

*Research Article*

# Chaos Synchronization Using Adaptive Dynamic Neural Network Controller with Variable Learning Rates

**Chih-Hong Kao,<sup>1</sup> Chun-Fei Hsu,<sup>2</sup>  
Chih-Hu Wang,<sup>2</sup> and Hon-Son Don<sup>1</sup>**

<sup>1</sup> Department of Electrical Engineering, National Chung-Hsing University, Taichung 402, Taiwan

<sup>2</sup> Department of Electrical Engineering, Chung Hua University, Hsinchu 300, Taiwan

Correspondence should be addressed to Chun-Fei Hsu, fei@chu.edu.tw

Received 22 July 2010; Accepted 20 January 2011

Academic Editor: E. E. N. Macau

Copyright © 2011 Chih-Hong Kao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper addresses the synchronization of chaotic gyros with unknown parameters and external disturbance via an adaptive dynamic neural network control (ADNNC) system. The proposed ADNNC system is composed of a neural controller and a smooth compensator. The neural controller uses a dynamic RBF (DRBF) network to online approximate an ideal controller. The DRBF network can create new hidden neurons online if the input data falls outside the hidden layer and prune the insignificant hidden neurons online if the hidden neuron is inappropriate. The smooth compensator is designed to compensate for the approximation error between the neural controller and the ideal controller. Moreover, the variable learning rates of the parameter adaptation laws are derived based on a discrete-type Lyapunov function to speed up the convergence rate of the tracking error. Finally, the simulation results which verified the chaotic behavior of two nonlinear identical chaotic gyros can be synchronized using the proposed ADNNC scheme.

## 1. Introduction

Radial basis function (RBF) networks are characterized by a simple structure with rapid computation time and superior adaptive performance [1]. There have been considerable interests in exploring the applications of RBF network to deal with the nonlinearity and uncertainty in control systems [2–5]. One main advantage of these RBF-based adaptive neural controllers is that the online parameter adaptive laws were derived without the requirement of offline training. Though the favorable control performance can be achieved in [2–5], the structure

of the used RBF network should be determined by some trial-and-error tuning procedure. It is difficult to consider the balance between the number of hidden neurons and desired performance. To solve this problem, a dynamic RBF (DRBF) network was proposed for the structural adaptation of the RBF network [6–9]. However, some structural learning algorithms are complex and some structural learning algorithms cannot avoid the structure of RBF network growing unboundedly.

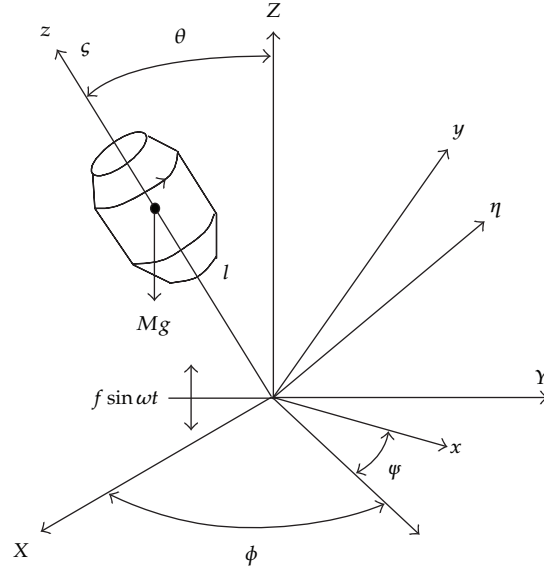
Another drawback of the RBF-based adaptive neural controller is how to determine the learning rates of the parameter adaptive laws. For a small value of the learning rates, the convergence of the tracking error can be easily guaranteed but with slow convergence speed. If the learning rates are large, the parameter adaptive laws may become system unstable. To attack this problem, a variable learning rate was studied in [10–13]. A discrete-type Lyapunov function was utilized to determine the optimal learning rates in [10, 11]; however, the exact calculation of the Jacobian term cannot be determined due to the unknown control dynamics. A genetic algorithm and a particle swarm optimization algorithm were used to determine the optimal learning rates [12, 13]; however, the computation loading is heavy and their scheme lacks the real-time adaptation ability.

In the last decade, control and synchronization of chaotic systems have become an important topic. Chaos synchronization can be applied in the vast areas of physics and engineering systems such as in chemical reactions, power converters, biological systems, information processing, and secure communication [14–16]. Many different methods have been applied to synchronize chaotic systems. Chang and Yan [17] proposed an adaptive robust PID controller using the sliding-mode approach; however, the phenomenon of chattering will appear. An adaptive sliding mode control was proposed to cope with the fully unknown system parameters [18]. To eliminate the chattering, a continuous control law is used; however, the system stability cannot be guaranteed. The adaptive control techniques are applied to chaos synchronization in [19]; however, adaptive control requires the structural knowledge of the chaotic dynamic functions. Yau [20] proposed a nonlinear rule-based controller for chaos synchronization. The fuzzy rules should be preconstructed by a time-consuming trial-and-error tuning procedure to achieve the required performance.

This paper proposes an adaptive dynamic neural network control (ADNNC) system to synchronize two nonlinear identical chaotic gyros. The proposed ADNNC system is composed of a neural controller and a smooth compensator. The neural controller uses a DRBF network to approximate an ideal controller and the smooth compensator is designed to dispel the approximation error introduced by the neural controller. This paper has successfully developed a low-computation loading requirement of the online structural learning algorithm for the DRBF network. To speed up the convergence rate of the tracking errors, an analytical method based on a discrete-type Lyapunov function is proposed to determine the variable learning rates of the parameter adaptive laws. Finally, some simulations are provided to verify the effectiveness of the proposed ADNNC system.

## 2. Problem Formulation

In this paper, a symmetric gyro with linear-plus-cubic damping as shown in Figure 1 [15] is considered. The dynamics of a gyro is a very interesting nonlinear problem in classical



**Figure 1:** A schematic diagram of a symmetric gyroscope.

mechanics. According to the study by Chen [15], the dynamics of the symmetrical gyro with linear-plus-cubic damping of the angle  $\theta$  can be expressed as

$$\ddot{\theta} + \alpha^2 \frac{(1 - \cos \theta)^2}{\sin^3 \theta} - \beta \sin \theta + c_1 \dot{\theta} + c_2 \dot{\theta}^3 = f \sin \omega t \sin \theta, \quad (2.1)$$

where  $\theta$  is the angle;  $f \sin \omega t$  is the parametric excitation;  $c_1 \dot{\theta}$  and  $c_2 \dot{\theta}^3$  are the linear and nonlinear damping, respectively;  $\alpha^2 (1 - \cos \theta)^2 / \sin^3 \theta - \beta \sin \theta$  is a nonlinear resilience force. The open-loop system behavior was simulated with  $\alpha^2 = 100$ ,  $\beta = 1$ ,  $c_1 = 0.5$ ,  $c_2 = 0.05$ , and  $\omega = 2$  for observing the chaotic unpredictable behavior. For the phase trajectory with  $f = 33$ , an uncontrolled chaotic trajectory of period 2 motion can be found, and for the phase trajectory with  $f = 36$ , a quasiperiod motion in the uncontrolled chaotic trajectory happens [15]. The time responses of the uncontrolled chaotic gyro with initial condition  $(1, 1)$  with  $f = 33$  and  $f = 36$  are shown in Figures 2(a) and 2(b), respectively. It is shown that the uncontrolled chaotic gyro has different types of trajectories for different system parameters.

Generally, the two chaotic systems in synchronization are called the drive system and response system, respectively. The interest in chaos synchronization is the problem of how to design a controller to drive the response chaotic gyros system to track the drive chaotic gyros system closely. Consider the following two nonlinear gyros, where the drive system and response system are denoted with  $x$  and  $y$ , respectively. The systems are given as

#### *Drive System*

$$\ddot{x} = f_x \sin \omega t \sin x - \alpha^2 \frac{(1 - \cos x)^2}{\sin^3 x} + \beta \sin x - c_1 \dot{x} - c_2 \dot{x}^3 = g(x, \dot{x}). \quad (2.2)$$

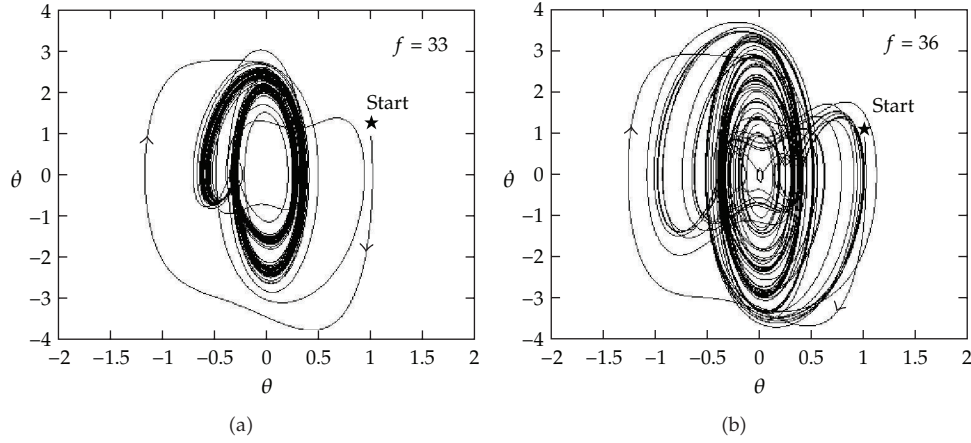


Figure 2: Uncontrolled chaotic trajectory for different system parameters.

### Response System

$$\begin{aligned}\ddot{y} &= f_y \sin \omega t \sin y - \alpha^2 \frac{(1 - \cos y)^2}{\sin^3 y} + \beta \sin y - c_1 \dot{y} - c_2 \dot{y}^3 + u + F(\dot{x}, \dot{y}) \\ &= g(y, \dot{y}) + u + F(\dot{x}, \dot{y}),\end{aligned}\quad (2.3)$$

where  $u$  is the control input and  $F(\dot{x}, \dot{y})$  is the coupling term. To achieve the control objective, the tracking error between the response system (2.3) and the drive system (2.2) is defined as

$$e = x - y. \quad (2.4)$$

The error dynamic equation can be obtained as

$$\ddot{e} = g(x, \dot{x}) - g(y, \dot{y}) - u - F(\dot{x}, \dot{y}). \quad (2.5)$$

If the system dynamics  $g(x, \dot{x})$ ,  $g(y, \dot{y})$ , and  $F(\dot{x}, \dot{y})$  can be obtained, there is an ideal controller as [21]

$$u^* = g(x, \dot{x}) - g(y, \dot{y}) - F(\dot{x}, \dot{y}) + k_1 \dot{e} + k_2 e, \quad (2.6)$$

where  $k_1$  and  $k_2$  are the nonzero constants. Applying the ideal controller (2.6) into error dynamic equation (2.5) obtains

$$\ddot{e} + k_1 \dot{e} + k_2 e = 0. \quad (2.7)$$

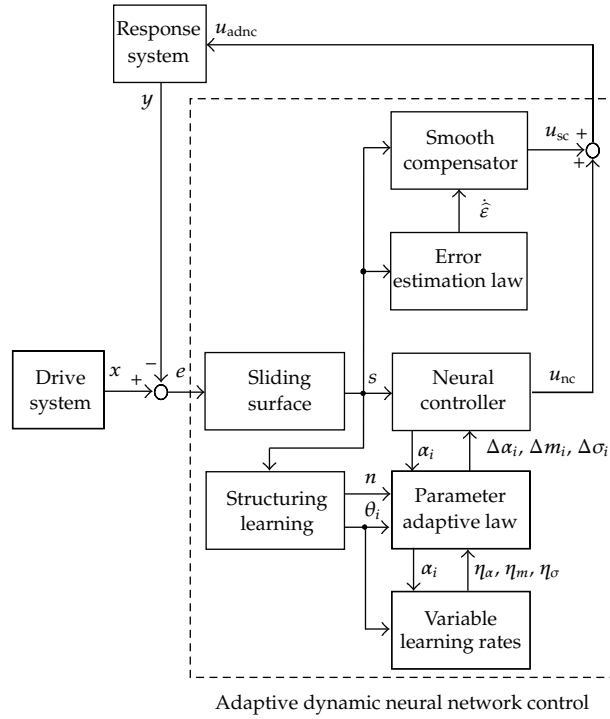


Figure 3: Block diagram of the ADNNC system for the chaos synchronization.

If  $k_1$  and  $k_2$  are chosen to correspond to the coefficients of a Hurwitz polynomial, it implies  $\lim_{t \rightarrow \infty} e(t) = 0$  [21]. The system dynamics of these chaotic systems are always unknown; thus the ideal controller  $u^*$  cannot be implemented.

### 3. Design of the ADNNC System

In this paper, an adaptive dynamic neural network control (ADNNC) system as shown in Figure 3 is introduced where a sliding surface is defined as

$$s = \dot{e} + k_1 e + k_2 \int_0^t e(\tau) d\tau \tag{3.1}$$

with  $k_1$  and  $k_2$  being nonzero positive constants. The proposed ADNNC system is composed of a neural controller and a smooth compensator, that is,

$$u_{adnc} = u_{nc} + u_{sc} \tag{3.2}$$

where the neural controller  $u_{nc}$  uses a DRBF network to mimic the ideal controller and the smooth compensator  $u_{sc}$  is designed to compensate for the differences between the ideal

controller and neural controller. The output of the DRBF network with  $n$  hidden neurons is given as

$$u_{nc} = \sum_{i=1}^n \alpha_i \exp\left(-\frac{(s - m_i)^2}{\sigma_i^2}\right) = \sum_{i=1}^n \alpha_i \theta_i, \quad (3.3)$$

where  $\alpha_i$  represents the  $i$ th connection weights between the hidden layer and output layer and  $\theta_i$ ,  $m_i$ , and  $\sigma_i$  are the firing weight, center, and width of the  $i$ th hidden neuron, respectively.

### 3.1. Structural Learning of DRBF Network

To attack the problem of the structure determination in RBF network, this paper proposed a simple structural learning algorithm. In the growing process, the mathematical description of the existing layers can be expressed as clusters [1, 22]. If a new input data falls within the boundary of the clusters, the DRBF network will not generate a new hidden neuron but update the parameters of the existing hidden neurons. Find the maximum degree  $\theta_{\max}$  defined as [1]

$$\theta_{\max} = \max_{1 \leq i \leq n} \theta_i. \quad (3.4)$$

It can be observed that the maximum degree  $\theta_{\max}$  is smaller as the incoming data is far from the existing hidden neurons. If  $\theta_{\max} \leq \theta_{th}$  is satisfied, where  $\theta_{th} \in (0, 1)$  a pre-given threshold, then a new hidden neuron is generated. The center and width of the new hidden neurons and the output action strength are selected as follows:

$$\begin{aligned} m_i^{\text{new}} &= s, \\ \sigma_i^{\text{new}} &= \bar{\sigma}, \\ \alpha^{\text{new}} &= 0, \end{aligned} \quad (3.5)$$

where  $\bar{\sigma}$  is a prespecified constant. Next, the structural learning phase is considered to determine whether or not to cancel the existing hidden neurons and weights which are inappropriate. A significance index is determined for the importance of the  $i$ th hidden neurons and can be given as follows [22]:

$$I_i(N+1) = \begin{cases} I_i(N) \exp(-\tau) & \text{if } \theta_i < \rho, \\ I_i(N) & \text{if } \theta_i \geq \rho, \end{cases} \quad (3.6)$$

where  $N$  denotes the number of iterations,  $I_i$  is the significance index of the  $i$ th hidden neurons whose initial value is 1,  $\rho$  is the reduction threshold value, and  $\tau$  is the reduction speed constant. If  $I_i \leq I_{th}$  is satisfied, where  $I_{th}$  a pre-given threshold, then the  $i$ th hidden neuron and weight are cancelled. If the computation loading is an important issue for

the practical implementation,  $I_{th}$  and  $\rho$  are chosen as large values so more hidden neurons and weights can be cancelled.

### 3.2. Parameter Learning of DRBF Network

Substituting (3.2) into (2.5) and using (2.6) yield

$$\dot{s} = \ddot{e} + k_1 \dot{e} + k_2 e = u^* - u_{nc} - u_{sc}. \quad (3.7)$$

Multiplying both sides by  $s$  of (3.7) gives

$$s\dot{s} = s(u^* - u_{nc} - u_{sc}). \quad (3.8)$$

According to the gradient descent method, the weights  $\alpha_i$  are updated by the following equation [23]:

$$\Delta\alpha_i = -\eta_\alpha \frac{\partial s\dot{s}}{\partial \alpha_i} = -\eta_\alpha \frac{\partial s\dot{s}}{\partial u_{nc}} \frac{\partial u_{nc}}{\partial \alpha_i} = \eta_\alpha s \theta_i, \quad (3.9)$$

where  $\eta_\alpha$  is the learning rate. Moreover, the center and width of the hidden neurons can be adjusted in the following equation to increase the learning capability:

$$\begin{aligned} \Delta m_i &= -\eta_m \frac{\partial s\dot{s}}{\partial m_i} = -\eta_m \frac{\partial s\dot{s}}{\partial u_{nc}} \frac{\partial u_{nc}}{\partial \theta_i} \frac{\partial \theta_i}{\partial m_i} = -2\eta_m s \alpha_i \frac{(s - m_i)}{\sigma_i^2} \theta_i, \\ \Delta \sigma_i &= -\eta_\sigma \frac{\partial s\dot{s}}{\partial \sigma_i} = -\eta_\sigma \frac{\partial s\dot{s}}{\partial u_{nc}} \frac{\partial u_{nc}}{\partial \theta_i} \frac{\partial \theta_i}{\partial \sigma_i} = 2\eta_\sigma s \alpha_i \frac{(s - m_i)^2}{\sigma_i^3} \theta_i, \end{aligned} \quad (3.10)$$

where  $\eta_m$  and  $\eta_\sigma$  are the learning rates. For given small values of the learning rates, the convergence can be guaranteed but the convergence speed of tracking error is slow. On the other hand, if the selection of learning rates is too large, the algorithm becomes unstable. To determine the learning rates of the parameter adaptive laws, a cost function is defined as

$$C = \frac{1}{2} s^2. \quad (3.11)$$

According to the gradient descent method, the adaptive law of the weight can be represented as

$$\Delta\alpha_i = -\eta_\alpha \frac{\partial C}{\partial \alpha_i} = -\eta_\alpha \frac{\partial C}{\partial u_{nc}} \frac{\partial u_{nc}}{\partial \alpha_i} = -\eta_\alpha \frac{\partial C}{\partial u_{nc}} \theta_i. \quad (3.12)$$

Comparing (3.9) with (3.12) yields the Jacobian term of the system  $\partial C / \partial u_{nc} = -s$ . Then, the convergence analysis in the following theorem derives the variable learning rates to ensure convergence of the output tracking error.

**Theorem 3.1.** Let  $\eta_\alpha$  be the learning rate for the weight of the DRBF network and define  $P_\alpha^{\max}$  as  $P_\alpha^{\max} \equiv \max\|P_\alpha\|$ , where  $P_\alpha = \partial u_{nc}/\partial \alpha_i$  and  $\|\cdot\|$  is the Euclidean norm. Then, the convergence of tracking error is guaranteed if  $\eta_\alpha$  is chosen as

$$0 < \eta_\alpha < \frac{2}{(P_\alpha^{\max})^2}. \quad (3.13)$$

**Theorem 3.2.** Let  $\eta_m$  and  $\eta_\sigma$  be the learning rates of the center and width of the DRBF network, respectively. Define  $P_m^{\max}$  and  $P_\sigma^{\max}$  as  $P_m^{\max} \equiv \max\|P_m\|$  and  $P_\sigma^{\max} \equiv \max\|P_\sigma\|$ , respectively, where  $P_m = \partial u_{nc}/\partial m_i$  and  $P_\sigma = \partial u_{nc}/\partial \sigma_i$ . The convergence of the tracking error is guaranteed if  $\eta_m$  and  $\eta_\sigma$  are chosen as

$$0 < \eta_m, \quad \eta_\sigma < \frac{\sigma_{\min}^2}{2\alpha_{\max}^2}, \quad (3.14)$$

where  $\alpha_{\max} = \max|\alpha_i|$  and  $\sigma_{\min} = \min|\sigma_i|$ .

### 3.3. Stability Analysis

Since the number of hidden neurons in the DRBF network is finite for the real-time practical applications, the approximation error is inevitable. So the ideal controller can be reformulated as

$$u^* = u_{nc}^* + \varepsilon, \quad (3.15)$$

where  $u_{nc}^*$  is the optimal neural controller and  $\varepsilon$  denotes an estimate approximation error between the ideal controller and optimal neural controller. This paper proposed a smooth compensator as

$$u_{sc} = \hat{\varepsilon} + \delta s, \quad (3.16)$$

where  $\hat{\varepsilon}$  denotes the estimated value of the approximation error and  $\delta$  is a small positive constant. Substituting (3.15) and (3.16) into (3.7) yields

$$\dot{s} = \varepsilon - u_{sc} = \varepsilon - \hat{\varepsilon} - \delta s. \quad (3.17)$$

Then, define a Lyapunov function candidate in the following form:

$$V(s, \tilde{\varepsilon}, t) = \frac{1}{2}s^2 + \frac{1}{2\eta_\varepsilon}\tilde{\varepsilon}^2, \quad (3.18)$$



where  $\eta_\varepsilon$  is the learning rate with a positive constant and  $\tilde{\varepsilon} = \varepsilon - \hat{\varepsilon}$ . Differentiating (3.18) with respect to time and using (3.17) obtain

$$\begin{aligned}\dot{V}(s, \tilde{\varepsilon}, t) &= s\dot{s} + \frac{1}{\eta_\varepsilon} \tilde{\varepsilon} \dot{\tilde{\varepsilon}} \\ &= s(\tilde{\varepsilon} - \delta s) + \frac{1}{\eta_\varepsilon} \tilde{\varepsilon} \dot{\tilde{\varepsilon}} \\ &= \tilde{\varepsilon} \left( s + \frac{1}{\eta_\varepsilon} \dot{\tilde{\varepsilon}} \right) - \delta s^2.\end{aligned}\quad (3.19)$$

For achieving  $\dot{V} \leq 0$ , the error estimation law is designed as

$$\dot{\tilde{\varepsilon}} = -\tilde{\varepsilon} = \eta_\varepsilon s; \quad (3.20)$$

then (3.19) can be rewritten as

$$\dot{V}(s, \tilde{\varepsilon}, t) = -\delta s^2 \leq 0. \quad (3.21)$$

Since  $\dot{V}(s, \tilde{\varepsilon}, t)$  is negative semidefinite, that is,  $V(s, \tilde{\varepsilon}, t) \leq V(s, \tilde{\varepsilon}, 0)$ , it implies that  $s$  and  $\tilde{\varepsilon}$  are bounded. Let function  $\Omega(\tau) \equiv \delta s^2 \leq -\dot{V}(s, \tilde{\varepsilon}, t)$  and integrate  $\Omega(t)$  with respect to time; then it is obtained

$$\int_0^t \Omega(\tau) d\tau \leq V(s, \tilde{\varepsilon}, 0) - V(s, \tilde{\varepsilon}, t). \quad (3.22)$$

Because  $V(s, \tilde{\varepsilon}, 0)$  is bounded and  $V(s, \tilde{\varepsilon}, t)$  is nonincreasing and bounded, the following result can be obtained:

$$\lim_{t \rightarrow \infty} \int_0^t \Omega(\tau) d\tau < \infty. \quad (3.23)$$

Moreover, since  $\dot{\Omega}(t)$  is bounded, by Barbalat's Lemma [21],  $\lim_{t \rightarrow \infty} \Omega(t) = 0$ . That is,  $s \rightarrow 0$  as  $t \rightarrow \infty$ . As a result, the stability of the proposed ADNNC system can be guaranteed. In summary, the design steps of ADNNC are summarized as follows.

*Step 1.* Initialize the predefined parameters of the DRBF network.

*Step 2.* The tracking error  $e$  and the sliding surface  $s$  are given in (2.4) and (3.1), respectively.

*Step 3.* Determine whether to add a new node by the  $\theta_{\max} \leq \theta_{\text{th}}$  condition and determine whether to cancel an existing node by significance index  $I_i$ .

*Step 4.* The control law is designed in (3.2), in which the neural controller and the smooth compensator are given as (3.3) and (3.16), respectively.

*Step 5.* Determine the variable learning rates  $\eta_\alpha^*$ ,  $\eta_m^*$ , and  $\eta_\sigma^*$  in (3.13) and (3.14), respectively.

**Table 1:** Performance measures.

(a)		
Methods	Error	
	Average	Standard deviation
Adaptive sliding mode control [18]	0.1082	0.2526
ADNNC with $\eta_\alpha = \eta_m = \eta_\sigma = 0.01$	0.3021	0.2369
ADNNC with $\eta_\alpha = \eta_m = \eta_\sigma = 0.1$	0.1186	0.1373
ADNNC with $\eta_\alpha = \eta_m = \eta_\sigma = 0.2$	0.0958	0.1218
ADNNC with variable learning rates	0.0930	0.1245
(b)		
Methods	Error	
	Average	Standard deviation
Adaptive sliding mode control [18]	0.1170	0.2740
ADNNC with $\eta_\alpha = \eta_m = \eta_\sigma = 0.01$	0.3036	0.2680
ADNNC with $\eta_\alpha = \eta_m = \eta_\sigma = 0.1$	0.1129	0.1068
ADNNC with $\eta_\alpha = \eta_m = \eta_\sigma = 0.2$	0.0663	0.0810
ADNNC with variable learning rates	0.0401	0.0851

*Step 6.* Update the parameters of the neural controller by (3.9), (3.10), and update the parameter of the smooth compensator by (3.20).

*Step 7.* Return to Step 2.

#### 4. Simulation Results

In this section, the proposed ADNNC system is applied to synchronize two identical chaotic gyros. To investigate the effectiveness of the proposed ADNNC system, two simulation cases including parameter variation and initial variation are considered as follows.

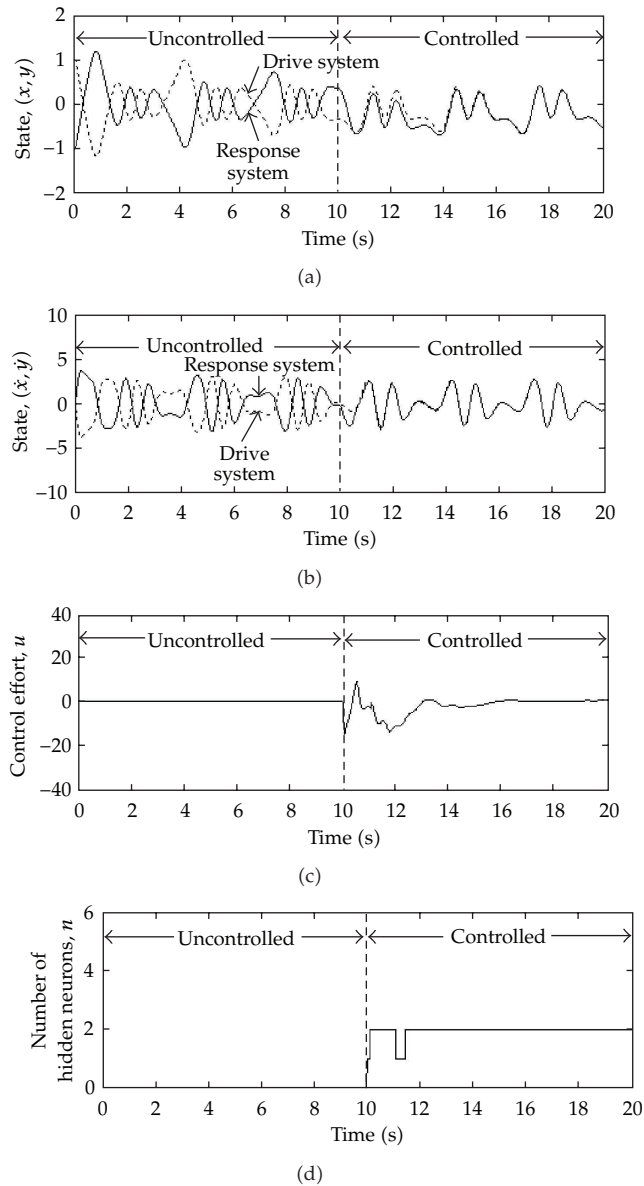
*Case 1.*  $(x, \dot{x}, y, \dot{y}) = (1, 1, -1, -1)$ ,  $f_x = 33$ , and  $f_y = 33$ .

*Case 2.*  $(x, \dot{x}, y, \dot{y}) = (1, 1, 1, 1)$ ,  $f_x = 33$ , and  $f_y = 36$ .

According to Theorems 3.1 and 3.2, respectively, the variable learning rates are chosen as

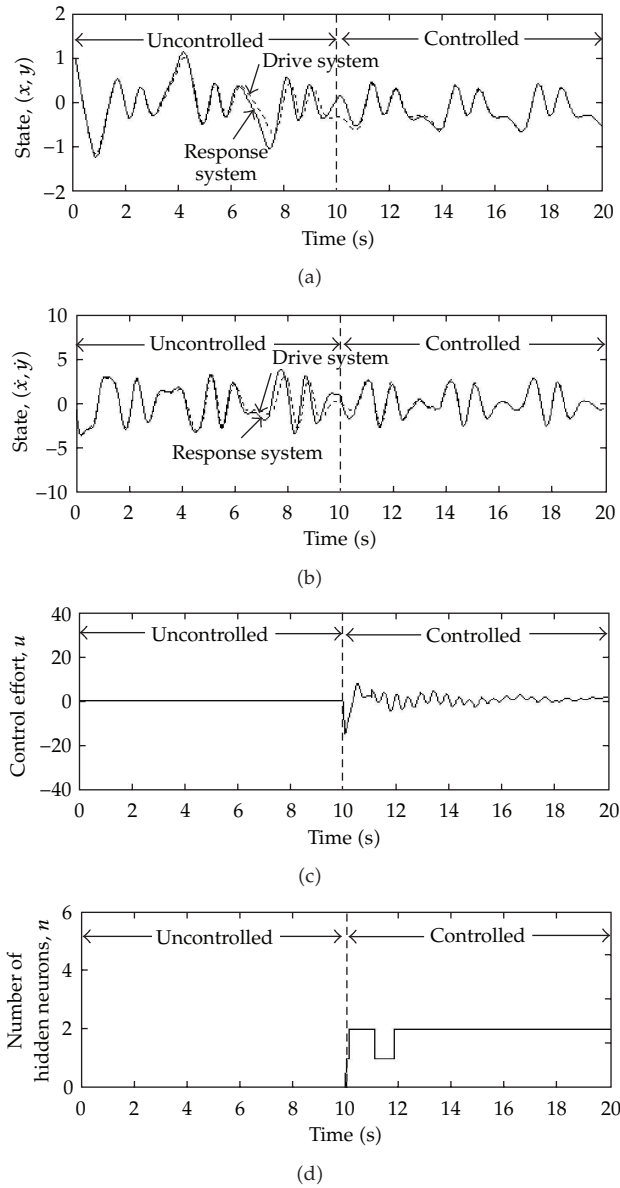
$$\begin{aligned} \eta_\alpha &= \frac{\lambda}{(P_\alpha^{\max})^2}, \\ \eta_m = \eta_\sigma &= \frac{\sigma_{\min}^2}{2\alpha_{\max}^2 + \gamma}, \end{aligned} \tag{4.1}$$

where  $\lambda = 1$  and  $\gamma = 0.001$ . The control parameters are chosen as  $k_1 = 2$ ,  $k_2 = 1$ ,  $\eta_\varepsilon = 0.1$ ,  $\delta = 0.5$ ,  $\bar{\sigma} = 2.5$ ,  $\theta_{\text{th}} = 0.6$ ,  $\tau = 0.01$ ,  $\rho = 0.2$ , and  $I_{\text{th}} = 0.01$ . All the gains are chosen in consideration of the requirement of stability. The simulation results of the proposed ADNNC system with variable learning rates are shown in Figures 4 and 5 for Cases 1 and 2, respectively. The tracking responses of states  $(x, y)$  are shown in Figures 4(a) and 5(a); the tracking responses of states  $(\dot{x}, \dot{y})$  are shown in Figures 4(b) and 5(b); the associated control



**Figure 4:** Simulation results of the ADNNC system with variable learning rates for Case 1.

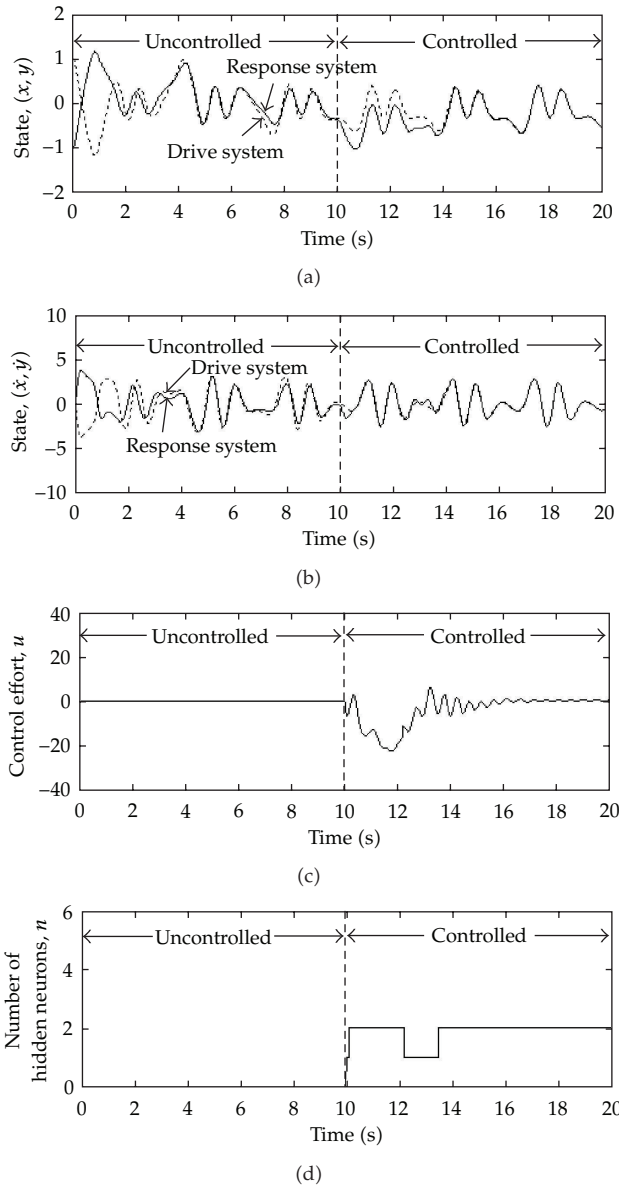
efforts are shown in Figures 4(c) and 5(c); the numbers of hidden neurons are shown in Figures 4(d) and 5(d). The simulation results show that the proposed ADNNC system with variable learning rates not only can achieve favorable synchronization performance but also an appropriate network size of the DRBF network can be obtained because the proposed self-structuring mechanism and the online learning algorithms are applied. To demonstrate the robust control performance of the proposed ADNNC system with variable parameter learning rates, a coupling term  $F(\dot{x}, \dot{y}) = 0.2[\exp(\dot{x} - \dot{y}) - 1]$  is examined here. The simulation results of the proposed ADNNC system with a coupling term are shown in Figures 6 and 7



**Figure 5:** Simulation results of the ADNNC system with variable learning rates for Case 2.

for Cases 1 and 2, respectively. The tracking responses of states  $(x, y)$  are shown in Figures 6(a) and 7(a); the tracking responses of states  $(\dot{x}, \dot{y})$  are shown in Figures 6(b) and 7(b); the associated control efforts are shown in Figures 6(c) and 7(c); the numbers of hidden neurons are shown in Figures 6(d) and 7(d). The simulation results show that the proposed ADNNC system with variable learning rates which can achieve favorable synchronization performance under a coupling term is examined.

In addition, since the selection of the learning rates  $(\eta_{\alpha}, \eta_m, \text{ and } \eta_{\sigma})$  for the online training of the DRBF network has a significant effect on the network performance,



**Figure 6:** Simulation results of the proposed ADNNC system for Case 1 with a coupling term.

the performance measures of various learning rates are summarized in Table 1. It shows that the proposed ADNNC system with variable parameter learning rates possesses the most accurate synchronization performance. To verify the effect of the varied learning rates outside the convergence range, the simulation results of the proposed ADNNC system with  $\eta_\alpha = \eta_m = \eta_\sigma = 0.4$  are shown in Figures 8 and 9 for Cases 1 and 2, respectively. The tracking responses of states  $(x, y)$  are shown in Figures 8(a) and 9(a); the tracking responses of states  $(\dot{x}, \dot{y})$  are shown in Figures 8(b) and 9(b); the associated control efforts are shown in Figures 8(c) and 9(c); the numbers of hidden neurons are shown in Figures 8(d) and 9(d).

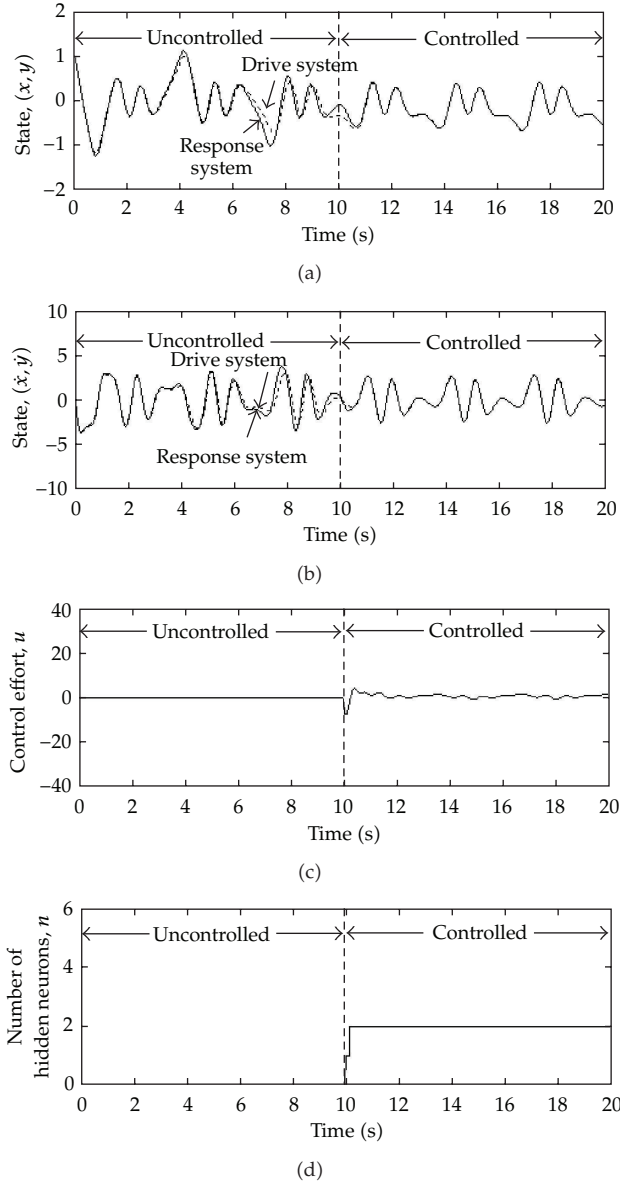
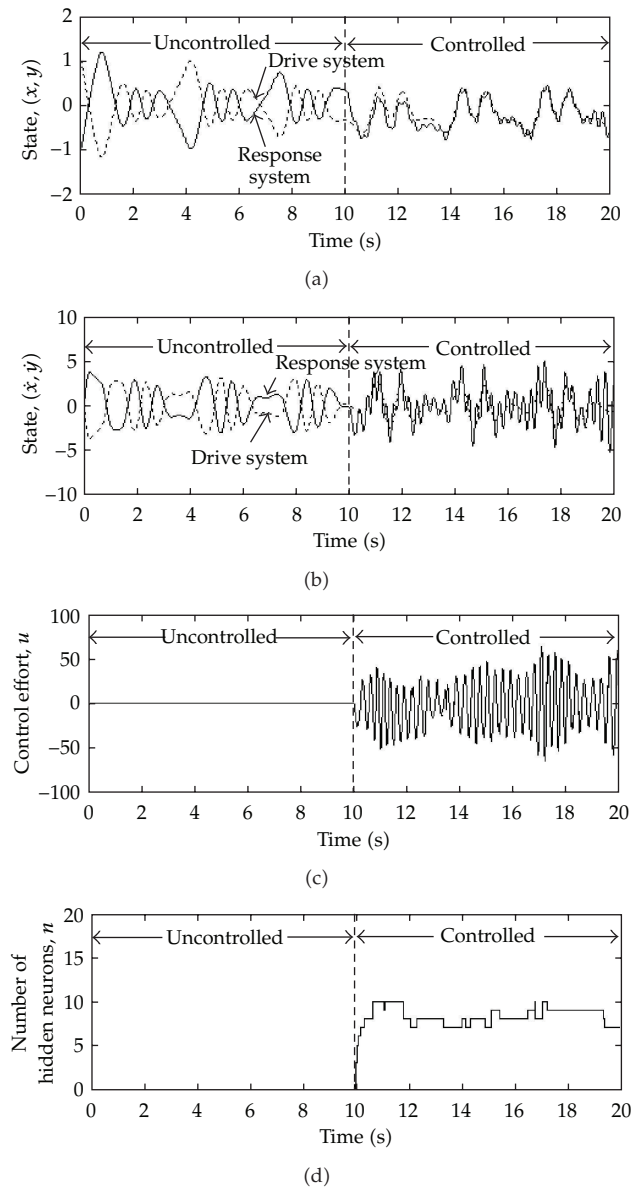


Figure 7: Simulation results of the proposed ADNNC system for Case 2 with a coupling term.

From the simulation results, the unstable tracking responses are induced due to the selection of learning rates outside the convergence region.

## 5. Conclusion

In this paper, an adaptive dynamic neural network control (ADNNC) system is proposed to synchronize chaotic symmetric gyros with linear-plus-cubic damping. The proposed ADNNC system is composed of a neural controller and a smooth compensator. The neural



**Figure 8:** Simulation results of the ADNNC system with large learning rates for Case 1.

controller uses a dynamic radial basis function (DRBF) network to mimic an ideal controller in which the DRBF network can automatically grow and prune the network structure. The smooth compensator is designed to dispel the approximation error between the ideal controller and neural controller. Moreover, to speed up the convergence of tracking error, a discrete-type Lyapunov function is utilized to determine the variable learning rates of the adaptation laws. Numerical simulations have verified the effectiveness of the proposed ADNNC method.

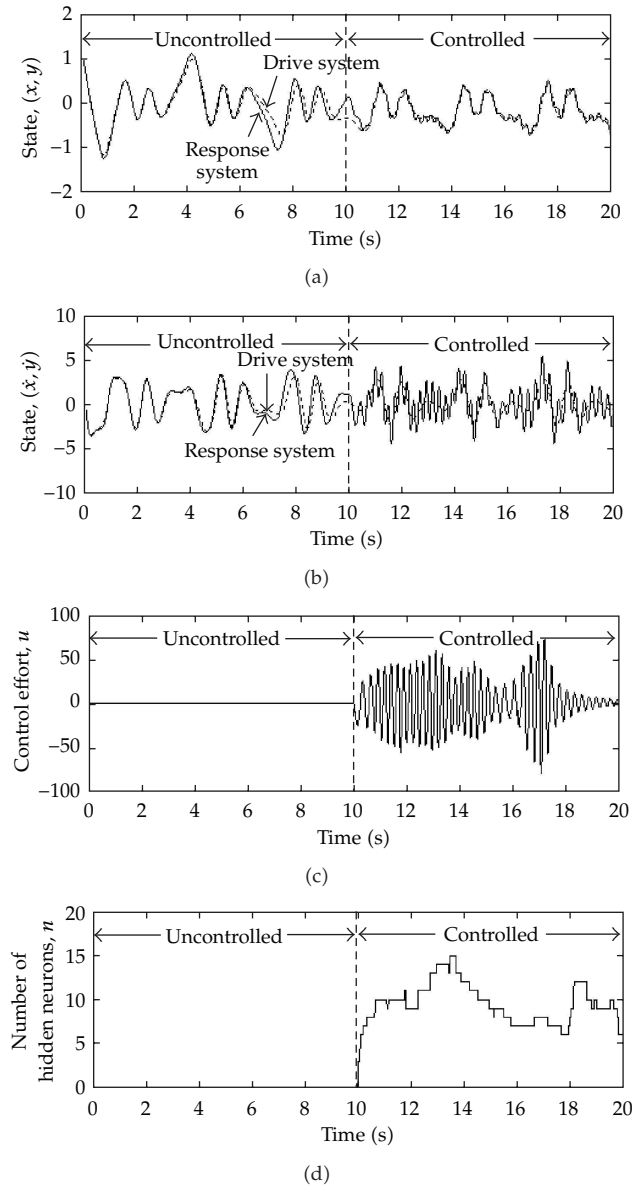


Figure 9: Simulation results of the ADNNC system with large learning rates for Case 2.

## Appendices

### A. Proof of Theorem 3.1

Since

$$P_\alpha = \frac{\partial u_{nc}}{\partial \alpha_i} = \theta_i, \quad (\text{A.1})$$



a discrete-type Lyapunov function is selected as

$$V_A = \frac{1}{2}s^2(N). \quad (\text{A.2})$$

The change in the Lyapunov function is expressed as

$$\Delta V_A = V_A(N+1) - V_A(N) = \frac{1}{2}[s^2(N+1) - s^2(N)]. \quad (\text{A.3})$$

Moreover, the sliding surface difference can be represented by

$$s(N+1) = s(N) + \Delta s(N) = s(N) + \left[ \frac{\partial s(N)}{\partial \alpha_i} \right]^T \Delta \alpha_i, \quad (\text{A.4})$$

where  $\Delta s(N)$  is the sliding surface change and  $\Delta \alpha_i$  represents the change of weights in the DRBF network. Using (3.11), (3.12), and (A.1), then

$$\frac{\partial s}{\partial \alpha_i} = \frac{\partial s}{\partial C} \frac{\partial C}{\partial u_{nc}} \frac{\partial u_{nc}}{\partial \alpha_i} = -P_\alpha. \quad (\text{A.5})$$

Then (A.4) becomes

$$s(N+1) = s(N) - P_\alpha^T \eta_\alpha s(N) P_\alpha. \quad (\text{A.6})$$

Thus,

$$\|s(N+1)\| = \|s(N) - P_\alpha^T \eta_\alpha s(N) P_\alpha\| \leq \|s(N)\| \|1 - P_\alpha^T \eta_\alpha s(N) P_\alpha\|. \quad (\text{A.7})$$

From (A.3) and (A.7),  $\Delta V_A$  can be rewritten as

$$\Delta V_A = \frac{1}{2} \eta_\alpha s^2(N) P_\alpha^T P_\alpha [\eta_\alpha P_\alpha^T P_\alpha - 2] \leq \frac{1}{2} \eta_\alpha s^2(N) (P_\alpha^{\max})^2 [\eta_\alpha (P_\alpha^{\max})^2 - 2]. \quad (\text{A.8})$$

If  $\eta_\alpha$  is chosen as  $0 < \eta_\alpha < 2/(P_\alpha^{\max})^2$ , then the discrete-type Lyapunov stability of  $V_A > 0$  and  $\Delta V_A < 0$  is guaranteed so the output tracking error will converge to zero as  $t \rightarrow \infty$ . This completes the proof of Theorem 3.1.

## B. Proof of Theorem 3.2

To prove Theorem 3.2, the following lemmas were used [9].

**Lemma B.1.** Let  $f(r) = r \exp(-r^2)$ ; then  $|f(r)| < 1$ , for all  $r \in R$ .

**Lemma B.2.** Let  $g(r) = r^2 \exp(-r^2)$ ; then  $|g(r)| < 1$ , for all  $r \in R$ .

(1) According to Lemma B.1,  $|[(s - m_i)/\sigma_i] \exp\{ -[(s - m_i)/\sigma_i]^2 \}| < 1$ , since

$$\begin{aligned} P_m &= \frac{\partial u_{nc}}{\partial m_i} = \frac{\partial u_{nc}}{\partial \theta_i} \frac{\partial \theta_i}{\partial m_i} = \alpha_i \frac{\partial \theta_i}{\partial m_i} \\ &\leq \max \left( \alpha_i \frac{\partial \theta_i}{\partial m_i} \right) \leq \alpha_{\max} \max \left( \frac{\partial \theta_i}{\partial m_i} \right) = \alpha_{\max} \max \left( -2 \frac{(s - m_i)}{\sigma_i^2} \theta_i \right) \\ &\leq 2\alpha_{\max} \frac{1}{\sigma_{\min}} \max \left| \frac{(s - m_i)}{\sigma_i} \theta_i \right| < 2 \frac{\alpha_{\max}}{\sigma_{\min}}. \end{aligned} \quad (\text{B.1})$$

Moreover, the sliding surface difference can be represented by

$$s(N + 1) = s(N) + \Delta s(N) = s(N) + \left[ \frac{\partial s(N)}{\partial m_i} \right]^T \Delta m_i, \quad (\text{B.2})$$

where  $\Delta m_i$  represents a change of the center in the  $i$ th hidden neuron. Using (3.11), (3.12), and (B.1), then

$$\frac{\partial s}{\partial m_i} = \frac{\partial s}{\partial C} \frac{\partial C}{\partial u_{nc}} \frac{\partial u_{nc}}{\partial m_i} = -P_m. \quad (\text{B.3})$$

Then using (B.3) and (B.2),

$$s(N + 1) = s(N) - P_m^T \eta_m s(N) P_m. \quad (\text{B.4})$$

Thus,

$$\|s(N + 1)\| = \|s(N) - P_m^T \eta_m s(N) P_m\| \leq \|s(N)\| \|1 - P_m^T \eta_m s(N) P_m\|. \quad (\text{B.5})$$

If  $0 < \eta_m < 2/(P_m^{\max})^2 = |\sigma_i|_{\min}^2/[2|\alpha_i|_{\max}^2]$ , the term  $\|1 - P_m^T \eta_m s(N) P_m\|$  in (B.5) is less than 1. Therefore, the discrete-type Lyapunov stability of  $V_A > 0$  and  $\Delta V_A < 0$  by (A.2) and (A.3) is guaranteed.

(2) According to Lemma B.2,  $|[(s - m_i)/\sigma_i]^2 \exp\{ -[(s - m_i)/\sigma_i]^2 \}| < 1$ , since

$$\begin{aligned} P_\sigma &= \frac{\partial u_{nc}}{\partial \sigma_i} = \frac{\partial u_{nc}}{\partial \theta_i} \frac{\partial \theta_i}{\partial \sigma_i} = \alpha_i \frac{\partial \theta_i}{\partial \sigma_i} \\ &\leq \max \left( \alpha_i \frac{\partial \theta_i}{\partial \sigma_i} \right) \leq \alpha_{\max} \max \left( \frac{\partial \theta_i}{\partial \sigma_i} \right) = \alpha_{\max} \max \left[ \frac{2}{\sigma_i} \left( \frac{s - m_i}{\sigma_i} \right)^2 \theta_i \right] \\ &\leq 2\alpha_{\max} \frac{1}{\sigma_{\min}} \max \left| \left( \frac{s - m_i}{\sigma_i} \right)^2 \theta_i \right| < 2 \frac{\alpha_{\max}}{\sigma_{\min}}. \end{aligned} \quad (\text{B.6})$$

Moreover, the sliding surface difference can be represented by

$$s(N+1) = s(N) + \Delta s(N) = s(N) + \left[ \frac{\partial s(N)}{\partial \sigma_i} \right]^T \Delta \sigma_i, \quad (\text{B.7})$$

where  $\Delta \sigma_i$  represents a change of the width in the  $i$ th hidden neuron. Using (3.11), (3.12), and (B.6), then

$$\frac{\partial s}{\partial \sigma_i} = \frac{\partial s}{\partial C} \frac{\partial C}{\partial u_{nc}} \frac{\partial u_{nc}}{\partial \sigma_i} = -P_\sigma. \quad (\text{B.8})$$

Then using (B.8) and (B.7) becomes

$$s(N+1) = s(N) - P_\sigma^T \eta_\sigma s(N) P_\sigma. \quad (\text{B.9})$$

Thus,

$$\|s(N+1)\| = \|s(N) - P_\sigma^T \eta_\sigma s(N) P_\sigma\| \leq \|s(N)\| \|1 - P_\sigma^T \eta_\sigma s(N) P_\sigma\|. \quad (\text{B.10})$$

If  $0 < \eta_\sigma < 2/(P_\sigma^{\max})^2 = |\sigma_i|_{\min}^2/[2|\alpha_i|_{\max}^2]$ , the term  $\|1 - P_\sigma(N)^T \eta_\sigma s(N) P_\sigma(N)\|$  in (B.10) is less than 1. Therefore, the discrete-type Lyapunov stability of  $V_A > 0$  and  $\Delta V_A < 0$  by (A.2) and (A.3) is guaranteed.

## Acknowledgments

The authors appreciate the partial financial support from the National Science Council of Republic of China under Grant NSC 98-2221-E-216-040. The authors are grateful to the reviewers for their valuable comments.

## References

- [1] C. T. Lin and C. S. G. Lee, *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1996.
- [2] Y. Li, S. Qiang, X. Zhuang, and O. Kaynak, "Robust and adaptive backstepping control for nonlinear systems using RBF neural networks," *IEEE Transactions on Neural Networks*, vol. 15, no. 3, pp. 693–701, 2004.
- [3] S. Kumarawadu and T. T. Lee, "Neuroadaptive combined lateral and longitudinal control of highway vehicles using RBF networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 500–512, 2006.
- [4] Y. S. Yang and X. F. Wang, "Adaptive H tracking control for a class of uncertain nonlinear systems using radial-basis-function neural networks," *Neurocomputing*, vol. 70, no. 4–6, pp. 932–941, 2007.
- [5] S. Wang and D. L. Yu, "Adaptive RBF network for parameter estimation and stable air-fuel ratio control," *Neural Networks*, vol. 21, no. 1, pp. 102–112, 2008.
- [6] G. B. Huang, P. Saratchandran, and N. Sundararajan, "A generalized growing and pruning RBF (GGAP-RBF) neural network for function approximation," *IEEE Transactions on Neural Networks*, vol. 16, no. 1, pp. 57–67, 2005.

- [7] J. Lian, Y. Lee, S. D. Sudhoff, and S. H. Zak, "Self-organizing radial basis function network for real-time approximation of continuous-time dynamical systems," *IEEE Transactions on Neural Networks*, vol. 19, no. 3, pp. 460–474, 2008.
- [8] C. F. Hsu, "Adaptive growing-and-pruning neural network control for a linear piezoelectric ceramic motor," *Engineering Applications of Artificial Intelligence*, vol. 21, no. 8, pp. 1153–1163, 2008.
- [9] M. Bortman and M. Aladjem, "A growing and pruning method for radial basis function networks," *IEEE Transactions on Neural Networks*, vol. 20, no. 6, pp. 1039–1045, 2009.
- [10] C. M. Lin and Y. F. Peng, "Adaptive CMAC-based supervisory control for uncertain nonlinear systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 34, no. 2, pp. 1248–1260, 2004.
- [11] C. H. Wang, C. S. Cheng, and T. T. Lee, "Dynamical optimal training for interval type-2 fuzzy neural network (T2FNN)," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 34, no. 3, pp. 1462–1477, 2004.
- [12] F. J. Lin, P. K. Huang, and W. D. Chou, "Recurrent-fuzzy-neural-network-controlled linear induction motor servo drive using genetic algorithms," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 3, pp. 1449–1461, 2007.
- [13] R.-J. Wai, K.-L. Chuang, and J.-D. Lee, "On-line supervisory control design for maglev transportation system via total sliding-mode approach and particle swarm optimization," *IEEE Transactions on Automatic Control*, vol. 55, no. 7, pp. 1544–1559, 2010.
- [14] L. M. Pecora and T. L. Carroll, "Synchronization in chaotic systems," *Physical Review Letters*, vol. 64, no. 8, pp. 821–824, 1990.
- [15] H. K. Chen, "Chaos and chaos synchronization of a symmetric gyro with linear-plus-cubic damping," *Journal of Sound and Vibration*, vol. 255, no. 4, pp. 719–740, 2003.
- [16] H. K. Chen and Z. M. Ge, "Bifurcations and chaos of a two-degree-of-freedom dissipative gyroscope," *Chaos, Solitons & Fractals*, vol. 24, no. 1, pp. 125–136, 2005.
- [17] W. D. Chang and J. J. Yan, "Adaptive robust PID controller design based on a sliding mode for uncertain chaotic systems," *Chaos, Solitons & Fractals*, vol. 26, no. 1, pp. 167–175, 2005.
- [18] J. J. Yan, M. L. Hung, and T. L. Liao, "Adaptive sliding mode control for synchronization of chaotic gyros with fully unknown parameters," *Journal of Sound and Vibration*, vol. 298, no. 1-2, pp. 298–306, 2006.
- [19] J. H. Park, "Synchronization of Genesio chaotic system via backstepping approach," *Chaos, Solitons & Fractals*, vol. 27, no. 5, pp. 1369–1375, 2006.
- [20] H. T. Yau, "Nonlinear rule-based controller for chaos synchronization of two gyros with linear-plus-cubic damping," *Chaos, Solitons & Fractals*, vol. 34, no. 4, pp. 1357–1365, 2007.
- [21] J. J. E. Slotine and W. P. Li, *Applied Nonlinear Control*, Prentice Hall, Englewood Cliffs, NJ, USA, 1991.
- [22] C. F. Hsu, "Self-organizing adaptive fuzzy neural control for a class of nonlinear systems," *IEEE Transactions on Neural Networks*, vol. 18, no. 4, pp. 1232–1241, 2007.
- [23] C. M. Lin and C. F. Hsu, "Supervisory recurrent fuzzy neural network control of wing rock for slender delta wings," *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 5, pp. 733–742, 2004.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

