Applied Soft Computing 9 (2009) 71–79

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Nonlinear systems control using self-constructing wavelet networks

Cheng-Jian Lin [a,b,*]

[a] Department of Electrical Engineering, National University of Kaohsiung, Kaohsiung City 811, Taiwan, ROC
[b] Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung County 411, Taiwan, ROC

ARTICLE INFO

Article history:
Received 26 January 2007
Received in revised form 4 March 2008
Accepted 11 March 2008
Available online 4 April 2008

Keywords:
Temperature control
Wavelet neural networks
Online learning
Back-propagation
Degree measure

ABSTRACT

This paper describes a self-constructing wavelet network (SCWN) controller for nonlinear systems control. The proposed SCWN controller has a four-layer structure. We adopt the orthogonal wavelet functions as its node functions. An online learning algorithm, structure learning and parameter learning, allows the dynamic determining of the number of wavelet bases, and adjusting the shape of the wavelet bases and the connection weights. The SCWN controller is a highly autonomous system. Initially, there are no hidden nodes. They are created and begin to grow as learning proceeds. Computer simulations have been conducted to illustrate the performance and applicability of the proposed learning scheme.

1. Introduction

Nonlinear system control is becoming an important tool, which can be used to improve control performance and achieve robust fault-tolerant behavior. Among the different nonlinear control techniques, methods based on artificial neural network (ANN) have been grown into a popular research topic in recent years [1–3]. The reason is that the classical control theory usually requires a mathematical model for designing the controller. The inaccuracy of mathematical modeling of the plants usually degrades the performance of the controller, especially for nonlinear and complex control problems [4]. ANN modeling has been admitted as a powerful tool, which can facilitate the effective development of models by combining information from different sources, such as data, records. However, the ANN lacks a systematic way to determine the appropriate model structure, has no localizability, and converges slowly. A suitable approach to overcoming the disadvantages of global approximation networks is the substitution of the global activation function with localized basis functions. In this type of local network, only a small subset of the network parameters is engaged at each point in the input space. The network transparency may be improved by adopting the wavelet decomposition technique from the field of adaptive signal processing. Due to the local properties of wavelets, arbitrary functions can be approximated by the truncated discrete wavelet transform.

Recently, many researches proposed wavelet neural networks for identification and control [5–14]. Ikonomopoulos and Endou [9] proposed the analytical ability of the discrete wavelet decomposition with the computational power of radial basis function networks. Members of a wavelet family were chosen through a statistical selection criterion that constructs the structure of the network. Ho et al. [10] used the orthogonal least squares (OLS) algorithm to purify the wavelets from their candidates, which avoided using more wavelets than required and often resulted in an overfitting of the data and a poor situation in ref. [6]. Lin et al. [11] proposed a wavelet neural network to control the moving table of a linear ultrasonic motor (LUSM) drive system. They chose an initialization for the mother wavelet based on the input domains defined by the examples of the training sequence. Huang and Huang [12] proposed an evolutionary algorithm for optimally adjusted wavelet networks. However, the selections of wavelet bases were based on practical experience or trial-and-error tests.

To steady control the nonlinear systems, a self-constructing wavelet network (SCWN) controller is proposed in this paper. It is a four-layered network structure, which is comprised of an input layer, wavelet layer, product layer, and output layer. We adopt the orthogonal wavelet functions as its node functions. Based on the self-learning ability, the on-line structure/parameter learning algorithm is performed concurrently in the SCWN controller. In the structure learning scheme, the degree measure method is used to find the proper wavelet bases and to minimize the number of wavelet bases generated from input space. In parameter learning scheme, the supervised gradient descent method is applied to adjust the shape of wavelet functions and the connection weights

[*] Correspondence address: Department of Electrical Engineering, National University of Kaohsiung, Kaohsiung City 811, Taiwan, ROC.
E-mail address: cjlin@nuk.edu.tw.

1568-4946/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2008.03.014

in SCWN controller. To a great extent, the learning quality of a network is related to the parameters of feedback error. The SCWN controller based on feedback error learning strategy has favorable control performance. Finally, the proposed SCWN controller is applied to two nonlinear control problems: control for backing up the truck, and control of water bath temperature system. The proposed SCWN model has the following advantages: (1) this study adopts the wavelet network to control nonlinear systems. The local properties of wavelets in the SCWN model enable arbitrary functions to be approximated more effectively. (2) We use an online learning algorithm to automatically construct the SCWN model. No nodes or wavelet bases exist initially. They are created automatically as learning proceeds, as online incoming training data are received and as structure and parameter learning are performed. The structure learning adopts partition-based clustering techniques to perform cluster analysis in a data set. The parameter learning, based on the gradient descent method, can adjust the wavelet functions and the corresponding weights of the SCWN. (3) As demonstrated in Section 4, the SCWN model is characterized by small network size and fast learning speed.

2. Structure of the SCWN controller

The structure of the SCWN controller is shown in Fig. 1. The proposed SCWN controller is designed as a four-layer structure, which is comprised of an input layer, wavelet layer, product layer, and output layer.

The input data in the input layer of the network is $x = [x_1, x_2, \ldots, x_i, \ldots, x_n]^{\mathrm{T}}$, where T is the transpose and n is the number of dimensions. Noted that in ordinary wavelet neural network model applications, it is often useful to normalize the input vectors x into the interval [0,1]. Then, the activation functions of the wavelet nodes in the wavelet layer are derived from the mother wavelet $\phi(x)$, with a dilation of d and a translation of t [6]:

$$\phi_{d.t}(x) = 2^{d/2}\phi(2^d x - t) \tag{1}$$

The mother wavelet is selected so that it constitutes an orthonormal basis in $L^2(\Re^n)$. The derivation of a differentiable *Mexican-hat* function is adopted as a mother wavelet herein,

$$\phi(x) = (1 - ||x||^2)\, \mathrm{e}^{-||x||^2/2}, \tag{2}$$

where $||x||^2 = x^{\mathrm{T}}x$. Therefore, the activation function of the jth wavelet node connected with the ith input data is represented as:

$$\begin{gathered}\phi_{d_j t_j}(x_i) = 2^{d_{ij}/2}(1 - ||2^{d_{ij}}x_i - t_{ij}||^2)\, \mathrm{e}^{-||2^{d_{ij}}x_i - t_{ij}||^2/2}, \\ i = 1, \ldots, n; \quad j = 1, \ldots, m,\end{gathered} \tag{3}$$

where n is the number of input-dimensions and m is the number of the wavelets. The wavelet functions of (3) with various dilations and translations are presented in Fig. 2. Then, each wavelet in the product layer is labeled Π, i.e., the product of the jth multi-dimensional wavelet with n input dimensions $x = [x_1, x_2, \ldots, x_i, \ldots, x_n]^{\mathrm{T}}$ can be defined as

$$\psi_j(x) = \prod_{i=1}^{n} \phi_{d_j t_j}(x_i). \tag{4}$$

According to the theory of multi-resolution analysis (MRA) [10,13], any $f \in L^2(\Re)$ can be regarded as a linear combination of wavelets at different resolution levels. For this reason, the function f is expressed as

$$Y(x) = f(x) \approx \sum_{j=1}^{m} w_j \psi_j(x) \tag{5}$$

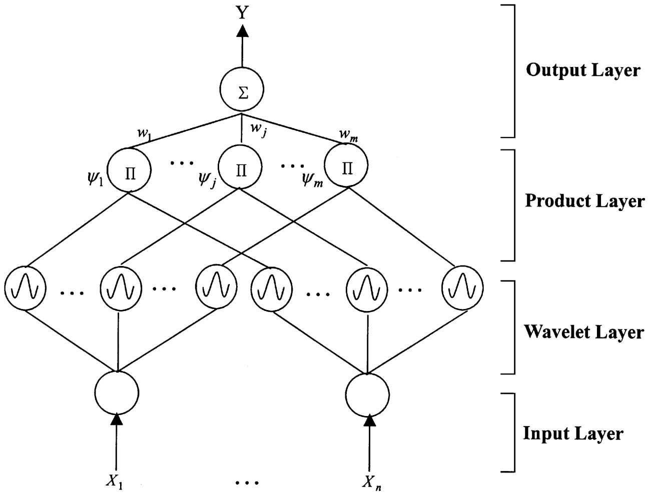

Fig. 1. The architecture of the SCWN controller.

If $\psi_j = [\psi_1, \psi_2, \ldots, \psi_m]$ is used as a nonlinear transformation function of hidden nodes and weight vectors and $w_j = w_1, w_2, \ldots, w_m$ defines the connection weights, then Eq. (5) can be considered the functional expression of the SCWN modeling function Y.

3. A self-constructing learning algorithm

In this section, the *degree measure* method and the well-known *back propagation* (BP) algorithm are used concurrently for constructing and adjusting the SCWN controller. The degree measure method is used to decide the number of wavelet bases in the wavelet layer and the product layer. On the other hand, the BP algorithm is used to adjust the parameters of the wavelet bases and connection weights. The details of the algorithm are presented below. Finally in this section, the stability analysis of the SCWN model based on the Lyapunov approach is performed the convergence property.

3.1. The structure learning scheme

Initially, there are no wavelet bases in the SCWN controller. The first task is to decide when a new wavelet base is generated. We adopt partition-based clustering techniques to perform cluster analysis in a data set. For each incoming pattern x_i, the firing strength of a wavelet base can be regarded as the degree of the incoming pattern belonging to the corresponding wavelet base. An input datum x_i with a higher firing strength means that its spatial location is nearer to the center of the wavelet base t_j than those with smaller firing strength. Based on this concept, the firing strength obtained from Eq. (4) in the product layer can be used as the degree measure

$$F_j = |\psi_j|, \quad j = 1, \ldots, q, \tag{6}$$

where q is the number of existing wavelet bases and $|\psi_j|$ is the absolute value of ψ_j. According to the degree measure, the criterion of a new wavelet base generated for new incoming data is described as follows:

Find the maximum degree $F_{\max}$

$$F_{\max} = \max_{1 \le j \le q} F_j \tag{7}$$

If $F_{\max} \le \bar{F}$, then a new wavelet base is generated, where $\bar{F}$ is a prespecified threshold that should decay during the learning process, limiting the size of the SCWN model.

In the structure learning scheme, the threshold parameter $\bar{F}$ is an important parameter. The threshold is set to between zero and one. A low threshold leads to the learning of coarse clusters (i.e., fewer hidden nodes are generated), whereas a high threshold leads to the learning of fine clusters (i.e., more hidden nodes are generated). If the threshold value equals zero, then all the training data belong to the same cluster in the input space. Therefore, the selection of the threshold value $\bar{F}$ will critically affect the simulation results. As a result of our extensive experiments and by carefully examining the threshold value $\bar{F}$, which uses the range [0,1], we concluded that the relationship between threshold value $\bar{F}$ and the number of input variables. Accordingly, $\bar{F}$ is defined as $[0.1^n, 0.5^n]$, where n is the number of input variables.

$$t_{q+1} = x_i \tag{8}$$

$$d_{q+1} = d_{\text{init}} = 0 \tag{9}$$

$$w_{q+1} = \text{random value}, \tag{10}$$

where x_i is the new incoming data; the connection weight w_{q+1} of the output layer is selected from the range between -1 and 1 randomly; and the dilation d_{q+1} is set to zero to obtain a suitable firing strength for the input value x_i (see Fig. 2).

The concise online degree measure method of the SCWN model is shown as follows:

```
Initialization;

do{
    IF x is the first incoming pattern,
        do{
            Generate a new wavelet base;
            with translation $t_{q+1} = x_i$;
            dilation $d_{q+1} = d_{init} = 0$;
            connection weight $w_{q+1} \in [-1, 1]$;
        }
    ELSE for each newly incoming pattern
        do{
            Executing the degree measure method
            IF $F_{\max} \le \overline{F}$
              do{
                  Generate a new wavelet base;
                  with translation $t_{q+1} = x_i$;
                  dilation $d_{q+1} = d_{init} = 0$;
                  connection weight $w_{q+1} \in [-1, 1]$;
              }
          }
} until the task is finished
```

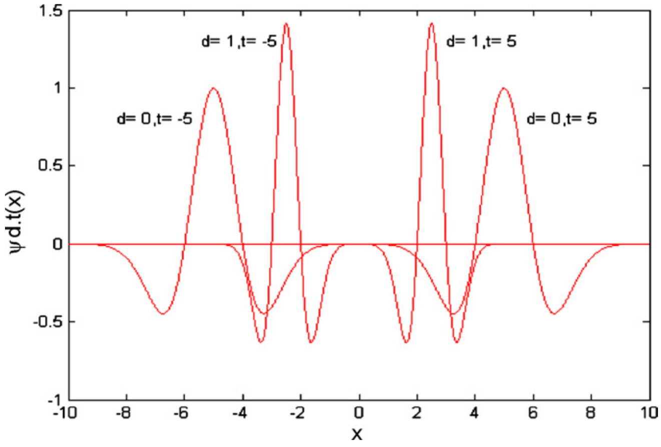

Fig. 2. Wavelet bases with various dilations and translations.

3.2. The parameter learning scheme

After the network structure has been adjusted according to the current training pattern, the network then enters the second learning step to adjust the parameters of the wavelet base and the connection weight (t, d, and w) with the same training pattern. The parameter-learning algorithm is based on a set of input/output pairs $\{x, y^{\mathrm{d}}(x)\}$. If the error function is

$$e = y(x) - y^{\mathrm{d}}(x), \tag{11}$$

where $y(x)$ is the model output and $y^{\mathrm{d}}(x)$ is the desired output, then the cost function E can be defined as

$$E = \frac{1}{2}e^2 \tag{12}$$

and can be minimized by all adjustable parameters using an iterative computational scheme.

Assuming that W is the adjustable parameter in the wavelet layer and the output layer, the general learning rule used is

$$W(k+1) = W(k) + \Delta W(k) = W(k) + \eta\left(-\frac{\partial E}{\partial W}\right), \tag{13}$$

where η and k represent the learning rate and the iteration number, respectively. The gradient of the cost function E in Eq. (12) with respect to the vector of arbitrarily adjustable parameter W is defined as

$$\frac{\partial E}{\partial W} = -e\frac{\partial y}{\partial W} \tag{14}$$

With the above equation defined, we can infer that the free parameters adjusted in the SCWN are as follows.

The connection weight of the output layer is updated by

$$w_j(k+1) = w_j(k) + \Delta w_j \tag{15}$$

where

$$\Delta w_j = -\eta_{\mathrm{w}}\frac{\partial E}{\partial w_j} = -\eta_{\mathrm{w}} e \psi_j \tag{16}$$

Similarly, the updated laws of t_{ij} and d_{ij} are shown as follows:

$$t_{ij}(k+1) = t_{ij}(k) + \Delta t_{ij} \tag{17}$$

$$d_{ij}(k+1) = d_{ij}(k) + \Delta d_{ij} \tag{18}$$

where

$$\Delta t_{ij} = -\eta_t \frac{\partial E}{\partial t_{ij}} = -\eta_t e w_j \psi_j \left\{ \frac{2^{d_{ij}}x_i - t_{ij}}{1-(2^{d_{ij}}x_i - t_{ij})^2}[2+(1-(2^{d_{ij}}x_i - t_{ij})^2)] \right\} \tag{19}$$

$$\Delta d_{ij} = -\eta_d \frac{\partial E}{\partial d_{ij}} = -\eta_d e w_j \psi_j \left\{ \frac{\ln 2}{2} - 2^{d_{ij}} x_i \ln 2 (2^{d_{ij}}x_i - t_{ij}) \left[1 + \frac{2}{1-(2^{d_{ij}}x_i - t_{ij})^2} \right] \right\} \tag{20}$$

3.3. Convergence analysis

The selection of suitable learning rates is very important. If the learning rate is small, convergence will be guaranteed. In this case, the speed of convergence may be slow. However, the learning rate is large, and then the system may become unstable. Therefore, we set a suitable learning rate as 0.05 in this study. Appendix A derives varied learning rates, which guarantee convergence of the output error based on the analyses of a discrete Lyapunov function, to train the SCWN model effectively. The convergence analyses in this study are performed to derive specific learning rate parameters for specific network parameters to ensure the convergence of the output error [15–17]. Moreover, the guaranteed convergence of output error does not imply the convergence of the learning rate parameters to their optimal values. The following simulation results demonstrate the effectiveness of the online learning SCWN model based on the proposed delta adaptation law and varied learning rates.

4. Illustrative examples

In this section, two control examples are given to demonstrate the validity of the presented SCWN controller. The first example is to control the truck backing-upper [18], the second example is to control the water bath temperature system [19].

4.1. Control for backing up the truck

Backing a truck to a loading dock is a nonlinear control problem for which no traditional control design methods exits. In this example, we develop a controller (called SCWN controller) to back up a simulated truck to a loading dock in a planar parking lot. This SCWN controller enables the truck to reach the desired position successfully.

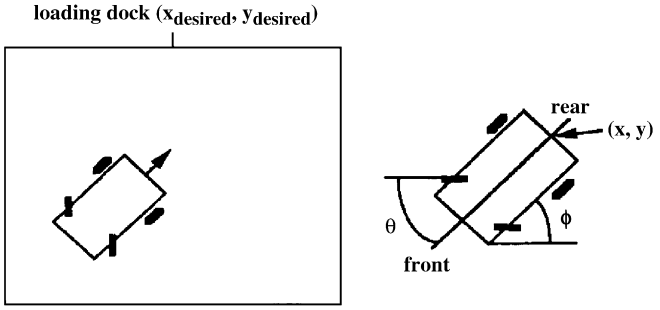

Fig. 3. Diagram of the simulated truck and loading zone.

The simulated truck and loading zone are shown in Fig. 3. The truck position is exactly determined by three state variables where ϕ is the angle of truck with the horizontal and the coordinate pair (x, y) specifies the position of the rear center of the truck in the plane. The steering angle θ of the truck is the controlled variable. The positive values of θ represent clockwise rotations of the steering wheel and negative values represent counterclockwise rotations. The truck is placed at some initial position and is backed up while being steered by the controller. The objective of this control problem is to use backward movements of the truck only make the truck arrive at the loading dock at right angle ($\phi_{\text{desired}} = 90^\circ$) and to have the position of the truck with the desired loading dock ($x_{\text{desired}} = 50$, $y_{\text{desired}} = 100$). The truck moves backward by fixed distance 'dis' of the movement of the steering wheel at every step. The loading region is limited to the plane [0,100] × [0,100].

The input and output variables of the SCWN controller must be specified. The controller has two inputs, truck angle ϕ and the crossposition x. Assuming enough clearance between the truck and the loading dock, the y coordinate not considered as an input variable. The output of controller is the steering angle θ. The range of the variables x, ϕ and θ are as follows:

$$\begin{aligned} &0 \le x \le 100 \\ &-90^\circ \le \phi \le 270^\circ \\ &-30^\circ \le \theta \le 30^\circ \end{aligned} \tag{21}$$

The equations of backward motion of the truck are given by

$$\begin{aligned} &x(K+1) = x(K) + \text{dis} \cos\theta(K) \cos\phi(K) \\ &y(K+1) = y(K) + \text{dis} \cos\theta(K) \cos\phi(K) \\ &\phi(K+1) = \tan^{-1}\left[\frac{l \sin\phi(K) + \text{dis}\cos\phi(K)\sin\theta(K)}{l\cos\phi(K) - \text{dis}\sin\phi(K)\sin\theta(K)}\right] \end{aligned} \tag{22}$$

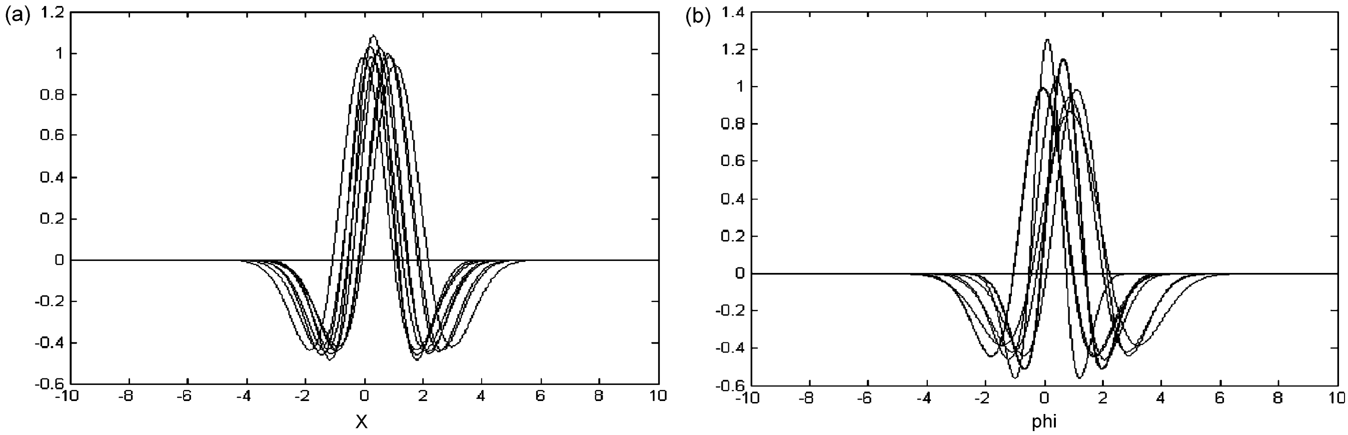

Fig. 4. The final distribution of the wavelet bases on the input variable.

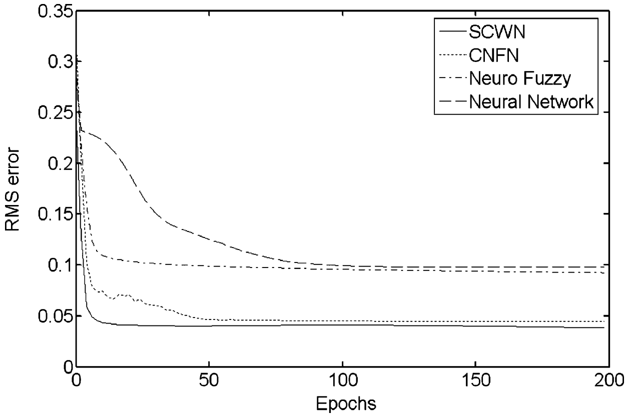

Fig. 5. Learning curve of various existing models in example 1.

where l is the length of the truck. Eq. (22) is used to obtain the next state when the present state is given. For the purpose of training the SCWN controller, learning takes place during six different tries, each starting from an initial state and terminate when the desired state is reached. The truck moves by a small fixed distance dis = 1.6 at every step and the length of the truck is set to as $l = 1$. The learning parameters are $\eta = 0.05$, $d_{\text{init}} = 0$, and $\bar{F} = 0.25$. The training process is continued for 200 epochs, and was performed 30 times. The lowest RMS (root mean square) error approximates 0.039. After the on-line structure-parameter learning, there are 9 wavelet nodes generated in our simulation as shown in Fig. 4a and b. Fig. 5 shows the RMS error for the trained SCWN controller. The control surface of the learned SCWN controller is shown in Fig. 6. In this figure, the steering signal outputs with respect to all the combinations of two input variable values x and ϕ after learning the six sets of training trajectories. After the training process is terminated, Fig. 7a–c shows the trajectories of the moving truck controlled by the SCWN controller starting from the initial position (x,y,ϕ) = (a) $(10^\circ,30^\circ,-30^\circ)$, (b) $(30^\circ,20^\circ,250^\circ)$, and (c) $(70^\circ,20^\circ,-30^\circ)$. We now compare the performance of our model with that of other existing methods [20–23]. The comparison results are tabulated in Table 1. Table 2 shows the CPU time of the

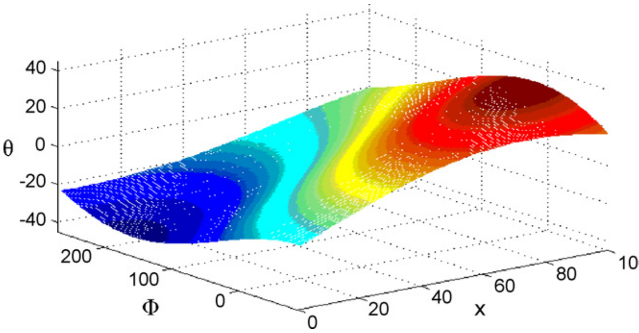

Fig. 6. Three-dimensional (3D) control surface of the learned SCWN controller in example 1.

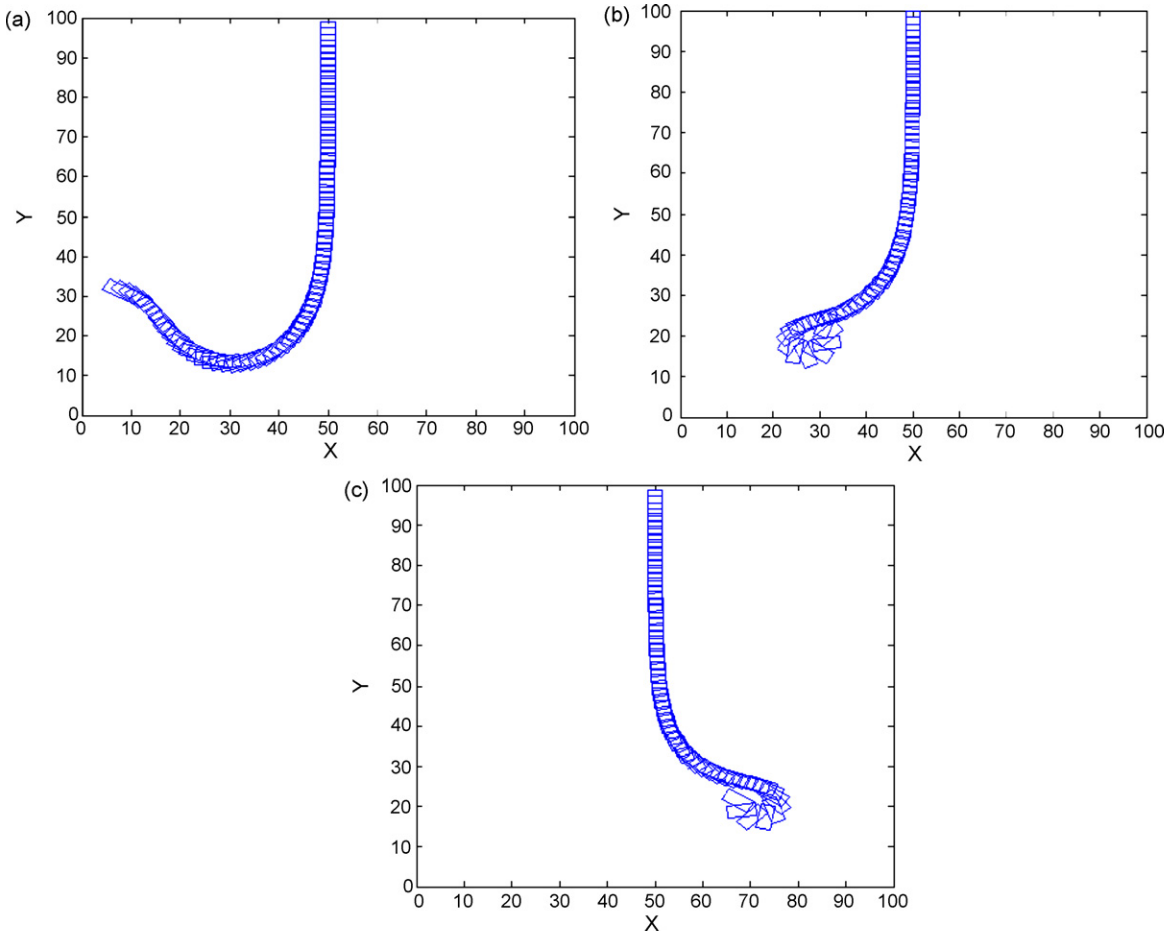

Fig. 7. Truck moving trajectories starting at three different initial positions under the control of the SCWN controller after learning six sets of training trajectories.

Table 1
Performance comparison of various existing models in example 1

	SCWN	CNFN [22]	FALCON [20]	Neuro-fuzzy [21]	Neural networks [23]
Training epochs	200	200	600	200	200
Number of rules/hidden nodes	9	13	19	35	25
RMS errors	0.039	0.0449	2.3	0.0924	0.0978

Table 2
The average training time using various methods in example 1

Method	Average training time (s)
SCWN	62.8281
CNFN [22]	63.7188
Neuro-fuzzy [21]	124.0313
Neural networks [23]	256.8438

cost of the SCWN controller, the CNFN model, the neuro-fuzzy model, and neural networks. The average training times of the SCWN controller, the CNFN model, the neuro-fuzzy model, and neural networks were 62.8281, 63.7188, 124.0313, 256.8438 s, respectively. The computation time was measured on a personal computer with an Intel Pentium 4 (2.8 GHz) CPU inside. Simulation results show that the proposed SCWN controller obtains a smaller RMS error and needs fewer CPU time by using fewer hidden nodes than other methods.

4.2. Control of water bath temperature system

Consider a discrete time temperature-control system:

$$y(t+1) = \mathrm{e}^{-\alpha T_s} y(k) + \frac{(\beta/\alpha)(1-\mathrm{e}^{-\alpha T_s})}{1+\mathrm{e}^{0.5y(k)-40}} u(k) + [1-\mathrm{e}^{-\alpha T_s}] y_0. \qquad (23)$$

The above equation models real water bath temperature-control systems give in ref. [19]. The parameters in this example are $\alpha = 1.0015\mathrm{e}^{-4}$ and $\beta = 8.67973\mathrm{e}^{-3}$ and $y_0 = 25$ (°C). The input $u(k)$ is limited to 0 and 5 V represent voltage unit. The sampling period is $T_s = 30$. The system configuration is shown in Fig. 8, where y_{ref} is the desired temperature of the controlled plant. By implement the on-line training scheme for SCWN controller, a sequence of random input signals $u_{\mathrm{rd}}(k)$ limited to 0 and 5 V is injected directly into the simulated system described in (23). The 120 training patterns are chosen from the input–outputs characteristic in order to cover the entire reference output. The initial temperature of the water is 25 °C, and the temperature rises progressively when random input signals are injected.

In this paper, we compare the SCWN controller to the PID controller, the manually designed fuzzy controller, neural networks (NN) controller and self-constructing fuzzy neural networks (SCFNN) [24] controller. For the PID control, a velocity-form discrete PID controller is described by

$$\begin{aligned}\Delta u(k) &= K\{e(k) - e(k-1) + \frac{T_s}{2T_i}[e(k) + e(k-1)] + \frac{T_d}{T_s}[e(k) \\ &\quad - 2e(k-1) + e(k-2)]\} \\ &= K_P[e(k) - e(k-1)] + K_I e(k) + K_D[e(k) - 2e(k-1) \\ &\quad + e(k-2)] \end{aligned} \qquad (24)$$

where $K_P = K - (1/2)K_I, K_I = (KT_s/T_i), K_D = (kT_d/T_s)$. The parameter $\Delta u(k)$ is the increment of the control input, $e(k)$ is the performance error at the sampling instant k, and K_P, K_I, and K_D are the proportional, integral, and derivative parameters, respectively. In order not to aggravate noise in the plant, only a two-term PID controller is used, i.e., K_D is set to zero in the water bath system. The other two parameters K_P and K_I are set as 80 and 70, respectively.

For the manually designed fuzzy controller, the input variables are chosen as $e(t)$ and ce(t), where $e(t)$ is the performance error indicating the error between the desired water temperature and the actual measured temperature and ce(t) is the rate of change in the performance error $e(t)$. The output or the controlled linguistic variable is the voltage signal $u(t)$ to the heater. Seven fuzzy terms are defined for each linguistic variable. These fuzzy terms consist of negative large (NL), negative medium (NM), negative small (NS), zero (ZE), positive small (PS), positive medium (PM), and positive large (PL). Each fuzzy term is specified by a Gaussian membership function. According to common sense and engineering judgment, 25 fuzzy rules are specified in Table 3. Like other controllers, a fuzzy controller has some scaling parameters to be specified. They are GE, GCE, and GU, corresponding to the process error, the change in error, and the controller's output, respectively. We choose these parameters as follows: GE = 1/15, GCE = 1/15, GU = 450.

Lin et al. [24] presented a self-constructing fuzzy neural network (SCFNN), which is suitable for practical implementation. The SCFNN is also using on-line learning scheme to decide the structure of fuzzy rules and to turn the adjustable parameters through the backpropagation algorithm in SCFNN model.

For the aforementioned controllers (SCWN controller, PID controller, manually designed fuzzy controller, NN controller and SCFNN controller), for groups of computer simulations are conducted on the water bath temperature control system. Each simulation is performed over 120 sampling time steps.

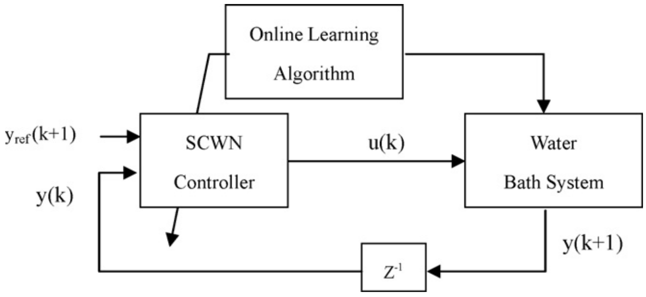

Fig. 8. Flow diagram of using SCWN controller for solving the temperature control problem.

Table 3
Fuzzy rule table formulated for the water bath temperature control system

		Error, $e(t)$						
		NL	NM	NS	ZE	PS	PM	PL
Change error, ce(t)	PL				PL	PL	PL	PL
	PM				PM	PM	PM	PL
	PS			PS	PS	PS	PM	PL
	ZE	NL	NM	NS	ZE	PS	PM	PL
	NS			NS	NS	NS		
	NM				NM			
	NL				NL			

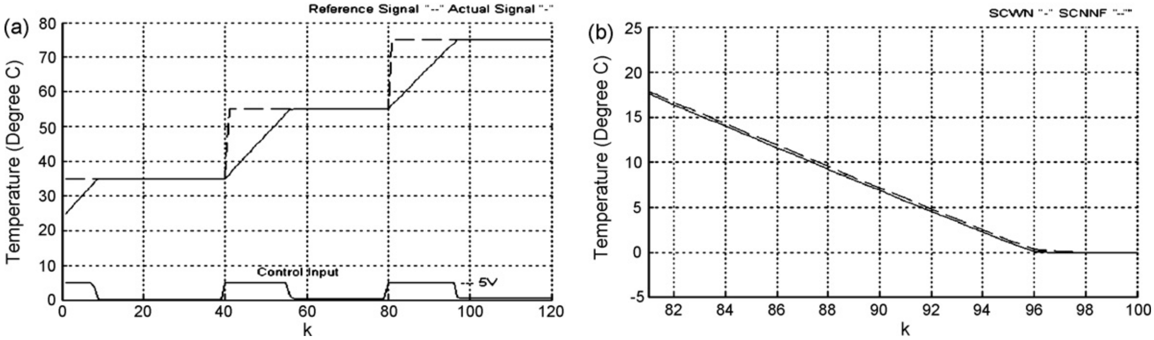

Fig. 9. (a) Final regulation performance of the SCWN controller for water bath system. (b) Corresponding errors of SCWN controller and SCFNN controller.

Table 4
Performance comparison of various controllers

$\mathrm{SAE} = \sum_{k=1}^{120} \lvert y_{\mathrm{ref}}(k) - y(k) \rvert$	SCWN controller	PID controller	Fuzzy controller	NN controller	SCFNN controller [24]
Regulation performance	353.33	418.5	401.5	364.62	356.41
Influence of impulse noise	270.5	311.5	275.8	272.17	280.50
Effect of change in plant dynamics	261.84	322.2	273.5	262.80	268.21
Number of adjustable parameters	195	3	150	241	205

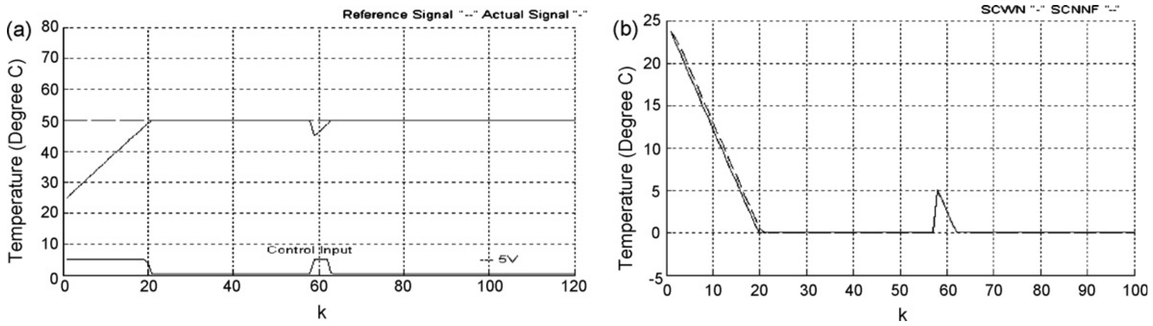

Fig. 10. (a) Behavior of the SCWN controller under the impulse noise for water bath system. (b) Corresponding errors of SCWN controller and SCFNN controller.

The first task is to control the simulated system to follow three set-points.

$$y_{\mathrm{ref}}(k) = \begin{cases} 35\,^{\circ}\mathrm{C}, & \text{for}\, k \le 40 \\ 55\,^{\circ}\mathrm{C}, & \text{for}\, 40 < k \le 80 \\ 75\,^{\circ}\mathrm{C}, & \text{for}\, 80 < k \le 120. \end{cases} \tag{25}$$

For the proposed SCWN controller, the learning rate $\eta = 0.05$, the initial variance $d_{init} = 0$ and the prespecified threshold $\bar{F} = 0.02$ are chosen. The learning proceeded for 10,000 epochs, and was performed 30 times. The regulation performance of the SCWN controller is shown in Fig. 9a, and the corresponding errors for SCWN controller and SCFNN controller [24] are shown in Fig. 9b. As shown in the error curves, the SCWN controller has much smaller error than SCNNF controller. To test their regulation performance, a performance index, sum of absolute error (SAE), is defined by

$$\mathrm{SAE} = \sum_{k} \lvert y_{\mathrm{ref}}(k) - y(k) \rvert \tag{26}$$

where $y_{\mathrm{ref}}(k)$ and $y(k)$ are the reference output and the actual output of the simulated system, respectively. The SAE values of the SCWN controller, the PID controller, the fuzzy controller, the NN controller and the SCFNN controller are 353.33, 418.5, 401.5, 364.52 and 356.41, which are shown in the first column of Table 4.

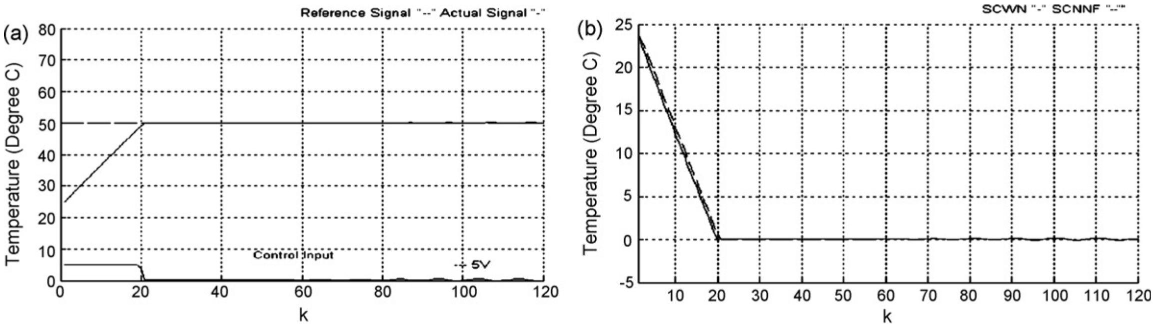

Fig. 11. (a) Behavior of the SCWN controller when a change occurs in the water bath system. (b) Corresponding errors of SCWN controller and SCFNN controller.

The proposed SCWN controller obtains much better SAE value of regulation performance than other methods.

The second set of simulations is carried out for the purpose of studying the noise-rejection ability of the four controllers when some unknown impulse noise is imposed on the process. One impulse noise value -5 °C is added to the plant output at the sixtieth sampling instant. A set-point of 50 °C is performed in this set of simulations. For the SCWN controller, the same training scheme, training data and learning parameters are used as those used in the first set of simulations. The behaviors of the SCWN controller under the influence of impulse noise and the corresponding errors are shown in Fig. 10a and b. The SAE values of the SCWN controller, the PID controller, the fuzzy controller, the NN controller and the SCFNN are 270.5, 311.5, 275.8, 272.17 and 280.5, which are shown in the second column of Table 4. It is observed that the SCWN controller performs quite well. It recovers very quickly and steadily after the presentation of the impulse noise.

One common characteristic of many industrial-control processes is that their parameters tend to change in an unpredictable way. To test the robustness of the four controllers, a value of $0.7u(k-2)$ is added to the plant input after the sixtieth sample in the fourth set of simulations. A set-point of 50 °C is used in this set of simulations. For the SCWN controller, the same training scheme, training data and learning parameters are used as those used in the first set of simulations. The behaviors of the SCWN controller when there is a change in the plant dynamics are shown in Fig. 11a. The corresponding errors of the SCWN controller and SCNNF controller [24] are shown in Fig. 11b. The SAE values of the SCWN controller, the PID controller, the fuzzy controller, the NN controller and the SCFNN controller are 261.84, 322.2, 273.5, 262.8 and 270.21, which are shown in the third column of Table 4. The results show the good control and disturbance rejection capabilities of the trained SCWN controller in the water bath system.

5. Conclusion

In this paper, we propose a self-constructing wavelet network (SCWN) for nonlinear systems control. The goal of SCWN controller is to improve control performance and achieve robust fault-tolerant behavior. An on-line structure/parameter learning algorithm is proposed concurrently in the SCWN controller. The degree measure method is used to find the proper wavelet bases from input space while the supervised gradient descent method is used to adjust the shape of wavelet functions and the connection weights in SCWN controller. Experimental results have been given to illustrate the performance and effectiveness of the proposed SCWN controller. We also compare the performance of SCWN controller with other existing models. The computer simulations demonstrate that the proposed SCWN controller can obtain a smaller RMS error and a quicker convergence than other methods by small network size for nonlinear systems control problems.

Acknowledgement

This work was supported by National Science Council, ROC, under grant NSC95-2221-E-390-040-MY2.

Appendix A

A.1. Proof of convergence theorem

Theorem 1. *Let η_{w} be the learning rate parameter of the SCWN weight, and let $P_{\mathrm{w\,max}}$ be defined as $P_{\mathrm{w\,max}} \equiv \max_k ||P_{\mathrm{w}}(k)||$, where $P_{\mathrm{w}}(k) = \partial y/\partial w_j$ and $||\cdot||$ is the Euclidean norm in $\Re^N$. The convergence is guaranteed if η_{w} is chosen as $\eta_{\mathrm{w}} = \lambda/(P_{\mathrm{w\,max}})^2 = \lambda/q$, in which λ is a positive constant gain, and q is the number of existing wavelet bases in the SCWN model.*

Proof of Theorem 1: Since

$$P_{\mathrm{w}}(k) = \frac{\partial y}{\partial w_j} = \psi_j \tag{A.1}$$

and $\psi_j \le 1$, the following result holds;

$$||P_{\mathrm{w}}(k)|| \le \sqrt{q}. \tag{A.2}$$

Then, a discrete Lyapunov function is selected as

$$V(k) = \frac{1}{2} e^2(k). \tag{A.3}$$

The change in the Lyapunov function is obtained as

$$\Delta V(k) = V(k+1) - V(k) = \frac{1}{2}[e^2(k+1) - e^2(k)] \tag{A.4}$$

The error difference can be represented as [16]

$$e(k+1) = e(k) + \Delta e(k) = e(k) + \left[\frac{\partial e(k)}{\partial w_j}\right]^{\mathrm{T}} \Delta w_j \tag{A.5}$$

where Δe and Δw_j represent the output error change and the weight change in the output layer, respectively. Eqs. (15) and (A.5) yield

$$\frac{\partial e(k)}{\partial w_j} = \frac{\partial e(k)}{\partial y}\frac{\partial y}{\partial w_j} = P_{\mathrm{w}}(k) \tag{A.6}$$

$$e(k+1) = e(k) - P_{\mathrm{w}}^{\mathrm{T}}(k)\eta_{\mathrm{w}} e(k) P_{\mathrm{w}}(k). \tag{A.7}$$

Then,

$$\begin{aligned} ||e(k+1)|| &= ||e(k)[1 - \eta_{\mathrm{w}} P_{\mathrm{w}}^{\mathrm{T}}(k) P_{\mathrm{w}}(k)]|| \\ &\le ||e(k)|| ||1 - \eta_{\mathrm{w}} P_{\mathrm{w}}^{\mathrm{T}}(k) P_{\mathrm{w}}(k)|| \end{aligned} \tag{A.8}$$

is true. If $\eta_{\mathrm{w}} = (\lambda/(P_{\mathrm{w\,max}}{}^2)) = (\lambda/q)$ is chosen, then the term $||1 - \eta_{\mathrm{w}} P_{\mathrm{w}}^{\mathrm{T}}(k) P_{\mathrm{w}}(k)||$ in Eq. (A.8) is less than 1. Therefore, the Lyapunov stability of $V > 0$ and $\Delta V > 0$ is guaranteed. The output error between the reference model and actual plant converges to zero as $t \to \infty$. This fact completes the proof of the theorem.

The following lemmas are used to prove Theorem 2.

Lemma 1. *Let* $g(x) = (2^{d_{ij}} x_i - t_{ij}/1 - (2^{d_{ij}} x_i - t_{ij})^2)[2 + (1 - (2^{d_{ij}} x_i - t_{ij})^2)]$, *then* $g(x)| < 1$, if $|d| < 1$, $\forall x \in \Re$.

Lemma 2. *Let* $f(x) = (\ln 2/2) - 2^{d_{ij}} x_i \ln 2(2^{d_{ij}} x_i - t_{ij})[1 + (2/1 - (2^{d_{ij}} x_i - t_{ij})^2)]$, *then* $f(x)| < 1$, if$|d| < 1$, $\forall x \in \Re$.

Theorem 2. *Let η_{t} and η_{d} be the learning rate parameters of the translation and dilation of the wavelet function for the SCWN; let P_{tmax} be defined as $P_{\mathrm{tmax}} \equiv \max_k ||P_{\mathrm{t}}(k)||$, where $P_{\mathrm{t}}(k) = \partial_y/\partial t_{ij}$; let P_{dmax} be defined as $P_{\mathrm{dmax}} \equiv \max_k ||P_{\mathrm{d}}(k)||$, where $P_{\mathrm{d}}(k) = \partial_y/\partial t_{ij}$.* The convergence is guaranteed if *η_{t} and η_{d} are chosen as $\eta_{\mathrm{t}} = \eta_{\mathrm{d}} = \eta_{\mathrm{w}}[|w_j|_{\max}]^{-2}$, in which $w_j|_{\max} = \max_k |w_j(k)|$ and $|\cdot|$ is the absolute value.*

Proof of Theorem 2: According to Lemma 1, $|(\ln 2/2) - 2^{d_{ij}}x_i \ln 2(2^{d_{ij}}x_i - t_{ij})[1 + (2/(1 - (2^{d_{ij}}x_i - t_{ij})^2))]| < 1$. The upper bounds on $P_t(k)$ can be derived as follows;

$$
\begin{aligned}
&P_t(k)\\
&= \frac{\partial y}{\partial t_{ij}} < \left|\sum_{j=1}^{q} w_j\right| \left|\psi_j \left\{ \frac{2^{d_{ij}}x_i - t_{ij}}{1 - (2^{d_{ij}}x_i - t_{ij})^2} [2 + (1 - (2^{d_{ij}}x_i - t_{ij})^2))] \right\}\right|\\
&< \left|\sum_{j=1}^{q} w_j\right| < \sqrt{q}\ |w_j|_{\max} \qquad (A.9)
\end{aligned}
$$

Thus,

$$||P_t(k)|| < \sqrt{q}\ |w_j|_{\max}. \qquad (A.10)$$

The error difference can also be represented as [16]

$$e(k+1) = e(k) + \Delta e(k) = e(k) + \left[\frac{\partial e(k)}{\partial t_{ij}}\right]^{T} \Delta t_{ij} \qquad (A.11)$$

where Δt_{ij} represents the change of the translation of the wavelet function in the membership function layer. Eqs. (17) and (A.11) yield

$$\frac{\partial e(k)}{\partial t_{ij}} = \frac{\partial e(k)}{\partial y} \frac{\partial y}{\partial t_{ij}} = P_t(k) \qquad (A.12)$$

$$e(k+1) = e(k) - P_t^{T}(k)\eta_t e(k) P_t(k). \qquad (A.13)$$

Then,

$$
\begin{aligned}
||e(k+1)|| &= ||e(k)[1 - \eta_t P_t^{T}(k) P_t(k)]||\\
&\le ||e(k)||\,||1 - \eta_t P_t^{T}(k) P_t(k)|| \qquad (A.14)
\end{aligned}
$$

is true. If $\eta_t = \lambda/(P_{t\max})^2 = \eta_w [w_j|_{\max}]^{-2}$ is chosen, then the term $||1 - \eta_w P_w^{T}(k) P_w(k)||$ in Eq. (A.14) is less than 1. Therefore, the Lyapunov stability of $V > 0$ and $\Delta V < 0$ given by Eqs. (A.3) and (A.4), is guaranteed. The output error between the reference model and actual plant converges to zero as $t \to \infty$.

According to Lemma 2, $(\ln 2/2) - 2^{d_{ij}}x_i \ln 2(2^{d_{ij}}x_i - t_{ij})[1 + (2/(1- (2^{d_{ij}}x_i - t_{ij})^2))]| < 1$. The upper bounds on $P_d(k)$ can be derived as follows;

$$
\begin{aligned}
&P_d(k)\\
&= \frac{\partial y}{\partial d_{ij}} < \left|\sum_{j=1}^{q} w_j\right| \left|\frac{\ln 2}{2} - 2^{d_{ij}}x_i \ln 2(2^{d_{ij}}x_i - t_{ij}) \left[1 + \frac{2}{1 - (2^{d_{ij}}x_i - t_{ij})^2}\right]\right|\\
&< \left|\sum_{j=1}^{q} w_j\right| < \sqrt{q}\, |w_j|_{\max} \qquad (A.15)
\end{aligned}
$$

Thus,

$$||P_d(k)|| < \sqrt{q}\ |w_j|_{\max}. \qquad (A.16)$$

The error difference can be represented as

$$e(k+1) = e(k) + \Delta e(k) = e(k) + \left[\frac{\partial e(k)}{\partial d_{ij}}\right]^{T} \Delta d_{ij} \qquad (A.17)$$

where Δd_{ij} represents the change of the dilation of the wavelet function in the membership function layer. Eqs. (18) and (A.17) yield

$$\frac{\partial e(k)}{\partial d_{ij}} = \frac{\partial e(k)}{\partial y} \frac{\partial y}{\partial d_{ij}} = P_d(k) \qquad (A.18)$$

$$e(k+1) = e(k) - P_d^{T}(k)\eta_d e(k) P_d(k). \qquad (A.19)$$

Then,

$$
\begin{aligned}
||e(k+1)|| &= ||e(k)[1 - \eta_d P_d^{T}(k) P_d(k)]||\\
&\le ||e(k)||\,||1 - \eta_d P_d^{T}(k) P_d(k)|| \qquad (A.20)
\end{aligned}
$$

is true. If $\eta_d = \lambda/(P_{d\max})^2 = \eta_w [w_j|_{\max}]^{-2}$ is chosen, then the term $||1 - \eta_d P_d^{T}(k) P_d(k)|$ in Eq. (A.20) is less than 1. Therefore, the Lyapunov stability of $V > 0$ and $\Delta V < 0$ given by Eqs. (A.3) and (A.4) is guaranteed. The output error between the reference model and actual plant converges to zero as $t \to \infty$. This fact completes the proof of the theorem.

References

[1] M. Khalid, S. Omatu, A neural network controller for a temperature control system, IEEE Trans. Control Syst. (1992).

[2] J.D. Johnson, Neural networks for control, Neurocomputing 14 (3) (1997) 301–302.

[3] F.J. Lin, C.H. Lin, C.H. Hong, Robust control of linear synchronous motor servo drive using disturbance observer and recurrent neural network compensator, Electr. Power Appl. 147 (4) (2000) 263–272.

[4] K.J. Astrom, B. Wittenmark, Adaptive Control, Addison-Wesley, Reading, MA, 1989.

[5] M.R. Sanner, J.J.E. Slotine, Structureally dynamic wavelet networks for adaptive control of robotic systems, Int. J. Control 70 (3) (1998) 405–421.

[6] Q. Zhang, Using wavelet networks in nonparametric estimation, IEEE Trans. Neural Networks 8 (1998) 227–236.

[7] J. Zhang, G.G. Walter, Y. Miao, W.N. Wayne Lee, Wavelet neural networks for function learning, IEEE Trans. Signal Process. 43 (6) (1997) 1485–1497.

[8] J. Chen, D.D. Bruns, WaveARX neural network development for system identification using a systematic design synthesis, Ind. Eng. Chem. Res. 34 (1995) 4420–4435.

[9] A. Ikonomopoulos, A. Endou, Wavelet decomposition and radial basis function networks for system monitoring, IEEE Trans. Nucl. Sci. 45 (5) (1998) 2293–2301.

[10] D.W.C. Ho, P.A. Zhang, J. Xu, Fuzzy wavelet networks for function learning, IEEE Trans. Fuzzy Syst. 9 (1) (2001) 200–211.

[11] F.J. Lin, R.J. Wai, M.P. Chen, Wavelet neural network control for linear ultrasonic motor drive via adaptive sliding-mode technique, IEEE Trans. Ultrason. Ferroelectr. Freq. Control 50 (6) (2003) 686–698.

[12] Y.C. Huang, C.M. Huang, Evolving wavelet networks for power transformer condition monitoring, IEEE Trans. Power Deliv. 17 (2) (2002) 412–416.

[13] L. Jiao, J. Pan, Y. Fang, Multiwavelet neural network and its approximation properties, IEEE Trans. Neural Networks 12 (5) (2001) 1060–1066.

[14] I. Daubechies, Orthonormal bases of compactly supported wavelets, Comm. Pur. Appl. Math. 41 (1998).

[15] P. Niyogi, F. Girosi, On the relationship between generalization error, hypothesis complexity, and sample complexity for radial basis functions, Neural Comput. 8 (1996).

[16] C.C. Ku, K.Y. Lee, Diagonal recurrent neural networks for dynamic systems control, IEEE Trans. Neural Networks 6 (1995) 144–156.

[17] Y.C. Chen, C.C. Teng, A model reference control structure using a fuzzy neural network, Fuzzy Sets Syst. 73 (1995) 291–312.

[18] D. Nguyen, B. Widrow, The truck backer-upper: an example of self-learning in neural network, IEEE Conf. Syst. Mag. 10 (3) (1990) 18–23.

[19] C.T. Lin, C.F. Juang, C.P. Li, Temperature control with a neural fuzzy inference network, IEEE Trains. Syst. Man Cybernet. Part C: Appl. Rev. 29 (3) (1999).

[20] C.J. Lin, C.T. Lin, An ART-based fuzzy adaptive learning control network, IEEE Trans. Fuzzy Syst. 5 (4) (1997) 477–496.

[21] H. Nomura, I. Hayashi, N. Wakami, A learning method of fuzzy inference rules by descent method, IEEE Conf. Fuzzy Syst. (1992) 203–210.

[22] C.J. Lin, C.H. Chen, Nonlinear system control using compensatory neuro-fuzzy networks, IEICE Trans. Fundam. Electron. Commun. Comput. Sci. E86-A (9) (2003) 2309–2316.

[23] C.W. Anderson, Strategy learning with multilayer connectionist representations, in: Proceedings of the Fourth International Workshop on Machine Learning, Irvine, CA, (1987), pp. 103–114.

[24] F.J. Lin, C.H. Lin, P.H. Shen, Self-constructing fuzzy neural network speed controller for permanent-magnet synchronous motor drive, IEEE Trans. Fuzzy Syst. 9 (2001) 751–759.