CMAC_ based Integral Variable Structure Control of Nonlinear System

Wei-Song Lin, Chin-Pao Hung
Institute of Electrical Engineering
National Taiwan University
Taipei, Taiwan, R.O.C.
E-mail: cbhong@chinyi.ncit.edu.tw

Abstract- A CMAC-based controller with a compensating neural network and an update rule is proposed to design the integral variable structure control (IVSC) of nonlinear system. The control scheme comprises a soft supervisor controller and a CMAC neural network. Based on the Lyapunov theorem, the soft supervisor controller guarantees the global stability of the system. The CMAC neural network provides a compensatory signal to perform the equivalent control by a real-time learning algorithm. The new IVSC control scheme reduced the dependency to system parameters and eliminated the chattering of the control signal through learning. It is proved that the CMAC-based IVSC (CIVSC) scheme is globally stable in the sense that all signals involved are bounded and the tracking error will converge to zero. Simulation results of numerical example demonstrate the effectiveness and robustness of the proposed controller.

Keyword: integral variable structure control, CMAC, nonlinear system, neural network control, learning control, sliding mode, VSC

1. INTRODUCTION

Dr. Chern first proposed the integral variable structure control (IVSC) to solve the steady state error problem, and to improve the robustness of traditional variable structure control (VSC) in 1991[1]. Then Chern applied this control scheme to robot manipulator [2], brushless DC servo system [3-4][8], DC motor [5], and induction motor [6][9][11], to demonstrate the feasibility of IVSC. The related researchers also proposed a new design method for IVSC [7]. Other applications are such as the engine system [10], the UPS system [13], and the voltage regulator system [12]. Since the IVSC introduced the integral state variable into the controller design, it can solve the steady state error problem of conventional VSC because of the dead zone or the boundary layer [14-15]. However, regardless of either the traditional VSC or the IVSC scheme, the controller design is a parametric scheme that is more dependent on the system model or parameters.

In the past year, the authors developed a real time learning scheme to solve the VSC design problem of unknown parameters dc servo system [16]. Recently, we expanded the control scheme to the IVSC of nonlinear system and proved why the CMAC_based IVSC can work well [26]. In [26], a switching supervisor controller and a weight update rule was proposed to guarantee the system stable running. Also, the parameters selection and control algorithm was presented to acquire some special properties. In the paper, another type supervisor controller and learning rule are proposed. This soft switching supervisor controller and different learning law simplified the parameters selection and eliminate the chattering through learning.

2. PROBLEM FORMULATION

Consider the n-order nonlinear systems of the form [19]

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

where f is an unknown continuous function, b is a positive unknown constant, and $u \in R$ and $y \in R$ are the input and output of the system, respectively. We assume that the state vector $\underline{x} = (x_1, x_2, \cdots x_n)^T = (x, \dot{x}, \cdots, x^{(n-1)})^T \in R^n$ is available for measurement. Therefore, the control objective of this paper can be described as follows:

Determine a feedback control $u(\underline{x} \mid \underline{w})$ and a learning law for adjusting the vector $\underline{w} = (w_1, w_2, \cdots w_g)^T \in R^{g\times 1}$, g is the selected memory size, such that the following conditions are met:

1) The closed loop system must be globally stable in the sense that all variables, $\underline{x}(t), \underline{w}(t)$ and $u(\underline{x} \mid \underline{w})$, must be uniformly bounded; i.e., $|\underline{x}(t)| \le M_x < \infty$, $|\underline{w}(t)| \le M_w < \infty$ and $|u(t)| \le M_u < \infty$, M_x, M_w, M_u are design parameters specified by the designer.

2) The tracking error $e \equiv y_d - y$, should be as small as possible under constraints in 1) and $\underline{y}_d = (y_d, \dot{y}_d, \cdots, y_d^{(n-1)})^T$, $\underline{e} = (e, \dot{e}, \cdots, e^{(n-1)})^T = (e_1, e_2, \cdots, e_n)^T$ and $\underline{k} = (k_n, \cdots k_1)^T \in R^n$ be such that all roots of the polynomial $h(s) = s^n + k_1 s^{n-1} + \cdots + k_n$ are in the open left-half plane. Here, $k_i, i = 1, \cdots n$ are designed to satisfy the IVSC sliding surface function σ defined as following [1]

$$\sigma = K_I Z + \sum_{i=1}^{n} c_i e_i = 0 \tag{2}$$

$$\dot{Z} = y_d - x_1 = e \tag{3}$$

$$c_n = 1, c_{n-i} = k_i, i = 1, \cdots, n-1, k_n = K_I \tag{4}$$

3.THE CMAC-BASED INTEGRAL VARIABLE STRUCTURE CONTROL (CIVSC)

3.1 The structure of proposed controller

As described above, if the exact system model is known

$$\dot{\sigma} = K_I \dot{Z} + \sum_{i=1}^{n} c_i \dot{e}_i = 0 \tag{5}$$

$$K_I(y_d - x_1) + \sum_{i=1}^{n-1} c_i e_{i+1} + (y_d{}^{(n)} - x^{(n)}) = 0 \tag{6}$$

$$K_I e_1 + \sum_{i=1}^{n-1} c_i e_{i+1} + y_d{}^{(n)} - f(\underline{x}) - bu = 0 \tag{7}$$

Then the optimal equivalent control u^* is

$$u^* = \frac{1}{b}\left\{- f(\underline{x}) + y_d{}^{(n)} + K_I e_1 + \sum_{i=1}^{n-1} c_i e_{i+1}\right\}$$

$$= \frac{1}{b}\left\{- f(\underline{x}) + y_d{}^{(n)} + \underline{k}^T \underline{e}\right\} \tag{8}$$

where $\underline{k} = (k_n, \cdots k_1)^T = (K_I, c_1, \ldots, c_{n-1})^T$. Applied (8) to (1) results in

$$x^{(n)} = f(x) + \left\{- f(x) + y_d{}^{(n)} + \underline{k}^T \underline{e}\right\}$$

i.e.

$$e^{(n)} + k_1 e^{(n-1)} + \cdots + k_n e = 0 \tag{9}$$

which implies that $\lim_{t\to\infty} e(t) = 0$, the main objective is achieved. Since f and b are not exactly known, the optimal equivalent control cannot be implemented. Our purpose is to design an IVSC controller using CMAC-based learning approach to approximate the optimal equivalent control u^*.

Suppose the control $u(\underline{x} \mid \underline{w})$ is the summation of a CMAC network $u_N(\underline{x} \mid \underline{w})$ and a supervisor control $u_s(\underline{x})$:

$$u(\underline{x} \mid \underline{w}) = u_N(\underline{x} \mid \underline{w}) + u_s(\underline{x}) \tag{10}$$

Substituting (10) into (1), we have

$$x^{(n)} = f(\underline{x}) + b[u_N(\underline{x} \mid \underline{w}) + u_s(\underline{x})] \tag{11}$$

Now adding and subtracting $b(t)u^*$ to (11) and after some straightforward manipulation we obtain the following error equation governing the closed-loop system:

$$e^{(n)} = -\underline{k}^T \underline{e} + b[u^* - u_N(\underline{x} \mid \underline{w}) - u_s(\underline{x})] \tag{12}$$

or, equivalently,

$$\underline{\dot{e}} = \Lambda_c \underline{e} + \underline{b}_c [u^* - u_N(\underline{x} \mid \underline{w}) - u_s(\underline{x})] \tag{13}$$

where

$$\Lambda_c = \begin{bmatrix} 0 & 1 & 0 & 0\cdots & 0 \\ 0 & 0 & 1 & 0\cdots & 0 \\ \vdots & & & & \\ -k_n & -k_{n-1} & \cdots & & -k_1 \end{bmatrix}, \underline{b}_c = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ b \end{bmatrix} \tag{14}$$

Define a Lyapunov function candidate

$$V_e = 0.5\,\sigma^2 \tag{15}$$

Then

$$\dot{V}_e = \sigma\dot{\sigma} \tag{16}$$

Using (7) and the error equation (13), we have

$$\dot{\sigma} = K_I \dot{Z} + \sum_{i=1}^{n} c_i \dot{e}_i$$

$$= K_I e_1 + \sum_{i=1}^{n-1} c_i e_{i+1} + \dot{e}_n$$

$$= K_I e_1 + \sum_{i=1}^{n-1} c_i e_{i+1} - \underline{k}^T \underline{e} + b[u^* - u_N(\underline{x} \mid \underline{w}) - u_s(\underline{x})] \tag{17}$$

$$= b[u^* - u_N(\underline{x} \mid \underline{w}) - u_s(\underline{x})] \tag{18}$$

Therefore

$$\sigma\dot{\sigma} = \sigma \cdot b[u^* - u_N(\underline{x} \mid \underline{w}) - u_s(\underline{x})]$$

$$= \sigma \cdot b[u^* - u_N(\underline{x} \mid \underline{w})] - \sigma \cdot b \cdot u_s(\underline{x}) \tag{19}$$

$$\le |\sigma| \cdot b[|u^*| + |u_N(\underline{x} \mid \underline{w})|] - \sigma \cdot b \cdot u_s(\underline{x})$$

$$= |\sigma| \cdot b\left[\left|\frac{1}{b}\left\{-f(x) + y_d{}^{(n)} + \underline{k}^T \underline{e}\right\}\right| + |u_N(\underline{x} \mid \underline{w})|\right] - \sigma \cdot b \cdot u_s(\underline{x}) \tag{20}$$

$$\le |\sigma| \cdot b\left[\left|\frac{1}{b}\left\{|f(x)| + |y_d{}^{(n)}| + |\underline{k}^T \underline{e}|\right\}\right| + |u_N(\underline{x} \mid \underline{w})|\right] - \sigma \cdot b \cdot u_s(\underline{x}) \tag{21}$$

Our task now is to design u_s such that $\dot{V}_e \le 0$. In order to do so, we need the following assumption:

Assumption 1: We can determine a function $f^U(\underline{x})$ and a constant b_L such that $|f(\underline{x})| \le f^U$ and $0 < b_L \le b$.

Therefore, we construct the u_s as follows:

$$u_s = sat\left(\frac{\sigma}{\varepsilon}\right) \bullet \left\{|u_N| + \frac{1}{b_L}\left(|f^U(x)| + |\underline{k}^T \underline{e}| + |y_d{}^{(n)}|\right)\right\} \tag{22}$$

where $sat(\sigma/\varepsilon)$ is a saturation function, executing the soft switching and defined as follows.

$$sat\left(\frac{\sigma}{\varepsilon}\right) = \begin{cases} \operatorname{sgn}\left(\frac{\sigma}{\varepsilon}\right) & \left|\frac{\sigma}{\varepsilon}\right| > 1 \\ \frac{\sigma}{\varepsilon} & \left|\frac{\sigma}{\varepsilon}\right| \le 1 \end{cases}, \ \varepsilon \text{ is a small positive constant} \tag{23}$$

Substituting (22) and (23) into (21), and considering the $|\sigma/\varepsilon|>1$ case, we have

$$= |\sigma| \cdot b \cdot \left\{\left|\frac{1}{b}\left(|f(x)| + |y_d{}^{(n)}| + |\underline{k}^T \underline{e}|\right)\right| + |u_N|\right\} - \sigma \cdot b \cdot \operatorname{sgn}\left(\frac{\sigma}{\varepsilon}\right) \cdot \left\{|u_N| + \frac{1}{b_L}\left(|f^U(x)| + |\underline{k}^T \underline{e}| + |y_d{}^{(n)}|\right)\right\}$$

$$= |\sigma| \cdot b \cdot \left\{\frac{1}{b}\left(|f(x)| + |y_d{}^{(n)}| + |\underline{k}^T \underline{e}|\right) + |u_N|\right\} - |\sigma| \cdot b \cdot \left\{|u_N| + \frac{1}{b_L}\left(|f^U(x)| + |\underline{k}^T \underline{e}| + |y_d{}^{(n)}|\right)\right\}$$

$$= |\sigma| \cdot b \cdot \left\{\frac{1}{b}\left(|f(x)| + |y_d{}^{(n)}| + |\underline{k}^T \underline{e}|\right) + |u_N| - |u_N| - \frac{1}{b_L}\left(|f^U(x)| + |\underline{k}^T \underline{e}| + |y_d{}^{(n)}|\right)\right\}$$

$$\le 0 \tag{24}$$

Therefore, using the supervisor control u_s, we always have $|\sigma / \varepsilon| \le 1$. Choose a suitable ε value the sliding surface function $|\sigma|$ will less or equal than ε. Since sliding surface function σ satisfies the Hurwitz stability condition, the boundedness of σ implies the boundedness of error state $\underline{e}$. The bounded range can be found in [5]. If we replace the *sat*() term of u_s as $sgn(\sigma)$, then we can guarantee the error state $\underline{e}$ converging to zero. But u_s is a discontinuous function and is proportional to the function upper bound, which is usually very large. Using saturation function can smooth the control signal; that is so called soft supervisor control. Since u_s is used to force the system trajectory move along the sliding surface, small σ value implies the u_N part is well behaved. Large supervisor control signal is not necessary.

3.2 Learning process

Next we develop a learning law to adjust the weighting vector $\underline{w}$.

Define the optimal weighting vector [19]:

$$\underline{w}^* \equiv \arg\min_{|\underline{w}|\le M_w} \{\sup_{\underline{x}\le Mx} |u_N(\underline{x} \mid \underline{w}) - u^*|\} \tag{25}$$

and the minimum approximation error:

$$\xi \equiv u_N(\underline{x} \mid \underline{w}^*) - u^* \tag{26}$$

The CMAC output $u_N(\underline{x} \mid \underline{w})$ is described as [23]

$$u_N(\underline{x} \mid \underline{w}) = \underline{\mu}(\underline{x})\underline{w} \tag{27}$$

where $\underline{\mu}(\underline{x}) \in R^{1\times g}$ is the mapping function of CMAC network. Note that the $\underline{\mu}(\underline{x})$ only contains the elements of 0 and 1. The numbers of 1 are equal to the fired memory cells A^* and the $|\underline{\mu}(\underline{x})| = \sqrt{A^*}$.

Then the error equation (13) can be rewritten as

$$\begin{aligned}\dot{\underline{e}} &= \Lambda_c \underline{e} + \underline{b}_c[u_N(\underline{x} \mid \underline{w}^*) - u_N(\underline{x} \mid \underline{w})] - \underline{b}_c u_s(\underline{x}) - \underline{b}_c \xi \\ &= \Lambda_c \underline{e} + \underline{b}_c \underline{\phi}^T \underline{\mu}^T(\underline{x}) - \underline{b}_c u_s(\underline{x}) - \underline{b}_c \xi\end{aligned} \tag{28}$$

where $\underline{\phi} = \underline{w}^* - \underline{w}$.

Define the Lyapunov function candidate

$$V = \frac{1}{2}\sigma^2 + \frac{bA^*}{2\beta}\underline{\phi}^T \underline{\phi} \tag{29}$$

where β is a positive constant and A^* is the association memory cells number.

then

$$\dot{V} = \sigma\dot{\sigma} + \frac{bA^*}{\beta}\underline{\phi}^T \dot{\underline{\phi}} \tag{30}$$

Substituting (19) and (26) into (30), and using the fact $\underline{\mu}(\underline{x})\underline{\phi}(\underline{x}) = \underline{\phi}^T(\underline{x})\underline{\mu}^T(\underline{x})$ we have

$$\begin{aligned}\dot{V} &= \sigma \cdot b[u_N(\underline{x} \mid \underline{w}^*) - u_N(\underline{x} \mid \underline{w})] - \sigma \cdot b \cdot u_s(\underline{x}) - \sigma \cdot \beta \cdot \xi + \frac{bA^*}{\beta}\underline{\phi}^T \dot{\underline{\phi}} \\ &= \sigma \cdot b\underline{\phi}^T \underline{\mu}^T(\underline{x}) - \sigma \cdot b \cdot \xi - \sigma \cdot b \cdot u_s(\underline{x}) + \frac{bA^*}{\beta}\underline{\phi}^T \dot{\underline{\phi}} \qquad (31) \\ &= \sigma \cdot b\underline{\phi}^T[\underline{\mu}^T(\underline{x}) + \frac{A^*}{\sigma\beta}\dot{\underline{\phi}}] - \sigma \cdot b \cdot \xi - \sigma \cdot b \cdot u_s(\underline{x}) \qquad (32)\end{aligned}$$

if we choose the learning law:

$$\dot{\underline{w}} = \frac{\sigma \cdot \beta \underline{\mu}^T(\underline{x})}{A^*} \tag{33}$$

then (32) becomes

$$\dot{V} = -\sigma \cdot b \cdot \xi - \sigma \cdot b \cdot u_s(\underline{x}) \le -\sigma \cdot b \cdot \xi \tag{34}$$

where we use the facts $\sigma \cdot b \cdot u_s(\underline{x}) \ge 0$ (cf.(22)) and $\dot{\underline{\phi}} = -\dot{\underline{w}}$. This is the best we can achieve. Fortunately, assume the memory size is large enough the CMAC can approximate function to arbitrarily accurate [23]. The last term of (34) will tend to zero. A large enough memory size, which can be obtained by programming the CMAC network with larger quantization level and more associated memory cells A^*, will easily have $\dot{V} \le 0$.

In order to guarantee the $|\underline{w}| \le M_w$, we modify the learning law as

$$\dot{\underline{w}} = \begin{cases} \dfrac{\sigma \cdot \beta \underline{\mu}^T(\underline{x})}{A^*}, & \text{if } |\underline{w}| < M_w \text{ or } |\underline{w}| = M_w \text{ and } \sigma \cdot \underline{w}^T \underline{\mu}^T(\underline{x}) \le 0 \\ T\left\{\dfrac{\sigma \cdot \beta \underline{\mu}^T(\underline{x})}{A^*}\right\}, & \text{if } |\underline{w}| = M_w \text{ and } \sigma \cdot \underline{w}^T \underline{\mu}^T(\underline{x}) > 0 \end{cases} \tag{35}$$

where the operator *T(*)* is define as [19]:

$$T\left\{\frac{\sigma \cdot \beta \underline{\mu}^T(\underline{x})}{A^*}\right\} = \frac{\sigma \cdot \beta \underline{\mu}^T(\underline{x})}{A^*} - \frac{\sigma \cdot \beta \underline{w}\underline{w}^T \underline{\mu}^T(\underline{x})}{A^* |\underline{w}|^2} \tag{36}$$

Remark1: The modification of learning law will lead to the (32) become

$$\dot{V} = -\sigma \cdot b \cdot \xi - \sigma \cdot b \cdot u_s(\underline{x}) + I_1 \cdot \sigma \cdot b \cdot \underline{\phi}^T \frac{\underline{w}\underline{w}^T \underline{\mu}^T(\underline{x})}{|\underline{w}|^2} \tag{37}$$

where $I_1 = 0(1)$ if the first (second) line of (35) is true. The last term of (37) can be proofed is nonpositive as follows: If $I_1 = 0$, the conclusion is trivial. Let $I_1 = 1$, which means that $|\underline{w}| = M_w$ and $\sigma \cdot \underline{w}^T \underline{\mu}^T(\underline{x}) > 0$. Then we have $\underline{\phi}^T \underline{w} = (\underline{w}^* - \underline{w})\underline{w} = 0.5\left[|\underline{w}^*|^2 - |\underline{w}|^2 - |\underline{w} - \underline{w}^*|^2\right] \le 0$, since $|\underline{w}| = M_w \ge |\underline{w}^*|$. Therefore, (34) is still established.

The following theorem guarantees the properties of the proposed control scheme.

Theorem 1: Consider the nonlinear plant (1) with the control (10), where $u_N(\underline{x} \mid \underline{w})$ is the CMAC output and $u_s(\underline{x})$ is given by (22). Let the weighting vector $\underline{w}$ be updated by the learning law (35) and let the assumption 1 be true. Then, the overall control scheme guarantees the following properties.

1. $|\underline{w}| \le M_w$, $|x(t)| \le |y_d| + \dfrac{\varepsilon}{(n-1)!}\left(\dfrac{n-1}{\alpha_{\min}}\right)^{n-1} e^{-(n-1)}$ (38)

   and

   $$|u(t)| \le 2\sqrt{A^*} \cdot M_w + \frac{1}{b_L}\left[f^U + |y_d^{(n)}| + |\underline{k}|\frac{\varepsilon}{(n-1)!}\left(\frac{n-1}{\alpha_{\min}}\right)^{n-1} e^{-(n-1)}\right] \tag{39}$$

   for all $t \ge 0$, where $\alpha_{\min}$ is the minimum characteristic root of $(s+\alpha_1)(s+\alpha_2)...(s+\alpha_n)$, which is derived as following.

2. $\lim_{t\to\infty}|e(t)| = 0$.

Proof: i) To prove $|\underline{w}| \le M_w$, we let $V_w = 0.5\underline{w}^T \underline{w}$. If the first line of (35) is true. We have either $|\underline{w}| \le M_w$ or $\dot{V}_w = \dfrac{\sigma \cdot \beta}{A^*} \cdot \underline{w}^T \underline{\mu}^T \le 0$ when $|\underline{w}| = M_w$; it is clear that we always have $|\underline{w}| \le M_w$. If the second line is true, we have

$|\underline{w}| = M_w$ and $\dot{V}_w = \sigma \cdot \beta \cdot \underline{w}^T \left[\frac{\mu^T(\underline{x})}{A^*} - \frac{\underline{w}\underline{w}^T \mu^T(\underline{x})}{A^* |\underline{w}|^2} \right] = 0$.

Therefore we always have $|\underline{w}| \le M_w, \forall t \ge 0$

As described above, the supervisor control u_s will curb the output trajectory to move along the sliding surface and obtain the $|\sigma(t)| \le \varepsilon$. Therefore

$$|\sigma(t)| = \left| K_I Z + \sum_{i=1}^{n} c_i e_i \right| \le \varepsilon \quad (40)$$

Take Laplace transform to both side of (40), we have

$$\left| \left(\frac{K_I}{s} + c_1 + c_2 s + \ldots + s^{n-1}\right) e(s) \right| \le \frac{\varepsilon}{s} \quad (41)$$

Though straightforward manipulation, then

$$|e(s)| \le \left| \frac{\varepsilon}{K_I + c_1 s + c_2 s^2 + \cdots + s^n} \right| = \left| \frac{\varepsilon}{(s+\alpha_1)(s+\alpha_2)\cdots(s+\alpha_n)} \right|$$

$$\le \frac{\varepsilon}{|(s+\alpha_1)||(s+\alpha_2)|\cdots|(s+\alpha_n)|} \quad (42)$$

Consider the $s = j\omega$ case (on the imagery axis, i.e. margin stable, system with maximum error output under stable condition.) we have

$$|e(s)| \le \frac{\varepsilon}{|(s+\alpha_1)||(s+\alpha_2)|\cdots|(s+\alpha_n)|} \le \frac{\varepsilon}{|(s+\alpha_{\min})|^n} \quad (43)$$

Take the Laplace inverse transform to both sides and find the extreme value, we have

$$|e(t)| \le \frac{\varepsilon}{(n-1)!} t^{n-1} e^{-\alpha_{\min} t} \le \frac{\varepsilon}{(n-1)!} \left(\frac{n-1}{\alpha_{\min}} \right)^{n-1} e^{-(n-1)} \quad (44)$$

Since $e = y_d - x$, we have

$|x| \le |y_d| + |e| \le |y_d| + \frac{\varepsilon}{(n-1)!} \left(\frac{n-1}{\alpha_{\min}} \right)^{n-1} e^{-(n-1)}$, which is (38).

Finally, we prove (39). Since $u_N(\underline{x} \mid \underline{w}) = \underline{\mu}(\underline{x})\underline{w}$ is only the sum of fired memory cells, we have $|u_N(\underline{x} \mid \underline{w})| \le |\underline{\mu}(\underline{x})||\underline{w}| = \sqrt{A^*} \cdot |\underline{w}| \le \sqrt{A^*} \cdot M_w$. Therefore, from (10) and (18), we have (39).

ii) From (44),

$$\lim_{t \to \infty} \frac{\varepsilon}{(n-1)!} t^{n-1} e^{-\alpha_{\min} t} = 0 \quad (45)$$

we have $\lim_{t \to \infty} |e(t)| = 0$. Q.E.D.

Remark 2: For real control system, the state $\underline{x}$ and control force u are required to be constrained within a certain region. Such as the robot system, the joint positions are constrained within workspace and the control signal should be smaller or equal to the output of the computer interface card (generally 10volt). Depending on the practical constraint, the designer specified M_x and M_u. Then based on (38) and (39) the sliding surface, M_w, and ε can be chosen to satisfy the required performance and let the state $\underline{x}$ and control force u be within the constraint sets.

3.3 Control algorithm

As described above, the structure of the proposed scheme is shown in Fig. 1 and the control algorithm is summarized as follows:

Step 1. Specify the design parameters M_x, M_u, based on practical constraints.

Step 2. Specify the integral sliding surface function σ_d and ε, then calculated the $\alpha_{\min}$.

Step 3. Determine the M_w, based on (38) and (39).

Step 4. Transform the sliding surface to the desired error states $\sigma_d(t) \Rightarrow \underline{e}_d(t)$

Step 5. Send the desired next error states $\underline{e}_d(t)$ to the CMAC network

Step 6. Perform a series of mappings to transform the input values to the CMAC output $u_N(t)$

Step 7. Calculate the supervisor control signal $u_s(t)$

Step 8. The $u_N(t)$ is assumed to be an optimal equivalent control signal and is added to the output of the $u_s(t)$ to form the IVSC control signal $u(t) = u_N(t) + u_s(t)$

Step 9. Send $u(t)$ to plant to produce the actual output states $\underline{e}_o(t)$

Step 10. Calculate the values of the actual integral sliding surface function $\sigma(t)$

Step 11. Update the fired weights using equation (35)

Step 12. If $t \ge T$, stop. (T is the control cycle.) Otherwise, go to step 4.

Noted that the step 1 to step 3 are off line processes and others are on line computing. The related parameters of CMAC, quantization levels and A^*, are also specified off line. They can be adjusted depending on the weighting distribution plot.

4. NUMERICAL EXAMPLE

Considering a simplified robot system model described as following [16]:

$$\dot{x}_1(t) = x_2(t)$$
$$\dot{x}_2(t) = 0x_1 - (33 + 10\sin 5t)x_2 + (300 + 80\sin 5t)u \quad (46)$$

where $x_1 = x$ is the end-effector output. Define the error output $e \equiv y_d - x = e_1, \dot{e}_1 = e_2$. y_d is the desired end-effector output. Without lose of generality, we let $y_d = 1$. We choose the integral sliding surface as following:

$$\sigma = k_i \int e_1 dt + c_1 e_1 + e_2, k_i = 1, c_1 = 10 \quad (47)$$

i.e. $\underline{k} = [1 \quad 10]$. The associated design parameters of CMAC network are listed as follows:

Quantization level 30, memory size 4096, fired memory cells A^*=10, M_x=5(rad), M_u=10(volt), M_w=2, ε=0.2, β=0.8, b_L=220. Fig. 2 shows the output states plots without the CMAC compensation, i.e. only the supervisor control. The system response is slow and with steady state error. Introducing the CMAC network, the simulation results are shown in Fig. 3. From the simulation results, it is clear that the CIVSC approach can give an almost accurate servo tracking response to partial known parameters system (Fig. 3(b)(c)). Fig. 3(d) shows the waveforms of control function, and we see that the CMAC output compensates a continuous signal. (See the zoom view plots on the right hand side). Fig. 3(e), the phase plane plot, also shows the output nearly tracks the integral sliding surface.

5. DISCUSSION

The selection of M_x, M_u and M_w are dependent on practical constraints. M_x and M_u represent the maximum rotated angle of robot arm and the maximum control signal of the D/A card output. It is easy to specify as 5 rad and 10 volt. Using (38) then the M_w can be determined as 2. The control results of Fig. 3(a) shows $|\underline{w}|=1.65$ that is smaller than M_w.

We wish the learning behavior happened in the neighborhood of the desired surface, multiple segments sliding surface [22] is better than only one sliding surface. Totally invariant sliding surface [25] guarantees all the transient states are under control and avoid the abrupt change of sliding surface. Simultaneously, the learning law is related to the sliding surface function. Under supervisor control, multiple segments sliding surface avoid the over tuning because of the output state far away the sliding surface ($\sigma(t)$ too large), i.e. in the reaching mode the learning is blind and easy to cause over tuning. It will give rise to vibration near the hitting point. (See the phase plane plot Fig. 3(e)) The phase plane plot of multiple segments sliding surface is shown in Fig. 4. But rigorous theoretic proofs are under studying.

The training gain β is related to the learning effect. In generally, the value of β is between 0 and 1 for supervised learning [18], i.e.$0<\beta\leq 1$. The memory size of CMAC also affects the control performance. The memory size depends on the desired input signal and the performance requirement. It is hard to make a clear decision but we can obtain some idea from the activity of memory address. For example, the weights value distribution plot (Fig. 3(a)) shows that not all the addresses excited, in this case the memory size is enough. We can reduce the quantization level or association cells number to decrease the memory size. On the other hand, if all the addresses have been excited, the memory size may be not enough. We need to adjust the quantization level or association cells number to increase the memory size. When the memory size is too large beyond the computer specification, the Hash coding is needed.

6. CONCLUSION

In this paper, a new integral variable structure control scheme is proposed to solve the little model information problem of IVSC. Based on the Lyapunov synthesis approach, the combination of a soft supervisor controller and CMAC network guarantee the stability of the proposed scheme in the sense that all signals involved are uniformly bounded. The supervisor controller just provides a rough stable control signal. And CMAC network generates a compensatory control signal through learning to guide the output trajectory moving along the sliding surface. Using the generalization property of neural network, the CMAC can learn the approximated equivalent control signal on line and yield a robust sliding mode control. It can keep all the advantages of IVSC and can alleviate the dependency to system parameters. Applying this scheme to a simplified robot manipulator with large uncertainties, the simulation results demonstrate the success of the proposed scheme.

6. REFERENCE

[1] T. L. Chern, Y. C. Wu," Design of integral variable structure controller and application to electrohydraulic velocity servosystems", *IEE Proc. D* vol. 138, no. 5, pp. 439-444, 1991.

[2] T. L. Chern, Y. C. Wu," Integral variable structure control approach for robot manipulator", *IEE Proc. D* vol. 139, no. 2, pp. 161-166, 1992.

[3] T. L. Chern, Y. C. Wu," Design of brushless DC position servo systems using integral variable structure approach", *IEE Proc. B* vol. 140, no. 1, pp. 27-34, 1993.

[4] S. K. Chung, J. H. Lee, J. S. Ko, M. J. Youn," Robust speed control of brushless direct-driver motor using integral variable structure control", *IEE Proc. Electr. Power Appl.*, vol. 142, no. 6, pp. 361-370, 1995.

[5] T. L. Chern, J. S. Wong," DSP based integral variable structure control for DC motor servo drivers", *IEE Proc. Control Theory Appl.* vol. 142, no. 5, pp. 444-450, 1995.

[6] T. L. Chern, C. S Liu, C. F. Jong, and G. M Yan, "Discrete integral variable structure model following control for induction motor drivers", *IEE Proceedings-Electric Power Applications*, vol. 143, no. 6, Nov., pp. 467-474, 1996.

[7] J. D. Wang, T. L. Lee, Y. T. Juang," New methods to design an integral variable structure controller", *IEEE Trans. Automation Control*, vol. 41, no. 1, pp. 140-143, 1996.

[8] T. L. Chern, J. Chang, G. K. Chanf," DSP based integral variable structure model following control for brushless DC motor drivers", *IEEE Trans. Power Electronics,* vol. 12, no. 1, pp. 53-63, 1997.

[9] T. L. Chern, J. Chang, " DSP based induction motor drivers using integral variable structure model following control approach", *IEEE Trans. Power Electronics,* vol. 12, no. 1, pp. 53-63, 1997.

[10] T. L. Chern, C. W Chuang, "Design of optimal MIMO DIVSC systems and its application to idle speed control of spark ignition engine", ASME Journal of Dynamic Systems Measurement and Control, vol. 119, June, pp.175-182, 1997.

[11] T. L. Chern, J. Chang, K. L. Tsai, "IVSC-based Adaptive Speed Estimator and Resistance Identifier for an Induction Motor", International Journal of Control, vol. 69, no. 1, pp.31-47, 1998.

[12] T. L. Chern, G. K. Chang, "Automatic voltage regulator design by modified discrete integral variable structure model following control", *Automatica*, pp.1575-1582, 1998.

[13] T. L. Chern, J. Chang, C. H. Chern, H. T. Su, "Microprocessor-based Modified Discrete Integral Variable Structure Control for UPS," *IEEE Trans. Industrial Electronics*, Apirl, 1999.

[14] K. K. D. Young, " Controller design for a manipulator using theory of variable structure systems", *IEEE Trans. Syst., Man, Cybern.* , vol. SMC-8 , no. 2, pp.101-109, 1978.

[15] J. J. Slotine, S. S. Sastry, "Tracking control of nonlinear systems using sliding surface, with application to robot manipulators", *INT. J. Control*, vol.38, no. 2, pp. 465-492, 1983.

[16] W. S. Lin, C. P. Hung," Variable structure control of unknown parameter dc servo systems using CMAC-based learning approach", *Proc. Am. Control Conf.*, pp. 5016-5021, 2001.

[17] J. S. Albus, "A new approach to manipulator control: the cerebeller model articulation controller (CMAC)[1] , *Trans. ASME J. Dynam., Syst., Meas., and Contr.*, vol. 97, pp.220-227, 1975

[18] D. A. Handeiman, S. H. Lane, and J. J. Gelfand, "Integrating neural networks and knowledge-based systems for intelligent robotic control" ,

IEEE Control System Magazine, pp. 77-86, 1990.
[19] L. X. Wang, "Stable adaptive fuzzy control of nonlinear system", *IEEE Trans. on Fuzzy Sytem,* vol. 1,no. 2, pp. 146-155, 1993.
[20] Y. F. Wong, A. Sideris, "Learning convergence in the cerebellar model articulation controller", *IEEE Trans. on Neural Network,* vol. 3,no. 1, pp. 115-121, 1992.
[21] K. S. Fu, R.C. Gonzalez, C.S.G. Lee," ROBOTICS: control, sensing, vision, and intelligence", McGRAW-Hill, 1987.
[22] A. Davari, and Z. Zhang, "Application of the three-segments variable structure systems", *Proc. Am. Control Conf.,* 1, pp. 62-63, 1991.
[23] S. Commuri, S. Jagannathan, F. L. Lewis, "CMAC neural network control of robot manipulators", Journal of Robotic System, 14(6), pp.465-482, 1997.
[24] W. S. Lin, and C. H. Tsai," Neurofuzzy-model-following control of MIMO nonlinear systems", *IEE Proc. Control Theory Appl.* vol. 146, no. 2, pp. 157-164, 1999.
[25] K.K. Shyu, C. K. Lai, J. Y. Hung, "Totally invariant state feedback controller for position control of synchronous reluctance motor", *IEEE Trans. Industrial Electronics,* vol. 48, no.3, pp.615-624, June, 2001.
[26] W. S. Lin, C. P. Hung," Integral variable structure control of nonlinear system using CMAC-based learning approach", *Proc. Am. Control Conf.,* 2002 (accepted).

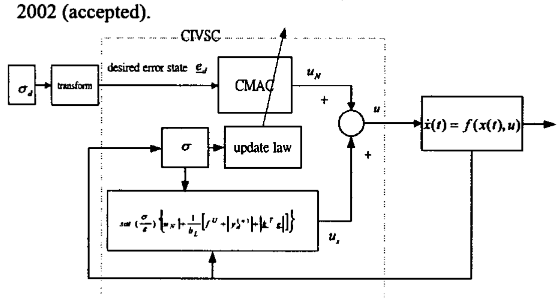

Fig.1 Block diagram of CIVSC

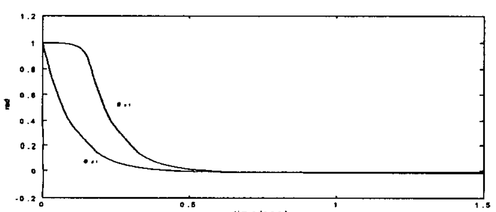
Fig. 2(a) Desired and actual output state e_1 (without the CMAC network)

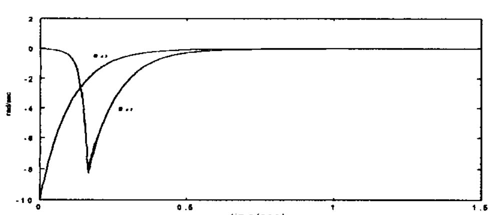
Fig.2(b) Desired and actual output state e_2(without the CMAC network)

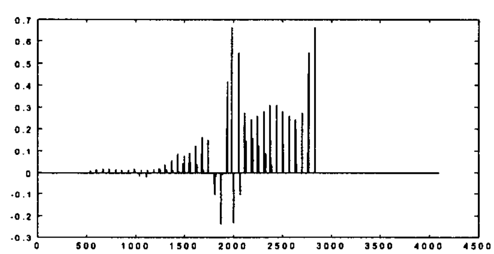
Fig. 3(a) Weight value of memory cells for CMAC network

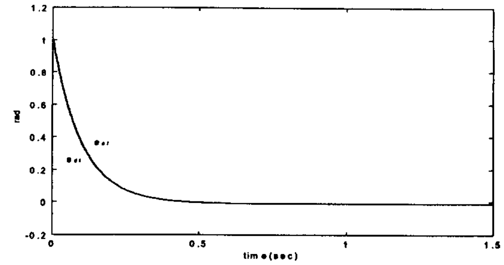
Fig. 3(b) Desired and actual output state e_1

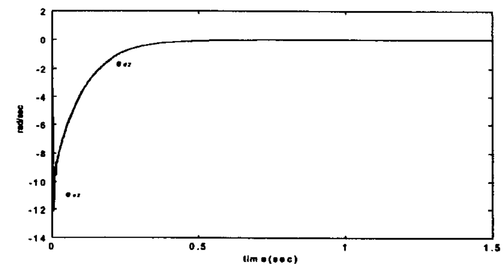
Fig. 3(c) Desired and actual output state e_2

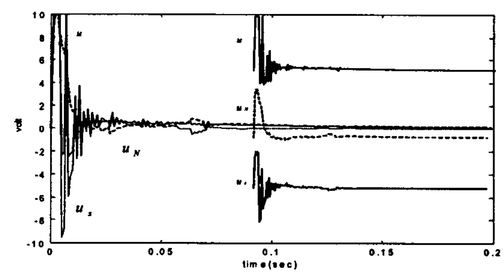
Fig. 3(d) Zoom control signal u, u_N and u_s

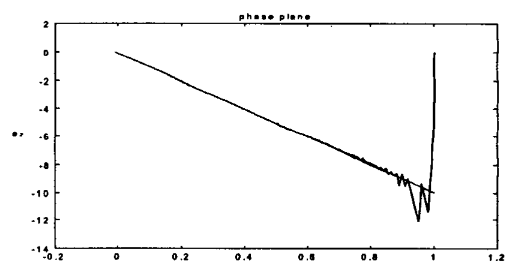
Figure 3(e) Phase plane plot of desired and actual output states

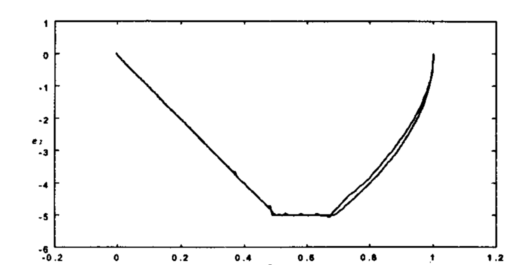
Fig. 4 Phase plane plots of desired and actual output states with three segments sliding surface