Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Adaptive CMAC Control System Design for a Class of Nonlinear Systems

Chih-Min Lin* Chao-Ming Chung
Department of Electrical Engineering
Yuan Ze University
Chung-Li, Tao-Yuan, 320, Taiwan, Republic of China
*cml@saturn yzu.edu.tw

Chun-Fei Hsu
Department of Electrical Engineering
Chung Hua University
Hsinchu, 300, Taiwan, Republic of China
fei@chu.edu.tw

***Abstract*—Cerebellar model articulation controller (CMAC) has been already validated that it can approximate a nonlinear function over a domain of interest to any desired accuracy. This paper proposes an adaptive CMAC (PIACMAC) system with a PI-type learning algorithm. The PIACMAC system is composed of a CMAC and a compensation controller. CMAC is used to mimic an ideal controller and the compensation controller is designed to dispel the approximation error between CMAC and ideal controller. The Lyapunov stability theorems is utilized to derive the parameter learning algorithm, so that the uniformly ultimately bounded of PIACMAC system can be guaranteed. Then, the PIACMAC system is applied to a Duffing-Holmes chaotic system. Simulation results verify that the proposed PIACMAC system with a PI-type learning algorithm can achieve better control performance than other control methods.**

***Keywords*—CMAC, Lyapunov stability theorems, Uniformly ultimately bounded, Chaotic system.**

I. INTRODUCTION

The cerebellar model articulation controller (CMAC) is classified as a non-fully connected perceptron-like associative memory network with overlapping receptive-fields, and it has been already validated that it can approximate a nonlinear function over a domain of interest to any desired accuracy [1]. Since CMAC has the advantages such as fast learning property, good generalization capability and information storing ability, it has been shown that the CMAC-based adaptive control systems can achieve better control performance than neural-network-based adaptive control systems in some applications [2-6].

Since the neuron's number of neural network including CMAC is finite, the approximation error is inevitable when it is used to approximate an ideal controller. In order to ensure the control system stability, a compensation controller is needed to be designed to dispel the approximation error. The most frequently used compensation controller is a sliding-mode type control, which requires the bound of the approximation error [7, 8]. A large bound of approximation error will result in weary chattering of the control effort. Besides, some researchers have proposed neural-network-based adaptive control designs based on the H^{∞} control scheme. Combing the H^{∞} control [9, 10], the neural-network-based robust adaptive control approaches have been proposed to attenuate the effects of approximation error to a prescribed level. However, it is a trade-off between the amplitude of control effort and the performance of tracking error by choosing the specified attenuation level.

Though the CMAC-based adaptive control system can guarantee the system's stability, they used a conventional I-type learning algorithm [2-6]. By using I-type learning algorithm, the convergence of the controller parameters and tracking errors may be slow. This paper proposes an adaptive CMAC (PIACMAC) system with a PI-type learning algorithm to speed up the convergence of tracking error and controller parameters. The proposed PIACMAC system is composed of a CMAC and a compensation controller. CMAC is used to mimic an ideal controller, and the compensation controller is designed to guarantee the uniformly ultimately bound stability of the closed-loop system in the Lyapunov sense. Finally, the proposed PIACMAC system is applied to control a Duffing-Holmes chaotic system. Simulation results validate the good tracking performance of the proposed control system. Moreover, the proposed control system can speed up the convergence of the tracking error by using the developed PI-type learning algorithm.

II. PROBLEM STATEMENT

Consider a class of n-th order nonlinear systems described by the following form

$$x^{(n)} = f(\mathbf{x}) + u \tag{1}$$

where $\mathbf{x} = [x, \dot{x}, \ldots, x^{(n-1)}]^T$ is the state vector of the system, which is assumed to be available for measurement, $f(\mathbf{x})$ is the nonlinear system dynamics which can be unknown, and u is the input of the system. The tracking control problem of the system is to find a control law so that the state trajectory x can track a reference command x_d closely. Define the tracking error vector as

$$e = x_d - x . \tag{2}$$

Assume that all the parameters in (1) are well known, there exists an ideal controller [11]

$$u^* = -f(\mathbf{x}) + x_d^{(n)} + k_1 e^{(n-1)} + \cdots + k_{n-1}\dot{e} + k_n e \tag{3}$$

where $k_i, i=1,2,...,n$ are positive constant. Applying the ideal controller (3) into (1), yields

$$e^{(n)} + k_1 e^{(n-1)} + \cdots + k_{n-1}\dot{e} + k_n e = 0 . \qquad (4)$$

Suppose the feedback gain $k_i, i=1,2,\cdots,n$ are chosen to correspond with the coefficients of a Hurwitz polynomial, it implies that $\lim_{t\to\infty} e = 0$ for any starting initial conditions. However, since the system dynamics $f(\mathbf{x})$ may be unknown or perturbed in practical applications, the ideal controller in (3) can not be precisely obtained.

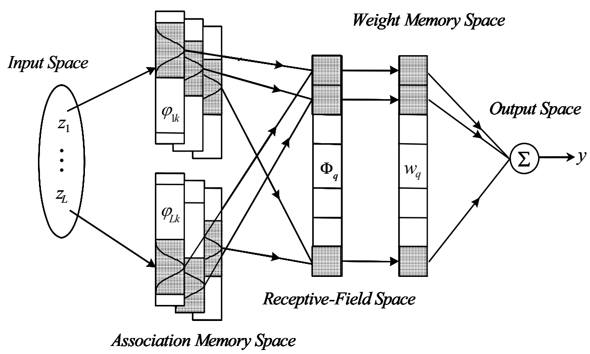

Fig. 1. The architecture of CMAC.

III. PIACMAC System Design

A. Description of CMAC

The network structure of CMAC is shown in Fig. 1. The architecture of CMAC includes input space, association memory space, receptive-field space, weight memory space and output space. The network structure of CMAC is shown in Fig. 1. The output of CMAC can be expressed as [4]

$$y = \sum_{q=1}^{N} w_q \, \Phi_q(\mathbf{z}) \qquad (5)$$

where $\mathbf{z} = [z_1, z_2, \cdots, z_L]^T$ is the input vector, Φ_q is the receptive-field basis function of the qth receptive-field, w_q denotes the connecting weight value of the qth receptive-field, N is the number of receptive-field, and the receptive-field basis function is defined as

$$\Phi_q(\mathbf{z}) = \prod_{j=1}^{L} \varphi_{jk}(z_j) \text{ , for } q = 1,2,\ldots,N \qquad (6)$$

where Gaussian function is adopted as the receptive-field basis function which can be represented as

$$\varphi_{jk}(z_j) = \exp\left[\frac{-(z_j - m_{jk})^2}{\sigma_{jk}^2}\right] \text{, for } k = 1,2,\cdots M \qquad (7)$$

where $\varphi_{jk}(z_j)$ presents the kth block of the jth input z_j with the mean m_{jk} and variance σ_{jk} and M is the number of block. For ease of notation, the output can be expressed in a vector notation as

$$y = \mathbf{w}^T \mathbf{\Phi}(\mathbf{z}, \mathbf{m}, \boldsymbol{\sigma}) \qquad (8)$$

where

$$\mathbf{w} = [w_1, ..., w_q, ..., w_N]^T \qquad (9)$$

$$\mathbf{\Phi} = [\Phi_1, ..., \Phi_q, ..., \Phi_N]^T \qquad (10)$$

$$\mathbf{m} = \left[m_{11}, ..., m_{L1}, m_{12}, ..., m_{L2}, ..., m_{1M}, ..., m_{LM}\right]^T \qquad (11)$$

$$\boldsymbol{\sigma} = \left[\sigma_{11}, ..., \sigma_{L1}, \sigma_{12}, ..., \sigma_{L2}, ..., \sigma_{1M}, ..., \sigma_{LM}\right]^T . \qquad (12)$$

This implies that there exists a CMAC of (5) such that it can uniformly approximate a nonlinear even time-varying function Ω. Theoretically, there exists optimal weight vectors such that [4, 12]

$$\Omega = y^* + \Delta = \mathbf{w}^{*T}\mathbf{\Phi}(\mathbf{z}, \mathbf{m}^*, \boldsymbol{\sigma}^*) + \Delta = \mathbf{w}^{*T}\mathbf{\Phi}^* + \Delta \qquad (13)$$

where Δ denotes the approximation error, $\mathbf{w}^*$ and $\mathbf{\Phi}^*$ are the optimal parameter vectors of $\mathbf{w}$ and $\mathbf{\Phi}$, respectively, and $\mathbf{m}^*$ and $\boldsymbol{\sigma}^*$ are the optimal parameter vectors of $\mathbf{m}$ and $\boldsymbol{\sigma}$, respectively. In fact, the optimal parameter vectors that are needed to best approximate a given nonlinear function Ω cannot be determined. Thus, an estimation function is defined as

$$\hat{y} = \hat{\mathbf{w}}^T \mathbf{\Phi}(\mathbf{z}, \hat{\mathbf{m}}, \hat{\boldsymbol{\sigma}}) = \hat{\mathbf{w}}^T \hat{\mathbf{\Phi}} \qquad (14)$$

where $\hat{\mathbf{w}}$ and $\hat{\mathbf{\Phi}}$ are the estimated parameter vectors of $\mathbf{w}$ and $\mathbf{\Phi}$, respectively, and $\hat{\mathbf{m}}$ and $\hat{\boldsymbol{\sigma}}$ are the estimated parameter vectors of $\mathbf{m}$ and $\boldsymbol{\sigma}$, respectively. Define the estimation error as

$$\begin{aligned}\tilde{y} &= \Omega - y = \mathbf{w}^{*T}\mathbf{\Phi}^* - \hat{\mathbf{w}}^T\hat{\mathbf{\Phi}} + \Delta \\ &= \tilde{\mathbf{w}}^T\tilde{\mathbf{\Phi}} + \hat{\mathbf{w}}^T\tilde{\mathbf{\Phi}} + \tilde{\mathbf{w}}^T\hat{\mathbf{\Phi}} + \Delta \end{aligned} \qquad (15)$$

where $\tilde{\mathbf{w}} = \mathbf{w}^* - \hat{\mathbf{w}}$ and $\tilde{\mathbf{\Phi}} = \mathbf{\Phi}^* - \hat{\mathbf{\Phi}}$. In the following, some tuning laws will be derived to online tune the parameters of CMAC to achieve favorable estimation of a nonlinear function. To achieve this goal, the Taylor expansion linearization technique is employed to transform the nonlinear function into a partially linear form, i.e. [4]

$$\tilde{\mathbf{\Phi}} = \mathbf{\Phi}_{\mathbf{m}}^T \tilde{\mathbf{m}} + \mathbf{\Phi}_{\boldsymbol{\sigma}}^T \tilde{\boldsymbol{\sigma}} + \mathbf{h} \qquad (16)$$

where $\tilde{\mathbf{m}} = \mathbf{m}^* - \hat{\mathbf{m}}$, $\tilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma}^* - \hat{\boldsymbol{\sigma}}$, $\mathbf{h}$ is a vector of higher-order terms, $\mathbf{\Phi}_{\mathbf{m}} = \left[\frac{\partial \Phi_1}{\partial \mathbf{m}} \frac{\partial \Phi_2}{\partial \mathbf{m}} \cdots \frac{\partial \Phi_N}{\partial \mathbf{m}}\right]\Big|_{\mathbf{m}=\hat{\mathbf{m}}}$, and $\mathbf{\Phi}_{\boldsymbol{\sigma}} = \left[\frac{\partial \Phi_1}{\partial \boldsymbol{\sigma}} \frac{\partial \Phi_2}{\partial \boldsymbol{\sigma}} \cdots \frac{\partial \Phi_N}{\partial \boldsymbol{\sigma}}\right]\Big|_{\boldsymbol{\sigma}=\hat{\boldsymbol{\sigma}}}$. Substituting of (16) into (15) yields

$$\tilde{y} = \tilde{\mathbf{w}}^T\tilde{\mathbf{\Phi}} + \hat{\mathbf{w}}^T(\mathbf{\Phi}_{\mathbf{m}}^T\tilde{\mathbf{m}} + \mathbf{\Phi}_{\boldsymbol{\sigma}}^T\tilde{\boldsymbol{\sigma}} + \mathbf{h}) + \tilde{\mathbf{w}}^T\hat{\mathbf{\Phi}} + \Delta$$
$$= \tilde{\mathbf{w}}^T\hat{\mathbf{\Phi}} + \tilde{\mathbf{m}}^T\mathbf{\Phi}_{\mathbf{m}}\hat{\mathbf{w}} + \tilde{\boldsymbol{\sigma}}^T\mathbf{\Phi}_{\boldsymbol{\sigma}}\hat{\mathbf{w}} + \hat{\mathbf{w}}^T\mathbf{h} + \tilde{\mathbf{w}}^T\tilde{\mathbf{\Phi}} + \Delta \quad (17)$$

To speed up the convergence of CMAC learning, the optimal vector $\mathbf{w}^*$ is decomposed into two parts as [13]

$$\mathbf{w}^* = \eta_P\mathbf{w}_{\mathbf{P}}^* + \eta_I\mathbf{w}_{\mathbf{I}}^* \quad (18)$$

where $\mathbf{w}_{\mathbf{P}}^*$ and $\mathbf{w}_{\mathbf{I}}^*$ are the proportional and integral terms of $\mathbf{w}^*$, respectively, η_P and η_I are positive constants, and $\mathbf{w}_{\mathbf{I}}^* = \int_0^t \mathbf{w}_{\mathbf{P}}^* d\tau$. The estimation vector $\hat{\mathbf{w}}$ is decomposed into two parts as

$$\hat{\mathbf{w}} = \eta_P\hat{\mathbf{w}}_{\mathbf{P}} + \eta_I\hat{\mathbf{w}}_{\mathbf{I}} \quad (19)$$

where $\hat{\mathbf{w}}_{\mathbf{P}}$ and $\hat{\mathbf{w}}_{\mathbf{I}}$ are the proportional and integral terms of $\hat{\mathbf{w}}$, respectively, and $\hat{\mathbf{w}}_{\mathbf{I}} = \int_0^t \hat{\mathbf{w}}_{\mathbf{P}} d\tau$. Thus, $\tilde{\mathbf{w}}$ can be expressed as

$$\tilde{\mathbf{w}} = \eta_I\tilde{\mathbf{w}}_{\mathbf{I}} - \eta_P\hat{\mathbf{w}}_{\mathbf{P}} + \eta_P\mathbf{w}_{\mathbf{P}}^* \quad (20)$$

where $\tilde{\mathbf{w}}_{\mathbf{I}} = \mathbf{w}_{\mathbf{I}}^* - \hat{\mathbf{w}}_{\mathbf{I}}$. Substituting (20) into (17), it can obtain that

$$\tilde{y} = (\eta_I\tilde{\mathbf{w}}_{\mathbf{I}} - \eta_P\hat{\mathbf{w}}_{\mathbf{P}} + \eta_P\mathbf{w}_{\mathbf{P}}^*)^T\hat{\mathbf{\Phi}} + \tilde{\mathbf{m}}^T\mathbf{\Phi}_{\mathbf{m}}\hat{\mathbf{w}} + \tilde{\boldsymbol{\sigma}}^T\mathbf{\Phi}_{\boldsymbol{\sigma}}\hat{\mathbf{w}}$$
$$+ \hat{\mathbf{w}}^T\mathbf{h} + \tilde{\mathbf{w}}^T\tilde{\mathbf{\Phi}} + \Delta$$
$$= \eta_I\tilde{\mathbf{w}}_{\mathbf{I}}^T\hat{\mathbf{\Phi}} - \eta_P\hat{\mathbf{w}}_{\mathbf{P}}^T\hat{\mathbf{\Phi}} + \tilde{\mathbf{m}}^T\mathbf{\Phi}_{\mathbf{m}}\hat{\mathbf{w}} + \tilde{\boldsymbol{\sigma}}^T\mathbf{\Phi}_{\boldsymbol{\sigma}}\hat{\mathbf{w}} + \varepsilon \quad (21)$$

where the uncertain term $\varepsilon = \eta_P\mathbf{w}_{\mathbf{P}}^{*T}\hat{\mathbf{\Phi}} + \hat{\mathbf{w}}^T\mathbf{h} + \tilde{\mathbf{w}}^T\tilde{\mathbf{\Phi}} + \Delta$ denotes the lump of approximation error, and is assumed to be bounded by $0 \le |\varepsilon| \le E$, in which E is a positive constant. However, this uncertainty bound E cannot be obtained in practice.

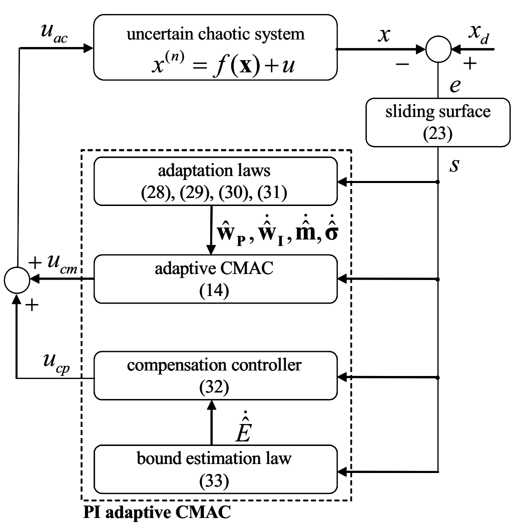

Fig. 2. The block diagram of PIACMAC system.

B. Design of PIACMAC

The PIACMAC system with a PI-type learning algorithm is shown in Fig. 2, where the control law is designed as

$$u_{ac} = u_{cm} + u_{cp} \quad (22)$$

in which a sliding surface is defined as

$$s = e^{(n-1)} + k_1e^{(n-2)} + \ldots + k_n\int_0^t e(\tau)d\tau\,. \quad (23)$$

The CMAC u_{cm} is used to approximate the ideal controller in (3), and the compensation controller u_{cp} is derived to compensate for the difference between the ideal controller and CMAC. Substituting (22) into (1) and using (3) and (23), the error dynamic equation can be obtained as

$$e^{(n)} + k_1e^{(n-1)} + \cdots + k_ne = u^* - u_{cm} - u_{cp} = \dot{s}\,. \quad (24)$$

By using the approximation error equation (21), (24) can be rewritten as

$$\dot{s} = \eta_I\tilde{\mathbf{w}}_{\mathbf{I}}^T\hat{\mathbf{\Phi}} - \eta_P\hat{\mathbf{w}}_{\mathbf{P}}^T\hat{\mathbf{\Phi}} + \tilde{\mathbf{m}}^T\mathbf{\Phi}_{\mathbf{m}}\hat{\mathbf{w}} + \tilde{\boldsymbol{\sigma}}^T\mathbf{\Phi}_{\boldsymbol{\sigma}}\hat{\mathbf{w}} + \varepsilon - u_{cp}\,. \quad (25)$$

To guarantee the uniformly ultimately bounded stability of the closed-loop PIACMAC control system, a Lyapunov function candidate is defined as

$$V = \frac{1}{2}s^2 + \frac{\eta_I}{2}\tilde{\mathbf{w}}_{\mathbf{I}}^T\tilde{\mathbf{w}}_{\mathbf{I}} + \frac{1}{2\eta_m}\tilde{\mathbf{m}}^T\tilde{\mathbf{m}} + \frac{1}{2\eta_\sigma}\tilde{\boldsymbol{\sigma}}^T\tilde{\boldsymbol{\sigma}} + \frac{1}{2\eta_E}\tilde{E}^2 \quad (26)$$

where η_m, η_σ and η_E are positive learning-rate constants, $\tilde{E} = E - \hat{E}$ in which $\hat{E}$ is the estimated approximation error bound. Taking the derivative of Lyapunov function in (26) and using (25), yields

$$\dot{V} = s\dot{s} - \eta_I\tilde{\mathbf{w}}_{\mathbf{I}}^T\dot{\hat{\mathbf{w}}}_{\mathbf{I}} - \frac{1}{\eta_m}\tilde{\mathbf{m}}^T\dot{\hat{\mathbf{m}}} - \frac{1}{\eta_\sigma}\tilde{\boldsymbol{\sigma}}^T\dot{\hat{\boldsymbol{\sigma}}} - \frac{1}{\eta_E}\tilde{E}\dot{\hat{E}}$$
$$= \eta_I\tilde{\mathbf{w}}_{\mathbf{I}}^T(s\hat{\mathbf{\Phi}} - \dot{\hat{\mathbf{w}}}_{\mathbf{I}}) + \tilde{\mathbf{m}}^T(s\mathbf{\Phi}_{\mathbf{m}}\hat{\mathbf{w}} - \frac{\dot{\hat{\mathbf{m}}}}{\eta_m}) + \tilde{\boldsymbol{\sigma}}^T(s\mathbf{\Phi}_{\boldsymbol{\sigma}}\hat{\mathbf{w}} - \frac{\dot{\hat{\boldsymbol{\sigma}}}}{\eta_\rho})$$
$$- \eta_P\hat{\mathbf{w}}_{\mathbf{P}}^Ts\hat{\mathbf{\Phi}} + s(\varepsilon - u_{cp}) - \frac{1}{\eta_E}\tilde{E}\dot{\hat{E}}\,. \quad (27)$$

If the adaptation laws are chosen as

$$\hat{\mathbf{w}}_{\mathbf{P}} = s\,\hat{\mathbf{\Phi}} - \mu_I(\hat{\mathbf{w}}_{\mathbf{I}} - \mathbf{w}_0) \quad (28)$$

$$\dot{\hat{\mathbf{w}}}_{\mathbf{I}} = s\,\hat{\mathbf{\Phi}} - \mu_I(\hat{\mathbf{w}}_{\mathbf{I}} - \mathbf{w}_0) \quad (29)$$

$$\dot{\hat{\mathbf{m}}} = \eta_m[s\mathbf{\Phi}_{\mathbf{m}}\,\hat{\mathbf{w}} - \mu_m(\hat{\mathbf{m}} - \mathbf{m}_0)] \quad (30)$$

$$\dot{\hat{\boldsymbol{\sigma}}} = \eta_\sigma[s\mathbf{\Phi}_{\boldsymbol{\sigma}}\,\hat{\mathbf{w}} - \mu_\sigma(\hat{\boldsymbol{\sigma}} - \boldsymbol{\sigma}_0)] \quad (31)$$

where μ_I, μ_m and μ_σ are small positive constants, $\mathbf{w}_0$, $\mathbf{m}_0$ and $\boldsymbol{\sigma}_0$ are initial estimation vectors of $\mathbf{w}^*$, $\mathbf{m}^*$ and $\boldsymbol{\sigma}^*$, respectively, and compensation controller is design as

$$u_{cp} = \hat{E}\tanh(\frac{s}{\delta}) \tag{32}$$

with bound estimation law as

$$\dot{\hat{E}} = \eta_E\left[s\tanh(\frac{s}{\delta}) - \mu_E(\hat{E} - E_0)\right] \tag{33}$$

where $\tanh(\cdot)$ is a tangent function, δ and μ_E are small positive constants, and E_0 is initial estimation vectors of E. Using (28)-(33), (27) can be obtained as

$$\begin{aligned}
\dot{V} &= \tilde{\mathbf{w}}_{\mathbf{I}}^T \eta_I \mu_I (\hat{\mathbf{w}}_{\mathbf{I}} - \mathbf{w}_0) + \tilde{\mathbf{m}}^T \mu_m (\hat{\mathbf{m}} - \mathbf{m}_0) + \tilde{\boldsymbol{\sigma}}^T \mu_\sigma (\hat{\boldsymbol{\sigma}} - \boldsymbol{\sigma}_0) \\
&\quad - \eta_P \hat{\mathbf{w}}_{\mathbf{P}}^T [\hat{\mathbf{w}}_P + \mu_I (\hat{\mathbf{w}}_{\mathbf{I}} - \mathbf{w}_0)] + s\varepsilon - s\hat{E}\tanh(\frac{s}{\delta}) \\
&\quad - \frac{1}{\eta_E}\tilde{E}\eta_E\left[s\tanh(\frac{s}{\delta}) - \mu_E(\hat{E} - E_0)\right] \\
&\le \tilde{\mathbf{w}}_{\mathbf{I}}^T \eta_I \mu_I (\hat{\mathbf{w}}_{\mathbf{I}} - \mathbf{w}_0) + \tilde{\mathbf{m}}^T \mu_m (\hat{\mathbf{m}} - \mathbf{m}_0) + \tilde{\boldsymbol{\sigma}}^T \mu_\sigma (\hat{\boldsymbol{\sigma}} - \boldsymbol{\sigma}_0) \\
&\quad - \hat{\mathbf{w}}_{\mathbf{P}}^T \eta_P \mu_I (\hat{\mathbf{w}}_{\mathbf{I}} - \mathbf{w}_0) + |s|E - s\hat{E}\tanh(\frac{s}{\delta}) \\
&\quad - \tilde{E}\left[s\tanh(\frac{s}{\delta}) - \mu_E(\hat{E} - E_0)\right].
\end{aligned} \tag{34}$$

It can be found that the following inequality holds for any $\delta > 0$ [14]

$$0 \le |s| - s\tanh(\frac{s}{\delta}) \le \kappa\delta \tag{35}$$

where κ is a constant satisfying $\kappa = \exp(-(\kappa+1))$. Using the inequality (35), (34) can be rewritten as

$$\begin{aligned}
\dot{V} &\le \tilde{\mathbf{w}}_{\mathbf{I}}^T \eta_I \mu_I (\hat{\mathbf{w}}_{\mathbf{I}} - \mathbf{w}_0) + \tilde{\mathbf{m}}^T \mu_m (\hat{\mathbf{m}} - \mathbf{m}_0) + \tilde{\boldsymbol{\sigma}}^T \mu_\sigma (\hat{\boldsymbol{\sigma}} - \boldsymbol{\sigma}_0) \\
&\quad - \hat{\mathbf{w}}_{\mathbf{P}}^T \eta_P \mu_I (\hat{\mathbf{w}}_{\mathbf{I}} - \mathbf{w}_0) + Ek\delta + \tilde{E}\mu_E(\hat{E} - \hat{E}_0) \\
&= \eta_I \mu_I (\mathbf{w}_{\mathbf{I}}^* - \hat{\mathbf{w}}_{\mathbf{I}})^T (\hat{\mathbf{w}}_{\mathbf{I}} - \mathbf{w}_0) + \mu_m (\mathbf{m}^* - \hat{\mathbf{m}})^T (\hat{\mathbf{m}} - \mathbf{m}_0) \\
&\quad + \mu_\sigma (\boldsymbol{\sigma}^* - \hat{\boldsymbol{\sigma}})^T (\hat{\boldsymbol{\sigma}} - \boldsymbol{\sigma}_0) - \eta_P \mu_I (\mathbf{w}_{\mathbf{P}}^* - \tilde{\mathbf{w}}_{\mathbf{P}})^T (\hat{\mathbf{w}}_{\mathbf{I}} - \mathbf{w}_0) \\
&\quad + \mu_E (E - \hat{E})(\hat{E} - E_0) + Ek\delta \\
&= -\frac{1}{2}\eta_I \mu_I [\|\mathbf{w}_{\mathbf{I}}^* - \hat{\mathbf{w}}_{\mathbf{I}}\|^2 + \|\mathbf{w}_{\mathbf{I}}^* - \mathbf{w}_0\|^2 - \|\hat{\mathbf{w}}_{\mathbf{I}} - \mathbf{w}_0\|^2] \\
&\quad - \frac{1}{2}\mu_m [\|\mathbf{m}^* - \hat{\mathbf{m}}\|^2 + \|\mathbf{m}^* - \mathbf{m}_0\|^2 - \|\hat{\mathbf{m}} - \mathbf{m}_0\|^2] \\
&\quad - \frac{1}{2}\mu_\sigma [\|\boldsymbol{\sigma}^* - \hat{\boldsymbol{\sigma}}\|^2 + \|\boldsymbol{\sigma}^* - \boldsymbol{\sigma}_0\|^2 - \|\hat{\boldsymbol{\sigma}} - \boldsymbol{\sigma}_0\|^2] \\
&\quad - \frac{1}{2}\eta_P \mu_I [\|\mathbf{w}_{\mathbf{P}}^* - \hat{\mathbf{w}}_{\mathbf{I}}\|^2 + \|\mathbf{w}_{\mathbf{P}}^* - \mathbf{w}_0\|^2 - \|\tilde{\mathbf{w}}_{\mathbf{P}} - \hat{\mathbf{w}}_{\mathbf{I}}\|^2 \\
&\quad - \|\tilde{\mathbf{w}}_{\mathbf{P}} - \mathbf{w}_0\|^2] \\
&\quad - \frac{1}{2}\mu_E [|E - \hat{E}|^2 + |E - E_0|^2 - |\hat{E} - E_0|^2] + Ek\delta \\
&\le -\frac{1}{2}\left[\eta_I \mu_I \|\tilde{\mathbf{w}}_{\mathbf{I}}\|^2 + \mu_m \|\tilde{\mathbf{m}}\|^2 + \mu_\sigma \|\tilde{\boldsymbol{\sigma}}\|^2 + \mu_E |\tilde{E}|^2\right] \\
&\quad + \frac{1}{2}\Big[\eta_I \mu_I \|\hat{\mathbf{w}}_{\mathbf{I}} - \mathbf{w}_0\|^2 + \mu_m \|\hat{\mathbf{m}} - \mathbf{m}_0\|^2 + \mu_\sigma \|\hat{\boldsymbol{\sigma}} - \boldsymbol{\sigma}_0\|^2 \\
&\quad + \eta_P \mu_I (\|\tilde{\mathbf{w}}_{\mathbf{P}} - \hat{\mathbf{w}}_{\mathbf{I}}\|^2 + \|\tilde{\mathbf{w}}_{\mathbf{P}} - \mathbf{w}_0\|^2) + \mu_E |E - E_0|^2\Big] \\
&\quad + Ek\delta
\end{aligned} \tag{36}$$

where $\|\cdot\|$ denotes an induced norm. Considering the Lyapunov function (26), (36) can be obtained as

$$\dot{V} \le -aV + b \tag{37}$$

where a and b are positive constants given by

$$a = \min(\mu_I, \mu_m \eta_m, \mu_\sigma \eta_\sigma, \mu_E \eta_E) \tag{38}$$

$$\begin{aligned}
b &= \frac{1}{2}\Big[\eta_I \mu_I \|\hat{\mathbf{w}}_{\mathbf{I}} - \mathbf{w}_0\|^2 + \mu_m \|\hat{\mathbf{m}} - \mathbf{m}_0\|^2 + \mu_\sigma \|\hat{\boldsymbol{\sigma}} - \boldsymbol{\sigma}_0\|^2 \\
&\quad + \eta_P \mu_I (\|\tilde{\mathbf{w}}_{\mathbf{P}} - \hat{\mathbf{w}}_{\mathbf{I}}\|^2 + \|\tilde{\mathbf{w}}_{\mathbf{P}} - \mathbf{w}_0\|^2) + \mu_E |E - E_0|^2 \\
&\quad + as^2\Big] + E\kappa\delta\,.
\end{aligned} \tag{39}$$

If (39) satisfies

$$0 \le V(t) \le \rho + (V(0) - \rho)\exp(-at) \tag{40}$$

where $\rho = \frac{b}{a} > 0$, then $\mathbf{e}$, $\mathbf{w}_{\mathbf{I}}$, $\mathbf{m}$, $\boldsymbol{\sigma}$ and E are uniformly bounded. Using (26) and (40) and giving any $\zeta > 2\rho$, there exists T such that for all $t \ge T$ the error satisfies

$$|e(t)| \le \zeta\,. \tag{41}$$

Thus, the uniformly ultimately bound stability of the closed-loop system can be guaranteed [14].

IV. Simulation Results

To illustrate the performance and efficiency of the proposed PIACMAC system, it is applied to control a Duffing-Holmes chaotic system. Recently, the issue of chaotic control system design has become a significant research topic in the physics, mathematics and engineering communities. Chaotic system is a nonlinear deterministic system that displays complex, noisy-like and unpredictable dynamic behavior; it has been found in many engineering systems such as in biological system, chemical reactions, laser physics, secure communication and biomedical. The dynamic equation of Duffing-Holmes chaotic system is described as [15, 16]

$$\ddot{x} = x - 0.25\dot{x} - x^3 + 0.3cos(t) + u = f(\mathbf{x}) + u\,. \tag{42}$$

The time response of the uncontrolled Duffing-Holmes chaotic system with initial conditions (0, 0) is shown in Fig. 3. It is clearly shown that the uncontrolled Genesio chaotic system has complex chaotic trajectories.

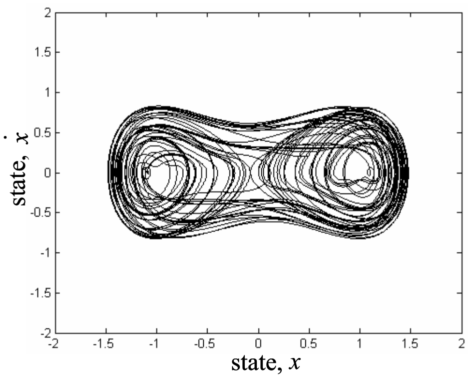

Fig. 3. The uncontrolled Duffing-Holmes chaotic system.

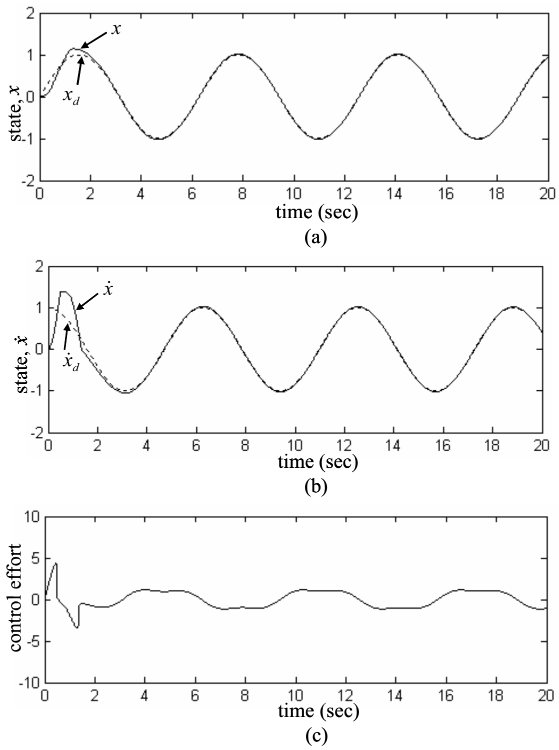

Fig. 4. Simulation results of the PIACMAC system with $\eta_P = 0$.

It should be emphasized that the development of the PIACMAC system does not need to know the dynamics of the controlled system. The parameters of the PIACMAC system can be online tuned by the proposed adaptive laws. The parameters for the PIACMAC system are selected as $k_1 = 2$, $k_2 = 1$, $\eta_I = 10$, $\eta_m = \eta_\sigma = \eta_E = 0.1$, $\mu_I = \mu_m = \mu_\sigma = \mu_E = 0.01$ and $\delta = 0.1$. All the parameters are chosen through some trials to achieve good transient control performance in the simulation considering the requirement of stability and possible operating conditions. The simulation results of the PIACMAC system with $\eta_P = 0$ for initial conditions (0, 0) are shown in Fig. 4. As $\eta_P = 0$, this results in the I-type learning algorithm. The tracking responses of state x and $\dot{x}$ are shown in Figs. 4(a) and 4(b), and the associated control effort is shown in Fig. 4(c). The simulation results show that it can achieve trajectory tracking performance; however, the convergence of tracking error is slow. Therefore, to achieve faster convergence performance, the PIACMAC system with $\eta_P = 1$ is applied to Duffing-Holmes chaotic system again. The simulation results of the PIACMAC system with $\eta_P = 1$ for initial conditions (0, 0) are shown in Fig. 5. The tracking responses of state x and $\dot{x}$ are shown in Figs. 5(a) and 5(b), and the associated control effort is shown in Fig. 5(c). From the simulation results, it is seen that the tracking errors converge quickly by using the PI-type learning algorithm.

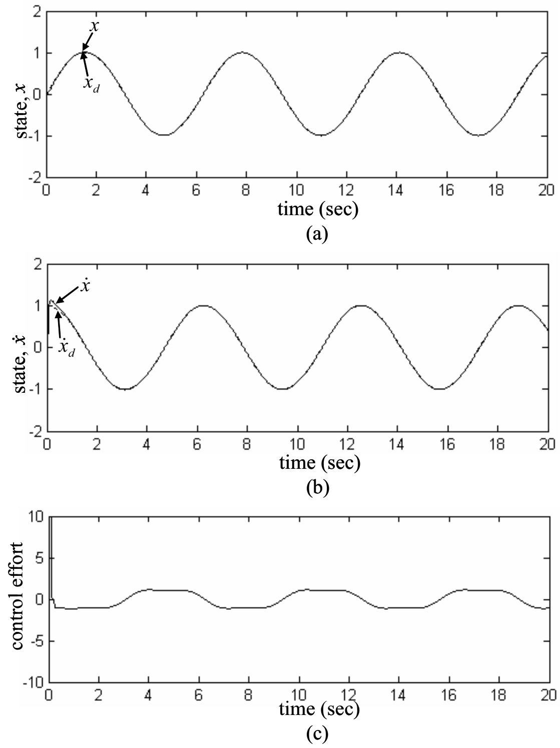

Fig. 5. Simulation results of the PIACMAC system with $\eta_P = 1$.

V. CONCLUSIONS

This paper has successfully developed an adaptive CMAC (PIACMAC) with a PI-learning algorithm. The proposed PIACMAC system is composed of a CMAC and a tangent function compensation controller. The Lyapunov stability theorem is utilized to derive the parameter tuning algorithms to guarantee the uniformly ultimately bounded of closed-loop system. This control system not only speeds up the convergence of tracking error by using the PI-type learning algorithm but also dispels the control chattering by using the tangent function compensation controller. The effectiveness of the proposed PIACMAC system is verified by the simulations of chaotic system control. Simulation results validate the good tracking performance of the proposed PIACMAC system.

The major contributions of this paper are: 1) The proposed PIACMAC system can achieve favorable tracking performance in controlling complex uncertain chaotic systems. 2) the Lyapunov stability theorem is utilized to derive the parameter tuning algorithms to guarantee that the closed-loop system is uniformly ultimately bounded; 3) the convergences of the tracking error can be speeded up by using the PI-type learning algorithm; and 4) the control chattering can be dispelled by using the hyperbolic tangent function compensation controller.

ACKNOWLEDGMENT

The authors appreciate the partial financial support from the National Science Council of Republic of China under grant NSC 96-2218-E-216-001.

REFERENCES

[1] J. C. Jan and S. L. Hung, "High-order MS CMAC neural network," *IEEE Trans. Neural Netw.*, vol. 12, no. 3, pp. 598-603, 2001.
[2] S. F. Su, T. Tao, and T. H. Hung, "Credit assigned CMAC and its application to online learning robust controllers," *IEEE Trans. Syst., Man, Cybern. B: Cybern.*, vol. 33, no. 2, pp. 202-213, 2003.
[3] C. M. Lin, Y. F. Peng and C. F. Hsu, "Robust cerebellar model articulation controller design for unknown nonlinear system," *IEEE Trans. Circ. Syst.-II*, vol. 51, no. 7, pp. 354-358, 2004.
[4] C. M. Lin and Y. F. Peng, "Adaptive CMAC-based supervisory control for uncertain nonlinear systems," *IEEE Trans. Syst., Man, Cybern. B: Cybern.*, vol. 34, no. 2, pp. 1248-1260, 2004.
[5] C. M. Lin and Y. F. Peng, "Missile guidance law design using adaptive cerebellar model articulation controller," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 636-644, 2005.
[6] S. F. Su, Z. J. Lee, and Y. P. Wang, "Robust and fast learning for fuzzy cerebellar model articulation controllers," *IEEE Trans. Syst., Man, Cybern. B: Cybern.*, vol. 36, no. 1, pp. 203-208, 2006.
[7] C. M. Lin and C. F. Hsu, "Neural network hybrid control for antilock braking systems," *IEEE Trans. Neural Netw.*, vol. 14, no. 2, pp. 351-359, 2003.
[8] 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, 2004.
[9] W. Y. Wang, M. L. Chan, C. C. J. Hsu, and T. T. Lee, " H^{∞} tracking-based sliding mode control for uncertain nonlinear systems via an adaptive fuzzy-neural approach," *IEEE Trans. Syst., Man, Cybern. B: Cybern.*, vol. 32, no. 4, pp. 483-492, 2002.
[10] C. F. Hsu, "Self-organizing adaptive fuzzy neural control for a class of nonlinear systems," *IEEE Trans. Neural Netw.*, vol. 18, no. 4, pp. 1232-1241, 2007.
[11] J. J. E. Slotine and W. P. Li, *Applied nonlinear control*. Englewood Cliffs, NJ: Prentice-Hall, 1991.
[12] L. X. Wang, *Adaptive fuzzy systems and control: design and stability analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1994.
[13] N. Golea, A. Golea, and K. Benmahammed, "Fuzzy model reference adaptive control," *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 4, pp. 436-444, 2002.
[14] J. H. Park, S. J. Seo, G. T. Park, "Robust adaptive fuzzy controller for nonlinear system using estimation of bounds for approximation errors," *Fuzzy Sets and Syst.*, vol. 133, no. 1, pp. 19-36, 2003.
[15] 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.
[16] C. F. Hsu, "Adaptive recurrent neural network control using a structure adaptation algorithm," *Neural Computing & Applications*, vol. 18, no. 2, pp. 115-125, 2009.