

Nonlinear System Control Using Adaptive Neural Fuzzy Networks Based on a Modified Differential Evolution

Cheng-Hung Chen, *Student Member, IEEE*, Cheng-Jian Lin, *Member, IEEE*, and Chin-Teng Lin, *Fellow, IEEE*

Abstract—This study presents an adaptive neural fuzzy network (ANFN) controller based on a modified differential evolution (MODE) for solving control problems. The proposed ANFN controller adopts a functional link neural network as the consequent part of the fuzzy rules. Thus, the consequent part of the ANFN controller is a nonlinear combination of input variables. The proposed MODE learning algorithm adopts an evolutionary learning method to optimize the controller parameters. For design optimization, a new criterion is introduced. A hardware-in-the-loop control technique is developed and applied to the designed ANFN controller using the MODE learning algorithm. The proposed ANFN controller with the MODE learning algorithm (ANFN-MODE) is used in two practical applications—the planetary-train-type inverted pendulum system and the magnetic levitation system. The experiment is developed in a real-time visual simulation environment. Experimental results of this study have demonstrated the robustness and effectiveness of the proposed ANFN-MODE controller.

Index Terms—Differential evolution (DE), magnetic levitation system, neural fuzzy networks, planetary-train-type inverted pendulum.

I. INTRODUCTION

MANY control techniques used for designing control system are based on measured signals to estimate an unknown plant. However, a plant model of a physical system that is to be controlled is usually very complex and difficult. Adaptive control [1]–[8] is a model-free controller that can be used to control nonlinear systems. Most of the adaptive controllers involve certain types of function approximator from input/output experiments. Generally, the basic objective of adaptive control is to maintain consistent performance of a control system in the presence of the designed parameters. However, traditional adaptive control theory only deals with the systems by known

dynamic structures, not unknown parameters. Furthermore, traditional adaptive controllers cannot make use of human experience, which is usually expressed in linguistic terms.

Neural fuzzy networks [9]–[16] have become a popular research topic. They bring the low-level learning and computational power of neural networks to fuzzy systems and bring the high-level human-like thinking and reasoning of fuzzy systems to neural networks. In a typical Takagi–Sugeno–Kang (TSK) type neural fuzzy network [13]–[15], which is a linear polynomial of input variables, the model output is approximated locally by the rule hyperplanes. However, the traditional TSK-type neural fuzzy network does not take full advantage of the mapping capabilities that may be offered by the consequent part. Therefore, we adopted an adaptive neural fuzzy network (ANFN) controller, which is based on our previous research [16] and combines a fuzzy system with a functional link neural network (FLNN) [17]. The consequent part of the fuzzy rules that corresponds to an FLNN comprises the functional expansion of input variables.

Training of the parameters is the main problem in designing a neural fuzzy network. Backpropagation (BP) training is commonly adopted to solve this problem. It is a powerful training technique that can be applied to networks with a forward structure. Since the steepest descent approach is used in BP training to minimize the error function, the algorithms may reach the local minima very quickly and never find the global solution. The aforementioned disadvantages lead to suboptimal performance, even with a favorable neural fuzzy network topology. Therefore, technologies that can be used to train the system parameters and find the global solution while optimizing the overall structure are required. Recently, genetic fuzzy systems [18]–[22] have received increasing attention mainly because they combine the approximate reasoning method of fuzzy systems with the learning capabilities of evolutionary algorithms. However, the search is extremely time-consuming, which is one of the basic disadvantages of all genetic algorithms (GAs). Although the convergence in some special cases can be improved by hybridizing GAs with some local search algorithms, it is achieved at the expense of the versatility and simplicity of the algorithm. Similar to GAs, differential evolution (DE) has emerged as a robust numerical optimization algorithm and has been successfully applied to solve various difficult optimization problems [23]–[25]. Basically, DE is fast, easy to use, and not only astonishingly simple, but also performs extremely well in a wide variety of test problems. The basic strategy employs the difference of two randomly selected individuals as the source of random

Manuscript received June 12, 2008; revised November 13, 2008. First published April 14, 2009; current version published June 17, 2009. This work was supported in part by the National Science Council, Taiwan, under Grant NSC 97-2221-E-167-022. This paper was recommended by Editor X. Guan.

C.-H. Chen is with the Department of Electrical and Control Engineering, National Chiao-Tung University, Hsinchu 300, Taiwan (e-mail: chchen.ece93g@nctu.edu.tw).

C.-J. Lin is with the Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taiping City 411 Taiwan (e-mail: cjlin@ncut.edu.tw).

C.-T. Lin is with the Department of Computer Science and the Department of Electrical and Control Engineering, National Chiao-Tung University, Hsinchu 300, Taiwan, and also with the Brain Research Center, University System of Taiwan, Hsinchu 300, Taiwan (e-mail: ctlin@mail.nctu.edu.tw).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSMCC.2009.2016572

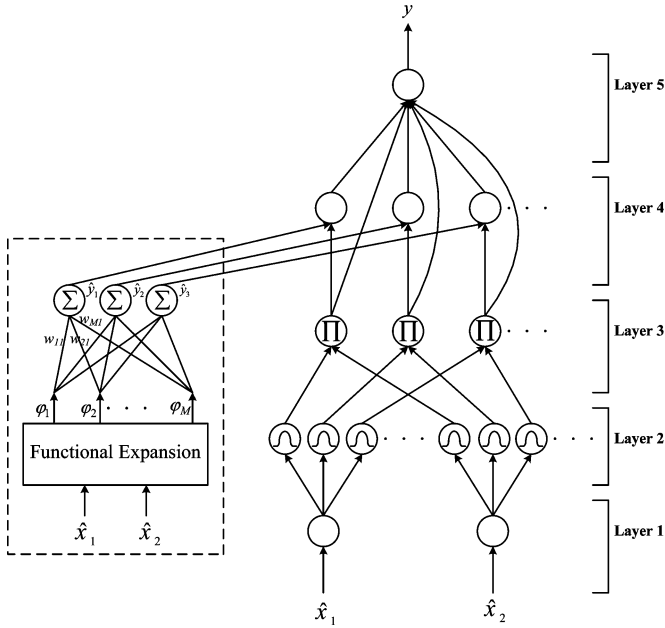


Fig. 1. Structure of the proposed ANFN controller.

variations for a third individual. However, DE usually explores too many search points before locating the global optimum. In addition, although DE is particularly simple to work with, having only a few control parameters, the choice of these parameters is often critical for the performance of DE [26].

This study proposes a modified differential evolution (MODE) for an ANFN (ANFN-MODE). The proposed MODE learning algorithm has two crucial ideas. First, MODE adopts a method to effectively search between the best-so-far individual and randomly chosen individuals, and employs an adaptive parameter to adjust the scaling factor of the traditional DE algorithm. Second, MODE also provides a cluster-based mutation scheme, which maintains useful diversity in the population to increase the search capability. The cluster-based mutation scheme prevents the MODE from being trapped in local optima of the search space. The population is clustered with similar features but that are unable to breed with individuals outside their cluster. Hence, for each cluster, the best individual will be reserved and other individuals will be suitably mutated to the next generation.

The rest of this paper is organized as follows. Section II presents the structure of the ANFN. Next, Section III presents the MODE. The experimental results of two practical applications are described in Section IV. Section V analyzes the stability of ANFN-MODE. Section VI gives conclusions.

II. STRUCTURE OF THE ANFN CONTROLLER

This section describes the ANFN controller [16], which uses a nonlinear combination of input variables (FLNN) [17]. Each fuzzy rule corresponds to an output of FLNN, comprising a functional link. Fig. 1 presents the structure of the proposed ANFN controller.

The ANFN controller realizes a fuzzy if-then rule in the following form:

Rule_j: IF \hat{x}_1 is A_{1j} and \hat{x}_2 is A_{2j} ... and \hat{x}_i is A_{ij} ... and \hat{x}_N is A_{Nj}

$$\begin{aligned} \text{THEN } \hat{y}_j &= \sum_{k=1}^M w_{kj} \phi_k \\ &= w_{1j} \phi_1 + w_{2j} \phi_2 + \dots + w_{Mj} \phi_M \end{aligned} \quad (1)$$

where \hat{x}_i and \hat{y}_j are the input and local output variables, respectively, A_{ij} is the linguistic term of the precondition part with Gaussian membership function, N is the number of input variables, w_{kj} is the link weight of the local output, ϕ_k is the basis trigonometric function of input variables, M is the number of basis function, and *Rule-j* is the j th fuzzy rule.

The operation functions of the nodes in each layer of the ANFN controller are now described. In the following description, $u^{(l)}$ denotes the output of a node in the l th layer.

No computation is performed in layer 1. Each node in this layer only transmits input values to the next layer directly:

$$u_i^{(1)} = x_i. \quad (2)$$

Each fuzzy set A_{ij} is described here by a Gaussian membership function. Therefore, the calculated membership value in layer 2 is

$$u_{ij}^{(2)} = \exp\left(-\frac{[u_i^{(1)} - m_{ij}]^2}{\sigma_{ij}^2}\right) \quad (3)$$

where m_{ij} and σ_{ij} are the mean and variance of the Gaussian membership function, respectively, of the j th term of the i th input variable \hat{x}_i .

Nodes in layer 3 receive 1-D membership degrees of the associated rule from the nodes of a set in layer 2. Here, the product operator described before is adopted to perform the precondition part of the fuzzy rules. As a result, the output function of each inference node is

$$u_j^{(3)} = \prod_i u_{ij}^{(2)} \quad (4)$$

where the $\prod_i u_{ij}^{(2)}$ term of a rule node represents the firing strength of its corresponding rule.

Nodes in layer 4 are called consequent nodes. The input to a node in layer 4 is the output from layer 3, and the other inputs are calculated from an FLNN. For such a node,

$$u_j^{(4)} = u_j^{(3)} \sum_{k=1}^M w_{kj} \phi_k \quad (5)$$

where w_{kj} is the corresponding link weight of FLNN and ϕ_k is the functional expansion of input variables. The functional expansion uses a trigonometric polynomial basis function, given by $[1, \hat{x}_1, \sin(\pi \hat{x}_1), \cos(\pi \hat{x}_1), \hat{x}_2, \sin(\pi \hat{x}_2), \cos(\pi \hat{x}_2), \hat{x}_1 \hat{x}_2]$ for 2-D input variables. Therefore, M is the number of basis functions, $M = 3 \times N + 2$, where N is the number of input variables. Moreover, the output nodes of FLNN depend on the number of fuzzy rules of the ANFN controller.

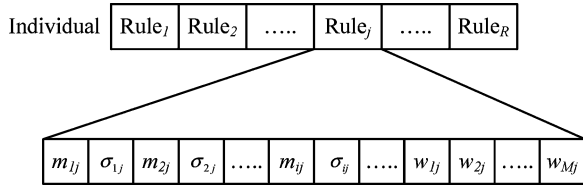


Fig. 2. Coding ANFN controller into an individual in the proposed MODE.

The output node in layer 5 integrates all of the actions recommended by layers 3 and 4 and acts as a defuzzifier with

$$y = u^{(5)} = \frac{\sum_{j=1}^R u_j^{(4)}}{\sum_{j=1}^R u_j^{(3)}} = \frac{\sum_{j=1}^R u_j^{(3)} \left(\sum_{k=1}^M w_{kj} \phi_k \right)}{\sum_{j=1}^R u_j^{(3)}} = \frac{\sum_{j=1}^R u_j^{(3)} \hat{y}_j}{\sum_{j=1}^R u_j^{(3)}} \quad (6)$$

where R is the number of fuzzy rules and y is the output of the ANFN controller. As described before, the number of tuning parameters for the ANFN controller is known to be $(2 \times N + (3 \times N + 2) \times P) \times R$, where N , R , and P denote the number of inputs, existing rules, and outputs, respectively.

III. MODE FOR THE ANFN CONTROLLER

This section describes a MODE for the ANFN controller. The MODE learning algorithm consists of four major phases—the initialization phase, the evaluation phase, the reproduction phase, and the cluster-based mutation phase. First, the initialization phase creates an initial population. Second, the evaluation phase evaluates the performance of each individual using an objective function. Third, the reproduction phase generates new individuals and select survivors to advance to the next phase. Fourth, the cluster-based mutation phase ensures diversity and prevents a population from converging to a suboptimal solution. The whole learning process is described step-by-step shortly.

A. Initialization Phase

1) *Coding Step*: The foremost step in MODE is the coding of the neural fuzzy network into an individual. Fig. 2 shows an example of the coding of parameters of the neural fuzzy network into an individual, where i and j represent the i th input variable and the j th rule, respectively. In this study, a Gaussian membership function is used with variables that represent the mean and variance of the membership function. Fig. 2 represents the neural fuzzy network given by (1), where m_{ij} and σ_{ij} are the mean and variance of a Gaussian membership function, respectively, and w_{kj} represents the corresponding link weight of the consequent part that is connected to the j th rule node. In this study, a real number represents the position of each individual.

2) *Create Initial Population*: Before the MODE learning algorithm is applied, every individual $x_{i,g}$ must be created randomly in the range $[0, 1]$, where $i = 1, 2, \dots, PS$ represents the i th individual for each generation g and PS denotes the population size.

TABLE I
PARAMETER SETTINGS BEFORE TRAINING

Parameter	Value
Population Size	50
Maximum Number of Generation	2000
Crossover Rate	0.9
Coding Type	Real Number

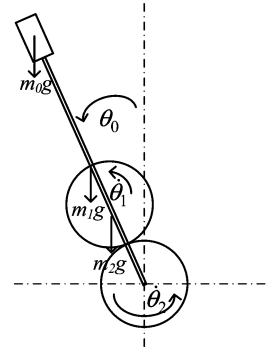


Fig. 3. A physical model geometry of the planetary train-type inverted pendulum.

B. Evaluation Phase

In this study, we adopt a fitness function (i.e., objective function) to evaluate the performance of each individual. The fitness function is defined as follows:

$$\text{fitness} = \frac{1}{1 + \sqrt{(1/N_t) \sum_{k=1}^{N_t} (y_k - \bar{y}_k)^2}} \quad (7)$$

where y_k represents the model output of the k th data, \bar{y}_k the desired output of the k th data, and N_t the number of the training data.

C. Reproduction Phase

1) *Parent Choice*: Each individual in the current generation is allowed to breed through mating with other randomly selected individuals from the population. Specifically, for each individual $x_{i,g}$, $i = 1, 2, \dots, PS$, where g denotes the current generation, three other random individuals, $x_{r_1,g}$, $x_{r_2,g}$, and $x_{r_3,g}$, are selected from the population such that r_1, r_2 , and $r_3 \in \{1, 2, \dots, PS\}$ and $i \neq r_1 \neq r_2 \neq r_3$. This way, a parent pool of four individuals is formed to produce an offspring.

2) *Offspring Generation*: After choosing the parents, MODE applies a differential operation to generate a mutated individual $v_{i,g}$ according to the following equation:

$$v_{i,g} = x_{r_1,g} + (1 - F)(x_{r_2,g} - x_{r_3,g}) + F(x_{\text{best}} - x_{r_1,g}) \quad (8)$$

where F , commonly known as the scaling factor, is defined as g/G to control the rate at which the population evolves, g denotes the current generation, G is the maximum number of generations, and x_{best} is the best-so-far individual (i.e., x_{best} keeps best fitness value up to now in the population). To

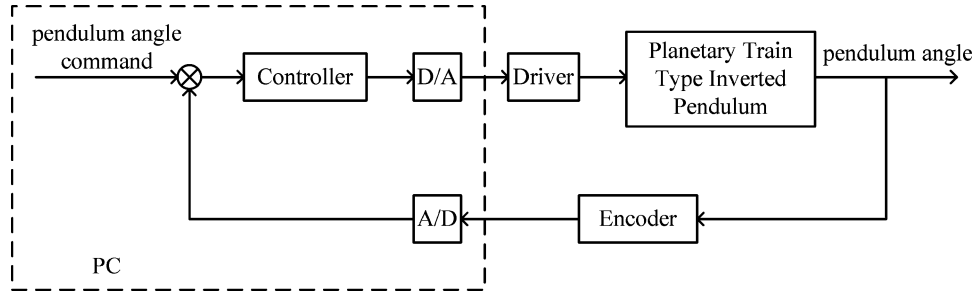


Fig. 4. Control block diagram for the planetary train-type inverted pendulum system.

complement the differential operation search strategy, the MODE uses a crossover operation, often referred to as discrete recombination, in which the mutated individual $v_{i,g}$ is mated with $x_{i,g}$ and generates the offspring $u_{i,g}$. The elements of an individual $u_{i,g}$ are inherited from $x_{i,g}$ and $v_{i,g}$, which are determined by a parameter called crossover probability ($CR \in [0, 1]$), as follows:

$$u_{i,d,g} = \begin{cases} v_{i,d,g}, & \text{if } \text{Rand}(d) \leq CR \\ x_{i,d,g}, & \text{if } \text{Rand}(d) > CR \end{cases} \quad (9)$$

where $d = 1, 2, \dots, D$ denotes the d th element of individual vectors. $\text{Rand}(d) \in [0, 1]$ is the d th evaluation of a random number generator.

3) *Survivor Selection*: MODE applies selection pressure only when selecting survivors. A knockout competition is played between each individual $x_{i,g}$ and its offspring $u_{i,g}$, and the winner is selected deterministically based on objective function values and is then promoted to the next phase. Moreover, the best individual also reserves to the next generation.

D. Cluster-Based Mutation Phase

To prevent the MODE from being trapped in the local optima of the search space (i.e., problems in which there are a number of points that are better than all their neighboring solutions, but do not have as good a fitness as the globally optimal solution), we adopt a cluster-based mutation scheme, which maintains diversity in the population to increase the search capability. We use an easy and fast self-cluster algorithm (SCA) [27] to cluster the population. Each cluster can be viewed as a subspace with similar biological features in the environment that can support different types of life, i.e., these similar individuals of each cluster direct the search toward the same local optima. Then, for each cluster, the best individual will be reserved and other individuals will be suitably mutated to the next generation.

A convex mutation operation is applied in this study. The basic concept of a convex mutation is derived from the convex set theory [28], [29]. Two elements in a single individual are randomly chosen to execute the mutation of a convex combination. The method is designed to enhance fine-tuning capabilities and generate high diversity of individuals to avoid early convergence. For a given $x = (x_1, x_2, \dots, x_{d-1}, x_d, x_{d+1}, \dots, x_D)$, if the element x_d is randomly selected, the resulting offspring is $x = (x_1, x_2, \dots, x'_{d-1}, x_d, x'_{d+1}, \dots, x_D)$. The two new elements x'_{d-1} and x'_{d+1} are

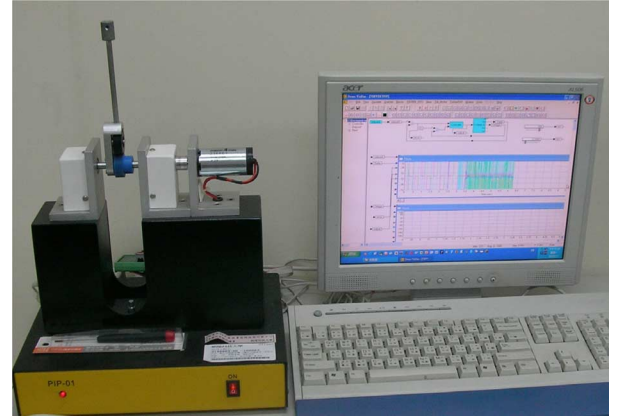


Fig. 5. The experimental planetary train-type inverted pendulum system.

$$x'_{d-1} = (1 - \gamma)x'_{d-1} + \gamma x'_{d+1} \text{ and}$$

$$x'_{d+1} = (1 - \gamma)x'_{d+1} + \gamma x'_{d-1} \quad (10)$$

where γ is a random number in $[0, 1]$.

IV. EXPERIMENTAL RESULTS

This experiment discusses the use of a real-time digital control environment with a hardware-in-the-loop (HIL) control application. We configure a real-time visual simulation (VisSim) environment, including an RT-DAC4/PCI motion control card and HIL systems to demonstrate the performance of the ANFN-MODE controller for practical control applications. VisSim is a Windows-based program for the modeling and simulation of complex nonlinear dynamic systems. VisSim combines an intuitive drag and drop block diagram interface with a powerful simulation engine. We can generate a VisSim diagram using a customizable ANSI C code directly. In this study, we applied the ANFN-MODE controller to the planetary-train-type inverted pendulum system and the magnetic levitation system in the VisSim. The experiment compares the performance with that of the ANFN-MODE controller, the ANFN-DE controller, and the ANFN-GA controller. Table I presents the parameter settings before training, which are used in the three computer simulations for the MODE. In the DE, the population size is set to 50, the maximum number of generation is set to 2000, and the crossover rate is set to 0.9. In the GA, the population size is

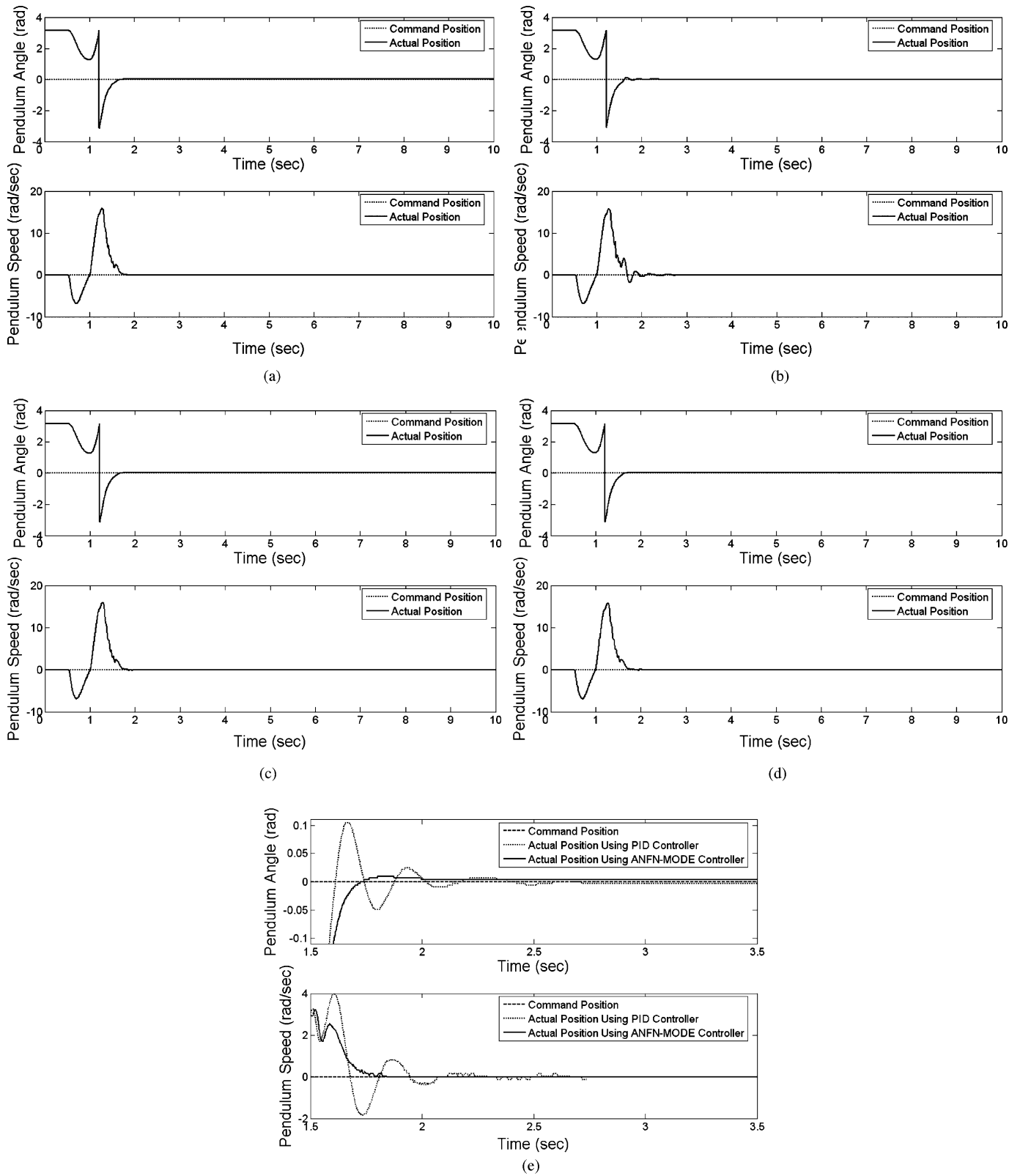


Fig. 6. (a)–(d) Final regulation performance of the ANFN-MODE controller, PID controller, ANFN-DE controller, and ANFN-GA controller. (e) Scaling curves of the ANFN-MODE controller and PID controller between the 1.5th second and the 3.5th second.

set to 50, the maximum number of generation is set to 2000, the crossover rate is set to 0.5, and the mutation rate is set to 0.3.

A. Control of the Planetary-Train-Type Inverted Pendulum System

In order to predict the dynamic behavior of a system from given input command and initial conditions of the system, it is necessary to make a mathematical model of the planetary-train-type inverted pendulum system [30]. The dynamic behavior of the system is helpful in sizing the actuator, choosing the amplifier power, designing the details of the mechanisms, and tuning the controller by computer simulation. To clarify the kinematic and dynamic relations, three major movable parts, the center gear, the planetary gear, and the pendulum, are depicted in Fig. 3.

The kinematic relations of the three components can be allotted to two mutual movements as follows. First, we assume the pendulum to be stationary ($\dot{\theta}_0 = 0$). The ratio of the movements of the planetary gear to the center gear is then

$$e = \frac{\dot{\theta}'_1}{\dot{\theta}_2} = -\frac{N_2}{N_1} = -\frac{r_2}{r_1} \quad (11)$$

where θ_1 and θ_2 are the angles of the center gear and the planetary gear, respectively; N_1 and N_2 are the numbers of the tooth of the center gear and the planetary gear, respectively; r_1 and r_2 are the radii of the center gear and the planetary gear; the dot denotes the time derivative; and $\dot{\theta}'_1$ refers to the angular velocity of the planetary gear in this case. Thus,

$$\dot{\theta}'_1 = e\dot{\theta}_2. \quad (12)$$

Second, we assume the center gear to be stationary ($\dot{\theta}_2 = 0$), and allow the pendulum and planetary gear to turn. The velocity of the planetary gear center can then be expressed as

$$v = (r_1 + r_2)\dot{\theta}_0. \quad (13)$$

This velocity lets the angular velocity of the planetary gear ($\dot{\theta}''_1$) in this case be

$$\dot{\theta}''_1 = \frac{v}{r_1} = \frac{r_1 + r_2}{r_1}\dot{\theta}_0. \quad (14)$$

Combining (13) and (14), we obtain

$$\begin{aligned} \dot{\theta}_1 &= \dot{\theta}'_1 + \dot{\theta}''_1 \\ &= e\dot{\theta}_2 + \frac{r_1 + r_2}{r_1}\dot{\theta}_0. \end{aligned} \quad (15)$$

For the purpose of obtaining the relations between the input motor torque τ_2 , the output responses of the pendulum θ_0 , and the center gear θ_2 , we will use Lagrangian mechanics. Using this method can ensure that we analyze the mechanism in a systematic approach. It starts with the findings of kinetic energy and potential energy of each movable part:

$$\begin{aligned} K_2 &= \frac{1}{2}I_2\dot{\theta}_2^2 \\ P_2 &= 0 \end{aligned}$$

TABLE II
COMPARISON OF PERFORMANCE OF VARIOUS CONTROLLERS TO CONTROL OF THE PLANETARY TRAIN-TYPE INVERTED PENDULUM SYSTEM WITH A 0.1 s SAMPLING RATE

	ANFN-MODE		PID		ANFN-DE		ANFN-GA	
	Controller		Controller		Controller		Controller	
	SAE_{θ_0}	SAE_{θ_1}	SAE_{θ_0}	SAE_{θ_1}	SAE_{θ_0}	SAE_{θ_1}	SAE_{θ_0}	SAE_{θ_1}
Case 1	33.3549	68.5454	34.0881	73.3770	33.4316	69.1548	33.5696	69.7101
Case 2	33.6101	72.3521	34.4442	80.4085	33.7968	72.9001	33.8245	73.3907

$$\begin{aligned} K_1 &= \frac{1}{2}m_1[(r_1 + r_2)\dot{\theta}_0]^2 + \frac{1}{2}I_1\dot{\theta}_1^2 \\ P_1 &= m_1g(r_1 + r_2)\cos\theta_0 \\ K_0 &= \frac{1}{2}\left(\frac{1}{3}m_0l^2\right)\dot{\theta}_0^2 \\ P_0 &= m_0g\frac{1}{2}\cos\theta_0 \end{aligned} \quad (16)$$

where K_2 , K_1 , and K_0 are the kinetic energies of the center gear, the planetary gear, and the pendulum, respectively; P_2 , P_1 , and P_0 are the potential energies of the center gear, the planetary gear, and the pendulum, respectively; I_2 and I_1 are the moments of inertia of the center gear and the planetary gear, respectively; m_1 and m_0 are the masses of the planetary gear and the pendulum, respectively; and l is the length of the pendulum.

Substituting (15) into (16), we obtain Lagrangian as follows:

$$\begin{aligned} L &= \sum K - \sum P \\ &= \left[\frac{1}{2}m_1(r_1 + r_2)^2 + \frac{1}{6}m_0l^2\right]\dot{\theta}_0^2 \\ &\quad + \frac{1}{2}I_1\left(e\dot{\theta}_2 + \frac{r_1 + r_2}{r_1}\dot{\theta}_0\right)^2 + \frac{1}{2}I_2\dot{\theta}_2^2 \\ &\quad - \left[m_1(r_1 + r_2) + \frac{1}{2}m_0\right]g\cos\theta_0. \end{aligned} \quad (17)$$

Because θ_0 and θ_2 are two independent variables, we regard them as generalized coordinates. Using Lagrange's equation, two of the dynamic equations are derived as follows:

$$\begin{aligned} \tau_2 &= \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_2}\right) - \frac{\partial L}{\partial \theta_2} \\ &= (e^2I_1 + I_2)\ddot{\theta}_2 + e\left(\frac{r_1 + r_2}{r_1}\right)I_1\ddot{\theta}_0 \end{aligned} \quad (18)$$

$$\begin{aligned} \tau_0 &= \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_0}\right) - \frac{\partial L}{\partial \theta_0} \\ &= \left[\left(\frac{r_1 + r_2}{r_1}\right)^2 I_1 + m_1(r_1 + r_2)^2 + \frac{1}{3}m_0l^2\right]\ddot{\theta}_0 \\ &\quad + e\left(\frac{r_1 + r_2}{r_1}\right)I_1\ddot{\theta}_2 - \left[m_1(r_1 + r_2) + \frac{1}{2}m_0\right]g\sin\theta_0. \end{aligned} \quad (19)$$

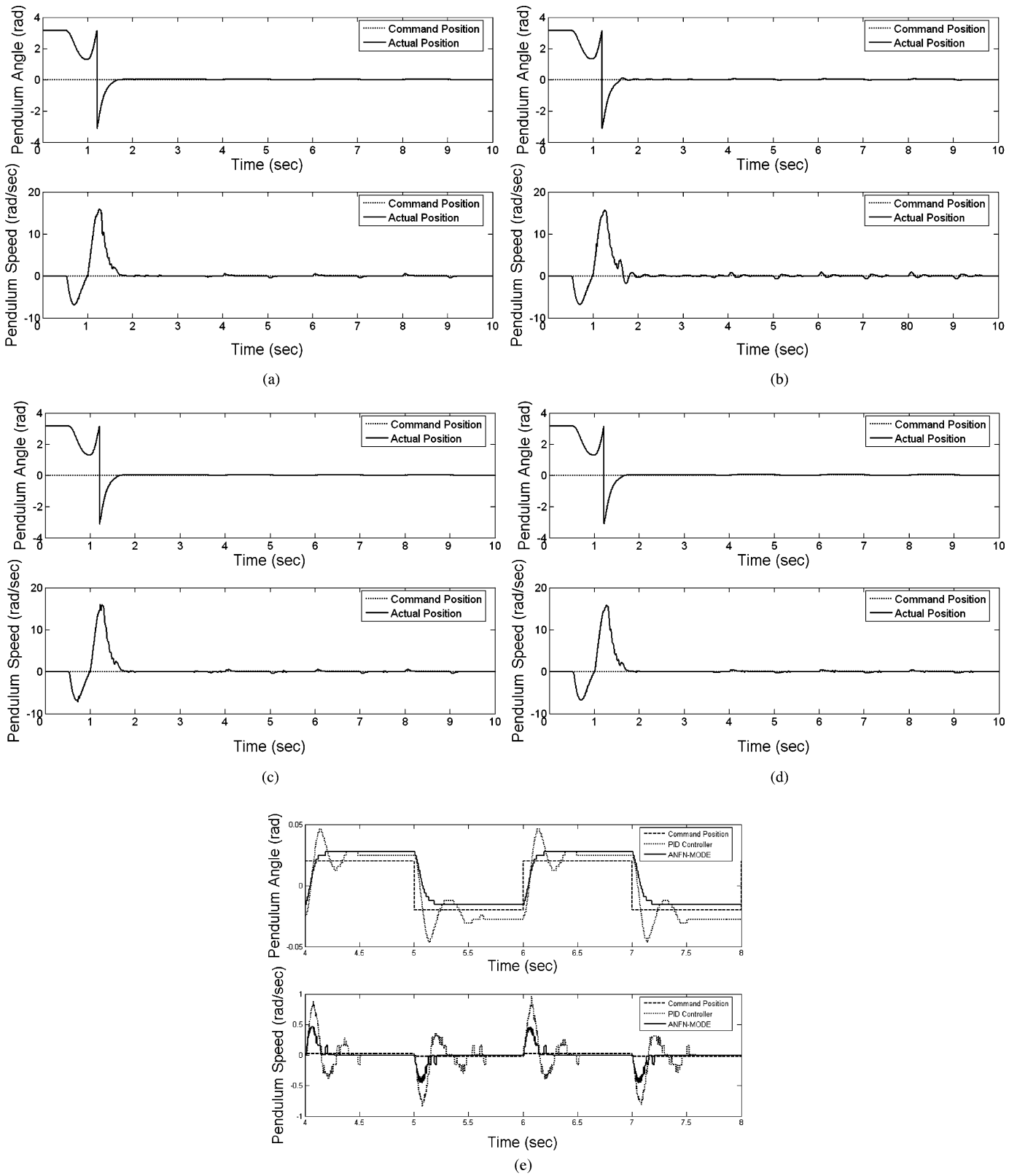


Fig. 7. (a)–(d) Tracking of the ANFN-MODE controller, PID controller, ANFN-DE controller, and ANFN-GA controller, respectively, for a square wave with amplitude ± 0.02 and frequency 0.5 Hz. (e) Tracking curves of the ANFN-MODE controller and PID controller between the fourth second and the eighth second.

In (19), there is no external torque applied in the pendulum, so we assign zero to the variable τ_0 .

In this study, the proposed control structure is shown in Fig. 4. The applied encoder is used to sense the angle of the pendulum and then to translate the signal as a feedback signal. The pendulum angle is controlled by a motor torque until the pendulum is balanceable. To validate the usefulness of the proposed control system under different reference trajectories, two cases, including the set point command (i.e., the stick angle command is equal to zero) and the periodic square command (i.e., the stick angle command is equal to the square wave) are used in this experiment.

This study compares the ANFN-MODE controller to the PID controller, the ANFN-DE controller, and the ANFN-GA controller. Each of these controllers is applied to the planetary-train-type inverted pendulum system. The PID controller is implemented as follows:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de}{dt} \quad (20)$$

where $u(t)$ is the control output and the error $e(t)$ is defined as desired value - measured value of the quantity being controlled. The control gains $K_p = 830$, $K_i = 0$, and $K_d = 0.3284$ are designed with a laboratory manual by our extensive experiments. The training patterns of the ANFN model are generated using the various PID controllers with different control gains.

Fig. 5 shows an experimental planetary-train-type inverted pendulum system test used to validate the experiment results. The performance measures include the set points regulation (case 1) and the square command tracking capability (case 2) of the controllers. In case 1, the proposed system is controlled to follow the set points, which is equal to zero. Fig. 6(a)–(d) presents the regulation performance of the ANFN-MODE controller, the PID controller, the ANFN-DE controller, and the ANFN-GA controller. Fig. 6(e) plots the scaling curves of the ANFN-MODE controller and the PID controller between the 1.5th second and the 3.5th second. To test their regulation performance, two performance indexes, the sum of absolute error (SAE) for the pendulum angle and the pendulum speed, are defined by

$$SAE_{\theta_0} = \sum |\theta_0^{\text{ref}} - \theta_0| \quad \text{and} \quad SAE_{\dot{\theta}_0} = \sum |\dot{\theta}_0^{\text{ref}} - \dot{\theta}_0| \quad (21)$$

where θ_0^{ref} and θ_0 are the referred pendulum angle and the actual pendulum angle, respectively, and $\dot{\theta}_0^{\text{ref}}$ and $\dot{\theta}_0$ are the referred pendulum speed and the actual pendulum speed, respectively. The SAE_{θ_0} and $SAE_{\dot{\theta}_0}$ of the experimental results are presented in Table II.

In case 2, the tracking capability of the proposed system is tested using a square wave with amplitude ± 0.02 and frequency 0.5 Hz. Fig. 7(a)–(d) presents the regulation performance of the ANFN-MODE controller, the PID controller, the ANFN-DE controller, and the ANFN-GA controller. Fig. 7(e) plots the scaling curves of the ANFN-MODE controller and the PID controller between the fourth second and the eighth second. A summary of the experimental results are presented in Table II.

TABLE III
PERFORMANCE COMPARISON OF VARIOUS EXISTING CONTROLLERS

	Our method		Huang <i>et al.</i> [31]		Lin and Chin [12]	
	SAE_{θ_0}	$SAE_{\dot{\theta}_0}$	SAE_{θ_0}	$SAE_{\dot{\theta}_0}$	SAE_{θ_0}	$SAE_{\dot{\theta}_0}$
Case 1	33.3549	68.5454	33.6172	70.2862	33.7316	71.8263
Case 2	33.6101	72.3521	33.9441	74.2963	34.0529	75.0147

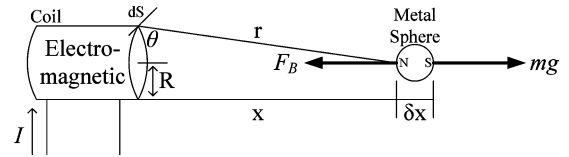


Fig. 8. Sphere and coil arrangement of the magnetic levitation system.

As presented in Table II, the proposed ANFN-MODE controller outperforms the other controllers.

Recently, Huang *et al.* [31] developed an intelligent neural sliding controller for planetary-train-type inverted pendulum system. The control methodology is based on the sliding mode control. The switching function in the normal control law is replaced with a bipolar sigmoid function. A fuzzy neural network is used to identify the pendulum dynamics. Adaptive tuning law is derived. The bipolar sigmoid function is thus adjusted according to the result of the identification process. We redid the same experiment to follow the set points equal to zero. In our experiment, compared with their method [31] that took about 2.5 s to reach and stay on the destination point, our method only took 2 s. To test their regulation performance, the experimental results are shown in Table III. As presented in Table III, the proposed ANFN-MODE controller has a better performance than the other controllers in tracking capability.

B. Control of the Magnetic Levitation System

In order to construct a physical model of the behavior of the magnetic levitation system [32], it is necessary to make some statements about the system and also some simplifying assumptions. The physical model of the sphere and coil of the magnetic levitation system is shown in Fig. 8. The applied control is voltage, which is converted into a current within the mechanical unit. The current passes through an electromagnet that creates the corresponding magnetic field in its vicinity. The sphere is placed along the vertical axis of the electromagnet.

Using the fundamental principle of dynamics, the behavior of the ferromagnetic ball is given by the following electromechanical equation:

$$m \frac{d^2 x}{dt^2} = mg - F_B(x, i) \quad (22)$$

where m is the mass of the levitated ball, g denotes the acceleration due to gravity, x is the distance of the ball from the electromagnet, i is the current across the electromagnet, and $F_B(x, i)$ is the magnetic control force.

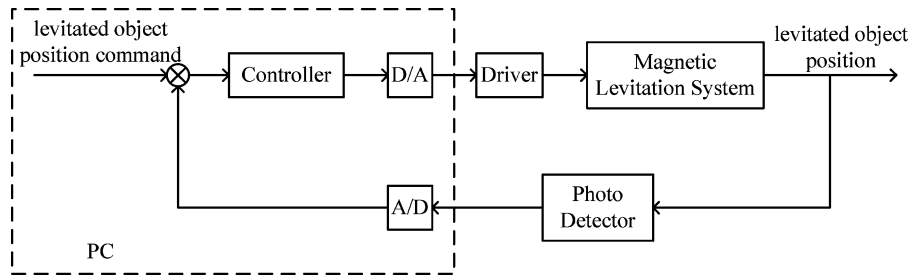


Fig. 9. Control block diagram for the magnetic levitation system.

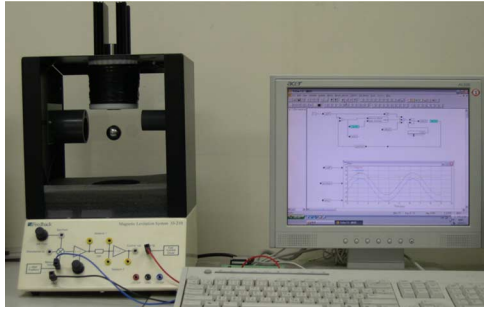


Fig. 10. Experimental magnetic levitation system.

The effect of the magnetic field from the electromagnetic is to introduce a magnetic dipole in the sphere, which itself becomes magnetized. The force acting on the sphere is then composed of gravity and the magnetic force acting on the induced dipole. The magnetic field at a distance of x from the end of the coil may be calculated from the *Biot-Savart law*. This states that the magnetic field produced by a small segment of wire, $d\mathbf{S}$, carrying a current I is given by

$$d\mathbf{B} = \frac{\mu_0}{4\pi} \frac{I d\mathbf{S} \times \mathbf{r}}{r^3} \quad (23)$$

where μ_0 is the permeability of free space and $d\mathbf{S} \times \mathbf{r}$ is the vector product of $d\mathbf{S}$ and \mathbf{r} .

We are interested in the field along the axis of the coil. Consideration of symmetry shows that the magnetic field perpendicular to the axis is zero. To evaluate the integral in (23), we position the current carrying element $d\mathbf{S}$ to lie horizontally on the top of the coil and specify it by its unit vector components; similarly, we specify the vector \mathbf{r} by its unit vector components. Then we have

$$d\mathbf{S} = dS[0 \ 1 \ 0] \quad \text{and} \quad \mathbf{r} = r[\sin \theta \ 0 \ \cos \theta]. \quad (24)$$

In (23), the vector product of $d\mathbf{S}$ and \mathbf{r} from (24) is given by

$$d\mathbf{S} \times \mathbf{r} = dSr[\cos \theta \ 0 \ \sin \theta]. \quad (25)$$

Now, from considerations of symmetry, the field component perpendicular to the coil axis must be zero on the axis. This is the $\sin \theta$ component in (25). Also

$$\text{the radius of the coil } R = r \cos \theta. \quad (26)$$

Hence, from (23), (25), and (26), the magnetic field component dB_x along the axis is given by

$$dB_x = \frac{\mu_0}{4\pi} \frac{I dS R}{r^3} \quad \text{and} \quad r^3 = (R^2 + x^2)^{3/2}. \quad (27)$$

Hence, integrating (27) round a single loop gives

$$\begin{aligned} B_x &= \frac{\mu_0}{4\pi} \frac{I 2\pi R^2}{(R^2 + x^2)^{3/2}} \\ &= \frac{\mu_0}{2} \frac{IR^2}{(R^2 + x^2)^{3/2}}. \end{aligned} \quad (28)$$

To evaluate the field due to the many turns along the axis of the coil, let n be the number of turns per meter and let L (in meters) be its length. Now, we sum all the contributions from all the turns of the coils, as shown in Fig. 8. When (28) is used, the total axial field from all the turns of the coil becomes

$$B_{\text{Total}} = \frac{\mu_0 n I R^2}{2} \int_X^{X+L} \frac{dx}{(R^2 + x^2)^{3/2}}. \quad (29)$$

Integrating (29) by parts gives

$$B_{\text{Total}} = \frac{\mu_0 n I R^2}{2} \left[\frac{X+L}{R^2(R^2 + (X+L)^2)^{1/2}} - \frac{X}{R^2(R^2 + X^2)^{1/2}} \right]. \quad (30)$$

We can rewrite (30) as

$$B(x) = K_1 I G(X). \quad (31)$$

The force on the ball due to the field is proportional to the induced dipole strength and the field strength. The induced dipole strength is itself proportional to the field strength, and hence, the upward force on the ball due to the field B is given by

$$F_B = K_1 K_2 I^2 (G(X) - G(X + \delta X)). \quad (32)$$

Therefore,

$$F_B \approx -K_3 I \delta X G'(X) \quad (33)$$

where $G'(X)$ denotes the derivation and δX is the dipole separation. On the assumption that the poles are located at the center of the mass of each hemisphere of the ball, δX is small compared to L and R and may be taken as a constant. Therefore, (33) becomes

$$F_B \approx K I^2 G'(X). \quad (34)$$

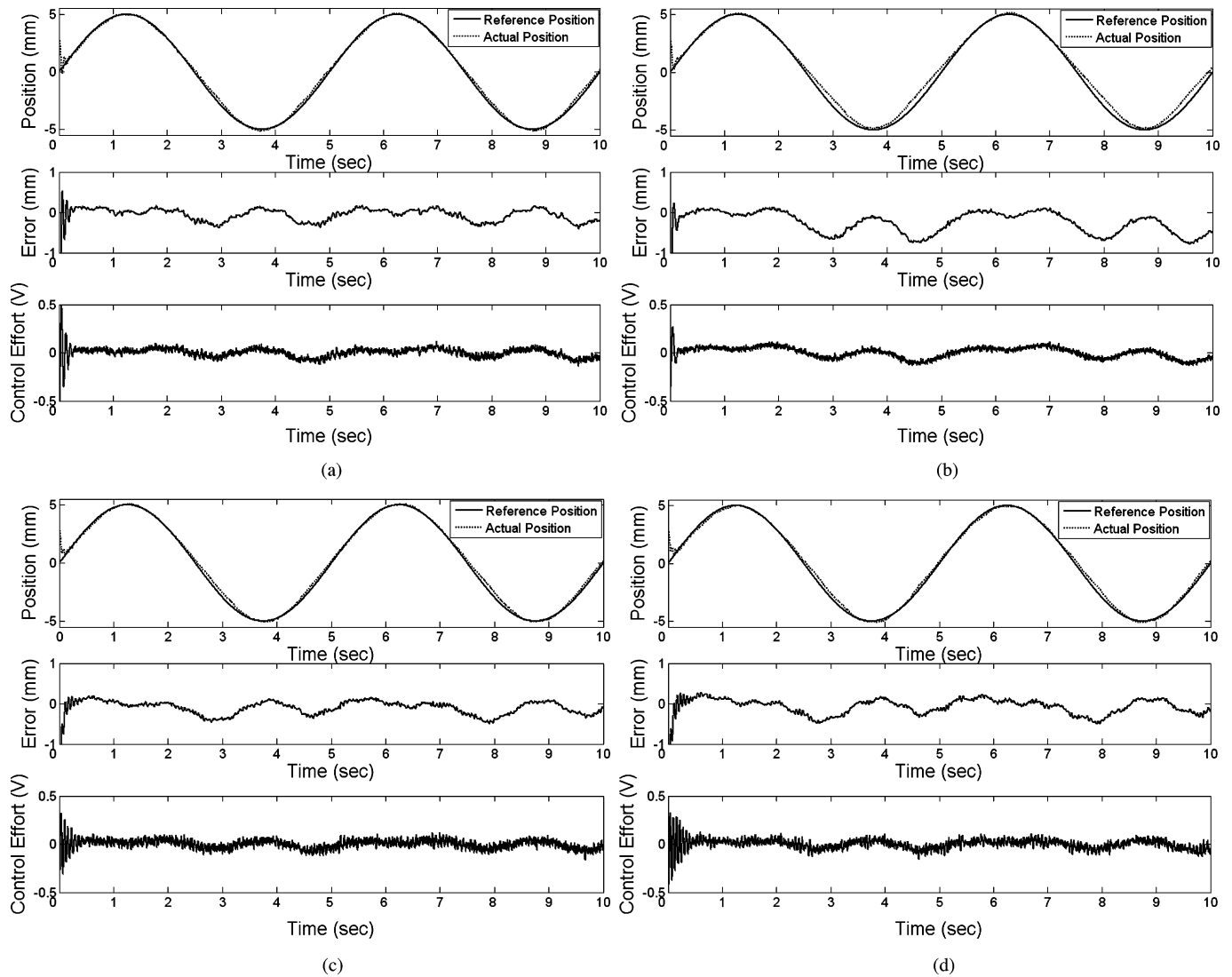


Fig. 11. (a)–(d) Experimental results of the ANFN-MODE controller, PID controller, ANFN-DE controller, and ANFN-GA controller due to the periodic sinusoidal command for the reference and actual positions, tracking error, and control effort.

In this study, the proposed control structure is shown in Fig. 9. The applied photodetector is used to detect the position of the levitated object and then to translate the signal as feedback signal.

In this experiment, the proposed ANFN-MODE controller is compared to the PID controller, the ANFN-DE controller, and the ANFN-GA controller. Each of the controllers is applied to control the magnetic levitation system. As given in the previous section, the PID controller with $K_p = 1.7$, $K_i = 0$, and $K_d = 0.031$ is designed. Fig. 10 shows an experimental magnetic levitation system that is tested to validate the experiment results. In the following four cases, the ANFN-MODE controller is demonstrated to have outperformed the other controllers.

The first case and the second case are used to verify the tracking capability of the controllers. In the first case, the reference signal is given by a sinusoidal wave with amplitude 0.5 and frequency 0.2 Hz, and in the second case, the reference signal is presented by a square wave with amplitude 0.5 and frequency 0.2 Hz. The final experimental results of the ANFN-MODE

controller, the PID controller, the ANFN-DE controller, and the ANFN-GA controller are shown in Figs. 11(a)–(d) and 12(a)–(d). To evaluate their performance, a performance index, the SAE, is defined by

$$SAE_P = \sum |P^{\text{ref}} - P| \quad (35)$$

where P^{ref} and P are the reference trajectory and the actual position of the simulated system, respectively. In the first case, the SAE_P values of the ANFN-MODE controller, the PID controller, the ANFN-DE controller, and the ANFN-GA controller are, respectively, 12.9002, 27.7017, 13.9169, and 15.1572, which are given in the second row of Table III. In the second case, the SAE_P values of the ANFN-MODE controller, the PID controller, the ANFN-DE controller, and the ANFN-GA controller are, respectively, 48.4033, 85.7310, 50.7233, and 53.5194, which are given in the third row of Table IV. The proposed ANFN-MODE controller has a smaller SAE_P value than the other controllers.

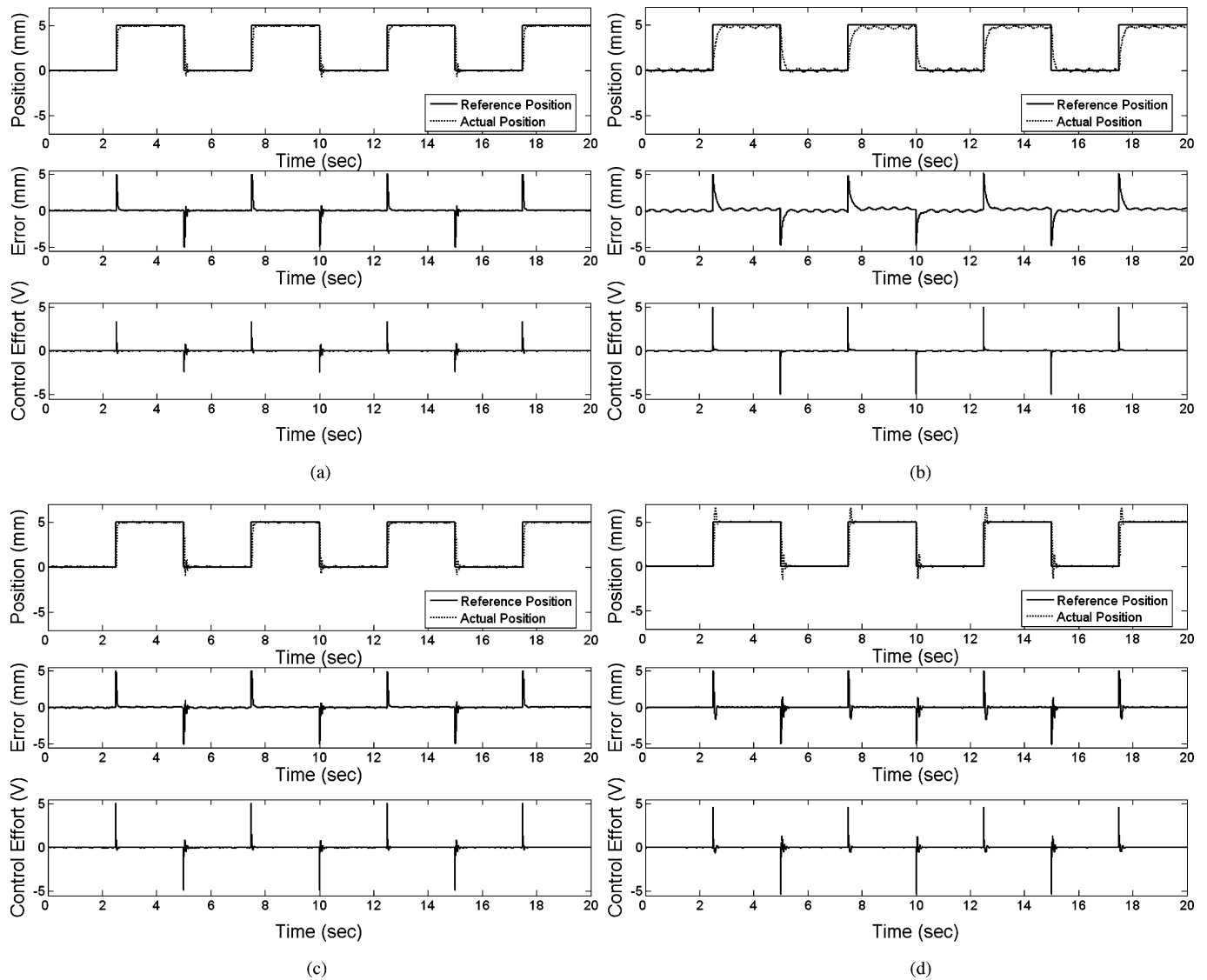


Fig. 12. (a)–(d) Experimental results of the ANFN-MODE controller, PID controller, ANFN-DE controller, and ANFN-GA controller due to the periodic square command for reference and actual positions, tracking error, and control effort.

TABLE IV
COMPARISON OF PERFORMANCE OF VARIOUS CONTROLLERS TO CONTROL OF A MAGNETIC LEVITATION SYSTEM WITH A 0.1 S SAMPLING RATE

$SAE_p = \sum P^{ref} - P $	ANFN-MODE Controller	PID Controller	ANFN-DE Controller	ANFN-GA Controller
Tracking sinusoidal wave	12.9002	27.70017	13.9169	15.1572
Tracking square wave	48.4033	85.7310	50.7233	53.5194
Influence of Impulse Noise	9.2709	12.7345	10.1515	10.8771
Effect of Change in Plant Dynamics	7.9469	24.0004	11.0672	14.0844

The third experiment is performed to demonstrate the noise-rejection ability of the four controllers when some unknown impulse noise is imposed on the process. One impulse noise value, -8 mm, is added to the plant output at the seventh second. A set point of 2.5 mm is adopted in this experimental case. The ANFN-MODE controller can recover from the dis-

turbance quickly after the occurrence of the impulse noise, as shown in Fig. 13(a). Fig. 13(b)–(d) presents the behaviors of the other three controllers under the influence of impulse noise. The SAE_p values of the ANFN-MODE controller, the PID controller, the ANFN-DE controller, and the ANFN-GA controller are, respectively, 9.2709 , 12.7345 , 10.1515 , and 10.8777 , which

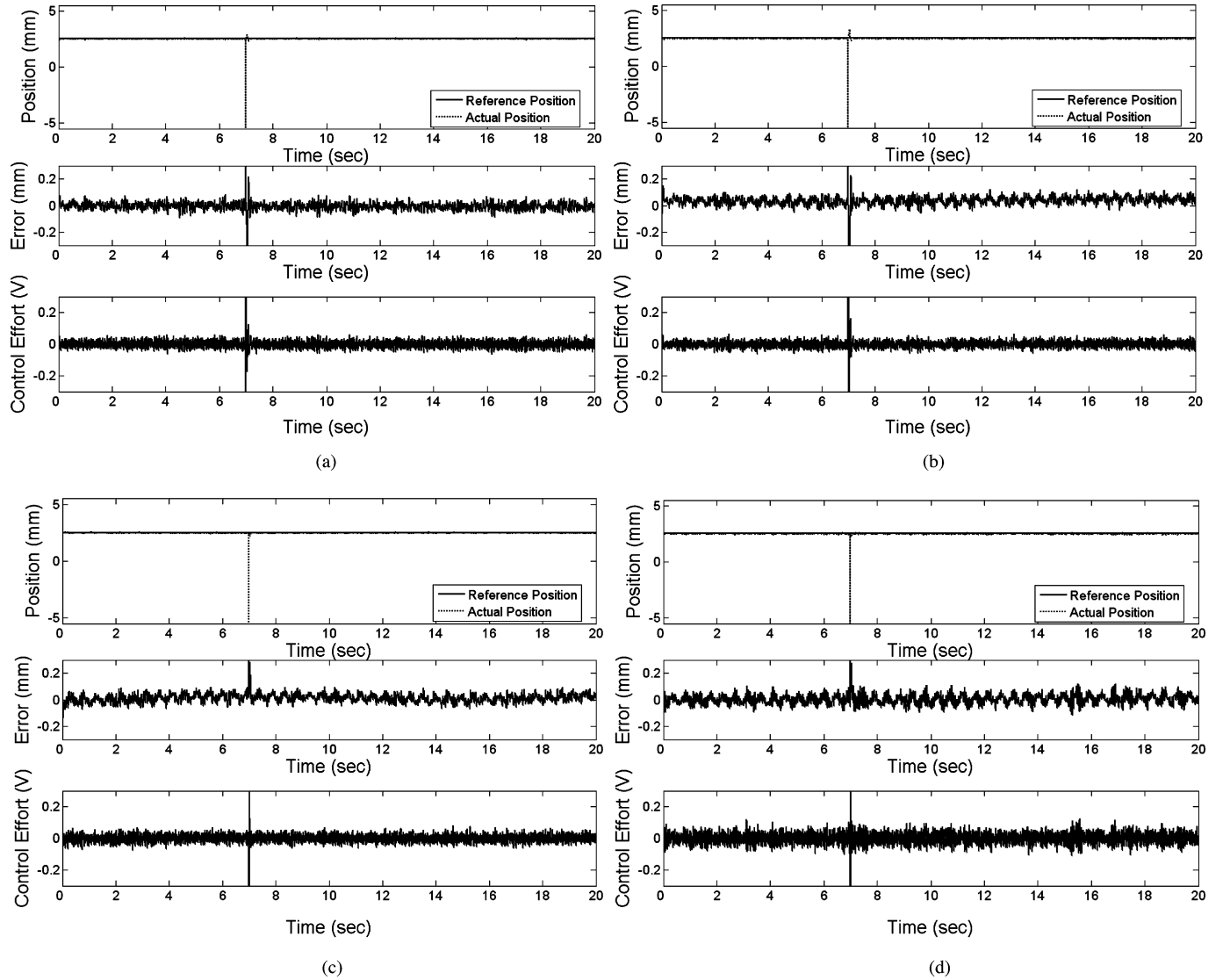


Fig. 13. (a)–(d) Behavior of the ANFN-MODE controller, PID controller, ANFN-DE controller, and ANFN-GA controller under impulse noise in a magnetic levitation system for the reference and actual positions, tracking error, and control effort.

are shown in the fourth row of Table III. The ANFN-MODE controller performs quite well.

One common characteristic of many industrial control processes is that their parameters tend to change in an unpredictable way. The signal $0.6u(t - 0.005)$ is added to the plant input between the 7th second and the 15th second in the fourth experiment to test the robustness of the four controllers. A set point of 2.5 mm is adopted in this fourth experiment. Fig. 14(a)–(d) presents the behaviors of the ANFN-MODE controller, the PID controller, the ANFN-DE controller, and the ANFN-GA controller when the plant dynamics change. The SAE_P values of the ANFN-MODE controller, the PID controller, the ANFN-DE controller, and the ANFN-GA controller are, respectively, 7.6136, 24.0004, 11.0672, and 14.0844, which are shown in the fifth row of Table IV. The results present the favorable control and disturbance rejection capabilities of the trained ANFN-MODE controller in the magnetic levitation system.

V. STABILITY ANALYSIS OF ANFN-MODE

Consider the n th-order nonlinear system (Fig. 15) that can be expressed in the canonical form

$$\begin{aligned} \dot{x}^{(n)} &= f(x, \dot{x}, \dots, x^{(n-1)}) + bu \\ y &= x \end{aligned} \quad (36)$$

where f is an unknown continuous function, b is the control gain (for simplicity, $b = 1$ is taken in the subsequent development), and $u \in \mathfrak{R}$, $y \in \mathfrak{R}$ are input and output of the plant, respectively. We assume that the state vector $\underline{x} = (x_1, x_2, \dots, x_n)^T = (x, \dot{x}, \dots, x^{(n-1)})^T \in \mathfrak{R}^n$ is available for measurement. The control objective is to force the plant state vector y to follow a specified desired trajectory y^d under the constraint that all signals involved must be bounded.

Defining the tracking error vector $e = y - y^d$, the problem is thus to obtain a feedback control law, $u = u_f + u_s$, where

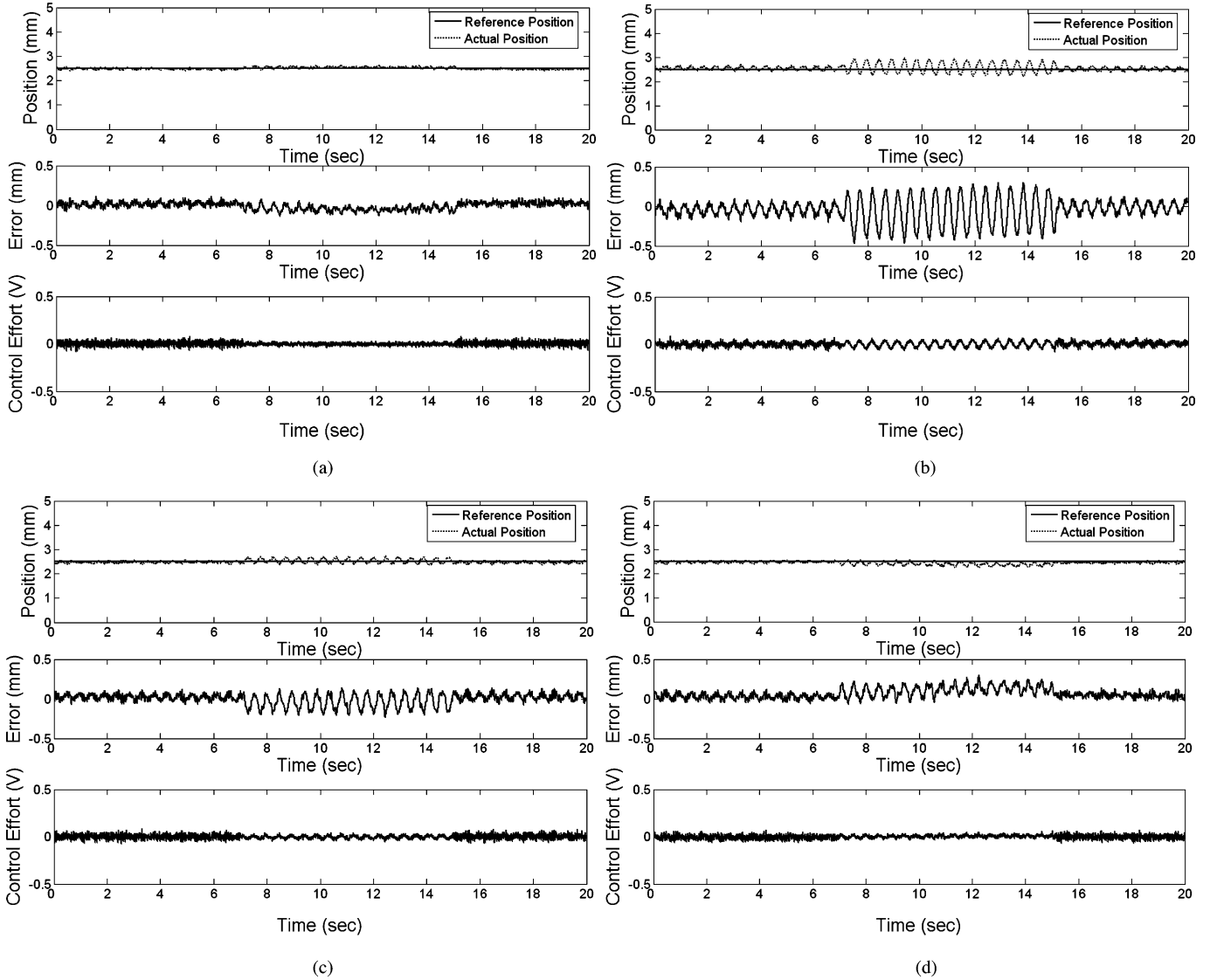


Fig. 14. (a)–(d) Behavior of the ANFN-MODE controller, PID controller, ANFN-DE controller, and ANFN-GA controller when a change occurs in the magnetic levitation system for reference and actual positions, tracking error, and control effort.

u_f and u_s are control output of the ANFN controller and the supervisor, respectively.

The global stability of the control system is a basic requirement for solving nonlinear control problems. Because general evolution algorithm has come characteristic of random search, some search points may cause learning process unstable. In this paper, the supervisor $u_s(t)$ is designed (Fig. 15) to guarantee global stable of the closed-loop system in the sense that the error state variables must be uniformly bounded, i.e., $|e(t)| \leq M < \infty$ for all $t \geq 0$, where M is a design parameter specified by the designer.

Let an error metric be $s(t) = \Lambda^T e(t)$ with $\Lambda^T = [\lambda^{(n-1)}, (n-1)\lambda^{(n-2)}, \dots, 1] \in \mathbb{R}^n$. The question $s(t) = 0$ defines a time-varying hyperplane in \mathbb{R}^n on which the tracking error vector decays exponentially to zero. The time derivative of the error metric can be rewritten as

$$\dot{s}(t) = -y^d(t) + \Lambda_v^T e(t) - f(x(t)) + u(t) \quad (37)$$

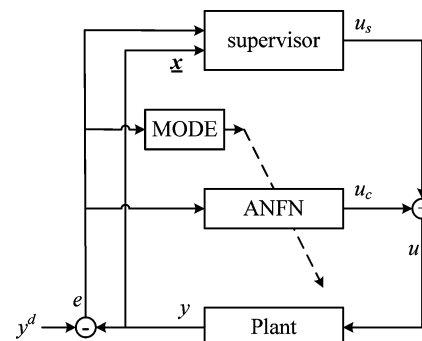


Fig. 15. Overall scheme of stable ANFN-MODE.

where

$$\Lambda_v^T = [0, \lambda^{(n-1)}, (n-1)\lambda^{(n-2)}, \dots, (n-1)\lambda] \in \mathbb{R}^n.$$

Considering $V = (1/2)s^2(t)$ as the Lyapunov function candidate, we have [33]

$$\begin{aligned}\dot{V}(t) &= s(t)\dot{s}(t) \\ &= s(t) [-y^d(t) + \Lambda_v^T e(t) - f(x(t)) + u_f(t) + u_s(t)] \\ &\leq s(t) [-y^d(t) + \Lambda_v^T e(t) - |f(x(t))| + u_f(t) + u_s(t)].\end{aligned}\quad (38)$$

If condition $|f(x(t))| \leq F$ ($F > 0$ is a constant) is assumed, then the supervisory $u_s(t)$ may be constructed as follows:

$$u_s(t) = k_s [-k_d s(t) + y^d(t) - \Lambda_v^T e(t) - \text{sgn}(s(t))F - u_f(t)] \quad (39)$$

where $k_d > 0$ is a constant; $k_s = 1$ if $|e(t)| \geq M$ and $k_s = 0$ if $|e(t)| < M$. Because all terms in (39) can be determined, the supervisory control $u_s(t)$ of (39) can be implemented, substituting (39) into (38) and considering the $k_s = 1$ case, we have

$$\begin{aligned}\dot{V}(t) &\leq s(t) [-k_d s(t) + |f(x(t))| - \text{sgn}(s(t)) \cdot F] \\ &\leq -k_d s^2(t) - |s(t)| [F - |f(x(t))|] \\ &\leq -k_d s^2(t) \leq 0.\end{aligned}\quad (40)$$

Therefore, using the supervisory $u_s(t)$ of (39), we always have $V(t) \rightarrow 0$, i.e., $s(t) \rightarrow 0$, which, in turn, implies $|e(t)| \leq M$.

VI. CONCLUSION

This study proposes an ANFN-MODE for nonlinear system control. The ANFN-MODE controller adopts a nonlinear combination of input variables to the consequent part of fuzzy rules and uses a MODE to optimize the system parameters. We applied the ANFN-MODE controller to the planetary-train-type inverted pendulum system and the magnetic levitation system in the VisSim. The experimental results demonstrate that the ANFN-MODE controller obtains a smaller SAE value than the generally used ANFN-DE, ANFN-GA, and PID controllers for solving nonlinear control problems.

REFERENCES

- [1] K. J. Astrom and B. Wittenmark, *Adaptive Control*. Reading, MA: Addison-Wesley, 1989.
- [2] S. S. Sastry and A. Isidori, "Adaptive control of linearization systems," *IEEE Trans. Autom. Control*, vol. 34, no. 11, pp. 1123–1131, Nov. 1989.
- [3] R. Marino and P. Tomei, "Global adaptive output-feedback control of nonlinear systems, Part I: Linear parameterization," *IEEE Trans. Autom. Control*, vol. 38, no. 1, pp. 17–32, Jan. 1993.
- [4] R. Marino and P. Tomei, "Global adaptive output-feedback control of nonlinear systems, Part II: Nonlinear parameterization," *IEEE Trans. Autom. Control*, vol. 38, no. 1, pp. 33–48, Jan. 1993.
- [5] A. Karniel and G. F. Inbar, "Human motor control: Learning to control a time-varying, nonlinear, many-to-one system," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 30, no. 1, pp. 1–11, Feb. 2000.
- [6] Y. H. Kim and F. L. Lewis, "Optimal design of CMAC neural-network controller for robot manipulators," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 30, no. 1, pp. 22–31, Feb. 2000.
- [7] J. J. Murray, C. J. Cox, G. G. Lendaris, and R. Saeks, "Adaptive dynamic programming," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 32, no. 2, pp. 140–153, May 2002.
- [8] Z. Ye, "Modeling, identification, design, and implementation of nonlinear automotive idle speed control systems—an overview," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 6, pp. 1137–1151, Nov. 2007.
- [9] S. Barada and H. Singh, "Generating optimal adaptive fuzzy-neural models of dynamical systems with applications to control," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 28, no. 3, pp. 371–391, Aug. 1998.
- [10] C. T. Lin and C. S. G. Lee, *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent System*. Englewood Cliffs, NJ: Prentice-Hall, 1996.
- [11] N. Kasabov, *Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering*. Cambridge, MA: MIT Press, 1996.
- [12] C. J. Lin and C. C. Chin, "Prediction and identification using wavelet-based recurrent fuzzy neural networks," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 5, pp. 2144–2154, Oct. 2004.
- [13] J.-S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 3, pp. 665–685, May/Jun. 1993.
- [14] C. F. Juang and C. T. Lin, "An on-line self-constructing neural fuzzy inference network and its applications," *IEEE Trans. Fuzzy Syst.*, vol. 6, no. 1, pp. 12–31, Feb. 1998.
- [15] C. Li and C. Y. Lee, "Self-organizing neuro-fuzzy system for control of unknown plants," *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 1, pp. 135–150, Feb. 2003.
- [16] C. H. Chen, C. J. Lin, and C. T. Lin, "A functional-link-based neuro-fuzzy network for nonlinear system control," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 5, pp. 1362–1378, Oct. 2008.
- [17] J. C. Patra, R. N. Pal, B. N. Chatterji, and G. Panda, "Identification of nonlinear dynamic systems using functional link artificial neural networks," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 29, no. 2, pp. 254–262, Apr. 1999.
- [18] O. Cordon, F. Herrera, F. Hoffmann, and L. Magdalena, *Genetic Fuzzy Systems*. Singapore: World Scientific, 2001.
- [19] F. Herrera, "Genetic fuzzy systems: Status, critical considerations and future directions," *Int. J. Comput. Intell. Res.*, vol. 1, no. 1, pp. 59–67, 2005.
- [20] E. Sanchez, T. Shibata, and L. A. Zadeh, *Genetic Algorithms and Fuzzy Logic Systems: Soft Computing Perspectives*. Singapore: World Scientific, 1997.
- [21] P. P. Angelov, *Evolving Rule-Based Models: A Tool for Design of Flexible Adaptive Systems*. Heidelberg, Germany: Physica-Verlag, 2002.
- [22] O. Cordon, F. Gomide, F. Herrera, F. Hoffmann, and L. Magdalena, "Ten years of genetic fuzzy systems: Current framework and new trends," *Fuzzy Sets Syst.*, vol. 141, no. 1, pp. 5–31, 2004.
- [23] R. Storn and K. V. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Opt.*, vol. 11, no. 4, pp. 341–359, Dec. 1997.
- [24] R. Storn, "System design by constraint adaptation and differential evolution," *IEEE Trans. Evol. Comput.*, vol. 3, no. 1, pp. 22–34, Apr. 1999.
- [25] K. V. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. Berlin, Germany: Springer-Verlag, 2005.
- [26] J. Liu and J. Lampinen, "A fuzzy adaptive differential evolution algorithm," *Soft Comput.—A Fusion Found., Methodol. Appl.*, vol. 9, no. 6, pp. 448–462, 2005.
- [27] C. H. Chen, C. J. Lin, and C. T. Lin, "An efficient quantum neuro-fuzzy classifier based on fuzzy entropy and compensatory operation," *Soft Comput.*, vol. 12, no. 6, pp. 567–583, Apr. 2008.
- [28] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Design*. New York: Wiley, 1997.
- [29] M. Bazaraa, J. Jarvis, and H. Sherali, *Linear Programming and Network Flows*. New York: Wiley, 1990.
- [30] Cho Chieh Tech. Enterprise Ltd. (1999). [Online]. Available: <http://www.chochieh.com.tw/>.
- [31] Y. J. Huang, C. Y. Hsu, T. C. Kuo, and J. Lin, "Intelligent neural sliding control for planetary gear type inverted pendulum mechanism," in *Proc. 22nd IEEE Int. Symp. Intell. Control*, Singapore, Oct. 1–3, 2007, pp. 493–498.
- [32] Feedback Instruments Ltd. (2001). [Online]. Available: <http://www.fb.com/>.
- [33] L. X. Wang, "Stable adaptive fuzzy control of nonlinear systems," *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 2, pp. 146–155, May 1993.



Cheng-Hung Chen (S'07) was born in Kaohsiung, Taiwan, in 1979. He received the B.S. and M.S. degrees in computer science and information engineering from Chaoyang University of Technology, Wufeng, Taiwan, in 2002 and 2004, respectively, and the Ph.D. degree in electrical and control engineering from the National Chiao-Tung University, Hsinchu, Taiwan, in 2008.

He is currently with the Department of Electrical and Control Engineering, National Chiao-Tung University. His current research interests include fuzzy systems, neural networks, evolutionary algorithms, image processing, intelligent control, and pattern recognition.



Cheng-Jian Lin (S'93–M'95) received the B.S. degree in electrical engineering from Ta-Tung University, Taipei, Taiwan, in 1986, and the M.S. and Ph.D. degrees in electrical and control engineering from the National Chiao-Tung University, Hsinchu, Taiwan, in 1991 and 1996.

From April 1996 to July 1999, he was an Associate Professor in the Department of Electronic Engineering, Nan-Kai College, Nantou, Taiwan. From August 1999 to January 2005, he was an Associate Professor in the Department of Computer Science and Information Engineering, Chaoyang University of Technology, where he was the Chairman from 2001 to 2005, a Full Professor from February 2005 to July 2007, and the Library Director of Poding Memorial Library from 2005 to 2007.

From August 2007 to July 2008, he was a Professor in the Department of Electrical Engineering, National University of Kaohsiung. He is currently a Full Professor in the Computer Science and Information Engineering Department, National Chin-Yi University of Technology, Taiping City, Taiwan. From 2002 to 2005, he was an Associate Editor of the *International Journal of Applied Science and Engineering*. His current research interests include soft computing, pattern recognition, intelligent control, image processing, bioinformatics, and field-programmable gate array (FPGA) design. He has authored or coauthored more than 150 papers published in the referred journals and conference proceedings.

Prof. Lin is a member of the Phi Tau Phi. He is also a member of the Chinese Fuzzy Systems Association (CFSA), the Chinese Automation Association, the Taiwanese Association for Artificial Intelligence (TAAI), the IEEE Systems, Man, and Cybernetics Society, and the IEEE Computational Intelligence Society. From 2003 to 2008, he was an executive committee member of the TAAI. From 2007 to 2008, he was an executive committee member of the CFSA. He has received several honors and awards, including the 2006 Outstanding Paper Award of the 11th Conference on Artificial Intelligence and Applications, the 2007 Outstanding Paper Award of the 12th Conference on Artificial Intelligence and Applications, and the 2006 Best Paper Award of the *International Transactions on Computer Science and Engineering* (vol. 32, no. 1).



Chin-Teng Lin (S'88–M'91–SM'99–F'05) received the B.S. degree in control engineering from the National Chiao-Tung University (NCTU), Hsinchu, Taiwan, in 1996, and the M.S.E.E. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, in 1989 and 1992, respectively.

Since August 1992, he has been with the College of Electrical Engineering and Computer Science, NCTU, where he was the Founding Dean from 2005 to 2007, and is currently the Provost of Academic Affairs and the Chair Professor of electrical and control engineering. He is also with the Brain Research Center, University System of Taiwan, Hsinchu. He is the author of *Neural Fuzzy Systems* (Prentice Hall) and *Neural Fuzzy Control Systems With Structure and Parameter Learning* (World Scientific). He has authored or coauthored more than 110 journal papers, including about 80 IEEE transaction papers. His current research interests include intelligent technology, soft computing, brain-computer interface, intelligent transportation systems, robotics and intelligent sensing, and nanobioinformation technologies and cognitive science (NBIC).

Prof. Lin was a member of the Board of Governors (BoG) of the IEEE Systems, Man, Cybernetics Society (SMCS) from 2003 to 2005. He is currently a BoG member of the IEEE Circuits and Systems Society (CASS). He was an IEEE Distinguished Lecturer from 2003 to 2005. He is the Deputy Editor-in-Chief (EIC) of the IEEE TRANSACTIONS OF CIRCUITS AND SYSTEMS, PART II. He was the Program Chair of the 2006 IEEE International Conference on Systems, Man, and Cybernetics, Taipei, Taiwan. He was the President of the Board of Government of Asia Pacific Neural Networks Assembly (APNNA) from 2004 to 2005. Since 1997, he has been receiving the Outstanding Research Award by the National Science Council (NSC), Taiwan, and also received the Outstanding Professor Award by the Chinese Institute of Engineering (CIE) in 2000 and the 2002 Taiwan Outstanding Information-Technology Expert Award. He was also elected to be one of 38th Ten Outstanding Rising Stars in Taiwan in 2000. He is a member of the Tau Beta Pi, the Eta Kappa Nu, and the Phi Kappa Phi honorary societies.