous Database (CGSPHD) Service [7] is placed in GUPTDSS to provide Grid Data Service (GDS). Besides, Database Connection Pool and Data Cache [13] are used to meliorate the efficiency of data access.

CGSPHDB Service aims to enable Grid users for acquiring and processing distributed data stored in various heterogeneous database more efficiently and conveniently. CGSPDB Service encapsulates an extended OGSA-DAI [14] service to provide a unified data access. This access makes the data integration possible and brings heterogeneity transparency, naming transparency and distribution transparency [7]. As the GUPTDSS documents are transmitted by GridFTP, in the narrow sense, the physical data resources accessed by CGSPHDB Service are heterogeneous database management systems. In GUPTDSS, GIS data is restored in PostgreSQL, while GPS and traffic operation data are stored in SQLServer. In particular, video is stored as files and SQLServer is used to store its physical address for index. Fig4 and fig5 show the architecture and workflow details of CGSPHDB.

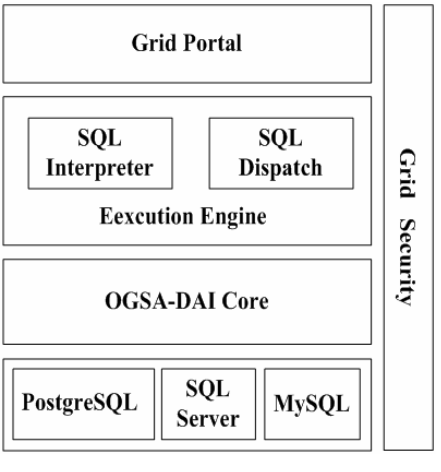


Fig. 4. (Left) The architecture of CGSPHDB

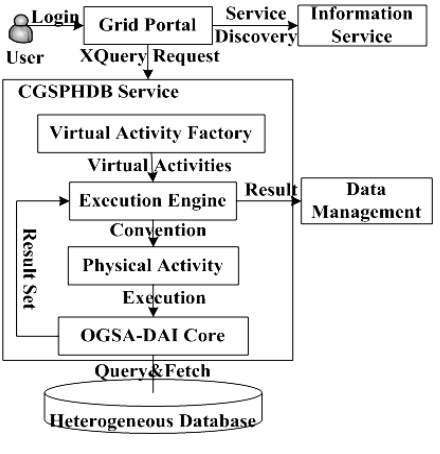


Fig. 5. (Right) The workflow of CGSPHDB

In GUPTDSS, users usually invoke the same transport service at a continuous time to access the same data. These frequent service invocations may continually establish plenty of database connections and seriously reduce the system performance. Therefore, connection pool is introduced in GUPTDSS to enhance the efficiency of data query. It takes following contents into account: 1) creates a special thread in charge of the creation, management and release of connection pool; 2) sets a connection queue to ensure that a connection can only be allocated to one thread at one time; 3) sets a overtime sign to prevent the over occupation of connection; 4) provides a unified interface to shield the concrete implementation of internal mechanism.

Data access in a distributed and dynamic environment can usually be efficiently optimized by implementing cache facilities. In GUPTDSS, domain information server maintains a recent data access queue in its memory for cache index, and clients can access the cache data through NFS [15] protocol. Cache stores the information of recent access data, including ID, logical name, size, last response time, average response time, last request times, priority, busy etc. In order to offer real-time data for applications, cache should be updated from time to time. When the cache queue is full-capacity, oldest and least-query data will be removed according to data priority. Hence, this mechanism could guarantee cache to store the most active data for usage.

3.3 Security Strategies

GUPTDSS contains potentially business-sensitive data and hence access to data and services should be restricted to authorized users. Therefore, all users have a certificate signed from a trusted Certificate Authority to identify his/her authentication. GUPTDSS offers two types of resource access control strategies to manage the access security: the default type and the defined type. In the case of the default type, resource provider defines a specific access list of allowed users to void illegal resource access. This simple strategy is suit for the access control of confidential resources. In the defined strategy, Community Authorization Service (CAS) [16] is introduced. CAS proposes a widely used set of group authentication and authorization mechanisms that address single login, delegation, and credential mapping issues arising in Virtual Organizations (VOs). Since GUPTDSS users of the same type often need to access to the same resources to meet the similar application demand, CAS is proved to be extraordinary suit for our system.

We also propose an improved Role Based Access Control (RBAC) [17] model associated with CAS to predigest the authorization management. RBAC model distributes users into different roles which have different permissions, and hence users can obtain suitable access control permissions indirectly by granted appropriate role. However, due to its static authorization mode, standard RBAC does not take the service lifecycle into account. It makes GUPTDSS unsafe since services can be invoked and changed no matter what life point it is at. In our extensible RBAC model, the permission of users is restricted to not only the delegated role of users but also the real-time status of applications. Its status properties include service time, location, and network condition etc. These properties are defined by eXtensible Access Control Markup Language (XACML) [18] while the application workflow is designed and will be set to the local resource provider while the workflow is executed. The user can access the

request resource using Security Assertion Markup Language (SAML) [19] only when he has the corresponding permission and the application status is enabled.

4 A Case Study: OP Service

Operation Planning (OP) Service aims to solve the typical problem for intelligent bus dispatch. It can be described as finding the best or approximate best alternative among all planning methods that can meet the restricted condition through a certain intelligent algorithm with limited calculation processes.

In GUPTDSS, Genetic Algorithm-based OP Service is coded by MPI and encapsulated as Grid service for high-performance parallel computing. It is deployed in CGSP Service Container [7] at each computing node. Meanwhile, its corresponding interface description in the form of a WSDL document is generated and annotated according to the ontology-based metadata schema and published in the cluster information center.

While invoking service, GUPTDSS offers two-layered job schedulers to manage the application implement: one is Grid scheduler and the other is local scheduler. Grid scheduler uses user specified requirements defined by GJDL to select satisfying services and resources from information center by Ontology Service and Information Service, and sends the sub service query to corresponding computing node. Local scheduler PBS manages the service implement at autonomous computing node, collects and distributes the corresponding data as service input from specified storage node by Data Management and Access Service, notifies the services status of execution processes by Monitor Service, and sends the sub results to Grid scheduler.

The separate sub results will be collected and transmitted to the final result database by GridFTP. As the application presentation layer is coded by C#.NET, GUPTDSS adopts SQLSERVER to store the final results. Grid Portal accesses SQLSERVER through ADO.NET interface, and presents the results to users by their user views.

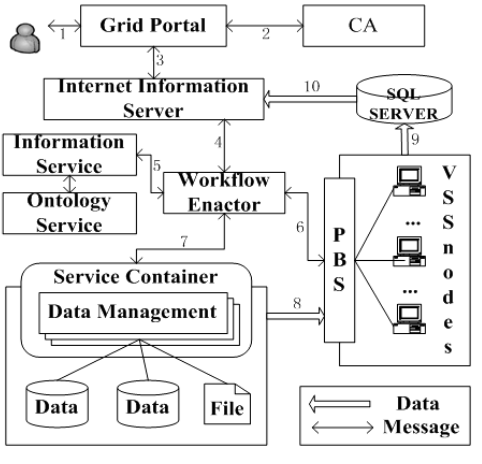


Fig. 6. The job executing flow of GUPTDSS

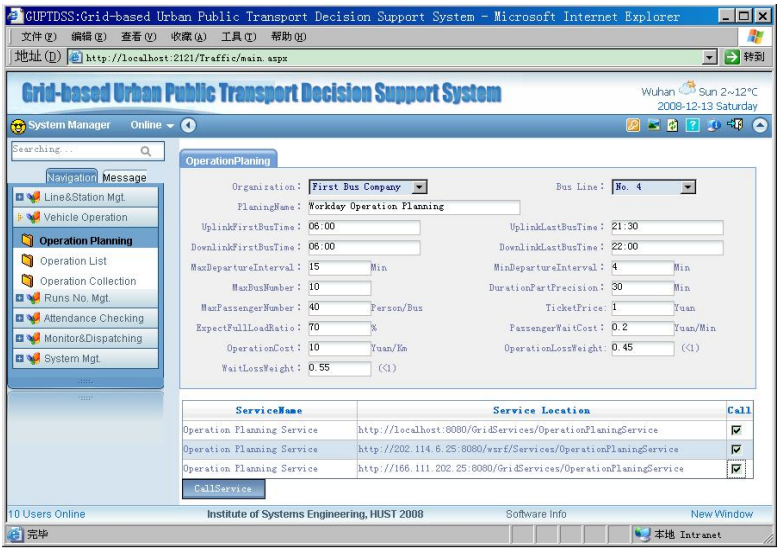


Fig. 7. OP Service in GUPTDSS accessed via a Web browser connected to the Internet

All activities of OP Service implement process are controlled by GUPTDSS security. Hence, users need to get the certificate signed from CA when he logins the Grid portal through a certain security protocol. While invoking OP Service, it is clear that users are only allowed to access the Grid resources and services if their verified credentials are accepted by the providers.

5 Conclusions and Future Work

In this paper we proposed an overview of the GUPTDSS infrastructure and its underlying service-oriented architecture. By making use of Grid Services, advanced semantic resource description and discovery technologies, novel heterogeneous data access mechanism and sophisticated security strategies, the developed GUPTDSS provides traffic organizations and users a suitable platform to manage urban transport much more easily and support the upper-level transport decision-making. We showed how the process takes place from the users' perspective and presented the underlying protocol implement.

In our early research, we recognized that the implement effect of GUPTDSS transport services is especially influenced by the efficiency of data access and processing. Therefore, future work will concentrate on the division of data domains, the strategy of data distribution and collection, and the creation and selection of data replicas. Besides, the design of workflow engine and algorithms of job scheduling will be discussed in future research papers as well.

Acknowledgements. The work is funded by National Natural Science Foundation of China (Grant 60773188), China Postdoctoral Science Foundation (Grant 20080430961) and the Key Technologies R&D Program of Wuhan (Grant

200710321090-2). The authors also like to thank Cluster and Grid Computing Lab of HUST for their support about CGSP.

References

1. Zhang, G.H., Li, M., Wang, J.X.: Application of the Advanced Public Transport System in Cities of China and the Prospect of Its Future Development. J. Transpn Sys. Eng. & IT 7, 24–30 (2007)
2. Wu, Z.H., Deng, S.G., Wu, J., Chen, H.J.: DartGrid II: A Semantic Grid Platform for ITS. IEEE Intelligent Systems 20, 12–15 (2005)
3. Foster, I., Kesselman, C.: The Grid2: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, San Fransisco (2003)
4. Foster, I., Kesselman, C., Nick, J., Tuecke, S.: The Physiology of the Grid: an Open Grid Services Architecture for Distributed Systems Integration. Technical report, Global Grid Forum (2002)
5. Welch, V., et al.: Security for Grid Services. In: 12th IEEE International Symposium on High Performance Distributed Computing, pp. 48–57. IEEE Press, Washington (2003)
6. MyProxy Credential Management Service, http://myproxy.ncsa.uiuc.edu/
7. CGSP Work Group: Design Specification of ChinaGrid Support Platform. Tsinghua University Press, Beijing (2004)
8. McGuinness, D.L., van Harmelen, F.: OWL Web Ontology Language Overview, http://www.w3.org/TR/2004/REC-owl-features-20040210/
9. Wordnet, http://www.cogsci.princeton.edu/~wn
10. Sinica BOW (Bilingual Ontological Wordnet), http://bow.sinica.edu.tw/
11. Suggested Upper Merged Ontology (SUMO), http://www.ontologyportal.org/
12. Tsinaraki, C., Polydoros, P., Christodoulakis, S.: Interoperability support between MPEG-7/21 and OWL in DS-MIRF. IEEE Transactions on Knowledge and Data Engineering 19, 219–232 (2007)
13. Tierney, B., Johnston, W., Lee, J., Thompson, M.: Data intensive distributed computing architecture for "Grid" applications. Future Generation Computer Systems 16, 473–481 (2000)
14. OGSA-DAI Web page, http://www.ogsadai.org
15. Smith, C.: Linux NFS Overview, http://nfs.sourceforge.net/
16. Pearlman, L., Welch, V., Foster, I., et al.: A Community Authorization Service for Group Collaboration. In: 3rd IEEE International Workshop on Policies for Distributed Systems and Net-works, pp. 50–59. IEEE Press, Washington (2002)
17. Sandehu, R., Coyne, E., Feinstein, H., et al.: Rule-Based Access Control Models. IEEE Computer 29, 38–47 (1996)
18. OASIS eXtensible Access Control Markup Language (XACML) V2.0 Specification Set, http://www.oasis-open.org/committees/xacml/
19. Oasis security services (SAML), http://www.oasis-open.org/committees/security/

Decision Making Based on Emergency Plan Templates

Pan Tang, Hongwei Wang, and Wei Zeng

National Simulation Training Center for National Economy Mobilization,
Huazhong University of Science and Technology,Wuhan 430074, China
tangpan001@163.com

Abstract. Decision making based on emergency plans is the main mechanism of problem solving in emergency management. According to characteristics of emergency plan texts, this paper extracts four main components and describes them with formal models. Based on emergency plan templates, we develop a dynamic decision making model to support the normative problem-solving process during emergencies. It provides foundation to develop decision support system for emergency managerst.

Keywords: Emergency plan template, Decision making.

1 Introduction

Emergencies, including planned events and unexpected incidents, are rare, uncertain, but create sudden and disastrous effects for natural environment and human society, presenting challenges to effective decision making. This makes multiple jurisdictions and disciplines work together for common goals of reducing damages. As an important part of emergency management, emergency managers must make decisions quickly with limited information and great loads under stress. To overcome these challenges, jurisdictions develop emergency plans, which describe purposes, possible situations, organization and assignment of responsibilities, tactic operations, resources for planned events based on experiences learned before [1]. When emergencies happen, critical and time-sensitive decision making based on emergency plans provide a standard problem-solving model to ensure rational and logical decisions.

Developing emergency plans before incidents happening have several key benefits. Firstly, they define possible states and provide probable actions. Therefore, it reduces uncertainty. Secondly, they facilitate us to use default activities rather than reason every time from the beginning, which decreases complexity and load of making decisions [2, 3]. Furthermore, it reduces errors induced by high stress on emergency managers. Finally, as a common knowledge of all responders involved, it make them coordinate with each other and provide them an explicit model to monitor teammates' activities in the response process.

Unfortunately, emergency plans are textual and unstructured documents. Computers cannot extract information and reason action plans based on them to adapt to dynamic emergency situations automatically. To account for that, Adriaan [4] developed an event-based task framework for them, which specifies events, tasks, and dependencies between them explicitly. Mark [5] formally specified emergency plans and compared those using tools from logic languages. Grathwohl [6] used description

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 1181–1190, 2009.

logic to model the flood emergency plan and reasoned action plans based on these formal models. Wenjun [7] constructed the emergency plan template ontology model based on ABC ontology, and store them using XML schema. Hongchen [8] proposed an entity and relationship model of organizations, resources and emergency response processes. However, these researchers mainly discussed how to extract components from these texts, formalize and store them in computers. They didn't discuss characteristics and requirements of dynamic decision making procedure based on emergency plans, and how to develop templates to support it. However, it is the goal of formalizing emergency plans indeed.

This paper extracts four components from emergency plan text, including incidents, organizations, tactic plans and resources, and formalizes them using logic frameworks respectively in section 2. Based on these models, we develop a normative decision making procedure to support the problem-solving processes in the complex and violate emergency situations in section 3. Finally, we conclude the paper and discuss the future work.

2 Modeling Emergency Plans

Jurisdictions and disciplines involving responsibilities of emergency management develop Emergency Operations Plans (EOP), Standard Operating Procedures (SOPs), and other procedural documents, which constitute a hierarchical and integrated document system, to support decision making during emergencies. These documents called emergency plans provide mechanisms to integrate multiple entities and functions, establish collaborative relationships, and ensure coordination of emergency response operations [9]. Decisions, which are made during development of them rather than after emergencies happening, can be deliberated, comprehensive and optimal.

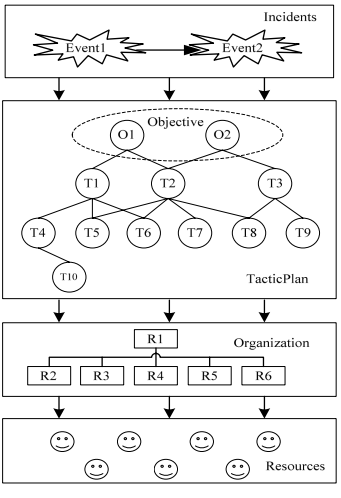


Fig. 1. Emergency plan template

Modeling emergency plans is to analyze the structural characteristics of paper-based documents, extract critical components and represent them with standard models, which are called emergency plan templates. Different kinds of emergency plans abstract multiple levels of emergency management businesses, which compliment with each other. Generally, emergency plan specifies incidents and assumed situations, organization structure and assignment of responsibilities, incident objectives, strategies and action plans to accomplish these objectives, available resources and other pertinent data. These characteristics make that it is possible to represent them with common templates. This is shown in fig. 1.

We extract four components, including specifications for incidents, emergency organizations, tactic plans and available resources respectively.

$$Emergencyplantemplate = \{Incidents, EO, TacticPlans, Resources\} \tag{1}$$

Incidents are models of planned events and relationships between them. *EO* specifies the organization structure and assigned responsibilities of responders. *TacticPlans* defines team plans to accomplish incident objectives for involved parties. Finally, pertinent data of available resources are defined in *Resources*. Based on *Emergencyplantemplate*, responders work as a team to communicate information, deploy resources, execute and modify operation plans for adapting to emergency situations rapidly and reducing damages effectively during emergencies.

2.1 Modeling Incidents

Incidents, which occur due to uncontrolled changes in the environment, always breakout unpredicted and evolve rapidly. Emergency managers must assess them accurately based on pertinent information and identify possible situations. While developing emergency plans, they analyze characteristics of incidents and assume possible situations. Formalizing and organizing information about them with common templates provide a simple and general representation to support information communicating in the response process [10]. They consist of a set of problem parameters delineating incidents, which are needed to access situations and predict probable courses. Furthermore, when an incident happens, there are usually several other incidents caused by it. We usually call it “incident chains”.

Generally, incidents and assumed situations are specified by a set of attributes and assigned reference values according to definitions in objection oriented programming language.

$$Incidents = \{incident_i(attribute, referValue)\} \tag{2}$$

$$\begin{aligned} &IncidentsRelationships = \{incidentsRelationship_j = (incidentA, incidentB)\} \\ &i, j = 1 \cdots n; incidentA, incidentB \in Incidents \end{aligned} \tag{3}$$

Attributes of incidents are the type, the occurring time, involved people, the impact area, and physical parameters, which is specified for different incidents. *IncidentsRelationship*s define causalities among them. For example, the physical attributes of flood are the water level, the flux, and so on. The *referValue* are the maximal value, minimal value, and other possible values assigned to them.

2.2 Modeling the Emergency Organization

During emergencies, personals from multiple parties assemble an organization to achieve the common goal of minimizing damages. Emergency plans provide an explicit organization model for them to establish, expand, and contract the emergency team rapidly and improve coordination effectively in the complex and dynamic task environment. It specifies organization members, relationships among them, and assigned responsibilities, which provides information about the organization awareness for responders to improve team coherence and effectiveness under assumed situations. Generally, emergency organization model specifies a set of roles and relationships between them.

$$EO = \{Roles, RoleRelationships\} \tag{4}$$

Roles in *Roles* are abstracted from emergency response operations, and represent a set of tasks and responsibilities. They are fundamental elements of the organization model and must be staffed once they're activated. *RoleRelationships* consists of two types of relationships between roles, which define the social structure of these roles, including *CommandandControlRelationship* and *CommunicationRelationship*.

$$CommandandControlRelationship = \{CmdCtrl_i(roleA, roleB, \phi)\}, i = 1 \cdots n; \tag{5}$$

Responders assigned to *roleA* can delegate authority to other responders assigned to *roleB* for executing the command described in ϕ. It defines orderly line of authority with ranks in the organization, which is used to communicate directions and maintain management control.

$$CommunicationRelationship = \{Commu_i(Rolesender,\ Rolereceiver, \phi, \varphi)\}, i = 1 \cdots n; \tag{6}$$

It describes reporting relationships between responders assigned to roles in *Roles*. *Rolesender* sends a message ϕ to *Rolereceiver* under given conditions φ. The message ϕ is real-time information about incidents, environment and response process. Executing communicating activities under conditions φ avoid information overwhelming.

Previous works, particularly in Incident Command System (ICS) [10], define a common organization structure designed for emergency management activities from small to complex incidents. Incident commander, operation section chief, planning section chief, logistics section chief, and finance section chief, consisting of the incident management team, are the most important roles. Each of them takes on specific incident response activities and commands his/her subordinators. The incident commanders at the highest rank position within ICS, is responsible for determining objectives, activating the organization models, ordering and coordinating responding agencies based on situations assessment. The planning section chief, assisting the incident commander in the tactic level, collects, evaluates, and disseminates information about incidents, develops action plans and monitors the implementing process to accomplish those incident objectives. The other roles at the operation level, account for executing assigned tasks, providing facilities and services, managing the cost, and so on.

2.3 Modeling Tactic Plans

Based on the organization model, emergency plans define standardized routines, policies and instructions for organization members, which are called tactic plans. Abstracted from experiences learned from previous disaster relief operations, they specify the determined action plans for emergency response. Due to critical time constraints, responders don't have enough time to communicate and negotiate extensively. Fortunately, explicit models of tactic plans improve implicit coordination effectively, and make operations robust and reliable for flexible organization and complex environment.

While developing tactic plans, emergency managers must analyze incident objectives based on evaluation of assumed situations, and define alternative strategies conformed to them under different constraints. Finally, they continue to program a list of action plans to accomplish these strategies until concise and simple actions for each organization unit are got. Therefore, tactic plans consist of incident objectives, strategies, actions, relevant contexts and constraints to take on them.

As the key work of formalizing emergency plans, there are several requirements of developing the logic framework for tactic plans. Firstly, it is multiple levels of response operations for each organization units defined in *EO*. Secondly, because of the complexity, *tacticPlans* can't specify all the details. It should provide the standard fields for determined characteristics of tasks and relationships, and custom fields for specified information, which will be instantiated by the real time information. Finally, it should provide alternative strategies to accomplish a task to overcome contingencies. Therefore, the overall task structure of tactic plans is top-down, contingent and partial structuring. Nodes in the structure are complex tasks and simple tasks, which are formalized based on joint intension model and parallel the SharedPlans theory [11].

Generally, incident objectives, strategies and actions, which have alternative solution methods to accomplish them, are represented by complex tasks. Actions performed in a straightforward, prescriptive manner by executing a direct action, are formalized using simple tasks. Complex task is specified by: (a) name; (b)role, having the responsibility to execute it; (c)priority; (d) preconditions, under which the node can be spitted; (e) $Connector \in \{AND,OR\}$,specifying relationships between the node and its sub nodes;(f)sub nodes and (g) end time. In the same way, simple task is defined by: (a) name; (b) role; (c) a list of input parameters and (d) a list of output parameters, specifying its effects. Additionally, these are precedence constraints between these nodes described in Constraints. For example, *nodeB* can't be executed until *nodeA* is finished.

$$\begin{gathered} TacticPlans = \{complexTask_i, simpleTask_j, constraints_k\} \\ ComplexTask_i = (name_i, role_i, priority_i, precondition_i, connector_i, subnodes_i, endTime_i) \\ SimpleTask_j = (name_j, role_j, inputParas_j, outParas_j) \\ Constraits_k = (nodeA, nodeB), i, j, k = 1 \cdots n \end{gathered} \tag{7}$$

2.4 Modeling Resources

Emergency plans and mutual aid agreements list available resources, including personals, teams, facilities, equipments, and supplies. Modeling resources is to establish common terminology to categorize them and describe capabilities of them within and across multiple jurisdictions and disciplines. Standard and uniform methods of describing resources provide the foundation of identifying, acquiring, allocating, and tracking them. Based on the model, emergency manager can order and allocate available resources to support response operations effectively.

The descriptions of resources in NIMS include category, kind, components, metrics, and type, which define standardized mechanisms for describing resources required over the life cycle of incidents [9]. Base on this theory and characteristics of resources listed in emergency plans, we define descriptions of resources with *Resources*, including name, category, kind, components, metric, types, capabilities, input parameters, output parameters, address, and phone numbers.

$$Resources = \{resources_i = (name, category, kind, components, metric, types, capabilities, inputParas, outputParas, address, phoneNumeber)\}, i = 1 \cdots n \quad (7)$$

The typing of Resources is specified by *category* and *kind*. Components describe units consisting of it. Metric is the measurement standard. Types refer to the level of resource capability. Capabilities refer to tasks it can execute alone. *InputParas* refers to parameters which are needed to perform relevant capabilities. *OutpuParas* is objects created by relevant capabilities. All of them will be used to avoid confusion and enhance interoperability during the emergency resource management process.

3 Decision Making Based on Emergency Plan Templates

Emergency response involves responsibilities of multiple jurisdictions and disciplines, and requires massive services and resources from them. Due to high consequence and time pressure, responders need to make decisions quickly on inadequate information to address these complex, ill structured problems. Emergency plans templates provide basis to analyze situations systematically, reason, implement and evaluate solutions during the process of decision making.

When incidents happen, responders at strategic, operational and tactical levels make interrelated and complementary decisions rationally and logically to ensure working coordinately. Generally, they make decisions based on the common procedure, including: (a) understanding the situation, (b) establishing and prioritizing incident objectives, (c) developing tactic plans to achieve adopted strategies, (d) establishing organization to implement these plans, (e) evaluating and revising plans to overcome contingencies and improve effectiveness. Each phase should be followed in sequence to ensure coordinated operations to achieve those incident objectives. It is a typical team decision problem, which involve a team G that has decided to perform a set of actions A. During the process of decision making, effective coordination is an essential ingredient. However, two sources of difficulties, including poorly shared mental models and possible conflicts of distributed decision-making [12], prohibit coordination and reduce effectiveness.

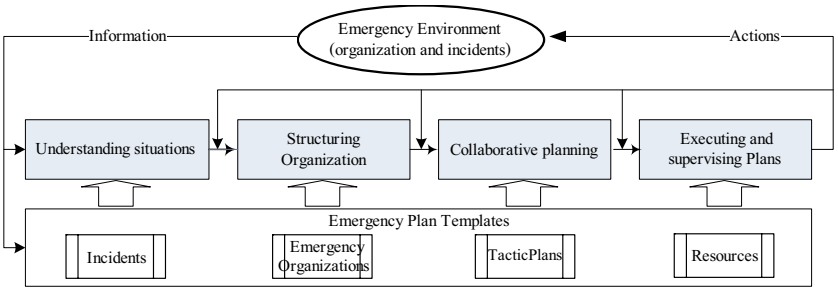


Fig. 2. Decision making procedure based on emergency plan templates

To overcome these challenges, we propose a dynamic and coordinated decision making procedure based on emergency plan templates modeled in section 2. It consists of four sub processes, consisting of understanding situations, structuring organization, developing tactic plans collaboratively, executing and supervising these plans, which is shown in fig 2.

3.1 Understanding Situations

When incident happen, personals should identify the organization and environment information, construct and maintain flexible and changing operational representations as incidents unfolds and evolves [13]. Acquiring critical information about situations accurately by observing the environment and communicating is the precondition of effective decision making. All operational representations are limited and hierarchical in term of scope and detail, which are different levels of abstraction of information about situations and are consistent with the lines of authority and responsibility.

Incidents delineate parameters of decision making problems. Based on pertinent information and evaluations, responders bind constants to these parameters, instantiate these models and formulate the specific decision making problem finally.

3.2 Structuring Organizations

Establishing emergency organization is an effective method to control large scale disaster. The number of responders and the organizational structure are totally dependent on the size of incidents and specifics of hazardous environment created by them. After incidents happen, first responders are loose, chaotic, and confusions about their roles and responsibilities, which can result disastrous outcomes. Subsequently, based on the identified incident objectives, they settle into the formal organization model defined in section 2.2 under severe time constraints according to procedures defined by emergency plans. While emergency situations change unpredictably and drastically, the organization should expand and contract flexibly.

EO provides a modular organization model to plot the control lines and establish organization for incidents from small routine events to large disasters. Structuring organizations is to formulate the organization and alter its structure to ensure effective response capability on scene. It consists of at least four basic processes to support the flexibility, including structure elaborating, role switching, authority migrating, and

system resetting [13], which are compliment and interrelated during emergencies. We abstract two mechanisms to support these processes.

(1)Establishing the organization structure. During multiple operational periods, organization structures are different. When incidents changes greatly, responders reconfigured the organization to get better performance. The process is to evaluate whether conditions of activating roles defined in *Roles* are activated according to real-time emergency situations.

(2)Roles assignment. The assignment of roles is the process of staffing positions of *EO*. Responders assigned to the roles require specific expertise and training. Available resources defined in *Resoures* have a continuum level of skills, which is range from rudimentary to highly specific according to types. Further more, boundaries of the organization structure fluctuate, and the task environment is volatile. Therefore, roles assignment is a sequence of decisions, which locate the best resources based on requirements of them.

3.3 Collaborative Planning

Structuring organizations is the process of coordination in the level of organization structure. Organization processes in operational periods should be modeled as teamwork to improve the coordination in the level of actions. This requires scenario-based planning and exercises with constant communications and coordination among responders, which is support by collaborative planning.

Collaborative planning is an ongoing, cyclical and directed process toward incident objectives. Generally, it consists of binding parameters of tasks, and selecting the optimal method to decompose complex tasks, which requires high level decisions. Based on determined objectives and priorities of them, staffs involving responsibilities of commanding should take on collaborative planning to develop action plans. During the process, incident management team select proper strategies, and decide which resources should be ordered and how to use them effectively. The result is an Incident Action Plan (IAP), which prioritizes incident objectives and provides tactical operations and resource assignments for unified efforts. Conducting activities based on it resolves interagency policy and procedural conflicts. As a distributed decision making process, it provides timely commitment and resource for critical emergency operations to ensure coordination among responders.

When planned events happen, incident commanders select nodes defined in *TacticPlans* to determine objectives, and binds parameters such as *endTime* and *priority* to refine them. Then, they evaluate situations, split it into sub nodes, and propagate constraints on parameters of them to parameters of those sub nodes. Then, they delegate these tasks to their subordinators. In the same way, they evaluate the task model, and continue the process of decomposition, until a set of simple tasks is got. Finally, they add and propagate constraints to these simple tasks.

When finding a method of splitting the delegated task with lower cost or a problem prevents them from achieving it, they try to re-decompose it. At the end of the "means end analysis" process, a full and concise IAP consisting of simple tasks and constraints among them will be finished and assigned to relevant responders.

3.4 Executing and Supervising Plans

After formulating the IAP, the process of decision making continues with the implementation of formalized steps defined in it. A concise and full IAP is essential to execute emergency management activities during the initial process. Executing and supervising plans is the process of directing and controlling efforts of resources to achieve specified incident objectives. In the process, contingencies can cause disruptions, which may result in additional control problems, greater loss, and increased expense and risk. Therefore, emergency managers should regularly compare planned progress with actual progress. When deviations occur and new information emerges, they evaluate them and decide whether to modify the current plan and develop a new one for subsequent operational periods by executing this decision making model from the first step.

4 Conclusion and Future Work

Decision making in emergency management is complex, distributed, and time critical problems, which requires geographically and temporally dispersed responders take on operations coordinately and rapidly. Emergency plan templates provide basis for the problem solving process. It is not only used for solving planed problems, but also as a starting point for solving novel problems during emergencies. This paper describes the initial PhD research work for developing decision making model and technology based on emergency plan templates.

Our future work is on studying task decomposition and task allocation algorithms to support decision making in the emergency response process. As a team decision making problem, it is about how to decompose received complex tasks into a set of subtasks with constraints by multiple responders and allocate available resources to execute them in the complex and volatile task environment. This theory will support the high-level decisions to improve effectiveness during emergencies.

References

1. Lee, J., Bui, T.: A Template-based Methodology for Disaster Management Information Systems. In: 33rd Annual Hawaii International Conference on System Sciences 2, pp. 1050–1057. IEEE Press, New York (2000)
2. Doug, D., Steve, C.: Planning with Templates. IEEE Intelligent Systems 20, 13–15 (2005)
3. Bui, T., Tan, A.: A Template-based Methodology for Large-Scale HA/DR involving Ephemeral Groups-A workflow Perspective. In: 40th Annual Hawaii International Conference on System Sciences, pp. 34–42. IEEE Press, New York (2007)
4. Adriaan, M., Jeroen, V., Cees, W.: An Event Based Task Framework for Disaster Planning and Decision Support. In: 2nd International ISCRAM Conference, pp. 151–153. IEEE Press, New York (2005)
5. Mark, H., Catholijn, J., Viara, P., Alexei, S., Lai, X.: Formal Modeling and Comparing of Disaster Plans. In: 2nd International ISCRAM Conference, pp. 97–107. ISCRAM, Brussels (2005)

6. Grathwohl, M., Beuvron, F., Rousselot, F.: A New Application for Description Logics: Disaster Management. In: International Workshop on Description Logics 1999, pp. 17–22. Linköpings Universitet, Linköpings (1999)
7. Wenjun, W., Fankuo, M., Yingwei, L., Yuelong, W., Zhuoqun, X.: Research on Ontology-Based Emergency Response Plan Template. Gongcheng/Computer Engineering 19, 170–172 (2006)
8. Hongchen, L., Yunfeng, D., Yanjun, L.: Formal description of emergency plans. Journal of Safety Science and Technology 4, 30–35 (2006)
9. National Incident Management System, http://www.fema.gov/pdf/emergency/nims/nims_doc_full.pdf
10. Zhengchuan, X., Yufei, Y., Shaobo, J.: A Decision Analysis Framework for Notification. In: 41st Annual Hawaii International Conference on System Sciences, pp. 26–34. IEEE Press, New York (2008)
11. Barbara, G., Sarit, K.: Collaborative Plans for complex Group Action. Artificial Intelligence 86, 269–357 (1996)
12. Wally, S., John, D.: A Case Study of Coordinative Decision-making in Disaster Management. Ergonomics 43, 1153–1166 (2000)
13. Bigley, G., Roberts, K.: The Incident Command System: High-Reliability Organizing for Complex and Volatile Task Environments. Academy of Management Journal 44, 1281–1300 (2001)

Computational Intelligence Techniques for a Smart Electric Grid of the Future

Zhenhua Jiang

Department of Electrical and Computer Engineering
University of Miami
Coral Gables, FL 33146, USA
zjiang1@miami.edu

Abstract. The electric grid of the future is required to become smarter so as to provide an affordable, reliable, and sustainable supply of electricity. Under such circumstances, considerable research activities have been carried out in the U.S. and Europe to formulate and promote a vision for the development of the future smart electric grid. However, how to achieve those smart features has been less reported. This article aims at identifying several specific areas of application of computational intelligence techniques in the smart electric grid of the future. The computational intelligence techniques that are capable of achieving certain favorable features of various components in a smart electric grid are elaborated.

Keywords: Smart Electric Grid, Adaptive and Self-Healing Systems, Learning, Computational Intelligence, Communications, Information Infrastructure.

1 Introduction

The U.S. electric grid, which has been progressively developed for over a century, is aging, inefficient, and congested, and incapable of meeting the future energy and security needs of the digital economy [1]. The driving forces towards lower-carbon generation technologies, improved efficiency on the power delivery grid [2],[3], and demand response, which will enable customers to become much more interactive with the grid, present great challenges to the grid. This will demand fundamental changes on both the supply and demand sides of electric energy. In addition, with centralized control techniques and traditional communication technologies, the present electric grid is no longer able to meet the new customer-centric network needs. Recent advances in the areas of power electronics, sensing, communications, control, data management, and computational intelligence will make it possible to modernize the power industry and make the power delivery system smarter [4]-[7].

There have been a variety of large research programs that attempt to define a smart electric grid for the 21 century's power delivery. Electric Power Research Institute, Inc. (EPRI) initiated IntelliGrid^{SM} program to create the technical foundation for a smart power grid that employs an open-standard, requirements-based approach to link electricity with communications and computer control to achieve tremendous gains in reliability, capacity, and customer services [8]-[9]. The U.S. Department of Energy recently established a Federal Smart Grid Task Force under Title XIII of the Energy Independence and Security Act of 2007 to coordinate the activities in the nationwide

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 1191–1201, 2009.

Grid 2030 program, which calls for the construction of a smart grid that would integrate advanced functions into the nation's electric grid to enhance reliability, efficiency, and security, and would also contribute to the climate change strategic goal of reducing carbon emissions [1]. The European Technology Platform (ETP) SmartGrids program was set up in 2005 to create a joint vision for the European networks of 2020 and beyond [10]-[11].

In addition, there are considerable research and development activities undergoing in both the industry and academia [12]-[20]. References [12] and [13] present some features of the smart grid for future power delivery. Roncero discusses the integration issue in the smart grid in [14]. Tsoukalas presents in [15] some interesting and promising concepts such as energy Internet. In [16], Hart presents some specific technologies such as smart metering infrastructure that help realize a smart grid.

A majority of previous work placed great emphasis on identifying the features and functions of a smart electric grid. However, how to achieve all of these smart features using intelligent technologies has been less reported. This paper aims at identifying specific application areas of computational intelligence techniques in the future smart electric grid. A vision of smart electric grid is given in Section 2. The computational intelligence techniques that are potentially capable of achieving certain favorable features of a variety of components in a smart electric grid are discussed in detail in Sections 3 through 7. Conclusions will be drawn in Section 8.

2 A Vision of Smart Electric Grid

The smart electric grid is a networked delivery system that provides customers with an affordable, reliable, and sustainable supply of electric energy. A set of intelligent technologies will smartly control a variety of components in the vast complex system, which must interconnect hundreds of millions of asynchronous houses and businesses through regional generation, transmission, distribution and storage systems. The new smart electric grid must improve efficiency by 50% or more in order for this power technology revolution to be affordable. In addition, it must be far more sophisticated from a computerized control standpoint in order to deal with unpredictable and time-varying green power sources, such as giant wind and solar farms located thousands of miles from metropolitan users, and mobile energy resources situated in moving plug-in hybrid vehicles. Distributed generation and local energy storage at consumer and manufacturing sites must be designed and tested to further fortify grid stability and safety from terrorism, as well as better defend it from unusual weather conditions. To achieve these, the smart electric grid must have the following features.

The smart electric grid provides a consistent, digital platform that allows for rapid and reliable sensing, measurement, communication, computation, control, protection, visualization, maintenance, and enterprise management of the entire system. This is a fundamental feature that facilitates the realization of other smart features.

The smart electric grid operates resiliently. It should be invulnerable to component failures, physical and cyber attacks, security breaches, and natural disasters such as hurricanes and snowstorms. A fast self-healing capability enables the electric grid to dynamically reconfigure itself and rapidly recover from power disturbances, which can be achieved through a combination of coordinated protection and control

schemes, embedded intelligence and human being's expertise, and self-awareness of the system operation state. In addition, the smart electric grid ensures high power quality.

The smart electric grid is flexible. The flexibility can be manifested in four aspects: accommodating all generation and storage options; enabling active participation by consumers in demand response; enabling new products, services, and plug-and-play capabilities that accommodate progressive technology upgrades of hardware and software components; and enabling seamless compatibility with various market operation styles. In addition, the design of the smart electric grid will be client-tailored for operators' convenience without loss of functions and interoperability, and cater to customers with more energy consumption options for high quality/price ratio.

The smart electric grid fosters sustainability. The growth of electricity demand should be sufficiently satisfied by deploying alternative and renewable energy sources, promoting energy-saving technologies, mitigating network congestion, optimizing assets and operating efficiently.

3 Smart Sensing, Metering and Communication Infrastructures

In a smart electric grid, computational intelligence technology will be incorporated in the sensing, measuring and metering circuits to reduce the burden of communications. Traditional electromechanical CT (current transducers) and PT (potential transducers) will be replaced by optical or electronic CT and PT whose advantages include wide bandwidth, high accuracy, and low maintenance costs. Khorashadi-Zadeh proposes in [21] an intelligence based data acquisition system to correct CT and CVT (capacitive voltage transducer) secondary waveform distortions. The key is to use artificial neural network to achieve the inverse transfer functions of CTs and CVTs.

All measurement signals are attached with a high-accuracy time stamp by using a GPS (global positioning system) signal. With this in hand, the RTU (remote terminal unit) function will be replaced by high-speed PMUs (phasor measurement units), which are distributed throughout the power network and used to monitor the power quality and in some cases respond automatically to them. The PMU's ability to rapidly sense system conditions supports the kind of automated self-healing of anomalies in the network needed to not only deliver lower cost power with fewer blackouts, but to support more complex power generation scenarios required by alternative energy resources. A Wide-Area Measurement System (WAMS), which is a network of PMUs, can provide real-time monitoring on a regional or national scale. Paper [22] discusses measurement approaches for adaptive protection and control in a competitive market.

A smart electric grid uses advanced digital electronic metering techniques instead of analog mechanical meters to capture the electricity use and the enables time-of-use pricing. Smart meters provide a communication path extending from generation plants to electrical outlets (smart sockets) and other smart grid enabled devices attached to the grid. Intelligence will be embedded in the digital meters. Core technologies of this metering platform include advanced microprocessor based meters, time-of-use pricing tools, electromagnetic signature measurement, wide-area monitoring systems, advanced switches and cables, backscatter radio technology, and digital relays.

A smart electric grid should be based on a self-healing communication network to significantly improve the reliability of monitoring and control of substations and control centers. Integrated power and communication networks will allow for real-time control, information and data exchange to optimize the system reliability, asset utilization, and security. The communication protocol of a smart electric grid should be standardized and open. A good option is the IEC 61850 standard [23], which provides an open interface not only among the intelligent electronic devices (IED) inside a substation but also between substations and between substations and control centers. This improves the interoperability of communication networks significantly.

In addition, with artificial intelligence embedded in a variety of equipment items, it is possible to develop intelligent transformers, transmission lines, circuit breakers, etc. Reference [24] uses a combination of a learning method with a fuzzy interference system to predict the position of an energy meter adjuster to fit in with the errors of the energy meter under calibration.

4 Autonomous Control and Adaptive Protection

Control and protective actions are critical for reliable operation of an electric grid. A smart electric grid employs autonomous distributed control schemes with multiagent technology and adaptive protection with embedded intelligence.

4.1 Multiagent Based Control Paradigm

A smart electric grid contains a large number of intelligent agent based decentralized controllers for normal operation, auto-restoration, remedial or predictive actions, and optimization. A smart electric grid, although complex and dynamic, can be viewed as a collection of individual intelligent agents that adapt to events and surroundings, acting both competitively and cooperatively for the good of the entire system. If the agents sense any anomalies in their surroundings, they can work together, essentially reconfiguring the system, to keep the problem local. Thus, the agents would prevent the cascading effect, the main source of vulnerability in critical infrastructure systems. An agent can be either a physical entity that acts in the environment such as a controller that controls a piece of equipment, or a virtual one such as a piece of software that makes bids to the energy market or stores data in a distributed database.

A multiagent based smart electric grid may use two types of agents: cognitive and reactive agents. Each cognitive agent has a knowledge base that comprises all the data and know-how required to carry out its task and to handle interactions with the other agents and its environment. Cognitive agents are also intentional, in that they have goals and explicit plans that let them achieve their goals. The reactive agent, in contrast, claims that it is not necessary that agents are individually intelligent for the system to exhibit intelligent behavior overall. The reactive agents work in a hard-wired, stimulus-response manner. The reactive agent's goals are only implicitly represented by rules, so it must consider every situation in advance. The reactive agent's advantage lies in its ability to react fast. In addition, collaborative agent societies with intelligent user interfaces provide a complementary style of human-computer interaction, where the computer becomes an intelligent, active and personalized collaborator.

Interface agents are computer programs that employ artificial intelligence methods to provide active assistance to a user of a particular computer application. Underlying software has characteristics that learn from data and then adjust themselves to better handle those types of data.

A variety of efforts have been given to study agent-based control frameworks for infrastructure operations. Amin explains in [25] that agents have the appropriate capabilities to operate the national power grid and presents in [26] a fairly detailed view of a multi-layered, multi-species multiagent system that includes, among others, fault isolation agents, command interpretation agents, and event identification agents. References [27] and [28] together present a relatively complete picture of the agent operated microgrid. They recognize and address the impossibility of centralized control among several owners and present the operation of a multi-agent system that uses agents advantageously to execute a classical distributed algorithm. The agents are developed on the JADE framework and interact with a small operating microgrid connected to the main power grid. In [29], Tolbert presents a scalable multi-agent paradigm to control distributed energy resources with the intent of achieving higher reliability, better power quality, and more efficient generation and consumption.

4.2 Adaptive Protection

In a smart electric grid, a great improvement is that the settings of protective relays can be remotely modified in real time to adapt to the changes in the grid configuration. A variety of computational intelligence techniques can be used to make substations smarter. As such, a smart substation will serve as an intelligent unit of some special protective schemes to improve the reliability of power grid.

In [30], Song proposes an adaptive protection scheme, based on neural networks, for controllable series-compensated EHV transmission lines. The main idea is to employ an artificial neural network (ANN) to make a decision based on extracting useful features in the desired spectra within a certain frequency range under fault conditions. Paper [31] presents new strategies for adaptive out-of-step protection of synchronous generators based on neural networks. A feed forward model of the neural network based on the stochastic back propagation training algorithm is used to predict the out-of-step conditions. Thorp presents in [32] simulation results of “protection agents” engaged in realistic protection and control scenarios using a multiagent technology. The agents increase system protection performance by exchanging basic information, primarily contributing to fault identification and isolation.

5 Advanced Data Management and Visualization

In a smart electric gird, widely-deployed decentralized applications require a strong distributed database management system, which will manage and share all the data in the substations and control centers and communicate with other substations or control centers by publishing those data to the communication network. All these data from PMU units, relays, fault recorders, power quality monitors, equipment monitors, etc., should be efficiently managed and displayed. Real-time data visualization gives the operators a clear picture of the current operation status of the gird.

A real-time model of the grid should be built for better control inside and outside a control center or substation. In order to get a reliable and consistent real-time model, the substation level topology processor will build the substation topology while the substation level state estimator will estimate the substation states to provide a more reliable and full view of the substation. Some previous work focusing on distributed state estimation has already provided the idea of building the substation-level state estimator and the related filter technology. Whenever changes happen in the power system, such as the substation topology changes or addition of a new substation into the power grid, the system-wide model can be rebuilt automatically in the control center by merging the substation models. It is easy to build a backup control center model or even rebuild a new control center model under emergency to significantly improve the operating resilience of control centers against physical and cyber attacks and natural disasters.

Information technology will be used to reduce complexity of data management so that the operators and managers have tools to effectively and efficiently operate a grid with an increasing number of variables. Technologies include visualization techniques that reduce a large amount of data into easily understood visual formats, software systems that provide multiple options when systems operator actions are required, and simulators for operational training and "what-if" analyses.

Schreiner presents in [33] innovative techniques for intelligent power system mobile data management, which include a solution to documentation optimization with automatic synchronization of the mobile data throughout the enterprise. In [34], McDonald addresses a standard intelligent electronic device (IED) protocol, as well as a standard substation local area network (LAN) technology. The efforts also include a framework to integrate the substation information into the utility enterprise system. Reference [35] proposes a model of an intelligent short-term demand side management system (DSM) based on a distributed measurement and management data system. The system is designed to avoid peaks of power request greater than a given threshold and to give maximum comfort to users.

Traditional real-time monitoring tools, real-time hardware-software architectures, and user interfaces developed for vertical integrated environments have proven to be inadequate for the new wide-area intelligent management of the deregulated electric grid. Paper [36] describes new reliability management needs, hardware-software user-interface configurations; intelligent alerts and alarming, and geographic multi-view visualization technologies for wide-area real-time monitoring. This paper also focuses on how these new wide-area technologies are helping to create wide-area real-time intelligent tools which can be adapted to the reliability management needs of the North American Electricity Reliability Corporation (NERC) Reliability Coordinators and regulatory and reliability standards compliance monitoring organizations.

6 Intelligent Interfaces with Distributed Resources and Market

Given the increasing interest in utilizing renewable energy and distributed generation and storage to meet future demand, the smart electric grid should provide advanced power electronics and control interfaces for these distributed resources so that they can be integrated into the grid in a large scale at the transmission and distribution

levels. The central clearing algorithms should be robust enough to accommodate the volatile nature of certain renewables such as wind generators with finer forecasting and scheduling methods. Demand-side participants should have access to the market through certain communications, control and information channels. Congestion management is an important feature of the control centers. The control centers should forecast and identify the potential congestions in the network and alleviate it with the help from wide-area GIS systems. By incorporating microgrids, the electric grid can deliver quality power to the customers in a manner that the power supply degrades gracefully after a major commercial outage, as opposed to a catastrophic loss of power, allowing more of the installations to continue operations. Microgrids should have a capability to operate in the islanding mode taking into account the transmission capacity, load demand, and stability limit, and provide mechanisms for a seamlessly transition to islanding operation.

Reference [37] presents an agent-based control framework for distributed energy resources microgrids. Each energy resource unit, including any controllable load, is represented by an agent and all the agents will collaborate and cooperate to achieve some global objectives. In [38], Hutson proposes an intelligent method for scheduling usage of available energy storage capacity in plug-in hybrid electric vehicles (PHEV) and electric vehicles (EV). The batteries on these vehicles can either provide power to the grid when parked, known as vehicle-to-grid (V2G) concept or take power from the grid to charge the batteries on the vehicles. To study algorithms for the interfaces, a scalable parking lot model is developed with different parameters assigned to fleets of vehicles.

In [39], Praca proposes an improved multi-agent simulation tool to study the negotiations in electricity spot markets based on different market mechanisms and behavior strategies, in order to take account of decentralized players such as virtual power plants. Paper [40] aims at assessing the economic benefits achievable by a group of industrial and commercial customers aggregated in a microgrid controlled with a central controller that uses a neural network to optimize the schedule of generators and responsive loads. The central controller receives market signals, load and generation bids, load and weather forecasts and determines hour by hour the correct dispatch of generators to maximize the value of the microgrid by minimizing the energy costs.

7 Decision Support Systems for System Operation and Planning

Real-time, intelligent decision support systems for grid operation, planning, economic dispatching, contingency analysis, and restoration from blackouts are part of the smart electric grid. Operation of smart electric grid will be supported by a series of online analyses such as voltage stability, transient angular stability, small-signal stability analysis, N-x contingence analysis, cascading failure analysis and probabilistic risk analysis. These online analyses shall perform dynamic model update and validation.

In addition, fast diagnosis and prognosis are necessary in a smart electric grid. Expert system based fault diagnosis technology provides intelligent maintenance and management of devices in a substation. While an increasing amount of data about fault conditions are gathered in a substation, an intelligent alarm management and

processing system should be developed to find the root cause of the fault based on artificial intelligent technologies such as expert system.

In [41], Fujiwara describes the detail of an intelligent load flow engine is an expert system embedded in the personal workstation that is used by the power system planners. The objective of the engine is to provide both an expert and novice user with a friendly working environment, aiming at enhancing a user's creativity in making operational plans. To this end, the personal scientific workstation that was developed for power system planning is equipped with an expert system whose knowledge is acquired from the experienced power system planners.

Paper [42] discusses the applications of machine learning approaches in security assessment of power systems. This paper also describes a framework that integrates several of these techniques so that users can extract relevant information tailored to their decision-making needs.

Reference [43] proposes a hierarchical case-based reasoning method for decision-making in power system restoration. In contrast to conventional case-based reasoning methods, the restoration problem is not represented by single cases, but instead by collections of cases at different levels. A flexible combinatorial strategy makes the hierarchical case-based reasoning possible to solve a wide range of restoration problems without the need for huge case bases of complex adaptation mechanisms.

In [44], Mellit introduces an approach based on an adaptive neuro-fuzzy inference scheme for the optimal sizing of standalone photovoltaic power systems especially in isolated sites where meteorological data are not available. In addition, several artificial intelligence based techniques such as feed-forward, radial basis function network, recurrent network, modular network, and the adaptive wavelet-network, are studied and compared.

8 Conclusion

This paper has identified several areas of application of computational intelligence techniques in a smart electric grid of the future. With a common digitalized platform, the smart electric grid will enable increased flexibility in control, operation and expansion, allow for embedded intelligence, essentially foster the resilience and sustainability of the electric grid, and eventually benefit the customers with lower costs, improved services and increased convenience. Implementation of such a vision demands a concerted effort to apply the intelligent technologies in a variety of aspects of the grid, including smart sensing, measurement and metering, distributed control and adaptive protection, advanced data management and visualization, intelligent interfaces with distributed energy resources and markets, and real-time decision support system for system operation and planning. Given the scale of the effort required and the enormity of the challenges ahead, collaboration among researchers from different fields is essential and should be developed through various channels in order to ensure and accelerate the success of realizing the smart electric grid.

Acknowledgments. This work was supported in part by the U.S. National Science Foundation under grants ECCS-0652300, ECCS-0748032 and ECCS-0821126.

References

1. Grid 2030: A National Version for Electricity's Second 100 Years. Office of Electric Transmission and Distribution, United State Department of Energy (2003)
2. Kuri, B., Li, F.: Valuing Emissions from Electricity towards a Low Carbon Economy. In: EEE PES General Meeting 2005, pp. 53–59. IEEE Press, New York (2005)
3. Hoffman, M., Daim, T.U.: Building Energy Efficiency Technology Roadmaps: a Case of Bonneville Power Administration (BPA). In: Technology Management for the Global Future, pp. 1503–1527. IEEE Press, New York (2006)
4. Chuang, McGranaghan, M.: Functions of a Local Controller to Coordinate Distributed Resources in a Smart Grid. In: IEEE PES General Meeting 2008, pp. 1–6 (2008)
5. Lambert, E., Fremont, J., Bouquet, C.: Method and applications of IEC Common Information Model Standard for Distribution Operations: A path towards smart grids development. In: IET-CIRED Seminar on SmartGrids for Distribution, pp. 1–4. IEEE Press, New York (2008)
6. King, R.L.: Information Services for Smart Grids. In: IEEE PES General Meeting 2008, pp. 1–6. IEEE Press, New York (2008)
7. Lobo, F., Cabello, A., Lopez, A., Mora, D., Mora, R.: Distribution Network as Communication System. In: IET-CIRED Seminar on SmartGrids for Distribution, pp. 1–4. IEEE Press, New York (2008)
8. EPRI Intelligrid, `http://intelligrid.epri.com/`
9. McGranaghan, M., Von Dollen, D., Myrda, P., Gunther, E.: Utility Experience with Developing a Smart Grid Roadmap. In: IEEE PES General Meeting, pp. 1–5. IEEE Press, New York (2008)
10. European Commission, European Smart Grids Technology Platform (2006)
11. European Commission, Towards Smart Power Networks (2005)
12. Garrity, T.F.: Getting Smart. IEEE Power and Energy Magazine 6, 38–45 (2008)
13. Amin, S.M., Wollenberg, B.F.: Toward a Smart Grid: Power Delivery for the 21st Century. IEEE Power and Energy Magazine 3, 34–41 (2005)
14. Roncero, J.R.: Integration is Key to Smart Grid Management. In: IET-CIRED Seminar on SmartGrids for Distribution, pp. 1–4. IEEE Press, New York (2008)
15. Tsoukalas, L.H., Gao, R.: From Smart Grids to an Energy Internet: Assumptions, Architectures and Requirements. In: Third International Conference on Electric Utility Deregulation and Restructuring and Power Technologies, pp. 94–98. IEEE Press, New York (2008)
16. Hart, D.G.: Using AMI to Realize the Smart Grid. In: IEEE PES General Meeting 2008, pp. 1–6. IEEE Press, New York (2008)
17. Sheble, G.B.: Smart Grid Millionaire. IEEE Power and Energy Magazine 6, 22–28 (2008)
18. Lauby, M.G., Malcolm, W.P.: North American Industry Trends Supporting Intelligent Grids. In: International Conference on Intelligent Systems Applications to Power Systems, pp. 1–6. IEEE Press, New York (2007)
19. Walton, C., Green, T., Woods, T.: Supergen FutureNet and FlexNet. In: IET-CIRED Seminar on SmartGrids for Distribution, pp. 1–4. IEEE Press, New York (2008)
20. Krebs, R., Buchholz, B.M., Styczynski, Z.A., Rudion, K., Heyde, C., Sassnick, Y.: Vision 2020 — Security of the Network Operation Today and in the Future. German Experiences. In: IEEE PES General Meeting 2008, pp. 1–6. IEEE Press, New York (2008)
21. Khorashadi-Zadeh, H., Li, Z.: Intelligence Based Data Acquisition System for Protective Relays. In: IEEE PES Transmission and Distribution Conference and Exposition, pp. 1–7. IEEE Press, New York (2008)

22. Phadke, A.G., Hadjsaid, N.: Measurements for Adaptive Protection and Control in a Competitive Market. In: 33rd Annual Hawaii International Conference on System Sciences, pp. 1–7. IEEE Press, New York (2000)
23. Sidhu, T.S., Gangadharan, P.K.: Control and Automation of Power System Substation using IEC61850 Communication. In: 2005 IEEE Conference on Control Applications, pp. 1331–1336. IEEE Press, New York (2005)
24. Wahyudiati, M.H., Shigeta, K., Shimizu, E.: The Adjuster Position Prediction in Energy Meter Calibration System using Fuzzy Learning Method. In: IEEE Instrumentation and Measurement Technology Conference, pp. 1289–1291. IEEE Press, New York (1996)
25. Amin, M.: National Infrastructures as Complex Interactive Networks. In: Automation, Control, and Complexity: New Developments and Directions, Samad and Weyrauch. John Wiley and Sons, New York (1999)
26. Amin, M.: Toward Self-Healing Energy Infrastructure Systems. IEEE Computer Applications in Power 14, 20–28 (2001)
27. Dimeas, A., Hatziargyriou, N.: A Multiagent System for Microgrids. In: IEEE Power Engineering Society General Meeting 2004, pp. 447–455. IEEE Press, New York (2004)
28. Dimeas, A.L., Hatziargyriou, N.D.: Operation of a Multiagent System for Microgrid Control. IEEE Transactions on Power Systems 20, 1447–1455 (2005)
29. Tolbert, L., Qi, H., Peng, F.: Scalable Multi-Agent System for Real-Time Electric Power Management. In: Power Engineering Society Summer Meeting, pp. 1676–1679. IEEE Press, New York (2001)
30. Song, Y.H., Johns, A.T., Xuan, Q.Y.: Artificial Neural-network-based Protection Scheme for Controllable Series-compensated EHV Transmission Lines. IEE Proceedings- Generation, Transmission and Distribution 143, 535–540 (1996)
31. Abdelaziz, A.Y., Irving, M.R., Mansour, M.M., El-Arabaty, A.M., Nosseir, A.I.: Adaptive Protection Strategies for Detecting Power System Out-of-step Conditions Using Neural Networks. IEE Proceedings - Generation, Transmission and Distribution 145, 387–394 (1998)
32. Thorp, J., Wang, X., Hopkinson, K., Denis, V., Coury, D., Giovanini, R.: Agent Technology Applied to the Protection of Power Systems. In: Conference on Securing Critical Infrastructures, Grenoble (2004)
33. Schreiner, Z., Middleton, A., Bizjak, J.J.: Innovative Techniques for Intelligent Power System Mobile Data Management. In: IET 9th International Conference on Developments in Power System Protection, pp. 459–464. IEEE Press, New York (2008)
34. McDonald, J.D.: Managing the Data and Choosing the Right Protocol. In: IEEE PES Transmission and Distribution Conference and Exposition, pp. 970–973. IEEE Press, New York (2001)
35. Amato, A., Di Lecce, V., Piuri, V.: A Smart Distributed Measurement Data Management System for DSM. In: 2007 IEEE Instrumentation and Measurement Technology Conference, pp. 1–7. IEEE Press, New York (2007)
36. Martinez, C.: Intelligent Real-time Tools and Visualizations for Wide-area Electrical Grid Reliability Management. In: IEEE Power and Energy Society General Meeting, pp. 1–4. IEEE Press, New York (2008)
37. Jiang, Z.: Agent-Based Control Framework for Distributed Energy Resources Microgrids. In: IEEE/WIC/ACM International Conference on Intelligent Agent Technology, pp. 646–652. IEEE Press, New York (2006)
38. Hutson, G., Venayagamoorthy, G., Corzine, K.: Intelligent Scheduling of Hybrid and Electric Vehicle Storage Capacity in a Parking Lot for Profit Maximization in Grid Power Transactions. In: IEEE Energy 2030 Conference, pp. 1–8. IEEE Press, New York (2008)

39. Praca, I., Morais, H., Ramos, C., Vale, Z., Khodr, H.: Multi-agent Electricity Market Simulation with Dynamic Strategies & Virtual Power Producers. In: IEEE Power and Energy Society General Meeting, pp. 1–8. IEEE Press, New York (2008)
40. Pilo, F., Pisano, G., Soma, G.: Neural Implementation of MicroGrid Central Controllers. In: IEEE International Conference on Industrial Informatics, pp. 1177–1182. IEEE Press, New York (2007)
41. Fujiwara, R., Sakaguchi, T., Kohno, Y., Suzuki, H.: An Intelligent Load Flow Engine for Power System Planning. IEEE Transactions on Power Systems 1, 302–307 (1986)
42. Wehenkel, L.: Machine Learning Approaches to Power-System Security Assessment. IEEE Intelligent Systems and Their Applications 12, 60–72 (1997)
43. Mellit, A.: Artificial Intelligence Based-Modeling for Sizing of a Stand-Alone Photovoltaic Power System: Proposition for a New Model using Neuro-Fuzzy System (ANFIS). In: 3rd International IEEE Conference on Intelligent Systems, pp. 606–611. IEEE Press, New York (2006)
44. Wang, H., Liu, Y.: Hierarchical Case-Based Decision Support System for Power System Restoration. In: IEEE Power Engineering Society General Meeting, pp. 1115–1119. IEEE Press, New York (2004)

Impact of Non-schedulability on Embedded System Performance

Jiafu Wan[1, 2], Di Li[3], Hehua Yan[2], and Ping Zhang[1]

[1] College of Computer Science & Engineering, South China University of Technology, Guangzhou 510006, China
[2] Guangdong Vocational College of Mechanical and Electrical Technology, Guangzhou 510515, China
[3] School of Mechanical & Automotive Engineering, South China University of Technology, Guangzhou 510641, China
Jiafuwan_76@163.com

Abstract. The uncertainty factors of real-time tasks during runtime affect the control performance owing to system resources and processor utilization restriction. The impact of non-schedulability on embedded system performance is deeply researched in this paper. First, the time characteristics such as sampling jitter, input-output jitter and non-schedulability are discussed. Then the schedulability analyses of the rate monotonic (RM) algorithm and the earliest deadline first (EDF) algorithm are introduced. Finally, using RM algorithm, an example, DC servo motor controller, is use to illuminate the impact of non-schedulable jobs on system performance. The experiment results indicate that it is extremely important to reduce, even eliminate the non-schedulable jobs for improving embedded system performance.

Keywords: Embedded system, Time characteristic, Schedulability.

1 Introduction

Nowadays, embedded computing is playing an increasingly important role in the embedded control system [1]. More and more real-time applications are built on embedded systems. In this system where the workload displays large fluctuations due to resource competition, a traditional real-time design, based on worst-case assumptions, may be infeasible. On one hand, assuming that all tasks execute at their maximum rates and according to their worst-case execution times (WCETs), the developer may be forced to choose a very powerful CPU, which will be under-utilized most of the time. On the other hand, basing the design on average-case conditions may give rise to temporary CPU overloads which degrades system performance. Besides, for traditional design method, the WCETs of tasks are assumed. In practice, it is very difficult for WCET estimation. An under-estimation or over-estimation may lead to CPU overloads or resource wastes respectively. Therefore, in order to provide the valuable reference for embedded system design, the impact of non-schedulability on embedded system performance is deeply researched.

The impact of jitter on stepper motor control is analyzed in [2]. Jitter will contribute additional torque load to the motor that has the potential to exceed the available

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part I, LNCS 5551, pp. 1202–1209, 2009.

torque at high speeds and cause a loss of position steps. The magnitude of the additional torque is determined for several typical stepper motors at moderate to high speeds. For motion control system, Feng [3] uses a model based on simulation to quantify the relationship between sampling jitter and controller performance. In [4], for analyzing the relationship between uncertainties of software CNC real-time tasks and manufacturing accuracy, a mismachining tolerance analysis method based on simulation model is presented.

The rest of this paper is outlined as follows. The time characteristic we consider here is described in Section 2. The schedulability analyses for the rate-monotonic (RM) algorithm and the earliest deadline first (EDF) algorithm are introduced in Section 3. Section 4 evaluates the impact of non-schedulability on embedded system performance. And section 5 concludes this paper.

2 Time Characteristics

Timing uncertainties during task executions owing to resource competition lead to input-output jitter, sampling jitter and even non-schedulability. In this section, the causes of jitter and non-schedulability are deeply researched on the basis of analyzing the related parameters.

2.1 Jitter

Generally, a control task consists of three distinct operations: input data collection, control algorithm computation, and output signal transmission, see Fig.1[1]. The timing of the operations is crucial to the performance of the controller. Ideally, the control algorithm should be executed with perfect periodicity, and there should be zero delay between the reading of the inputs and the writing of the outputs. This will not be the case in a real implementation, where the execution and scheduling of tasks introduce latencies.

The basic timing parameters of a control task are shown in Fig.2. It is assumed that the control task is released periodically at times given by $r_k = kT$, where T is the sampling interval of the controller, . Due to preemption from other tasks in the system, the actual start of the task may be delayed for some time L_s. This is called the sampling latency of the controller. A dynamic scheduling policy will introduce variations in this interval. The sampling jitter is quantified by the difference between the maximum and minimum sampling latencies in all task instances,

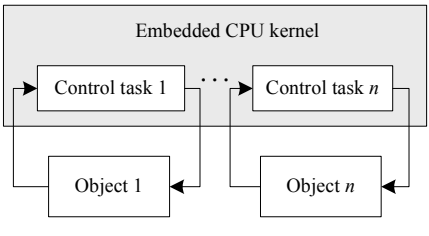


Fig. 1. Sampling control system

$$J_s \overset{def}{=} L_s^{\max} - L_s^{\min}. \tag{1}$$

Normally, it can be assumed that the minimum sampling latency of a task is zero, in which case we have $J_s = L_s^{\max}$. Jitter in the sampling latency will of course also introduce jitter in the sampling interval T. It is obvious that T is only a nominal sampling period. From Fig.2, it is seen that the actual sampling interval T_k in period k is given by

$$T_k = T + L_s^k - L_s^{k-1}. \tag{2}$$

The sampling interval jitter J_T is quantified by

$$J_T \overset{def}{=} T^{\max} - T^{\min}. \tag{3}$$

We can see that the sampling interval jitter is upper bounded by

$$J_h \leq 2J_s, \tag{4}$$

where J_h is the maximum sampling jitter.

After some computation time and possibly further preemption from other tasks, the controller will actuate the control signal. The delay from the sampling to the actuation is the input-output latency, denoted L_{io}. Varying execution times or task scheduling will lead to variations in this interval. The input-output jitter J_{io} is quantified by

$$J_{io} \overset{def}{=} L_{io}^{\max} - L_{io}^{\min} \tag{5}$$

According to the definitions of sampling jitter and input-output jitter, combined with timing parameters of hybrid tasks, sampling jitter and input-output jitter display varieties of nominal sampling period and task executions respectively. Additionally, in order to review the relation between jitter and control performance, and describe that jitter variation leads to system performance degradation, percent of nominal sampling period is used to denote jitter variation, named jitter range.

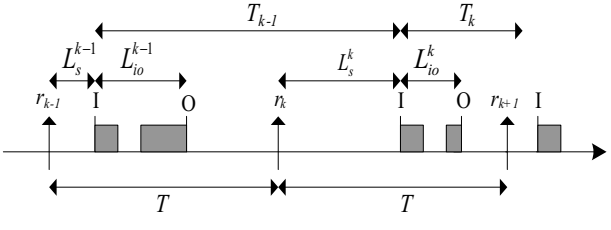


Fig. 2. The basic timing parameters of a control task

A Input-output jitter

In an embedded control system, the control task during the very execution period can be divided into three parts, namely: data sampling, control algorithm execution and signal output. In a general way, assuming that scheduling policy is based on priority, threads at a lower priority do not run until all threads with a higher priority have either finished or have been blocked. If one thread is running and a thread of higher priority is unblocked, the lower-priority thread is immediately suspended and the higher-priority thread is scheduled. Therefore, the control task may possibly be interrupted so that latency τ_c may be caused between sampling start and control signal output. If latency is variable value, then input-output jitter is introduced.

In the view of task implementation, whether it's a system task or an application task, at any time each task exists in one of a small number of states, including ready, running, or blocked. The relationship between state transitions and input-output jitter is shown in Fig.3. It can be seen that latency τ_c is the interval of state transitions.

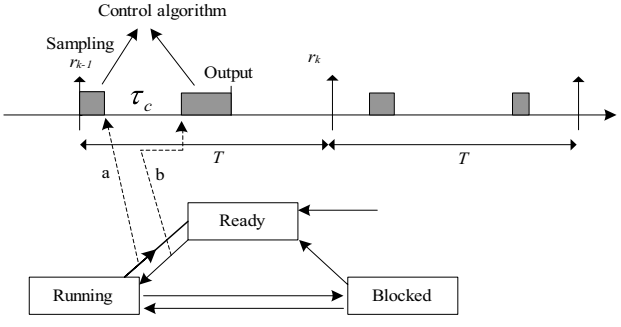


Fig. 3. Input-output jitter and task state

B Sampling jitter

To some extent, sampling jitter and task period are extremely interrelated. From the view of real-time scheduling, sampling period is namely task period. Short sampling period leads to probability increase of system resource competition, which results in some jobs that can not execute in the desired time point. On the other hand, in the control field, sampling period can generically be decided by dynamic characteristic, closed loop performance, etc. An experience theorem for determining sampling period is given by

$$4 \le \frac{T_r}{T} \le 10 \ , \tag{6}$$

where T_r is rise time.

Obviously, due to Eq.(6), the factor of embedded system implementation is neglected. In order to design high-performance system, we should further consider nondeterminacy of task timing characteristic. As shown in Fig.4, τ_s is the latency from sampling execution start to control signal output, the starting point of signal output is that the corresponding task becomes running state. Objectively speaking, the impact of τ_s on system performance is non-neglectable.

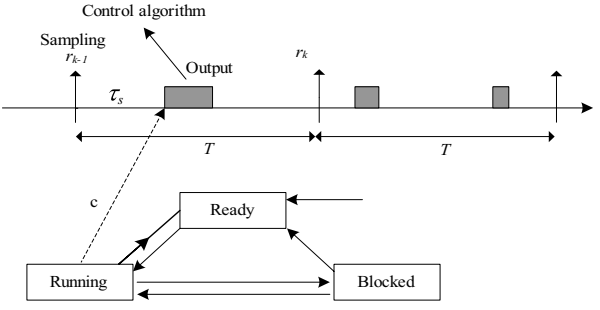


Fig. 4. Sampling jitter and task state

2.2 Non-schedulability

In the resource-constrained embedded system, the noticeable input-output jitter and sampling jitter are introduced if the period of key real-time task is selected irrationally and there are not special schedulers for management and control of tasks. In the extreme situation, some jobs can't be finished before the desired time point thus to degrade system performance. Fig.5 shows non-schedulable jobs during runtime. The non-schedulable job is given by

$$L_s^{k-1} + L_{io}^{k-1} > T \,. \tag{7}$$

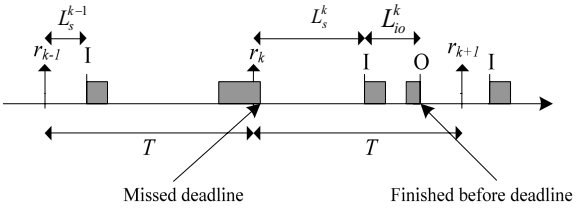


Fig. 5. Non-schedulable jobs

3 Schedulability Analysis

The best known examples of static- and dynamic-priority algorithms are RM algorithm and EDF algorithm, respectively. In this section, we will discuss schedulability of each algorithm.

For traditional RM algorithm, the task set consists of periodic, preemptible tasks whose deadlines equal the task period. RM is an optimal static-priority uniprocessor scheduling algorithm and is very popular. A task set of n tasks is schedulable under RM if its total processor utilization is no greater than $n\,(2^{1/n}\text{-}1)$ [5].

For EDF algorithm, tasks are preemptible and the task with the earliest deadline has the highest priority. EDF is an optimal uniprocessor algorithm. If a task set is not

schedulable on a single processor by EDF, there is no other processor that can successfully schedule that task set [5].

Define $u = \sum_{i=1}^{n} (e_i / P_i)$, $d_{\max} = \max_{1 \le i \le n} \{d_i\}$ and $P = lcm(P_1, P_2, \cdots, P_n)$ (Here "*lcm*" stands for least common multiple). Define $h_T(t)$ to be the sum of the execution times of all tasks in set T whose absolute deadlines are less than t. A task set of n tasks is not EDF-feasible if

- $u > 1$ or
- There exists

$$t < \min\{P + d_{\max}, \frac{u}{1-u} \max_{1 \le i \le n}(P_i - d_i)\} \tag{8}$$

such that $h_T(t) > 1$.

4 Example

The following is used to illuminate the impact of non-schedulable jobs on control system performance. Assume that all control loops in this system are independent of each other. The controlled processes are the DC servo motor modeled as [6]:

$$G(s) = \frac{1000}{s^2 + s} \quad . \tag{9}$$

The timing parameters, i.e. sampling period T and execution time c, of each control task are given in Table 1.

Table 1. Timing parameters for simulations

Task	Sampling period T t/ms	c t/ms
Task 1	6	2
Task 2	5	2
Task 3	4	2

Using RM algorithm, we can get the results for system step response and task execution in Fig.6 It can clearly be seen that task 2 and task 3 are stable, but task 1 is not stable. The reason for this is that the period of task 1 is longer than the others. So the priority of task 1 is the lowest among all tasks. According to table 1, the CPU utilization can be quantified by 2/6+2/5+2/4≈1.23>1, which means that the system is overloaded and some jobs can not be finished before their deadlines. As we can see from Fig.6, in many situations, task 1 is preempted by task 2 and task 3, which leads to the high ratio of non-schedulable jobs and affects the system balance. In contrast, the execution of Task 3 will not be affected by Task1 and Task 2 as its highest priority. However Task 2 is still possible to be preempted by Task 3.

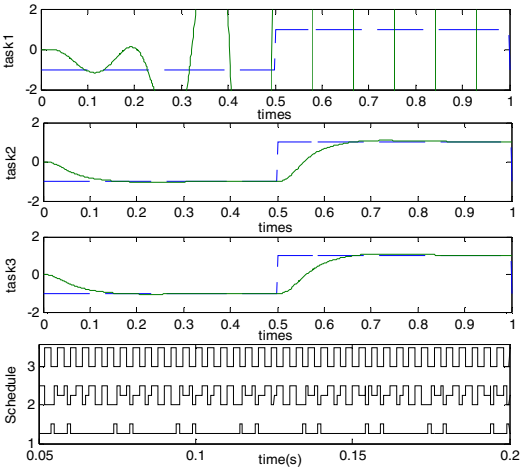


Fig. 6. Impact of non-schedulable jobs on embedded system performance

Based on above analysis, we can draw the conclusion that it is extremely important to reduce, even eliminate the non-schedulable jobs for improving control system performance.

5 Conclusions

According to the analysis of embedded system implementation, the reasons of leading to sampling jitter and input-output jitter are deeply researched. The schedulability analyses for RM algorithm and EDF algorithm are introduced. For non- schedulable jobs, an example of DC servo system is employed to testify that it is extremely important to reduce, even eliminate the non-schedulable jobs for improving system performance. This is an important reference for embedded system design.

Acknowledgements. This work was supported in part by Chinese National Natural Science Foundation under Grant 50875090, Guangdong Province Natural Science Foundation under Grant 05103543 and Program for New Century Excellent Talents in University.

References

1. Cervin, A.: Integrated Control and Real Time Scheduling. Department of Automatic Control, Lund Institute of Technology, Sweden (2003)
2. Proctor, F.M., Shackleford, W.P.: Real-time Operating System Timing Jitter and Its Impact on Motor Control. In: SPIE 2001, pp. 10–16. SPIE, Bellingham (2001)
3. Feng, S.T.: Research on Motion Control System Design Based Component, Real-time Performance Analysis and Optimized Design Method. College of Mechanical Engineering, South China University of Technology, China (2007)

4. Wan, J.F.: Study on Hybrid Task Scheduling Theory for Software CNC. College of Mechanical Engineering. South China University of Technology, China (2008)
5. Krishna, C.M., Kang, G.S.: Real-time Systems. Tsinghua Press, Beijing (2001)
6. TrueTime 1.5-reference Manual, http://www.control.lth.se/truetime/

Author Index