ROBUST ADAPTIVE FUZZY CMAC CONTROL FOR UNKNOWN SYSTEMS

Ter-Feng Wu, Pu-Sheng Tsai, Fan-Ren Chang

Dept. of Electrical Engineering, National Taiwan University, Taipei, Taiwan

Abstract: This work presents an integrated robust adaptive control scheme that merges the fuzzy control algorithm with the Cerebellar Model Arithmetic Control (CMAC) for unknown systems. The presented adaptation mechanism is used to tune the weight parameters in the CMAC, such that a given ideal stable controller will be best approximated without prior off-line learning phase required. A robust controller is appended to compensate the approximation error of fuzzy CMAC for improving the robustness. Based on the Lyapunov stability analysis the tracking stability can be guaranteed. Demonstrative examples show that the performance of the proposed control schemes is satisfied.

Keywords: CMAC, Fuzzy control, Adaptive control, Robust control

1. INTRODUCTION

Controller design can typically be roughly classified into the following two categories- (a) model-based orientation, (b) model-free or partially known orientation. Various control theories in the former category, based on the mathematical model of the controlled systems, have successfully been applied to a large class of control systems (SLOTINE and LI, 1991; GOODWIN, *et al.*, 2001). However, for complicated even in unknown systems, conventional methods may require much time to schedule their model. By contrast, the arising intelligent control methodologies provide an advisable solution based on human faculty of heuristics or learning (KIM and LEWIS, 2000; PENG and WOO, 2002; TAKAGI and SUGENO, 1985; LIN and CHIANG, 1997). Although this is an alternative way to handle ill-modeled systems, extensive knowledge and experience is required. Two popular areas of research - fuzzy control and neural networks - are generally applied to the preceding control issues (JANG, *et al.*, 1997).

Fuzzy control is a powerful control technique for representing human experience to control nonlinear and complicated processes (BERSTECHER, *et al.*, 2001). However, some difficulties in capturing the input-output membership functions exist, and especially in capturing the output membership function related to a control action. Accordingly, the membership functions should be automatically adjusted to compensate for the system uncertainty and the model approximated error when to adopt Fuzzy control scheme alone.

The Cerebellar Model Arithmetic Control (CMAC) is one type of neural network control (ALBUS, 1975). The main advantage is its effective architecture and the general capacity for operation using table-look-up. From the perspective of practical implementation, can be easily coded using a microcontroller or FPGA/CPLD when the weight parameters of CMAC are determined. Two important issues related to CMAC parameters are the partitioning of the input variables and the determination of weight variables. The input variable partition has not been systematically studied as yet. Clearly, fine partitioning will yield good precision. The first question concerns the existence of a coarse but acceptable partition that does not increase the cost of realization or the computational load, and keeps the system performance independent of the partitioning. The second question regards how to determine properly the weights of CMAC. In general, the weight parameters can be best determined by an off-line learning algorithm. Therefore, a high computational burden is required to achieve the desired accuracy. However, some pre-work on real-time CMAC control without the learning phase has been performed to overcome this shortcoming (ALBUS, 1975; COMMURI and LEWIS, 1995). For example, initial weight values are determined beforehand, according to the sliding mode control, to meet a real-time control need. Furthermore, the optimal control design can also be applied to the CMAC neural network with no learning requirement (KIM and LEWIS, 2000).

The purposes of this study were to merge the fuzzy control algorithm with the CMAC neural network, so as to construct a integrated robust adaptive fuzzy CMAC control system. Make it have the following several characteristics: (1) only two input variables for the CMAC to simplify the control design procedure, (2) no prior off-line learning phase required, (3) improving the robustness enough for resisting the presence of modeling uncertainties and external disturbance, (4) the system tracking stability and the error convergence can be guaranteed.

2. STRUCTURE OF FUZZY CMAC CONTROLLER

2.1 Basic CMAC

In this study, it is assumed that the plant has only one control input and all its state variables are available. Thus, it is considered a CMAC function as:

$$y_{CMAC} = \mathbf{F}(\mathbf{s}) \tag{1}$$

where $\mathbf{F} : \mathbf{R}^{L} \to \mathbf{R}^{1}$ is a nonlinear function. Using the following input-output mappings

$$\mathbf{G} : \mathbf{S} \Rightarrow \mathbf{A} \tag{2}$$

$$P : \mathbf{A} \Rightarrow y_{CMAC} \tag{3}$$

where $\mathbf{S}$ is a continuous L-dimensional input space; $\mathbf{A}$ is an M-dimensional association space, and u is a scalar output. Function $\mathbf{G}(\mathbf{s})$ transforms each point $\mathbf{s}$ in the input space to an association vector $\mathbf{a} = \mathbf{G}(\mathbf{s}) \in \mathbf{A}$, which comprises M constant value "1". The function $P(\mathbf{a})$ projects the association vector $\mathbf{a}$ onto a vector of weights $\mathbf{w}$ to yield a scalar output y_{CMAC}

$$y_{CMAC} = P(\mathbf{a}) = \mathbf{a}^T \mathbf{w} \tag{4}$$

Figure 1 illustrates two state variables of CMAC, assuming that the variation of interest of each state variable is divided into two regions. Variable s_1 is divided into A and B, and variable s_2 is divided into a and b. Notation Aa, Ab, Ba and Bb denote hypercubes. Shifting by a small interval yields different hypercubes. For example, C and D in the second row of s_1, c and d in the second column of s_2 are potential shifted regions, yielding hypercubes Cc, Cd, Dc and Dd. Such shifting further yields Ee, Ef, Fe and Ff. Three layers of hypercubes have 12

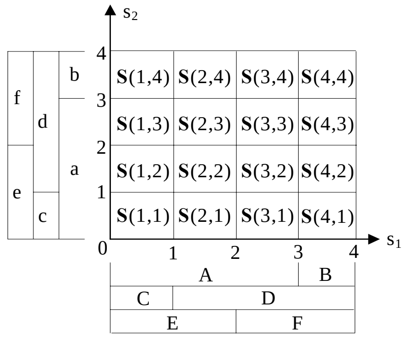

Fig. 1. Hybercubes example of two input variables of CMAC.

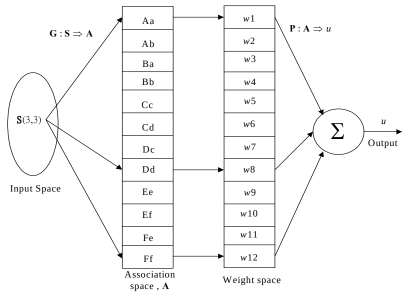

Fig. 2. Example of the mapping of CMAC.

elements which forms an assembly vector, $\mathbf{a}_{m,n}^T \triangleq$ [Aa Ab Ba Bb Cc Cd Dc Dd Ee Ef Fe Ff].

Additionally, the continuous variables are divided into 16 blocks. Block (m,n), a continuous input subspace was marked "**S**(m,n)" quantifying and covering the regions of variation, $(m-1) \le s_1 \le m$ and $(n-1) \le s_2 \le n$. If (s_1 , s_2) = (0.1,0.5) ∈ S(1,1), then Aa =1, Cc =1 and Ee = 1 with $\mathbf{a}_{1,1}^T$ = [1 0 0 0 1 0 0 0 1 0 0 0] . Clearly, the function **G** maps the input space **S** to an association space **A**. Figure 2 shows an example of the input (s_1 , s_2) = (2.2, 2.5) with a scalar output u.

2.2 Fuzzy CMAC controller

Suppose that a fuzzy system has N fuzzy rules, each of which has two input variables s_1, s_2.

$$R^{(i)} : \text{IF } s_1 \text{ is } F_1^i \text{ and } s_2 \text{ is } F_2^i \text{ THEN } u \text{ is } \mathbf{a}_i^T \mathbf{w} \tag{5}$$

where the THEN-part is extracted from the conventional CMAC above. For instance, as for the above structure, the size of the association vector $\mathbf{a}_i, i = 1, 2, \cdots, N$ as 12×1 is provided with reference to the weight vector $\mathbf{w} = [w_1\ w_2\ \cdots w_{12}]^T$. Defuzzification yields the output y_{FCMAC} as

$$y_{FCMAC} = \frac{v_1 \mathbf{a}_1^T \mathbf{w} + \cdots + v_N \mathbf{a}_N^T \mathbf{w}}{v_1 + \cdots + v_N} = \frac{\sum_{i=1}^{N} \mathbf{a}_i^T \mathbf{w} \left[\prod_{k=1}^{2} \mu_{F_k^i}(s_k) \right]}{\sum_{i=1}^{N} \left[\prod_{k=1}^{2} \mu_{F_k^i}(s_k) \right]} \tag{6}$$

where $v_i = \mu_{F_1^i}(s_1)\cdot\mu_{F_2^i}(s_2),\ i = 1,2,\cdots,N$. Define the following nonlinear mapping

$$g_i(\mathbf{s}) = \frac{\prod_{k=1}^{2}\mu_{F_k^i}(s_k)}{\sum_{i=1}^{N}\left[\prod_{k=1}^{2}\mu_{F_k^i}(s_k)\right]} \tag{7}$$

The following parametric form of fuzzy CMAC is then

$$y_{FCMAC} = u_{FCMAC}(s \mid w) = \mathbf{GAw} \tag{8}$$

where

$$\mathbf{G} = \left[g_1\ g_2 \cdots g_N\right] \tag{9}$$

$$\mathbf{A} = \begin{bmatrix}\mathbf{a}_1^T\\ \mathbf{a}_2^T\\ \vdots\\ \mathbf{a}_N^T\end{bmatrix} \tag{10}$$

Matrix **A** (determined by CMAC) and **G(s)** (IF-part of fuzzy rules) are typically fixed, but the weight vector **w** is adjustable herein.

The determination of the rules (5) of this fuzzy system is now explicated. The fuzzy system is first developed as

$$IF\quad s_1 = P\quad and\quad s_2 = P,\quad THEN\quad u = \mathbf{a}_1^T\mathbf{w} \tag{11}$$

$$IF\quad s_1 = N\quad and\quad s_2 = P,\quad THEN\quad u = \mathbf{a}_2^T\mathbf{w} \tag{12}$$

$$IF\quad s_1 = N\quad and\quad s_2 = N,\quad THEN\quad u = \mathbf{a}_3^T\mathbf{w} \tag{13}$$

$$IF\quad s_1 = P\quad and\quad s_2 = N,\quad THEN\quad u = \mathbf{a}_4^T\mathbf{w} \tag{14}$$

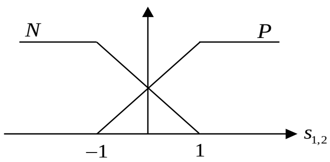

Fig. 3. Membership functions of $s_{1,2}$.

Figure 3 shows the membership functions of fuzzy sets P (Positive) and N (Negative). Figure 4 depicts the relationship between the fuzzy sets and CMAC. Matrix **A** is given by

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1\\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1\\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{15}$$

based on Fig. 4. The THEN-part of the fuzzy rules (11)-(14) is derived from Table 1, where $\mathbf{a}_i^T, i = 1,2,3,4$ is taken the logical 'or' operation for each distinct class in CMAC, to include the information in the fuzzy rules. For example, the $(s_1, s_2) = (\mathrm{P,P})$ class, four hypercubes should be considered in fuzzy rule (11); therefore $\mathbf{a}_1^T = [111100011111]$ for this rule.

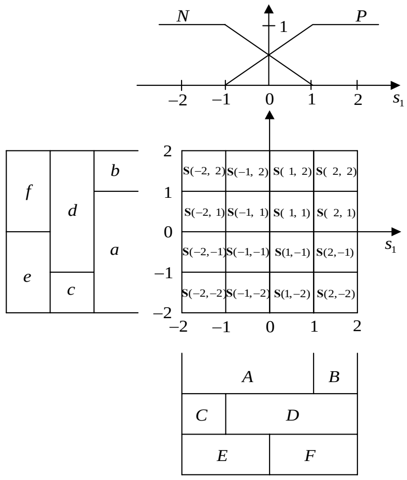

Fig. 4. Relationship between fuzzy sets and CMAC.

Table 1 Relationship between fuzzy rules and CMAC.

Continuous Input (s_1, s_2)	[Aa,Ab,Ba,Bb,Cc,Cd,Dc,Dd,Ee,Ef,Fe,Ff]		
Class	Quantif-ication	Hypercubes	$\mathbf{a}_i^T, i = 1,2,3,4$
(P,P)	S(-1,-1)	[100000011000]	[111100011111]
	S(-1, 1)	[100000010100]	
	S(-1, 2)	[010000010100]	
	S(1,-1)	[100000010010]	
	S(1, 1)	[100000010001]	
	S(1, 2)	[010000010001]	
	S(2,-1)	[001000010010]	
	S(2, 1)	[001000010001]	
	S(2, 2)	[000100010001]	
(N,P)	:	:	[110001011111]
(N,N)	:	:	[100011111111]
(P,N)	:	:	[101100111111]

3. ROBUST ADAPTIVE FUZZY CMAC CONTROLLER DESIGN

Consider the following a nth-order system

$$x^{(n)} = f(x, \dot{x}, \cdots, x^{(n-1)}) + bu \tag{16}$$

$$y = x \tag{17}$$

where f is an unknown function; b is a given nonzero gain constant; u is the system input; y is the system output, and $\mathbf{x} = \left[x_1, x_2, \cdots, x_n\right]^T = \left[x, \dot{x}, \cdots, x^{(n-1)}\right]^T$ is the state vector. The control goal is to design an adaptive fuzzy CMAC controller u such that the system output y can follow a desired output y_d where the desired output y_d and its derivatives are given and bounded. Define the error vector as

$$\mathbf{e} = \left[e_1, e_2, \cdots, e_n\right]^T = \left[e, \dot{e}, \cdots, e^{(n-1)}\right]^T \tag{18}$$

where $e_i = (y_d - y)^{(i-1)},\ i = 1,2,\cdots,n$. The following ideal control law under a given vector $\mathbf{k} = \left[k_n, k_{n-1}, \cdots, k_1\right]^T$ with positive constant elements $k_i, i = 1,2,\cdots,n$ can be easily found.

$$u^* = b^{-1}\left(-f + y_d^{(n)} + \mathbf{k}^T\mathbf{e}\right) \tag{19}$$

Applying the above control law to system (16) and suitably selecting **k** yields the error dynamic

$$e^{(n)} + k_1 e^{(n-1)} + \cdots + k_n e = 0 \tag{20}$$

such that all roots of the characteristic polynomial $h(\lambda) = \lambda^n + k_1\lambda^{n-1} + \cdots + k_n$ are in the open left half of the complicated plane. This result shows that the tracking error asymptotically approaches zero given a desired trajectory. Unfortunately, the function *f* and the gain *b* are not generally known exactly. Hence, the ideal control law (19) cannot be realized in practical applications.

For the system output can follow a given desired trajectory, a fuzzy CMAC controller will be examined to perform the control task. According to the approximation theory, we use the fuzzy CMAC controller in (6) to rewrite the ideal control law as

$$u^* = u_{FCMAC}(\mathbf{s}|\mathbf{w}^*) + \varepsilon \tag{21}$$

where $\mathbf{w}^*$ is the optimal weight vector, and ε is the approximation error, satisfying $|\varepsilon| \le D$ with a smallest positive constant *D*. It is worth noting that there are often uncertainties in *f*, and hence, the optimal weight vector w^* and approximation error ε may be unknown. In this case, we modify the control law as

$$u = u_{FCMAC}(\mathbf{s}|\hat{\mathbf{w}}) + u_r \tag{22}$$

where the weight vector $\hat{\mathbf{w}}$ is updated on-line by the adaptive law, and the robust compensation u_r is in a variable-structure (i.e., switching) manner. Both the adapaive law and robust compensation will be introduced in the following.

To simplify the u_{FCMAC}, define a switching variable as one input of the fuzzy CMAC,

$$s_1 = \mathbf{c}^T\mathbf{e} \tag{23}$$

where $\mathbf{c} = [c_1, \cdots, c_n]^T$ is the coefficient vector, which can be derived by the concept of a sliding mode control. Define another input of fuzzy CMAC as

$$s_2 = \dot{s}_1 = \mathbf{c}^T\dot{\mathbf{e}} \tag{24}$$

The fuzzy CMAC developed above, can be employed in such a system because an n-dimensional state variable is transferred to only two input variables. Therefore, the fuzzy CMAC controller is more easily implemented than the conventional CMAC control. The robust controller must be designed to handle this approximation error and to guarantee the stability of the fuzzy CMAC control system. Additionally, the fuzzy CMAC must be constructed to approximate the ideal control (19).

To achieve this objective, system (16) can be expressed as

$$\begin{aligned} x^{(n)} &= f + b[u + u^* - u^*] \\ &= f + b[(u_{FCMAC} + u_r) - u^* + b^{-1}(-f + y_d^{(n)} + \mathbf{k}^T\mathbf{e})] \\ &= y_d^{(n)} + \mathbf{k}^T\mathbf{e} + b[u_{FCMAC} + u_r - u^*] \end{aligned} \tag{25}$$

From Eq. (18), the equivalent vector has the form

$$\dot{\mathbf{e}} = \mathbf{\Lambda e} + \mathbf{b}[u^* - u_{FCMAC} - u_r] \tag{26}$$

where

$$\mathbf{\Lambda} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 \\ -k_n & -k_{n-1} & \cdots & \cdots & \cdots & \cdots & -k_1 \end{bmatrix}, \ \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ b \end{bmatrix}.$$

Manipulating Eq. (26) yields

$$\begin{aligned} \dot{\mathbf{e}} &= \mathbf{\Lambda e} + \mathbf{b}(u^* - u_{FCMAC} - u_r + u^*_{FCMAC} - u^*_{FCMAC}) \\ &= \mathbf{\Lambda e} + \mathbf{b}(u^*_{FCMAC} - u_{FCMAC}) + \mathbf{b}\varepsilon - \mathbf{b}u_r \end{aligned} \tag{27}$$

Consider the following Lyapunov function candidate;

$$V = \frac{1}{2}\mathbf{e}^T\mathbf{P}\mathbf{e} + \frac{1}{2\gamma}\boldsymbol{\varphi}^T\boldsymbol{\varphi} \tag{28}$$

where γ is a constant positive convergence rate, $\boldsymbol{\varphi} \triangleq \mathbf{w}^* - \mathbf{w}$, and **P** is an $n \times n$ symmetric positive definite matrix that satisfies the following Lyapunov equation.

$$\mathbf{\Lambda}^T\mathbf{P} + \mathbf{P\Lambda} = -\mathbf{Q} \tag{29}$$

Q is an $n \times n$ positive definite matrix. Differentiating *V* with respect to time yields

$$\begin{aligned} \dot{V} &= \frac{1}{2}\dot{\mathbf{e}}^T\mathbf{P}\mathbf{e} + \frac{1}{2}\mathbf{e}^T\mathbf{P}\dot{\mathbf{e}} + \frac{1}{2\gamma}\dot{\boldsymbol{\varphi}}^T\boldsymbol{\varphi} + \frac{1}{2\gamma}\boldsymbol{\varphi}^T\dot{\boldsymbol{\varphi}} \\ &= \frac{1}{2}\mathbf{e}^T(\mathbf{\Lambda}^T\mathbf{P} + \mathbf{P\Lambda})\mathbf{e} + \mathbf{e}^T\mathbf{Pb}(u^*_{FCMAC} - u_{FCMAC} + \varepsilon - u_r) + \frac{1}{\gamma}\dot{\boldsymbol{\varphi}}^T\boldsymbol{\varphi} \\ &= -\frac{1}{2}\mathbf{e}^T\mathbf{Q}\mathbf{e} + \mathbf{e}^T\mathbf{PbGA}\boldsymbol{\varphi} + \frac{1}{\gamma}\dot{\boldsymbol{\varphi}}^T\boldsymbol{\varphi} + \mathbf{e}^T\mathbf{Pb}\varepsilon - \mathbf{e}^T\mathbf{Pb}u_r \\ &= -\frac{1}{2}\mathbf{e}^T\mathbf{Q}\mathbf{e} + \frac{1}{\gamma}(\gamma\mathbf{e}^T\mathbf{PbGA} + \dot{\boldsymbol{\varphi}}^T)\boldsymbol{\varphi} - \mathbf{e}^T\mathbf{Pb}(u_r - \varepsilon) \end{aligned} \tag{30}$$

From (30), we select the adaptation law for the unknown weight *w* as

$$\dot{\boldsymbol{\varphi}} = -\dot{\mathbf{w}} = -(\gamma\mathbf{e}^T\mathbf{PbGA})^T \tag{31}$$

and the robust controller as

$$u_r = D \cdot \operatorname{sgn}(\mathbf{e}^T\mathbf{Pb}) \tag{32}$$

such that we have $\dot{V} \le -\frac{1}{2}\mathbf{e}^T\mathbf{Q}\mathbf{e}$, which indicates that the function *V* is non-increasing. So we have $V \in \mathcal{L}_\infty$ and hence meet $\mathbf{e}, \boldsymbol{\varphi} \in \mathcal{L}_\infty$. From (27), we have $\dot{\mathbf{e}} \in \mathcal{L}_\infty$ further. By integrating the inequality $\dot{V}$ above, we can conclude that $\mathbf{e} \in \mathcal{L}_2$. Using the properties $\dot{\mathbf{e}} \in \mathcal{L}_\infty$, $\mathbf{e} \in \mathcal{L}_2$ and applying **Barbălat** lemma, it is guaranteed that the tracking error **e** converges to zero asymptotically (i.e., $\mathbf{e}(t) \to 0$ as

$t \to \infty$). The closed-loop systems subject to the adaptive fuzzy CMAC control scheme are illustrated in Fig. 5

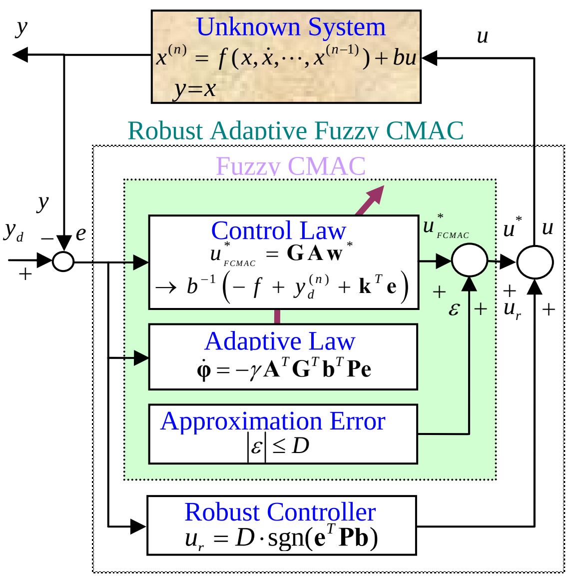

Fig. 5. The closed-loop system architecture.

4. ILLUSTRATIVE EXAMPLES

This section applies the proposed robust adaptive fuzzy CMAC to control a one-link rigid robotic manipulator, the Duffing forced-oscillation system and a three-order process.

4.1 One-link rigid robotic manipulator

The dynamic model of this system is given in (LIN and PENG, 2004).

$$ml^2\ddot{q} + b\dot{q} + ml\, g_v \cos(q) = u \qquad (33)$$

where l is the link length; m is the mass, and q is the angular position under initial conditions $q(0) = 0.2$ and $\dot{q}(0) = 0$. Let the state variables be $x_1 = q$ and $x_2 = \dot{q}$; the model (33) can be rewritten as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} (f + bu) \qquad (34)$$

where $f = \left(-1/ml^2\right) x_2 - \left(g_v/l\right)\cos(x_1)$ and $b = (1/ml^2)$. The parameters in Eq. (34) are $m = l = g_v = 1$. In many applications, the system parameters, such as the friction constant b, are usually unknown, so it is preferred here that the adaptive fuzzy CMAC control scheme is applied.

$$\begin{bmatrix} \dot{x}_{d1} \\ \dot{x}_{d2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -16 & -8 \end{bmatrix} \begin{bmatrix} x_{d1} \\ x_{d2} \end{bmatrix} + \begin{bmatrix} 0 \\ 16 \end{bmatrix} r(t) \qquad (35)$$

under the initial condition $\left[x_{d1}\ x_{d2}\right] = \left[0\ \ 0\right]$ and $r(t)$ is a periodic rectangular input with period $T = 9$ sec.

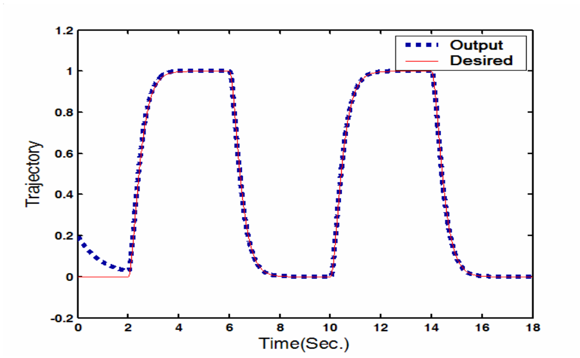

Fig. 6. Output-desired trajectories of the one-link rigid robotic manipulator.

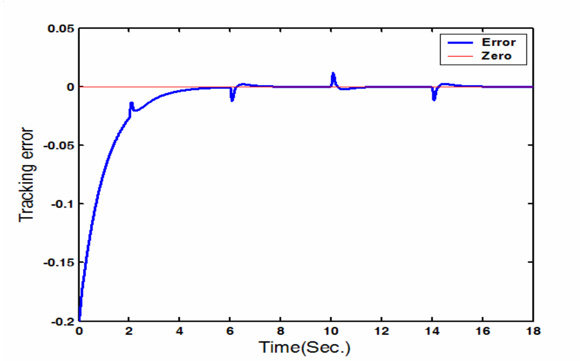

Fig. 7. Tracking error.

The parameters of the controller are $\gamma = 10$, $D = 1$, $\mathbf{c} = [1, 0]^T$, $\mathbf{k} = [1, 2]^T$ and $\mathbf{P} = \begin{bmatrix} 15 & 5 \\ 5 & 5 \end{bmatrix}$. Figure 6 and 7 show the output-desired trajectories and tracking error, respectively.

4.2 Duffing forced-oscillation system

The dynamic model for this example, is given in (JIANG, 2002)

$$\ddot{x} + 0.1\dot{x} + x^3 - 12\cos(t) = u \qquad (36)$$

This dynamic equation can be written as

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -0.1x_2 - x_1^3 + 12\cos(t) + u \end{aligned} \qquad (37)$$

where the state variables are $x_1 = x$ and $x_2 = \dot{x}$. The proposed controller is now applied to cause the state x_1 to follow the desired trajectory $x_d = \sin(t)$ under the initial condition $x_1(0) = x_2(0) = 2$. The parameters of controller are $\gamma = 30$, $D = 3$, $\mathbf{c} = [1, 0]^T$, $\mathbf{k} = [1, 2]^T$ and $\mathbf{P} = \begin{bmatrix} 15 & 5 \\ 5 & 5 \end{bmatrix}$, respectively. Figure 8 and 9 demonstrate the output-desired trajectories and tracking error, respectively.

4.3 Three-order process

The dynamic equation of this system is taken from in

$$\dddot{x} + 2.14\ddot{x} + 1.276\dot{x} + 0.228x = 4.228u \qquad (38)$$

Its state space model is

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -0.228 & -1.276 & -2.14 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 4.228 \end{bmatrix} u \quad (39)$$

Clearly, this process is stable. The proposed controller can handle a system of order greater than three: three state variables are integrated into only one switching variable according to $s_1 = 3e_1 + 2e_2 + e_3$; moreover, $s_2 = 3\dot{e}_1 + 2\dot{e}_2 + \dot{e}_3$. The control task is to track a square reference. The parameters of the controller are selected as $D = 3$, $\gamma = 0.1$, $\mathbf{c} = [3,2,1]^T$, $\mathbf{k} = [1,2,3]^T$ and

$$\mathbf{P} = \begin{bmatrix} 23 & 21 & 5 \\ 21 & 46 & 13 \\ 5 & 13 & 6 \end{bmatrix}. \quad (40)$$

Figure 10 depicts the output-desired trajectory. From simulation results given above, can entirely reveal the simplification and effectiveness of the proposed control scheme.

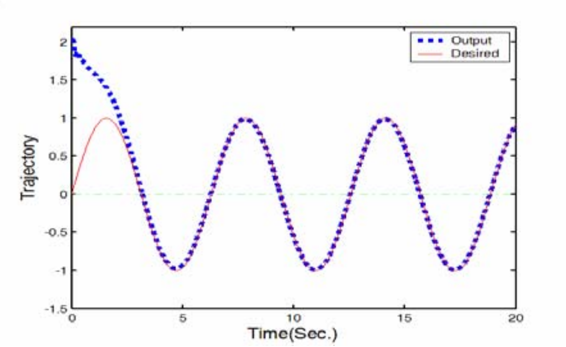

Fig. 8. Output-desired trajectory of the Duffing sys.

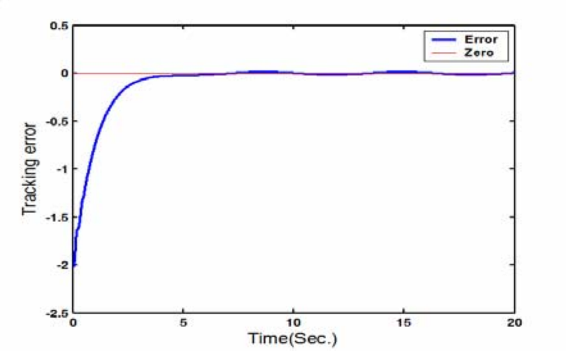

Fig. 9. Tracking error.

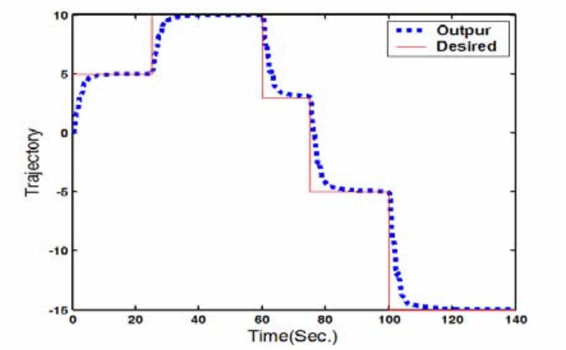

Fig. 10. Output trajectory of a third-order system with distinct reference setpoint.

5. CONCLUSIONS

A robust adaptive controller, which integrates a fuzzy algorithm and a conventional CMAC, was proposed for unknown systems. The conventional CMAC is incorporated in the THEN-part of the fuzzy rules, and is used to approximate an ideal controller. The robust controller is designed based on the residual part of the approximation error. To simplify the partition space, the input variables are integrated into only one switching variable by the concept of sliding mode control. The developed architecture can therefore be applied to higher dimensional control systems. The proposed way to design the control procedure is easier than the conventional design scheme and requires no preliminary off-line learning phase. Simulation results show that the proposed robust adaptive fuzzy CMAC control performs satisfactorily even in the presence of modeling uncertainties and external disturbances.

REFERENCES

ALBUS, J.S. (1975). A new approach to manipulator control: The cerebellar model articulation controller (CMAC), *Trans. ASME, J. Dyn. Syst., Meas., Contr.*, **97**, pp. 220-227.

BERSTECHER R. G., PALM R., and UNBEHAUEN H.D. (2001). An adaptive fuzzy sliding-mode controller, *IEEE Trans. On Industrial Electronics*, **48**, (1), pp. 18-31.

COMMURI, S., and LEWIS F. L. (1995). Control of unknown nonlinear dynamical systems using CMAC neural networks: structure, stability, and passivity, *IEEE International Conference on Neural Networks*, pp. 123-129.

GOODWIN, G.C., GRAEBE, S.F., and SALGADO, M.E. (2001). *Control system design*, Prentice-Hall.

JANG, J. -S. R., SUN, C. –T., and MIZUTANI, E. (1997). *Neuro-fuzzy and soft computing* Prentice Hall.

JIANG, Z.P. (2002). Advanced feedback control of the chaotic Duffing equation, *IEEE Trans. on Circuits and Systems- I* , **49**, (2), pp. 244-249.

KIM, Y. H., and LEWIS, F. L. (2000). Optimal design of CMAC neural-network controller for robot manipulators, *IEEE Trans. On System, Man and Cybernetics, Part C,* **30**, (1), pp. 22-31.

LIN, C.M., and PENG, Y.F. (2004). Adaptive CMAC-based supervisory control for uncertain nonlinear systems, *IEEE Trans. On Systems, Man, and Cybernetics*, **34**, (2), pp. 1248-1260.

LIN, C.S., and CHIANG, C.T. (1997). Learning convergence of CMAC technique, *IEEE Trans. On Neural Networks*, **8**, (6), pp. 1281-1292.

PENG, L. and WOO, P.Y. (2002). Neural-fuzzy control system for robotic manipulators, *IEEE Control Systems Magazine*, **22**, (1), pp. 53-63.

SLOTINE, J.J.E. and LI, W. (1991). *Applied nonlinear control*, Prentice-Hall.

TAKAGI, T., and SUGENO, M. (1985). Fuzzy identification of systems and its applications to modeling and control, *IEEE Trans. on Systems, Man and Cybernetics*, **15**, (1), pp. 116-132