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

Sliding-Mode-Based Fuzzy CMAC Controller Design for a Class of Uncertain Nonlinear System

Chun-Sheng Chen
Department of Electronic Engineering
China University of Science and Technology
Taipei, Taiwan, R.O.C.
chunshen@cc.chit.edu.tw

***Abstract*—The maim idea proposed in this paper is integrating sliding mode control (SMC) theory and cerebellar model articulation controller (CMAC) neural network into fuzzy controller design and the fuzzy control rules can be determined systematically by the sliding condition of the SMC. The advantages of using fuzzy model into CMAC are to improve function approximation accuracy in terms of the weighting coefficients of CMAC. The proposed slide-mode-based fuzzy CMAC (SFCMAC), which results from the direct adaptive approach, has the ability to tune the adaptation parameters in the THEN-part of each fuzzy rule during real-time operation. The weight-update law is derived using a Lyapunov stability analysis that guarantees the stability of the closed-loop system. Simulation results show a satisfactory performance of the proposed control scheme.**

***Keywords*— CMAC, sliding mode, fuzzy control, nonlinear**

I. Introduction

The control of general nonlinear system has been a widely investigated problem because of its wide applications in practical systems. However, inevitably the constructed model comprises unmodeled nonlinearity and uncertain disturbance that conventional control strategies based on a mathematical model of the controlled system cannot be easily derived to deal with. Moreover, many practical systems may be so complex or, even, model-free that to construct a mathematical model or identify its parameters is difficult or even impossible.

There has been considerable attention over the years on researches using neural networks (NNs) based on human heuristic and learning algorithms [1-3]. The cerebellar model articulation controller (CMAC), developed by Albus in the 1970s [4,5], is a nonfully connected perceptron-like associative memory network that computes a nonlinear function by referring a look-up table over a domain of interest. Structurally, it is equivalent to a network architecture with three layers. The contents of these memory locations are referred as weights, and the output of this network is a linear combination of these weights in the memory addressed by the activated inputs [6]. Compared with the general multiplayer neural network with back-propagation algorithm, the CMAC has been applied to a wide variety of complex dynamical systems because of its simple computation, fast learning property and good generalization capability [7-9]

The maim idea proposed in this paper is integrating sliding mode control (SMC) theory and CMAC neural network into fuzzy controller design and the fuzzy control rules can be determined systematically by the sliding condition of the SMC. The advantages of using fuzzy model into CMAC are to improve function approximation accuracy in terms of the weighting coefficients of CMAC. The proposed slide-mode-based fuzzy CMAC (SFCMAC), which results from the direct adaptive approach, has the ability to tune the adaptation parameters in the THEN-part of each fuzzy rule during real-time operation. Each fuzzy rule corresponds to a sub-CMAC. Then, the constructed SFCMAC is employed, such that the closed-loop stability is guaranteed.

This paper will address the problem of controlling an unknown nonlinear affined system. First, a SFCMAC is used to approximate the equivalent control by using an on-line fuzzy adaptation scheme, and then the hitting control is appended to show that the proposed SFCMAC can result in a closed-loop system, which is stable.

II. Description of the FCMAC neural network

A. The basic CMAC model

The basic idea of CMAC is to store learned data into overlapping regions in a way that the data can easily be recalled but use less storage space [10]. The action of storing weight information in the CMAC model is similar to that of the cerebellum in humans. Generally speaking, the CMAC can be viewed as a lookup table. In this technique, each state variable is quantized and the problem space is divided into discrete states. A vector of quantized input values specifies a discrete state and is used to generate addresses for retrieving information from memory for this state. Fig. 1 depicts the structure of a two-dimensional (2-D) CMAC. The input vector (or the so-called state) is defined by two state variables, v_1 and v_2, which are quantized and the state space is divided into three discrete regions, called blocks. It is noted that the width of the blocks affects the generalization capability of the CMAC. In this example, 7 locations are to be distinguished for each variable. For each state variable, three kinds of segmentation are used, the variable v_1 is divided into blocks *A*, *B* and *C*, and the variable v_2 is divided into blocks *a*, *b* and *c*. Then, the areas *Aa*, *Ab*, *Ac*, *Ba*, *Bb*, *Bc*, *Ca*, *Cb* and *Cc* are the addresses or the locations that store data which are often called hypercubes.

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

When each block is shifted by a small interval (call an element), different blocks will be obtained. For instance, blocks *D*, *E* and *F* in the second row for v_1 and *d*, *e* and *f* in the second column for v_2 are possible shifted regions. Then, *Dd*, *De*, *Df*, *Ed*, *Ee*, *Ef*, *Fd*, *Fe* and *Ff* are new hypercubes from the shifted regions. Similarly, hypercubes *Gg*, *Gh*, *Gi*, *Hg*, *Hh*, *Hi*, *Ig*, *Ih* and *Ii* are defined in the third layer.

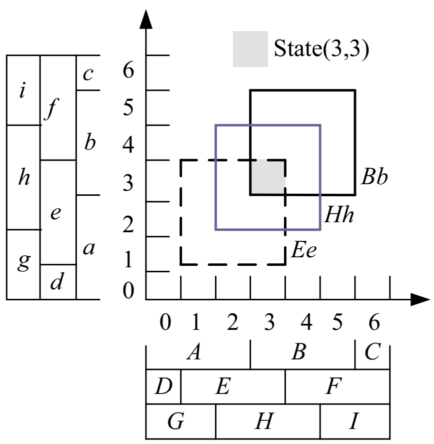


Figure 1. Block division of a 2-D CMAC.

Hypercubes in the *p*th layer are defined by the *p*th way of quantization of both variables. We restrict that hypercubes must be formed by the corresponding quantizations, e.g., the *p*th way of quantization (see Fig. 1) for v_1 with the *p*th way of quantization for v_2. Thus, the hypercubes, such as, *Ad* and *Db*, do not exist. With this kind of decomposition, one can imagine that there are N_h layers of hypercubes, where N_e is the number of elements in a complete block. Each state is covered by N_h different hypercubes, one from each layer. The total number of blocks on an input and that of elements in a block determine the number of layers and the degree of generalization. The CMAC associates each hypercube to a physical memory element.

CMAC uses a set of indices as an address in accordance with the current state to extract the stored data. Assume that N_n is the number of hypercubes; this is the same as the memory size in the case. Using the CMAC technique, a stored data y_i can be mathematically expressed as

$$y_i = \boldsymbol{a}_i \boldsymbol{w} = \sum_{j=1}^{N_n} a_{ij} w_j \tag{1}$$

where $\boldsymbol{w}$ indicates the column vector of memory contents and $\boldsymbol{a}_i$ is a memory element selection row vector that has N_e ones. Each hypercube is associated to an " arbitrary but deterministic" physical memory address. In Fig. 1, there are 27 hypercubes used to distinguish 49 different states in the 2-D CMAC. For example, let the hypercubes *Bb*, *Ee*, and *Hh* be addressed by the state $(v_1, v_2) = (3,3)$. Since each state addresses exactly N_n hypercubes, only these three hypercubes are set to 1, and the others are set to 0. Note that after the way of block division (see Fig. 1) is determined, the vector $\boldsymbol{a}_i$ for a specific quantized state is fixed.

B. Fuzzy CMAC neural network

A fuzzy CMAC (FCMAC) system can be controlled by the following linguistic rules

$$R^{(i_1,i_2)}: \text{ IF } x_1 \text{ is } B_1^{i_1} \text{ and } x_2 \text{ is } B_2^{i_2}, \text{ THEN } y = \boldsymbol{a}_{i_1 i_2} \boldsymbol{w} \tag{2}$$

where $i_j = 1,2,\cdots,p_j$, $j = 1,2$, and $\boldsymbol{a}_{i_1 i_2}$ is the memory element selection row vector, and $\boldsymbol{w}$ is the weighting column vector of the CMAC. The FCMAC system with center-average defuzzifier, product inference and singleton fuzzifier is defined as [11]

$$\begin{aligned} y &= \sum_{i_1=1}^{p_1}\sum_{i_2=1}^{p_2} \boldsymbol{a}_{i_1 i_2} \boldsymbol{w} \cdot \left(\prod_{j=1}^{2} \mu_{B_j^{i_j}}(x_j) \right) \Bigg/ \sum_{i_1=1}^{p_1}\sum_{i_2=1}^{p_2} \left(\prod_{j=1}^{2} \mu_{B_j^{i_j}}(x_j) \right) \\ &= \boldsymbol{\varphi}(\boldsymbol{x}) \boldsymbol{A} \boldsymbol{w} \end{aligned} \tag{3}$$

where $\prod_{j=1}^{2} \mu_{B_j^{i_j}}(x_j)$ is the membership function of fuzzy set $B_j^{i_j}$ and $\boldsymbol{A}$ is matrix constituted from the memory element selection vector $\boldsymbol{a}_{i_1 i_2}$, and $\boldsymbol{\varphi} = (\phi_1, \phi_2, \cdots, \phi_{p_1 p_2})$ is a row vector whose proper dimension depends on the number of fuzzy rules with $\phi_{i_1 i_2}$ is defined as

$$\phi_{i_1 i_2}(x) = \left(\prod_{j=1}^{2} \mu_{B_j^{i_j}}(x_j) \right) \Bigg/ \sum_{i_1=1}^{p_1}\sum_{i_2=1}^{p_2} \left(\prod_{j=1}^{2} \mu_{B_j^{i_j}}(x_j) \right) \tag{4}$$

III. DESIGN OF THE SFCMAC CONTROLLER

Consider a nonlinear system whose equation of motion can be governed by

$$y^{(n)} = f(\boldsymbol{x}) + bu \tag{5}$$

where y and $y^{(n)}$ denote the output and its derivative, respectively, u is the system input, $\boldsymbol{x} = [x, \dot{x}, \cdots, x^{(n-1)}]^T = [x_1, x_2, \cdots, x_n]^T$ is the state vector, $f(\boldsymbol{x})$ is an unknown continuous function, and b is a positive constant.

If we let y_d represents the known desired trajectory, the control aim is to determine a controller for the nonlinear system described by (5) so that the tracking error represented by

$$\boldsymbol{e} = [e, \dot{e}, \cdots, e^{(n-1)}]^T \tag{6}$$

With $e = y_d - y$ will be attenuated to an arbitrary small residual tracking error set. Further, we define a sliding surface in the error space passing through the origin to represent a sliding surface as follows:

$$s = e^{(n-1)} + c_{n-1} e^{(n-2)} + \cdots + c_2 \dot{e} + c_1 e \tag{7}$$

and $\boldsymbol{c} = [c_1, \cdots, c_{n-1}]^T$ be such that all roots of the characteristic polynomial describing the sliding surface

$$p(\lambda) = \lambda^{n-1} + c_{n-1}\lambda^{n-2} + \cdots + c_2 \lambda + c_1 \tag{8}$$

have negative real parts with desirable pole placement.

Since (8) guarantees (7) to satisfy the Hurwitz stability criterion, maintaining system states on sliding surface $s(t)$ for all $t>0$ is equivalent to the tracking problem $y=y_d$. The tracking control problem can be formulated by keeping the error vector $\boldsymbol{e}$ on the sliding surface defined as follows:

$$\dot{s}=y_d^{(n)}-f(x)-bu+\sum_{i=1}^{n-1}c_i e^{(i)} \tag{9}$$

If the system state is outside the sliding surface s, the controller must be designed such that it can force the system states to approach the sliding surface and then move along the sliding surface to the origin. By choosing the Lyapunov function candidate $V=\frac{1}{2}s^T s$, an equivalent control is given first such that each state Lyapunov-like condition holds for system stability [12]:

$$\frac{1}{2}\frac{d}{dt}(s^2)\le -\eta|s|,\ \eta>0 \tag{10}$$

Suppose that the control law is:

$$u^*=b^{-1}[\sum_{i=1}^{n-1}c_i e^{(i)}-f(\boldsymbol{x})+y_d^{(n)}+\eta\,\mathrm{sgn}(s)] \tag{11}$$

where η is a positive constant and $\mathrm{sgn}(s)$ is defined as:

$$\mathrm{sgn}(s)=\begin{cases}1 & s>0\\ 0 & s=0\\ -1 & s<0\end{cases} \tag{12}$$

Thus, the sliding condition (10) can be easily verified.

However, f is unknown, only estimation $\hat{f}$ can be used to construct u. To solve this problem, we propose the adaptive scheme using the fuzzy CMAC system. The fuzzy CMAC is constructed from the following fuzzy system:

$$\text{IF } x_1 \text{ is } B_1^{i_1} \text{ and } x_2 \text{ is } B_2^{i_2},\ \text{THEN } f \text{ is } \boldsymbol{a}_{i_1 i_2}\boldsymbol{w} \tag{13}$$

where we allow weight vector $\boldsymbol{w}$ to be adjustable. Furthermore, for the fuzzy CMAC neural network of (13), we assume that we have $f(\boldsymbol{x})=\hat{f}(\boldsymbol{x}\,|\,\boldsymbol{w})$. Then this estimated fuzzy CMAC can be expressed as:

$$\hat{f}(\boldsymbol{x}\,|\,\boldsymbol{w})=\boldsymbol{\varphi}(\boldsymbol{x})\boldsymbol{A}\boldsymbol{w} \tag{14}$$

where $\boldsymbol{\varphi}(\boldsymbol{x})$ is row vector of whose proper dimension depends on the number of fuzzy rules, $\boldsymbol{A}$ is matrixes constituted from the memory element selection $\boldsymbol{a}_i$, $\boldsymbol{w}$ is the corresponding parameter of the memory contents.

Motivated by the principle of SMC, the equivalent control u_{eq} is estimated by using an adaptive mechanism that forces the system state to slide on the sliding surface and the hitting control u_h that drives the states toward the sliding surface. Thus overall control law can be represented as:

$$u=u_{eq}+u_h \tag{15}$$

where u_{eq} and u_h are, respectively, yielded through fuzzy and non-fuzzy design modes. Note that the fuzzy CMAC neural network $\hat{f}$ is used to construct the partially section of the control law:

$$u_{eq}=b^{-1}[\sum_{i=1}^{n-1}c_i e^{(i)}-\hat{f}(\boldsymbol{x}\,|\,\boldsymbol{w})+y_d^{(n)}+\eta\,\mathrm{sgn}(s)] \tag{16}$$

IV. LEARNING ALGORITHM AND PERFORMANCE ANALYSIS

In this section, we show how to derive an adaptive law to adjust the weighting factor such that the estimated equivalent control u_{eq} can be optimally approximated to the equivalent control of the SMC under the situations of unknown function f. Then, we construct the hitting control to guarantee system's stability.

We define the control $u=u_{eq}+u_h$. Suppose there exists constant optimal parameter for the weighting vector $\boldsymbol{w}^*$. Thus (9) can be rewritten as:

$$\begin{aligned}\dot{s}&=[\hat{f}(\boldsymbol{x}\,|\,\boldsymbol{w})-f(\boldsymbol{x})]-bu_h-\eta\cdot\mathrm{sgn}(s)\\ &=\boldsymbol{\varphi A}(\boldsymbol{w}-\boldsymbol{w}^*)-bu_h-\eta\cdot\mathrm{sgn}(s)+\xi\\ &=\boldsymbol{\varphi A}\tilde{\boldsymbol{w}}-bu_h-\eta\cdot\mathrm{sgn}(s)+\xi\end{aligned} \tag{7}$$

where $\tilde{\boldsymbol{w}}=\boldsymbol{w}-\boldsymbol{w}^*$ denotes the parameter estimation error, and the minimum approximation errors as $\xi=\hat{f}(\boldsymbol{x}\,|\,\boldsymbol{w}^*)-f(\boldsymbol{x})$, with $\boldsymbol{w}=[w_1,w_2,\cdots,w_{N_n}]^T$. Our design objective is to specify the control and adaptive law for w_i such the sliding condition (10) is guaranteed.

Consider the Lyapunov function candidate

$$V=\frac{1}{2}s^2+\frac{1}{2r_1}\tilde{\boldsymbol{w}}^T\tilde{\boldsymbol{w}} \tag{18}$$

where r_1 is a positive constant. By the fact, $\dot{\tilde{\boldsymbol{w}}}=\dot{\boldsymbol{w}}$ and (17), we can obtain the time derivative of V as

$$\begin{aligned}\dot{V}&=s\dot{s}+\frac{1}{r_1}\tilde{\boldsymbol{w}}^T\dot{\tilde{\boldsymbol{w}}}\\ &=s[\boldsymbol{\varphi A}\tilde{\boldsymbol{w}}-\eta\cdot\mathrm{sgn}(s)+\xi-bu_h]+\frac{1}{r_1}\tilde{\boldsymbol{w}}^T\dot{\tilde{\boldsymbol{w}}}\\ &=\frac{1}{r_1}\tilde{\boldsymbol{w}}^T[sr_1(\boldsymbol{\varphi A})^T+\dot{\tilde{\boldsymbol{w}}}]+s\xi-s\eta\cdot\mathrm{sgn}(s)-sbu_h\end{aligned} \tag{19}$$

We then have the following adaptation law

$$\dot{\boldsymbol{w}}=\dot{\tilde{\boldsymbol{w}}}=-sr_1(\boldsymbol{\varphi A})^T \tag{20}$$

then

$$\dot{V}=s\xi-s\eta\cdot\mathrm{sgn}(s)-sbu_h \tag{21}$$

To complete the SFCMAC design, it is necessary to show that the hitting control is enough to force the state trajectory toward the sliding surface as well as to establish asymptotic convergence of the tracking error. Consider the Lyapunov function candidate

$$V=\frac{1}{2}s^2 \tag{22}$$

Taking the derivative of (22) and using (15), and (9), one has

$$\dot{V}=s(-f-bu_{eq}+\sum_{i=1}^{n-1}c_i e^{(i)}+y_d^{(n)})-sbu_h \tag{23}$$

To ensure that (23) is less than zero, the hitting control should be selected as

$$u_h=\operatorname{sgn}(s)\cdot b^{-1}\cdot[|f|_{\max}+b\cdot|u_{eq}|+|y_d^{(n)}|+|\sum_{i=1}^{n-1}c_i e^{(i)}| \tag{24}$$

This means that the inequality $\dot{V}=s\dot{s}<0$ is obtained and the hitting control actually achieves a stable SFCMAC system.

From the above discussion, we use a SFCMAC to estimate the equivalent control of the system. Conceptually, the equivalent control is desired when the state trajectory is near $s=0$, while the hitting control is determined in the case of $s\neq 0$ [13]. A fuzzy rule base is of the form

$$\text{If } s \text{ is } ZO, \quad \text{Then } u \text{ is } u=u_{eq} \tag{25}$$

$$\text{If } s \text{ is } NZ, \quad \text{Then } u \text{ is } u=u_{eq}+u_h \tag{26}$$

where *ZO* and *NZ* denote zero and nonzero fuzzy sets, respectively, and input variable *s* is given in (7). The control law of the fuzzy controller is

$$u=\frac{\mu_{ZO}(s)u_{eq}+\mu_{NZ}(s)[u_{eq}+u_h]}{\mu_{ZO}(s)+\mu_{NZ}(s)} \tag{27}$$

where $\mu_{ZO}(s)$ and $\mu_{NZ}(s)$ is the membership functions of fuzzy sets *ZO* and *NZ*, respectively. The membership functions of fuzzy sets *ZO* and *NZ* are selected to overlap and be symmetric to satisfy $\mu_{ZO}(s)+\mu_{NZ}(s)=1$.

If we choose the triangle membership functions as shown in Fig. 2 for the fuzzy sets *ZO* and *NZ* of *s*, the control law u will be continuously adjusted by the use of the fuzzy logic depending on "*ZO*" layer s^1. When holding the condition $|s|\geq s^1$, it can be seen that the control law is the same as the proposed SFCMAC. However, the amount of hitting control in region $|s|<s^1$ is dominated by the grade of the membership function of *NZ*, that is, the hitting control could be attenuated by the grade of *NZ*.

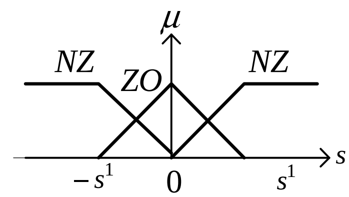


Figure 2. The fuzzy membership functions of *ZO* and *NZ*

V. SIMULATION

In this section, the proposed SMC-based FCMAC control system will be applied to control a Duffing forced oscillation system [14]. The oscillation system has chaotic phenomena if the control input equals zero. The dynamic equation of a Duffing forced oscillation system can be described as follows:

$$\begin{aligned}\dot{x}_1&=x_2\\ \dot{x}_2&=-0.1x_2+x_1^3+12\cos(t)+u\end{aligned} \tag{28}$$

The trajectory to be followed is described by the linear system from (7), the desired coefficients are specified to be $c_1=2$. The chaotic system is given by the following desired trajectory sin(t) with the initial states $\boldsymbol{x}=(2,-2)^T$ and weights are assigned as $\boldsymbol{w}=0$ for 27 discrete blocks. The fuzzy system for the proposed SFCMAC controller for system (28) is given by:

$$R^{(i_1,i_2)}: \text{IF } x_1 \text{ is } B_1^{i_1} \text{ and } x_2 \text{ is } B_2^{i_2}, \quad \text{THEN } f \text{ is } \boldsymbol{a}_{i_1i_2}\boldsymbol{w} \tag{29}$$

where $i_j=1,2,\cdots,7$, $j=1,2$, and $\boldsymbol{a}_{i_1i_2}$ is the memory element selection row vector, and $\boldsymbol{w}$ is the weighting column vector of the CMAC. The membership functions of states x_1 and x_2 for the qualitative statements (N=7^2=49 regular rule partitions) are defined as {*NB*, *NM*, *NS*, *ZE*, *PS*, *PM*, *PB*} where

$$NB: \frac{1}{0.5+\exp(0.1\times(x+3))}$$

$$NM: \exp(-0.5(x_i+2)^2)$$

$$NS: \exp(-0.5(x_i+1)^2)$$

$$ZE: \exp(-0.5x_i^2)$$

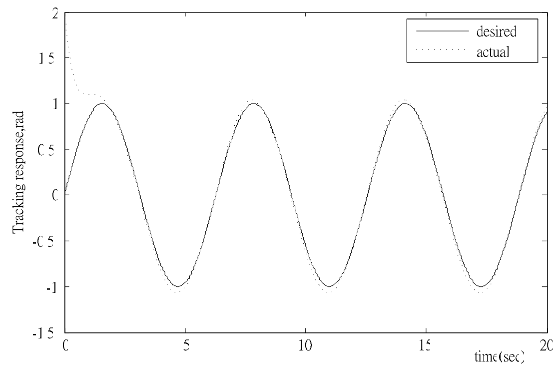
$$PS: \frac{1}{0.5+\exp(-0.1\times(x-3))}$$

$$PM: \exp(-0.5(x_i-2)^2)$$

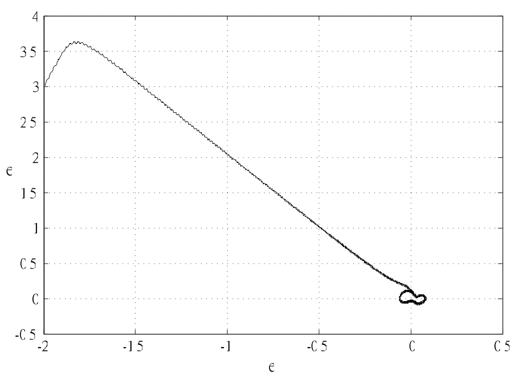
$$PB: \exp(-0.5(x_i-3)^2)$$

Consider the design parameters are given by $r_1=0.1$, $\eta=0.01$.

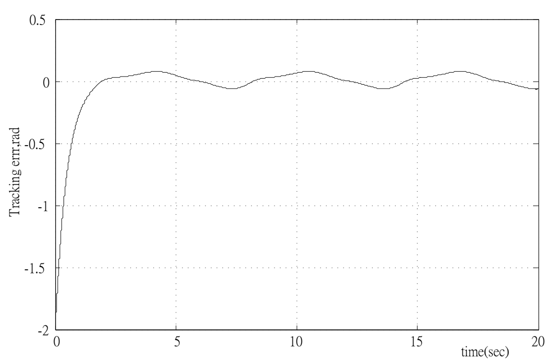
Simulation results with the initial condition $\boldsymbol{x}=(2,-2)^T$ are given in Fig. 3. From these simulation results, the tracking error has been attenuated efficiently. Thus we see that the proposed control scheme can control the system to follow the desired trajectory without using any linguistic information.



(a) Tracking response



(b) The state trajectory of the phase plane.



(c) Tracking error

Figure 3. Simulation results with the initial condition $\boldsymbol{x} = (2,-2)^T$

VI. CONCLUSIONS

In this paper, a sliding-mode-based fuzzy CMAC is proposed for the trajectory tracking of unknown nonlinear dynamics. When matching with the model occurs, the overall control system is equivalent to a stable dynamic system. The bounds of the fuzzy modeling error are estimated adaptively using a learning algorithm and the global asymptotic stability of the algorithm is established via Lyapunov function. The overall robust adaptive scheme is shown to guarantee that the output tracking error can converge to a residual set ultimately.

ACKNOWLEDGMENT

The financial support for this research from the National Science Council of Taiwan, R.O.C. under NSC94-2213-E-157-002 is gratefully acknowledged

REFERENCES

[1] R. M. Sanner and J.-J. E. Slotine, “Gaussian networks for direct adaptive control,” *IEEE Trans. Neural Networks*, vol. 3, pp. 837–863, Sept., 1992.

[2] K. S. Narendra and K. Parthasarathy, “Identification and control of dynamical systems using neural networks,” *IEEE Trans. Neural Networks*, vol. 1, pp. 4–27, March 1990.

[3] K. J. Hunt, D. Sbarbaro, R. Zbikowski, and P. J. Gawthrop, “Neural networks for control systems-A survey,” *Automatica*, vol. 28, pp. 1083–1112, 1992.

[4] J. S. Albus, “A new approach to manipulator control: The cerebellar model articulation controller (CMAC),” *Trans. ASME J. Dyn. Syst., Meas., Contr,* , vol. 97, pp. 220–227, Sept. 1975.

[5] J. S. Albus, “Data storage in the cerebellar model articulation controller (CMAC),” *Trans. ASME J. Dyn. Syst., Meas., Contr,* , vol. 97, pp. 228–233, 1975.

[6] S. H. Lane, D. A. Handelman, and J. J. Gelfand, “Theory and development of higher-order CMAC neural networks,” *IEEE Contr. Syst. Mag*., vol. 12, pp. 23–30, 1992.

[7] H. Shiraishi, S. L. Ipri, and D. D. Cho, “CMAC neural network controller for fuel-injection systems,” IEEE Trans. Contr. Sys. Technology, vol. 3, pp. 32–38, 1995.

[8] K. S. Hwang and C. S. Lin, “Smooth trajectory tracking of three-link robot: A self-organizing CMAC approach,” *IEEE Trans. Syst., Man, Cybern. B*, vol. 28, pp. 680-692, 1998.

[9] A. L. Kun and W. T. Miller, “Control of variable speed gaits for a biped robot,” *IEEE Robot. Automat*., vol. 6, pp. 19-29, 1999.

[10] C.-H. Lee and C.-C. Teng, “Identification and control of dynamic systems using recurrent fuzzy neural network,” *IEEE Trans. on Fuzzy Systems*, vol. 8, pp. 349-366, 2000.

[11] L.X.Wang, Stable Adaptive fuzzy Control of Nonlinear Systems, *IEEE Trans. Fuzzy Systems*, vol. 1, pp. 146-155, 1993.

[12] J.J.E. Slotine and J.A. Coetsee, “Adaptive Sliding Controller Synthesis for Non-linear Systems,” *Int. J. Control*, vol. 43, pp. 1631-1651, 1986.

[13] V.I. Utkin, “Variable Structure Systems with Sliding Modes,” *IEEE Trans. Automat. Control*, vol. 22, pp. 212-222, 1977.

[14] L. X. Wang, *A course in fuzzy systems and control*, Prentic-Hall, 1997