National Formosa University, Yunlin, Taiwan, Nov. 30, 2012-Dec. 2, 2012

Adaptive Hermite-Polynomial-Based CMAC Neural Control for Chaos Synchronization

Chun-Fei Hsu

Department of Electrical Engineering,
Tamkang University,
New Taipei City, Taiwan, ROC
e-mail: fei@ee.tku.edu.tw

***Abstract*—Gyros are a particularly interesting form of nonlinear systems that have attracted many researchers due to their applications in the navigational, aeronautical and space engineering domains. In this paper, a problem of synchronization between two chaotic gyros based on a mater-slave scheme is studied. An adaptive Hermite-polynomial-based CMAC neural control (AHCNC) system which is composed of a neural controller and a smooth compensator is proposed. The neural controller using a Hermite-polynomial-based CMAC neural network (HCNN) is main controller and the smooth compensator is designed to guarantee system stable in the Lyapunov stability theorem. Finally, the simulation results show that the proposed AHCNC scheme can achieve favorable chaos synchronization after the controller parameters learning.**

Keywords*—Adaptive control, Neural control, CMAC neural network, Hermite polynomial.

I. INTRODUCTION

Though favorable control performance can be achieved using a classical control theory, it requires that a mathematical model be used in designing a controller [1]. Since the system parameters and external load disturbance may be unknown or perturbed, the classical control systems are difficult to implement. To tackle this problem, many intelligent control techniques based on neural networks have become popular topics of research [2-6]. The most important feature of these intelligent control schemes is the self-learning ability that neural networks are used to approximate arbitrary linear or nonlinear mappings through online learning algorithms. By adequately choosing network structures, training methods and sufficient input data, the intelligent control techniques have been developed to compensate for the effects of nonlinearities and system uncertainties.

Cerebellar model articulation controller (CMAC) neural network, pro-posed by Albus, is a kind of supervised neural network inspired by the human cerebellum. Unlike multilayer perceptron networks, CMAC neural network learns input-output mappings based on the premise that similar inputs should produce similar outputs. The basis function of CMAC neural network can be categorized into two main types. One is the constant basis function and the other is the non-constant differentiable basis function such as Gaussian function. Recently, the CMAC neural network have been adopted widely for the control of complex dynamical systems owing to its fast learning property, good generalization capability, and simple computation compared with the neural network [7-10].

It is well-known that any series expansion can be expressed to any desired level of accuracy as long as sufficient number of terms is used. This well-known fact is very much analogous to the approximation capability of a FNN neural network. As a result, some researchers proposed a Hermite-polynomial-based neural network (HNN) which each hidden neuron employs a different orthogonal Hermite polynomial basis function for its activation function [11, 12]. Various test examples shown in [11-13] verify that the HNN with different orthogonal Hermite polynomial basis function as activation functions possesses a more efficient search space and improved performance compared to the ones of the conventional neural networks with identical sigmoid or radial basis functions as activation functions.

Gyros are one of the most attractive subjects of dynamic systems and have attracted intensive study [14]. Synchronization of two chaotic gyros has been widely investigated due to its great utility in areas of secure communications, attitude control of long duration spacecraft and signal processing in optical gyros. Until now, many different methods have been applied theoretically and experimentally to synchronize chaotic gyros [15-17]. Yau [15] proposed a nonlinear rule-based controller for chaos synchronization. However, the fuzzy rules should be pre-constructed by a time-consuming trial-and-error tuning procedure to achieve the required performance. Poursamad and Davaie-Markazi [16] proposed a robust adaptive fuzzy controller with a switching compensator to ensure system stable; however, the switching compensator will cause chattering phenomena. Hsu [17] proposed an adaptive fuzzy wavelet neural synchronization control system. All the control parameters are evolved in the Lyapunov sense to ensure the system stability, but how to build a suitable-sized wavelet neural network structure required a trial-and-error tuning procedure.

In this paper, an adaptive Hermite-polynomial-based CMAC neural control (AHCNC) system which is composed of a neural controller and a smooth compensator is proposed to synchronize two chaotic gyros based on a Hermite-polynomial-based CMAC neural network (HCNN). The HCNN uses different Hermite polynomials for different hidden neurons and a nonlinear combination of input variables for output layer, thus the developed HCNN provides a good discrimination capability. The smooth compensator is

designed to guarantee system stable in the Lyapunov stability theorem. Finally, in the simulation study, it is shown that the proposed AHCNC system can drive the slave gyro system to synchronize the master gyro system.

II. PROBLEM FORMULATION

Gyros as shown in Fig. 1 for sensing angular motion are used in airplane automatic pilots, rocket vehicle launch-guidance, space-vehicle attitude systems, ship gyrocompasses and submarine inertial auto navigators [14]. It has been proved in special situations a gyro may show chaotic dynamics. A symmetric gyro with linear-plus-cubic damping is given as [14]

$$\ddot{\theta} + \alpha^2 \frac{(1-\cos\theta)^2}{\sin^3\theta} - \beta\sin\theta + \gamma_1\dot{\theta} + \gamma_2\dot{\theta}^3 = f\sin\omega t\sin\theta \tag{1}$$

where θ is the angle, $f\sin\omega t$ is the parametric excitation, $\gamma_1\dot{\theta}$ and $\gamma_2\dot{\theta}^3$ are the linear and nonlinear damping, respectively, and $\alpha^2\frac{(1-\cos\theta)^2}{\sin^3\theta} - \beta\sin\theta$ is a nonlinear resilience force. It is known the solutions of (1) may perform complex dynamics. The open-loop system behavior was simulated with $\alpha^2 = 100$, $\beta = 1$, $\gamma_1 = 0.5$, $\gamma_2 = 0.05$ and $\omega = 2$ for observing the chaotic unpredictable behavior. The phase trajectory for $f = 33$ can find the uncontrolled chaotic trajectory is period 2 motion and the phase trajectory for $f = 36$ can find the uncontrolled chaotic trajectory is quasi-period motion [14]. The time responses of the uncontrolled chaotic gyro with initial condition (1,1) with $f = 33$ and $f = 36$ are shown in Figs. 2(a) and 2(b), respectively. It is shown that the uncontrolled chaotic gyro has different trajectories for different system values.

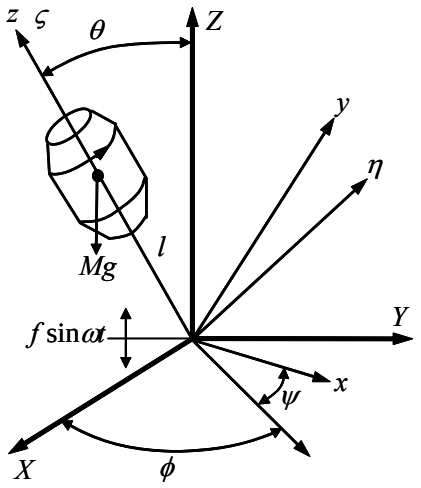

Fig. 1. A schematic diagram of the symmetric gyro system.

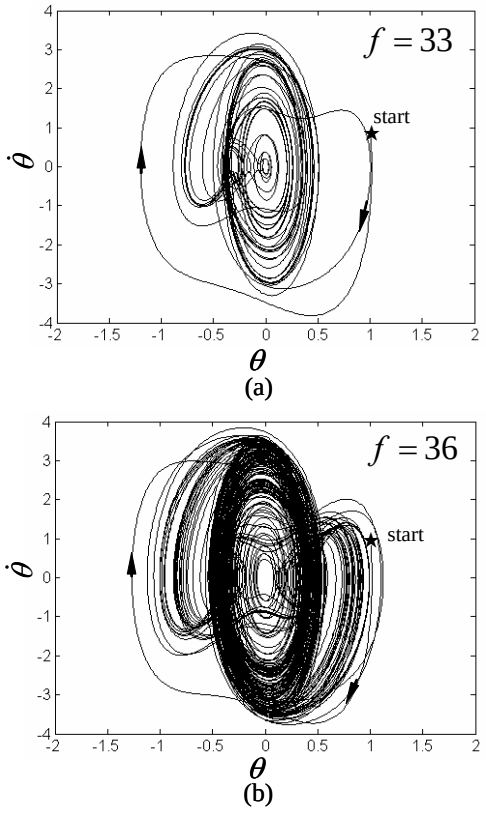

Fig. 2. Uncontrolled chaotic trajectories for different system parameters.

Generally, the two chaotic gyros in synchronization are called the drive (master) system and response (slave) system, respectively. The chaos synchronization problem has the following features: the trajectories of a slave system can track the trajectories of a master system. Consider two coupled chaotic gyros systems as [14]
Master system:

$$\ddot{x} = g_x \tag{2}$$

where

$$g_x = f_x\sin\omega t\sin x - \alpha^2\frac{(1-\cos x)^2}{\sin^3 x} + \beta\sin x - \gamma_1\dot{x} - \gamma_2\dot{x}^3$$

Slave system:

$$\ddot{y} = g_y + u \tag{3}$$

where

$$g_y = f_y\sin\omega t\sin y - \alpha^2\frac{(1-\cos y)^2}{\sin^3 y} + \beta\sin y - \gamma_1\dot{y} - \gamma_2\dot{y}^3$$

and u is the control input. If uncertainties occur, i.e., the parameters of the system deviate from the nominal value or an coupling term is added into the system, the two coupled chaotic gyros can be modified as

$$\ddot{x} = g_x + \Delta g_x \tag{4}$$

and

$$\ddot{y} = g_y + \Delta g_y + u + F \tag{5}$$

where Δg_x and Δg_y denote the system uncertainties of master system and slave system, respectively, and F denotes the coupling term. The interest in chaos synchronization is the problem of how to design a controller to drive a slave chaotic gyro to track a master

chaotic gyro closely. The control objective of the two coupled chaotic gyros are synchronized by designing an appropriate signal control input u even under different initial conditions and a coupling term is added. To achieve the control objective, a tracking error and a sliding surface are defined as

$$e = x - y \tag{6}$$

$$s = \dot{e} + k_1 e + k_2 \int_0^t e(\tau) d\tau \tag{7}$$

From (4)-(7), the error dynamic equation can be obtained as

$$\begin{aligned} \ddot{e} &= g_x - g_y - u + \Delta g_x - \Delta g_y - F \\ &= g_x - g_y - u + w \end{aligned} \tag{8}$$

where the lumped uncertainty w is defined as $w = \Delta g_x - \Delta g_y - F$. If the system dynamics g_x, g_y and w can be obtained, there is an ideal controller as [1]

$$u^* = g_x - g_y + w + k_1 \dot{e} + k_2 e + ks \tag{9}$$

where the k_1, k_2 and k are positive constants. Imposing the control law $u = u^*$ in (8) with (9) yields

$$\dot{s} = \ddot{e} + k_1 \dot{e} + k_2 e = -ks . \tag{10}$$

Define a Lyapunov function candidate in the following form

$$V_1(s,t) = \frac{1}{2} s^2 . \tag{11}$$

Differentiating (11) with respect to time and using (10) obtains

$$\dot{V}_1(s,t) = s\dot{s} = -ks^2 \le 0 . \tag{12}$$

As a result, the stability of the ideal controller is guaranteed. Since the system dynamics of the chaotic gyros are unknown, the ideal controller cannot be implemented.

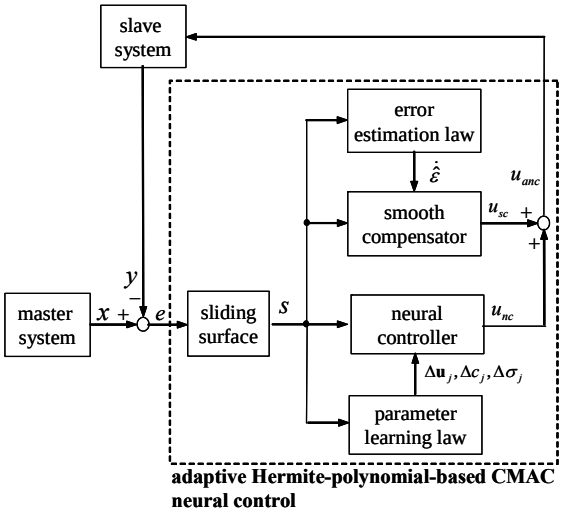

Fig. 3. Block diagram of the AHCNC system for chaos synchronization.

III. AHCNC System Design

This paper proposed an adaptive Hermite-polynomial-based CMAC neural control (AHCNC) system as shown in Fig. 3 which is composed of a neural controller and a smooth compensator. The controller output is defined as

$$u_{anc} = u_{nc} + u_{sc} . \tag{13}$$

The neural controller u_{nc} uses a Hermite-polynomial-based CMAC neural network (HCNN) to approximate an ideal controller and the smooth compensator u_{sc} is designed to cope with the influence of residual approximation error introduced by the neural controller. Substituting (13) into (8) and using (9) yield

$$\dot{s} = u^* - u_{nc} - u_{sc} - ks . \tag{14}$$

A. Description of HCNN

In Fig. 4, the structure of the HCNN is a multi-layer neural network which includes an input layer (Layer 1), a Hermite layer (Layer 2), a association layer (Layer 3), a receptive layer (Layer 4), a TSK layer (Layer 5) and an output layer (Layer 6). The signal propagation and basic function in each space are introduced in the following.

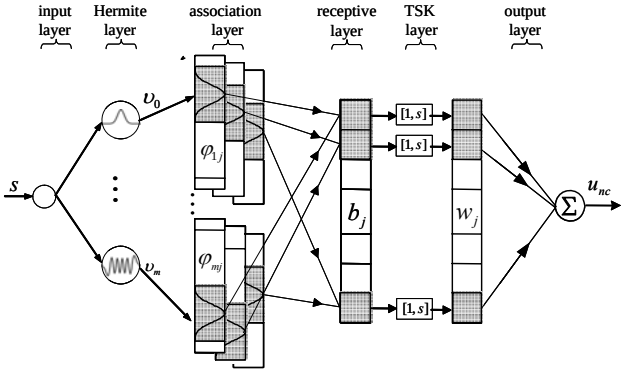

Fig. 4. Structure of HDWNN.

Layer 1 – Input layer: No function is performed in this layer. The node only transmits input values to layer 2.

Layer 2 – Hermite layer: In this layer, the orthogonal Hermite polynomial basis functions represented as [13]

$$\upsilon_i = \frac{1}{\sqrt{2^i i! \sqrt{\pi}}} e^{-s^2/2} H_i(s) , \text{ for } i = 0, 1, 2, ..., m \tag{15}$$

where the orthogonal Hermite polynomials are given recursively by

$$H_0(s) = 1 \tag{16}$$

$$H_1(s) = 2s \tag{17}$$

$$\vdots \tag{18}$$

$$H_i(s) = 2sH_{i-1}(s) - 2(i-1)H_{i-2}(s) , \text{ for } i \ge 2 . \tag{19}$$

In general, a better approximated performance can be obtained if a higher-order orthogonal Hermite polynomial basis functions is used.

Layer 3 – Association layer: In this space, several elements can be accumulated as a block. Each block performs a receptive-field basis function. The Gaussian function is adopted as the receptive-field basis function which can be represented as

$$\varphi_{ij}(\upsilon_i) = \exp\left[\frac{-(\upsilon_i - m_{ij})^2}{\sigma_{ij}^2}\right], \text{ for } j = 1, 2, \cdots n_B \tag{20}$$

where $\varphi_{ij}(\upsilon_i)$ presents the j-th block of the i-th input with the mean m_{ij} and variance σ_{ij} and n_B is the

number of blocks.

Layer 4 – Receptive layer: In this receptive-field space, the multidimensional receptive-field function is defined as

$$b_k = \prod_{i=1}^{m} \varphi_{ij}(\upsilon_i) = \exp\left[\sum_{i=1}^{m} \frac{-(\upsilon_i - m_{ij})^2}{\sigma_{ij}^2}\right], \quad \text{for } k = 1, 2, ..., n \tag{21}$$

where b_k is associated with the k-th receptive-field. The number of receptive-field n is equal to n_B in this paper.

Layer 5 – TSK layer: The TSK layer represents the linear combination function. Each node in this layer is denoted by [18, 19]

$$\mu_k = u_{k0} + u_{k1} s = \mathbf{u}_k^T \mathbf{\Phi} \text{ , for } k = 1, 2, ..., n \tag{22}$$

where u_{k0} and u_{k1} are the parameters designed by the designer, $\mathbf{u}_k = [u_{k0}, u_{k1}]^T$ and $\mathbf{\Phi} = [1, s]^T$.

Layer 6 – Output layer: The output node together with links connected it act. The output of the neural network can be represented as

$$u_{nc} = \sum_{j=1}^{n} \mu_j \Theta_j \ . \tag{23}$$

B. Parameter Learning

The online learning algorithm is a gradient descent algorithm in the space of network parameters and aims at minimizing $s\dot{s}$ for achieving fast convergence of s. Multiplying both sides of (14) by s gives

$$s\dot{s} = s(u^* - u_{nc} - u_{sc}) - ks^2 . \tag{24}$$

According to the gradient descent method, the learning rules are summarized as follows.

(1) The weights $\mathbf{u}_j$ are updated by the following equation [20]

$$\Delta \mathbf{u}_j = -\eta_u \frac{\partial s\dot{s}}{\partial \mathbf{u}_j} = -\eta_u \frac{\partial s\dot{s}}{\partial u_{nc}} \frac{\partial u_{nc}}{\partial \mathbf{u}_j} = \eta_u s \mathbf{\Phi} \Theta_j \tag{25}$$

where $\Delta \mathbf{u}_j$ is the updated value of the j-th connection weights between the TSK layer and output layer and η_u is a positive learning rate. The connective weights can be updated according to the following equation

$$\mathbf{u}_j(k+1) = \mathbf{u}_j(k) + \Delta \mathbf{u}_j(k) . \tag{26}$$

(2) The parameters of wavelet neurons can also be adjusted in the following equation to increase the learning capability

$$\Delta c_j = -\eta_c \frac{\partial s\dot{s}}{\partial c_j} = -\eta_c \frac{\partial s\dot{s}}{\partial u_{nc}} \frac{\partial u_{nc}}{\partial \Theta_j} \frac{\partial \Theta_j}{\partial c_j} = 2\eta_c s \mu_j \frac{(\xi - c_j)}{\sigma_j^2} \Theta_j \tag{27}$$

$$\Delta \sigma_j = -\eta_\sigma \frac{\partial s\dot{s}}{\partial \sigma_j} = -\eta_\sigma \frac{\partial s\dot{s}}{\partial u_{nc}} \frac{\partial u_{nc}}{\partial \Theta_j} \frac{\partial \Theta_j}{\partial \sigma_j} = 2\eta_\sigma s \mu_j \frac{(\xi - c_j)^2}{\sigma_j^3} \Theta_j \tag{28}$$

where Δc_j and $\Delta \sigma_j$ are the updated values of the j-th hidden neuron, respectively, and η_c and η_σ are the positive learning rates. The parameters of the hidden neurons are updated as following

$$c_j(k+1) = c_j(k) + \Delta c_j(k) \tag{29}$$

$$\sigma_j(k+1) = \sigma_j(k) + \Delta \sigma_j(k) . \tag{30}$$

C. Stability Analysis

Since the number of hidden neurons is finite, an approximation error is inevitable. There exists an approximation error between optimal neural controller and ideal controller as

$$u^* = u_{nc}^* + \varepsilon \tag{31}$$

where u_{nc}^* is the optimal neural controller and ε denotes an estimate approximation error between the ideal controller and optimal neural controller. To ensure the closed-loop control system's stability, a compensator should be designed to dispel the approximation error. For guaranteeing the system stability, the smooth compensator is designed as

$$u_{sc} = \hat{\varepsilon} + \lambda s \tag{32}$$

where $\hat{\varepsilon}$ denotes the estimated value of the approximation error and λ is a positive constant. Substituting (32) and (31) into (14) yields

$$\dot{s} = \varepsilon - \hat{\varepsilon} - ks = \tilde{\varepsilon} - (\lambda + k)s \tag{33}$$

where $\tilde{\varepsilon} = \varepsilon - \hat{\varepsilon}$. To guarantee the stability of the AHCNC system, consider a Lyapunov function candidate in the following form as

$$V_2(s, \tilde{\varepsilon}, t) = \frac{1}{2} s^2 + \frac{1}{2\eta_\varepsilon} \tilde{\varepsilon}^2 \tag{34}$$

where η_ε is a positive learning rate. Differentiating (34) with respect to time and using (33) obtains

$$\begin{aligned} \dot{V}_2(s, \tilde{\varepsilon}, t) &= s\dot{s} + \frac{1}{\eta_\varepsilon} \tilde{\varepsilon}\dot{\tilde{\varepsilon}} \\ &= s[\tilde{\varepsilon} - (\lambda + k)s] + \frac{1}{\eta_\varepsilon} \tilde{\varepsilon}\dot{\tilde{\varepsilon}} \\ &= \tilde{\varepsilon}(s + \frac{1}{\eta_\varepsilon} \dot{\tilde{\varepsilon}}) - (\lambda + k)s^2 . \end{aligned} \tag{35}$$

For achieving $\dot{V}_2(s, \tilde{\varepsilon}, t) \le 0$, the error estimation law is designed as

$$\dot{\hat{\varepsilon}} = -\dot{\tilde{\varepsilon}} = \eta_\varepsilon s \tag{36}$$

then (35) can be rewritten as

$$\dot{V}_2 = -(\lambda + k)s^2 \le 0 . \tag{37}$$

Similar to (12), the stability of the AHCNC system can be guaranteed [1].

IV. Simulation Results

In this section, the proposed AHCNC system is applied to synchronize two chaotic gyros with nonlinear damping. The development of the proposed AHCNC method does not require the knowledge of the system dynamics. For practical implementation, the controller parameters of the AHCNC system can be tuned online by the proposed adaptive laws. Two simulation cases

including parameter variation and initial variation are considered.

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 $f_y = 36$.

The control parameters of the proposed AHCNC system are chosen as $k_1 = 2$, $k_2 = 1$, $\lambda = 0.1$, $\eta_u = 0.1$, $\eta_c = \eta_\sigma = 0.002$, and $\eta_\varepsilon = 0.001$. All the gains are chosen in consideration of the requirement of stability.

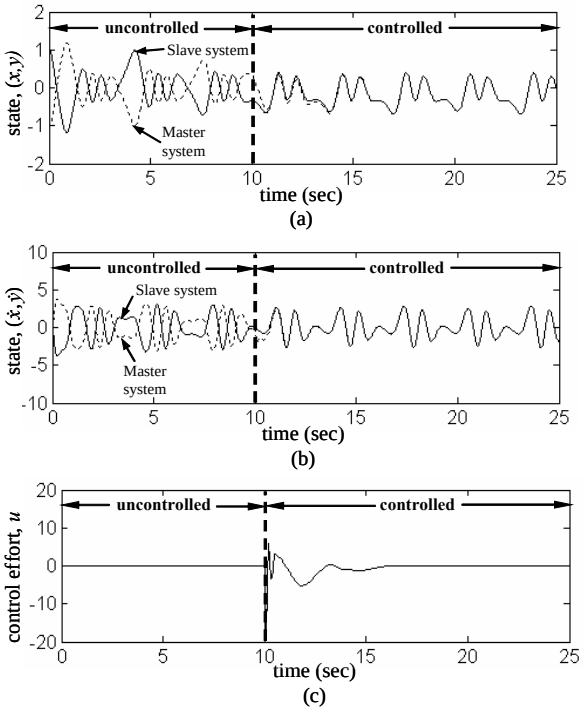

Fig. 5. Simulation results of the AHCNC system for Case 1.

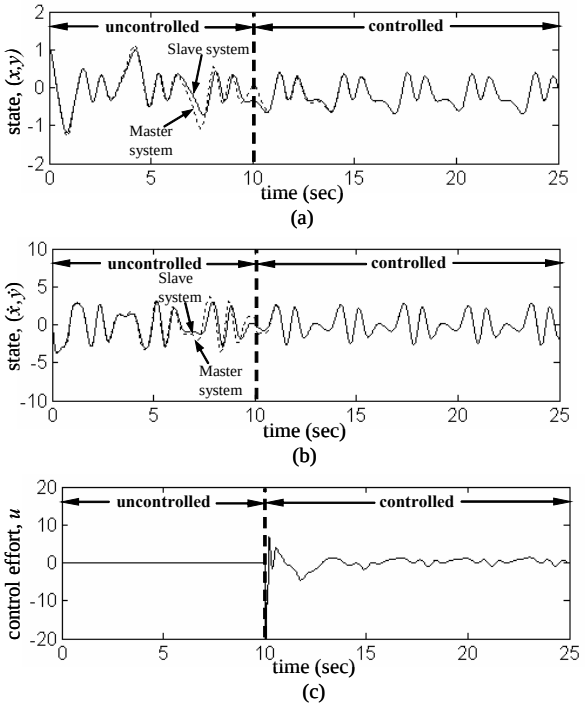

Fig. 6. Simulation results of the AHCNC system for Case 2.

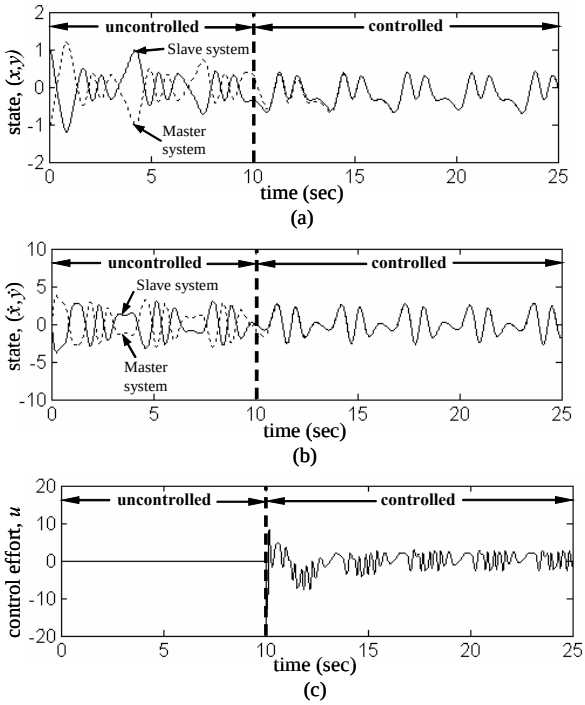

Fig. 7. Simulation results of the AHCNC system for Case 1 with a coupling term.

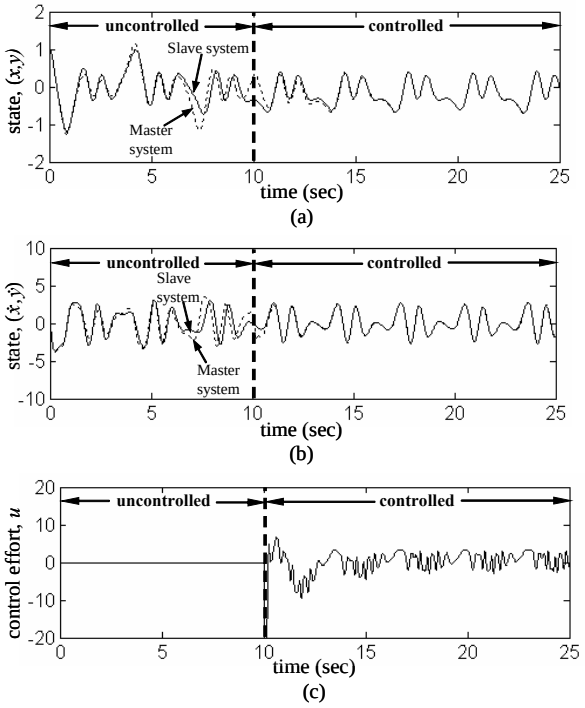

Fig. 8. Simulation results of the AHCNC system for Case 2 with a coupling term.

The simulation results of the proposed AHCNC system are shown in Figs. 5 and 6 for Cases 1 and 2, respectively. The tracking responses of states (x, y) are shown in Figs. 5(a) and 6(a), the tracking responses of states $(\dot{x}, \dot{y})$ are shown in Figs. 5(b) and 6(b), and the control inputs are shown in Figs. 5(c) and 6(c). The simulation results show that the proposed AHCNC system can achieve favorable synchronization

performance because the proposed online learning algorithms are applied.

To demonstrate the robust control performance of the proposed AHCNC system, a coupling term $F(x, y) = 2\sin(\dot{x} + \dot{y})$ is examined here. The simulation results of the proposed AHCNC system are shown in Figs. 7 and 8 for Cases 1 and 2, respectively. The tracking responses of states (x, y) are shown in Figs. 7(a) and 8(a), the tracking responses of states $(\dot{x}, \dot{y})$ are shown in Figs. 7(b) and 8(b), and the control inputs are shown in Figs. 7(c) and 8(c). The simulation results show that perfect tracking responses can be achieved.

V. Conclusions

An adaptive Hermite-polynomial-based CMAC neural control (AHCNC) system is proposed to synchronize two chaotic gyros. The proposed AHCNC system which is composed of a neural controller and a smooth compensator can automatically tune the controller parameters. The neural controller uses a Hermite-polynomial-based CMAC neural network (HCNN) to online approximate an ideal controller based on gradient descent algorithm and the smooth compensator is designed to eliminate the effect of the approximation error in the sense of Lyapunov stability. The HCNN which combines a CMAC neural network with a Hermite neural network is designed to improve the accuracy of functional approximation. Finally, the simulation results show that the high performance in chaos synchronization of gyro systems can be achieved by the proposed HCNN system after the controller parameters.

VI. Acknowledgment

The authors appreciate the partial financial support from the National Science Council of Republic of China under the grant NSC 100-2628-E-032-003.

VII. References

[1] J. J. E. Slotine and W. P. Li, *Applied Nonlinear Control*, Englewood Cliffs, NJ: Prentice Hall, 1991.

[2] K. H. Cheng, C. F. Hsu, C. M. Lin, T. T. Lee, and C. Li, "Fuzzy-neural sliding-mode control for DC-DC converters using asymmetric Gaussian membership functions," *IEEE Trans. Ind. Electron.*, vol. 54, no. 3, pp. 1528-1536, June 2007.

[3] 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, pp. 932-941, Jan. 2007.

[4] C. H. Chen, C. M. Lin, and T. Y. Chen, "Intelligent adaptive control for MIMO uncertain nonlinear systems," *Expert Syst. with Appl.*, vol. 35, no. 3, pp. 865-877, Oct. 2008.

[5] C. H. Chiu, "The design and implementation of a wheeled inverted pendulum using an adaptive output recurrent cerebellar model articulation controller," *IEEE Trans. Ind. Electro.*, vol. 57, no. 5, pp. 1814-1822, May 2010.

[6] Y. Wen and X. Ren, "Neural networks-based adaptive control for nonlinear time-varying delays systems with unknown control direction," *IEEE Trans. Neural Netw.*, vol. 22, no. 11, pp. 1783-1795, Nov. 2011.

[7] Y. F. Peng, R. J. Wai, and C. M. Lin, "Implementation of LLCC-resonant driving circuit and adaptive CMAC neural network control for linear piezoelectric ceramic motor, *IEEE Trans. Ind. Electron.*, vol. 51, no. 1, pp. 35-48, Feb. 2004.

[8] M. F. Yeh and K. C. Chang, "A self-organizing CMAC network with gray credit assignment," *IEEE Trans. Syst., Man, and Cybern., Part B: Cybern.*, vol. 36, no. 3, pp. 623-635, June 2006.

[9] C. M. Lin, L. Y. Chen, and C. H. Chen, "RCMAC hybrid control for MIMO uncertain nonlinear systems using sliding-mode technology," IEEE Trans. Neural Netw., vol. 18, no. 3, pp. 708-720, May 2007.

[10] M. F. Yeh and C. H. Tsai, "Standalone CMAC control system with online learning ability," *IEEE Trans. Syst., Man, and Cybern., Part B: Cybern.*, vol. 40, no. 1, pp. 43-53, Feb. 2010.

[11] M. Lagerholm, C. Peterson, G. Braccini, L. Edenbrandt, and L. Sornmo, "Clustering ECG complexes using Hermite functions and self-organizing maps," *IEEE Trans. Biomed. Eng.*, vol. 47, no. 7, pp. 838-848, July 2000.

[12] L. Ma and K. Khorasani, "Constructive feedforward neural networks using hermite polynomial activation functions," *IEEE Trans. Neural Netw.*, vol. 16, no. 4, pp. 821-833, July 2005.

[13] F. J. Lin, S. Y. Chen, and M. S. Huang, "Tracking control of thrust active magnetic bearing system via Hermite polynomial-based recurrent neural network," *IET Electr. Power Appl.*, vol. 4, no. 9, pp. 701-714, Nov. 2010.

[14] H. K. Chen, "Chaos and chaos synchronization of a symmetric gyro with linear-plus-cubic damping," *J. Sound Vibr.*, vol. 255, no. 4, pp. 719-740, Aug. 2002.

[15] H. T. Yau, "Nonlinear rule-based controller for chaos synchronization of two gyros with linear-plus-cubic damping," *Chaos Solitons Fract.*, vol. 34, no. 4, pp. 1357-1365, Nov. 2007.

[16] A. Poursamad and A. H. Davaie-Markazi, "Robust adaptive fuzzy control of unknown chaotic systems," *Appl. Soft Comput.*, vol. 9, no. 3, pp. 970-976, June 2009.

[17] C. F. Hsu, "Adaptive fuzzy wavelet neural controller design for chaos synchronization," *Expert Syst. with Appl.*, vol. 38, no. 8, pp. 10475-10483, Aug. 2011.

[18] C. F. Juang and C. Lo, "Zero-order TSK-type fuzzy system learning using a two-phase swarm intelligence," *Fuzzy Sets and Syst.*, vol. 59, no. 21, pp. 2910-2926, Nov. 2008.

[19] C. F. Hsu, "Intelligent tracking control of a DC motor driver using self-organizing TSK type fuzzy neural networks," *Nonlinear Dynamics*, vol. 67, no. 1, pp. 587-600, 2012.

[20] C. M. Lin and C. F. Hsu, "Supervisory recurrent fuzzy neural network control of wing rock for slender delta wings," *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 5, pp. 733-742, Oct. 2004.