# Fuzzy Sliding Mode Controller with CMAC Application Rigid Robot Manipulators

Lon-Chen Hung* and Hung-Yuan Chung**
Department of Electrical Engineering, National Central University,
Chung-Li, Tao-Yuan, 320, Taiwan, R.O.C
TEL: 886-3-4227151 ext 4475
Fax: 886-3-425-5830

*Abstract:* - In this paper to address a signed distance fuzzy sliding-mode control (SDFSMC) architecture which integrates cerebellar model articulation controller (CMAC). The proposed control consists of the SDFSMC and a feedforward compensation with modify CMAC network which to control the dynamics of the nonlinear systems with unknown structured nonlinearities without requiring a priori knowledge of the system parameter values. The effectiveness of the proposed control scheme is verified with an application the rigid robot manipulators.

*Key-Words:* - fuzzy control, sliding mode, CMAC

## 1 Introduction

In recent years, there have been attempts to design the FLC based on the sliding mode control (SMC) law [1-3]. They have shown that the boundary layer can reached in finite time and the ultimate boundedness of states is obtained asymptotically even though there exist some disturbance of dynamic uncertainties of the system. Palm showed that the analogy between a simple FLC and sliding mode controller with a boundary layer [4]. Hwang et al. proposed a fuzzy sliding mode controller and opened a way of designing an FLC for higher order nonlinear system [5]. The sliding mode control provides a good performance in tracking of some nonlinear systems. Nevertheless, a notorious characteristic of sliding mode control approach is the discontinuity around the switching hyperplane, that means some of the state variable are vibrant. One of the methods to cope with the problem is to utilize a feedforward compensator to offset unpredictable affect of system uncertainties.

That researches also show that the FSMC has the following advantages [6]: (1) It can well control most of complex systems without knowing their exact mathematical models. (2) The dynamic behavior of the controlled system can be approximately dominated by a fuzzified sliding surface. (3) FMSC can not only increase the robustness to system uncertainties but also decrease the chattering phenomenon in the conventional sliding mode controller. The fuzzy controller of second-order systems is designed on a phase plane built by error e and change of error that are produced from the states $x$ and $\dot{x}$. To reduce the fuzzy rules in the fuzzy controller, various method such as signed distance fuzzy sliding-mode control in Fig. 1. By this way [7-10], the number of fuzzy rules can be greatly reduced and tuning, and tuning of rules is more easily. Although a DSFMC approach without a knowledge of robot dynamics may be promising. However, the determination of the SDFSMC is also increased quickly as the number of sliding function increases.

In this paper proposed the control consists of the SDFSMC and a feedforward compensation with modify CMAC network which to control the dynamics of the nonlinear systems with unknown structured nonlinearities. The valid of the proposed control scheme is confirmed with an application the rigid robot manipulators. To control a nonlinear plant by proposed SDFSMC which integrates CMAC in Fig. 2. It is important to note that this approach will not be very practical due to high dimensionality of input–output space.

The CMAC is a perceptron-like associative memory that performs a table look-up of a nonlinear function over a particular region of the function space. The CMAC network has the capability to learn an unknown nonlinear mapping given the input-output set and to approximate the nonlinear function. The research results up to present have shown that the CMAC network has great potential in real-time adaptive control applications [11-12].

The rest of the paper is divided into five sections. In Section 2, the systems are described. In Section 3, the SDFSMC with CMAC controller is presented. In Section 4, the proposed controller is used to control the rigid robot manipulators. Finally, we conclude with Section 5.

## 2 System description

Consider the following *n*th-order nonlinear systems:

$$x^{(n)} = f(x,\dot{x},...,x^{(n-1)}) + g(x,\dot{x},...,x^{(n-1)})u + d(t)$$
$$y = x \tag{1}$$

where $f$ and $g$ are unknown continuous functions, $d(t)$ is the unknown external disturbance, $u \in \Re$ and $y \in \Re$ are the control input and the output of the system, respectively, and $\boldsymbol{x} = (x,\dot{x},...,x^{(n-1)})^T \in \Re$ is the state vector of the system, which is assumed to be measurable. Without loss of generality, we assume that $g > 0$ and $d(t)$ is bounded as $|d(t)| \le D$. Define the state vector to track a desired state $\boldsymbol{x} = (x_d,\dot{x}_d,...,x_d^{(n-1)})^T \in \Re$. Let $\boldsymbol{e} = \boldsymbol{x}\text{-}\boldsymbol{x}_d$ be tracking error, and let

$$\boldsymbol{e} = \boldsymbol{x}(t) - \boldsymbol{x}_d(t) = (e,\dot{e},...,e^{(n-1)})^T \tag{2}$$

be the error vector.

## 3 The FSMC with CMAC controller

### 3.1 Signed distance fuzzy sliding-mode control

In this section, the idea of named the signed distance is used, and the feasibility of the present approach will be demonstrated. The switching line is defined by:

$$s: \dot{e} + c_1 e = 0 \tag{3}$$

$$h: \{(e,\dot{e}) \mid S(e) = \dot{e} + ce = 0\} \tag{4}$$

First, we introduce a new variable called the signed distance. Let $A(e,\dot{e})$ be the intersection point of the switching line and the line perpendicular to the switching line from an operating point $B(e,\dot{e})$, as illustrated in Fig. 3. Next, $d$ is evaluated. The distance between $A(e,\dot{e})$ and $B(e,\dot{e})$ can be given by the following expression [13]:

$$d = \sqrt{[(e-e_1)^2 + (\dot{e}-\dot{e}_1)^2]} = \frac{|\dot{e}_1 + c_1 e_1|}{\sqrt{1+c_1^2}} \tag{5}$$

Without loss of generality, Eq. (5) can be rewritten as follows:

$$d = \frac{|\dot{e} + c_1 e|}{\sqrt{1+c_1^2}} \tag{6}$$

The signed distance $d_s$ is defined for an arbitrary point $B(e,\dot{e})$ as follows:

$$d_s = \mathrm{sgn}(s)\frac{|\dot{e} + c_1 e|}{\sqrt{1+c_1^2}} = \frac{\dot{e} + c_1 e}{\sqrt{1+c_1^2}} = \frac{s}{\sqrt{1+c_1^2}} \tag{7}$$

where

$$sgn(s) = \begin{cases} 1 & for \quad s > 0 \\ -1 & for \quad s < 0 \end{cases} \tag{8}$$

Now, we choose a Lyapunov function:

$$V = \frac{1}{2} d_s^2 \tag{9}$$

Then

$$\dot{V} = d_s \dot{d}_s = \frac{s\dot{s}}{1+c_1^2} \tag{10}$$

Hence, it is seen that if $s>0$, then $d_s>0$, decreasing $u$ will make $s\dot{s}$ decrease so that $\dot{V}<0$ and that if $s<0$, then $d_s<0$, increasing $u_1$ will make $s\dot{s}$ decrease so that $\dot{V}<0$. So we can ensure that the system is asymptotically stable. From the above relation, we can conclude that:

$$u_1 \propto -d_s \tag{11}$$

Hence, the fuzzy rule table can be established on a one-dimensional space of $d_s$ as shown in Table 1 instead of a two-dimensional space of $x$ and $\dot{x}$. The control action can be determined by $d_s$ only. Hence, we can easily add or modify rules for fine control. For implementation, a triangular type membership function is chosen for the aforementioned fuzzy variables, as shown in Fig. 4.

However, $d_s$ and $u_1$ are the input and output of the signed distance fuzzy logic control, respectively. The input of the proposed fuzzy controller which is a fuzzified variable of $d_s$. The output of the fuzzy controller which is the fuzzified variable of $u_1$. All the universes of discourse of $d_s$ and $u_1$ are arranged from –1 to 1.

Now, Extension to general case for DSFSMC, Consider a general n-input traditional FLC has rule of the following form:

$$R_{GN}^k: \text{If } e \text{ is } LE_k^1,...,\text{and } e^{(n-1)} \text{ is } LE_k^n \text{ then } u_1 \text{ is } LU_k$$
$$k = 1,2,...,M^n \tag{12}$$

where $M$ is the number of fuzzy sets for each fuzzy input variable and $LE_k^i$ ($i$=1,2,…,$n$) is the linguistic value taken by the process state variable $e^{(n-1)}$ in the $k$-th rule. In this case, the rule table is established on $n$-dimension space of $e,\dot{e},...,$and $e^{(n-1)}$. The number of rule $M^n$ is huge, which makes very difficult to generate reasonable control rules. Also, Define a general signed distance $D_s$ as follows:

$$D_s = \frac{e^{(n-1)} + c_{n-1}e^{(n-2)} + ... + c_2\dot{e} + c_1 e}{\sqrt{1+c_{n-1}^2 + ... + c_2^2 + c_1^2}} = \frac{S(\boldsymbol{e})}{\sqrt{\sum_{i=1}^{n} c_i^2}} \tag{13}$$

Consider the following Lyapunov function

$$\tilde{V} = \frac{1}{2}\frac{s^2(\boldsymbol{e})}{\sqrt{\sum_{i=1}^{n} c_i^2}} \tag{14}$$

To used (14) be rewritten as

$$\tilde{V} = \frac{\sqrt{\sum_{i=1}^{n} c_i^2}}{2} \bullet \frac{s^2(\boldsymbol{e})}{\sum_{i=1}^{n} c_i^2} = \frac{1}{2}\sqrt{\sum_{i=1}^{n} c_i^2}\, D_s^2(\boldsymbol{e}). \tag{15}$$

By taking the time derivative of (13), we obtain

$$D_s = \frac{(\sum_{i=1}^{n-1} c_i e^{(i)} + f(x,t) + g(x,t)u + d(t) - x_d^{(n)})}{\sqrt{\sum_{i=1}^{n} c_i^2}} \tag{16}$$

Differentiating (15) with respect to time. $\dot{\tilde{V}}$ along the system trajectory as

$$\begin{aligned}\dot{\tilde{V}} &= \sqrt{\sum_{i=1}^{n} c_i^2} \bullet D_s \bullet \dot{D}_s \\ &= s \bullet (\sum_{i=1}^{n} c_i e^{(i)} + f(\boldsymbol{x},t) + g(\boldsymbol{x},t)\hat{u} + d(t) - x_d^{(n)}) \\ &\le -\sigma \bullet \sqrt{\sum_{i=1}^{n} c_i^2} \bullet |D_s| = -\sigma \bullet |S|\end{aligned} \tag{17}$$

which $\sigma$ is positive number. Hence, if is known, the DFLC law can be designed as:

$$u_1 = g^{-1}(-\sum_{i=1}^{n-1} c_i e^{(i)} - f(\boldsymbol{x},t) - d(t) + x_d^{(n)} - \varphi \bullet \mathrm{sgn}(S) \bullet \sigma) \tag{18}$$

where

$$\varphi = \begin{cases} 1, & if \quad s \neq 0 \\ 0, & if \quad s = 0 \end{cases} \tag{19}$$

## 3.2 CAMC controller

The second part $u_2$ of the control law $u$ a block diagram is shown in Fig. 2 for CMAC neural network algorithm, which is a nonlinear function learner. CMAC is considered essentially as a table look up algorithm. It is adaptive because it can modify the contents in the table by a learning algorithm. It also has generalization capability due to the distributed storage of information. No hash coding is needed for this application.

The CMAC can be used as a general nonlinear function approximator in Fig. 5 as follows: Given a reference nonlinear function, $f(x_1, x_2)$, where $(x_1, x_2)$ to $N_g$ memory locations (generalization size).

1. Sample the input vector, which may contain both, desired and feedback signal information.
2. Find corresponding memory locations for that input vector using a mapping algorithm, use hash coding if necessary.
3. To speed up the initial learning and to achieve better generalization, employ the generalization technique, i.e., each input vector to CMAC is mapped into a number of memory locations instead of only one memory location.
4. Response from CMAC is then the summation of the contents of these active memory locations.
5. Update the contents of the active memory locations according to the learning algorithm.
6. Repeat until the learning accuracy criterion is satisfied. Herein is the derived mathematical model of CMAC algorithm.

The value of $x_i(k)$ is 1 when the $i$th memory element is activated while it is zero when the memory element $i$ is deactivated. The allocating vector $x_i(k)$ is determined using the formula [14]

$$x_i(k) = \{s_i(k)[av_i(k) + ml(k)]\} \tag{20}$$

where $av_i(k)$ is $i$th added value at instant $k$; $ml(k)$ represents the first active memory element location at instant $k$ and $s_i(k)$ has 1 or 0 value at step time $k$. $av_i(k)$ and $s_i(k)$ can be defined as follows:

$$av_i(k) = \sum_{i=1}^{m} x_{(i-1)}(k), \quad x_0(k) = 0 \tag{21}$$

where $m$ is the memory size, and

$$s_i(k) = \begin{cases} 1 & \text{if } [\mathrm{g}_i(k) \le ml\,\mathrm{g}(k)] \text{ and } [ml(k) \le i], \\ 0 & \text{otherwise} \end{cases} \tag{22}$$

where

$$g_i(k) = av_i(k) + ml(k)$$

$$ml\,\mathrm{g}(k) = ml(k) + N_g - 1$$

$N_g$ is the generalization size:

The currently active memory location contents are modified and updated using the following conventional learning algorithm

$$w_i(k+1) = w_i(k) + \eta \times e(k), \tag{23}$$

where $0 \le \eta \le 1$ is the learning rate, which determines the rate of convergence of the weight and $e(k)$ is the tracking error which is derived from

$$e(k) = r - y \tag{24}$$

where $r$ and $y$ are the reference input and system output, respectively.

The control action $u_2$ can be determined from

$$u_2 = \sum_{i=1}^{n} w_i(k) x_i(k) \tag{25}$$

where $w_i$(k) and $x_i(k)$ are the weight or the memory content and the allocating pointer of memory element $i$, respectively.

# 4 Experiment of the Rigid Robot System

In this section, we shall demonstrate that the SDFSMC with CMAC control design is applicable to the rigid robot manipulators to verify the theoretical development.

## 4.1 The application algorithm

*Step 1*. Initialize: At the beginning, all weights stored in CMAC memory are set all weights of CMAC to small random values.

*Step 2.* Compute to control from SDFSMC: Using Eqs. (18) compute the control $u_1$. We choose the sliding vector $c_1$ and $c_2$ are 5.0 and 1.0, respective.

*Step 3.* Compute the compensate control from CMAC: Compute up as explained in (13). Each input to CMAC is quantized into $x_i(k)$ = 400 levels (resolution of joint angles is 0.05 rad). According the size $N_g$ of virtual memory space is $3.73\times10^6$. Hash coding is employed to compress into a much smaller physical memory space $m$. For all subsequent simulations, if not stated otherwise, we choose $m$ =256 and the training factor $\eta = 0.8$ for this simulation. The summation of the weights stored in shared memory cells forms a good starting for the optimisation of current control interval.

*Step 4.* Apply control to robot: Sum the estimated main control and the compensate control to form the control signal and then apply this to the robot.

*Setp 5.* Go back to step (2) until the final condition is reached.

## 4.2 Robotic dynamics

The dynamic equation for a single-joint manipulator, as shown in Fig. 7, can be expressed as follows [15]:

$$I\ddot{\theta} + B_1\dot{\theta} + \mu g\cos(\theta) = \tau \qquad (26)$$

where $I = \frac{1}{3}M_0 l^2 + Ml^2$, $\mu = l(\frac{1}{2}M_o + M)$, $M_0 = 2.823kg$ is the mass of the link, $M$ is the load, $l$ = 0.33m is the manipulator length, $B_1$ is the damping coefficient, and the center of mass is located at the middle of link. If the single link moves in a horizontal plane with input torque $\tau = -k_d\dot{\theta} - k_p(\theta - \theta_0)$ provided by the local controller, the dynamic equation in Eq. (26) can be rewritten as

$$I\ddot{\theta} + B_d\dot{\theta} + K_p(\theta - \theta_0) = 0 \qquad (27)$$

where $k_p$ and $k_d$ are constants for the local controller, $B_d = K_d + B_1$, $\theta$ is the actual joint value, and $\theta_0$ is the joint variable specified by the motion command.

The values of $B_d$ and $K_p$ in Eq. (27) can be obtained via a linear scaling of the $B$ and $K$ used in the fuzzy system for a second-order system by comparing the inertia in Eq. (27) with the load used in the fuzzy system. The scaling of $B_d (= K_d + B_1)$ and $K_p$ in some sense implicates the adjustment of $K_d$ and $K_p$ of the local controller for matching the system's operating range.

## 4.3 Experimental results

The robot manipulator for civil engineering and construction in a tunnel is given in this chapter. Fig.6 is the picture of our experimental system. The hardware of a robot arm contains: (1) the forward and back motion of X-axis, range:0~29.4 cm; (2) the swirl motion of θ-axis, range +78 ~ -78 degrees; (3) the sliding motion of R-axis, range 0~20 cm. The diagrammatic sketch of a three-axis robot arm is shown in Fig.7. Each axis includes both limited-switch and home-tested sensor switch. The home-tested sensor benefits users to return the initial position, write programs, and measure exactly. Both limited-switch and home-tested sensor switch are made of magnetic induction.

The comparison of our proposed method with the SDFSMC method without the CMAC for the same plant is shown in Figs. 8-11 respectively. We can find that our proposed method not only reduces overshoot to near zero but also maintains a small extent of the setting time and the steady error.

# 5 Conclusion

In this paper, the SDFSMC with CMAC control design has been presented. The proposed control consists of the SDFSMC and a feedforward compensation with CMAC network which to control the dynamics of the nonlinear systems with unknown structured nonlinearities without requiring a priori knowledge of the system parameter values.

The experimental results demonstrated the effects of our design. The effects included faster rising time, better ability of resisting the disturbances, minimum overshoot in the set point, more accurate steady-state error and the most important is simplified the design process.

*References:*

[1] Jacob S. Glower, Jeffrey Munighan, Design Fuzzy Controllers from a Variable Structures Standpoint, *IEEE Trans.Fuzzy Sys.*, Vol. 5. No. 1., 1997, pp. 138-144.

[2] Ji-Chang Lo, Ya-Hui Kuo, Decoupled Fuzzy Sliding-Mode Control, *IEEE Trans.Fuzzy Sys*, Vol. 6. No. 3., 1998, pp. 426-435.
[3] Byung-Jae Choi, Seong-Woo Kwak, Byung Kook Kim, Design of a single-input fuzzy logic controller and its properties, *Fuzzy Sets and Systems*, Vol. 106, 1999, pp.299-308.
[4] Y. R. Hwang and M. Tomizuka, Fuzzy smoothing algorithms for variable structure systems, *IEEE Trans. Fuzzy Syst*., Vol. 2, 1994, pp. 277-284.
[5] Shiuh-Jer Huang, Wei-Cheng Lin, Adaptive fuzzy controller with sliding surface for vehicle suspension control, *IEEE Trans. Fuzzy Syst*., Vol.11, 2003, pp.550– 559.
[6] D.P. Filev and R. R. Yager, On the analysis of fuzzy logic controllers, *Fuzzy Sets Sys*, Vol. 68, 1994, pp. 39-66.
[7] Yu, X., Man, Z., and Wu, B., Design of Fuzzy Sliding-Mode Control Systems, *Fuzzy Sets Sys*, Vol. 95, No. 3, 1998, pp. 295-306.
[8] Sung-Woo Kim, Ju-Jang Lee, Design of a fuzzy controller with fuzzy sliding surface, *Fuzzy Sets Sys*, Vol.71, 1995, pp. 359-367.
[9] Ji-Chang Lo, Ya-Hui Kuo, Decoupled Fuzzy Sliding-Mode Control, *IEEE Trans. Fuzzy Syst*, Vol. 6. No. 3., 1998, pp. 426-435.
[10] Chung-Chun Kung, Tung-Yun Kao; Ti-Hung Chen, Adaptive fuzzy sliding mode controller design, *FUZZ-IEEE'02. Proc. of the 2002 IEEE International Conf. on Fuzzy Syst.*, Vol. 1, 2002, pp. 674-679.
[11] Albus J. Brains, *behavior, and robotics*. Byte Books; 1981.
[12] Serrano FJ, Vidal AR, Roriguez A., Generalizing CMAC architecture and training *IEEE Trans. on Neural Networks*, vol. 9, 1998, pp. 1509-1514.
[13] Berstecher, R.G.; Palm, R., Unbehauen, H.D., An adaptive fuzzy sliding-mode controller," *IEEE Transactions on Industrial Electronics*, Vol. 48, 2001, pp. 18-31.
[14] Magdy M. Abdelhameed, Unnat Pinspon, Sabri Cetinkunt, Adaptive learning algorithm for Cerebellar model articulation controller, *IEEE/ASME Trans. Mechatronics*, Vol.12, 2002, pp. 859-873.
[15] K.Y. Young and C.C. Fan, An approach to simplify the learning space for robot learning control, *Fuzzy Sets and Systems*, Vol.95, 23-28, 1998.


Fig. 1. The block diagram of the SDFLC.

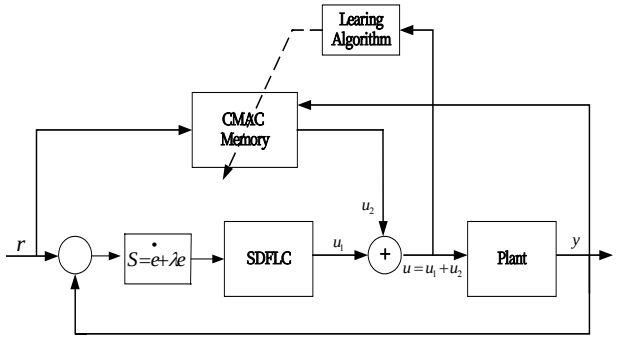


Fig. 2. The block of the FSMC with CMAC controller.

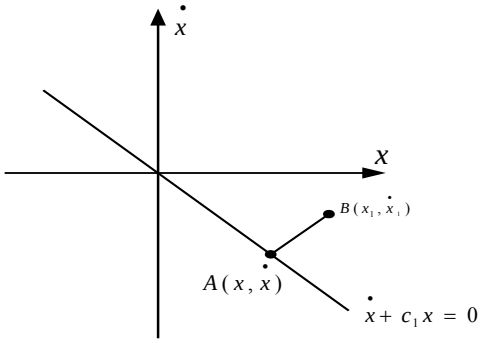


Fig. 3. Derivation of a signed distance.

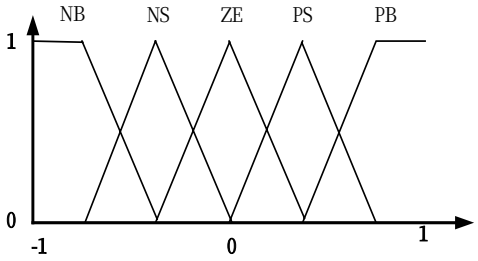


Fig. 4. Fuzzy variable of triangular type.

| $d_s$ | NB | NM | ZE | PM | PB |
|---|---|---|---|---|---|
| $u$ | PB | PM | ZE | NM | NB |

Table 1. Rule table for a signed distance fuzzy logic control.

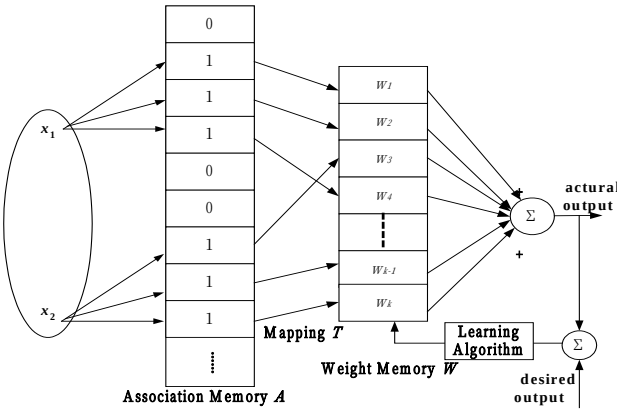


Fig. 5. Block diagram of CMAC neural network.

Fig. 6. A real robot manipulations for civil engineering and construction in a tunnel.

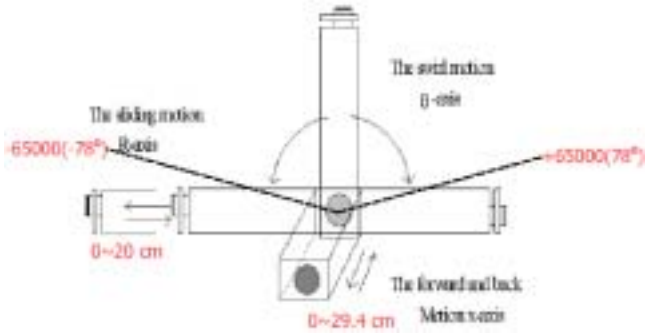


Fig.7. The diagrammatic sketch of a rigid robot arm.

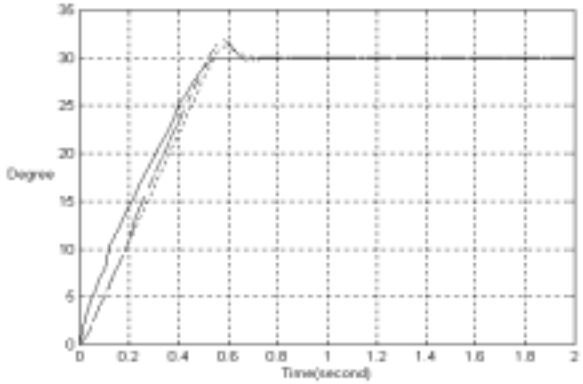


Fig. 8. The response of the position controllers for θ-axis is moved from 0 to 30 degrees. (solid line : ─ SDFSMC with CMAC, dotted line : … SDFSMC, dashed line : -- FSMC )

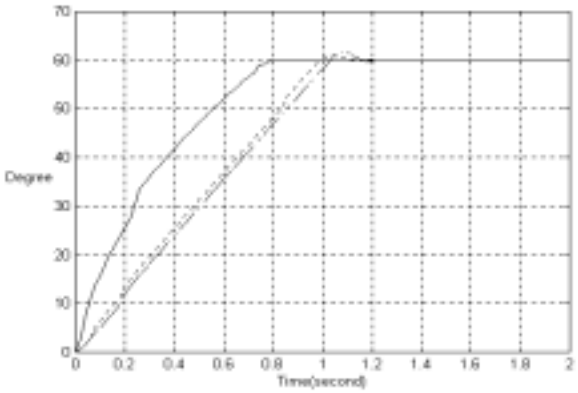


Fig. 9. The response of the position controllers for θ-axis is moved from 0 to 60 degrees. (solid line : ─ SDFSMC with CMAC, dotted line : … SDFSMC, dashed line : -- FSMC )

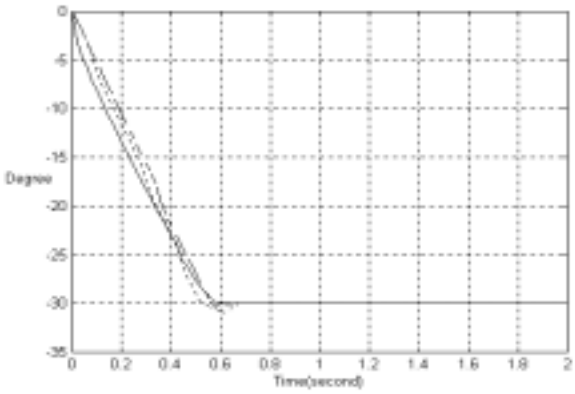


Fig. 10. The response of the position controllers for θ-axis is moved from 0 to -30 degrees. (solid line : ─ SDFSMC with CMAC, dotted line : … SDFSMC, dashed line : -- FSMC )

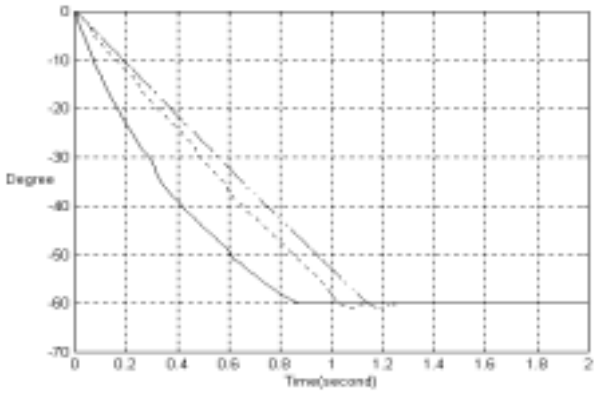


Fig. 11. The response of the position controllers for θ-axis is moved from 0 to -60 degrees. (solid line : ─ SDFSMC with CMAC, dotted line : … SDFSMC, dashed line : -- FSMC )