Variable structure control of unknown parameters dc servo systems using CMAC-based learning approach

Wei-Song Lin and Chin-Pao Hung
Institute of Electrical Engineering
National Taiwan University
Taipei, Taiwan, R.O.C.

Abstract

A CMAC-based controller with a compensating neural network and an update rule is proposed to design the variable structure control (VSC) of unknown parameters dc servo systems. By introducing a stabilizer controller and a CMAC neural network to construct the VSC control law, the new control scheme performs the equivalent control by a real-time learning algorithm. The stabilizer controller is designed by using Lyapunov stability theory and the updating rule of the CMAC weights is obtained by using the gradient descent method. Simulation results of a simplified robot link model demonstrate the effectiveness and robustness of the proposed controller.

1. Introduction

A variable structure controller (VSC) design usually needs the mathematical model of system. When the exact plant model is not available, a nominal model with bounded uncertainty is usually assumed so that a robust control law can be obtained [1-3]. In these conventional schemes, the control laws design need the system model. If the system models are unknown, these control laws cannot be implemented. To solve the unknown parameters problem, many researchers proposed fuzzy sliding mode control (FSMC) using IF-THEN linguistic rule to achieve a robust controller [4-6]. However, these fuzzy schemes usually need the expert experience, and the design process is tedious. In the past decade, several investigators have presented learning control schemes for improving the control performance of traditional control methods or to solve the unknown model problems [7-8]. Based on the repetitive operations, the control object can be achieved perfectly. Miller in [9] proposed a real-time learning scheme for industrial manipulator control using the CMAC network. In [9], the CMAC network is used to learn the inverse dynamic of industrial robot. The combination of CMAC and a fixed gain controller make the real-time control possible, whereas only few publications use neural network to solve the problems of VSC to system without the plant model. Until 1999, Bottura [10] introduced CMAC network into the VSC to reach the hitting condition for

system without the knowledge of system parameters. However, Bottura only proposed an elementary idea; systematic design steps and the theoretical demonstration are lacking.

CMAC (Cerebellar Model Articulation controller) is developed by Albus in the 1970's [11]. It is like the models of human memory, using local data to perform reflexive processing. Therefore its learning speed is very fast and this makes it attractive to on line application in control system [12]. This article considers the VSC controller design of unknown parameters DC servo systems. By introducing a stabilizer controller to guarantee the stable learning, a CMAC neural network with learning algorithm is constructed to learn the equivalent control of VSC. This is unlike the conventional VSC, which is a parametric scheme where any change in the performance requirements necessitates the design work to be re-performed. The CMAC approximates and performs the equivalent control depending on the uncertainties and the constraints of the desired sliding surface. Therefore, redesign is not necessary for any parameter change in the performance requirements or the controlled plant. With application of this scheme to the motion control of robot manipulator, computer simulations demonstrate the success of the proposed method.

2. Problem formulation

Consider a dc servo system described by the following error dynamic equation [15]:

$$\dot{x}_i(t) = x_{i+1}(t) \quad i = 1, \dots, n-1$$

$$\dot{x}_n(t) = -\sum_{i=1}^n a_i(t)x_i - b(t)u(t) - f \quad (1)$$

where $x_1(t)$ is the error state output, u is the control command, f is the disturbance, $a_i(t), b(t)$ are the unknown bounded plant parameters and $b(t) > 0$. Therefore, the control objective of this paper can be describe as follows [14]:

Determine a feedback control $u(\underline{x}|\underline{w})$ (based on

CMAC neural network) and a learning law for adjusting the vector $\underline{w}$ such that the following conditions are met:

- 1) The close loop system must be globally stable in the sense that all variables, $\underline{x}(t)$, $\underline{w}(t)$ and $u(\underline{x}|\underline{w})$, must be uniformly bounded; i.e., $|\underline{x}(t)| \leq M_x < \infty$, $|\underline{w}(t)| \leq M_w < \infty$ and $|u(\underline{x}|\underline{w})| \leq M_u < \infty$, M_x , M_w , M_u are design parameters specified by the designer.
- 2) The tracking error x_1 , should be as small as possible under the constraints in 1) and $\underline{x} = (x_1, x_2, x_3, \dots, x_n) = (x_1, \dot{x}_1, \dots, x_1^{(n-1)})^T$ satisfy the sliding surface model given by

$$\sigma = \sum_{i=1}^n c_i x_i = 0, \quad i = 1, \dots, n, \quad c_n = 1 \quad (2)$$

in which the c_i 's are all real and $h(s) = s^{n-1} + c_{n-1}s^{n-2} + \dots + c_1$ is Hurwitz polynomial.

3. The VSC design using CMAC-based learning approach

3.1 The structure of proposed controller

Generally, the VSC control law can be decomposed into [15]

$$u = u_{eq} + \Delta u \quad (3)$$

here u_{eq} , called the equivalent control, is defined as the solution of $\dot{\sigma} = 0$ under free uncertainty and Δu is constructed to satisfy the hitting condition, i.e. $\sigma \dot{\sigma} < 0$. In the conventional scheme, the equivalent control is derived from the system nominal model. If the system parameters are unknown, the control law cannot be implemented.

As described above, if the exact system model is known

$$\dot{\sigma} = \sum_{i=1}^n c_i \dot{x}_i = 0 \quad (4)$$

Substituting (1) into (4), we have

$$\sum_{i=1}^{n-1} c_i x_{i+1} - \sum_{i=1}^n a_i(t) x_i - b(t) u(t) - f = 0 \quad (5)$$

Then the optimal equivalent control u^* is

$$\frac{1}{b(t)} \left\{ \sum_{i=1}^{n-1} c_i x_{i+1} - \sum_{i=1}^n a_i(t) x_i - f \right\} \quad (6)$$

Applied to (1) results in

$$\begin{aligned} \dot{x}_i(t) &= x_{i+1}(t) \quad i = 1, \dots, n-1 \\ \dot{x}_n(t) &= -\sum_{i=1}^n a_i(t) x_i - b(t) \frac{1}{b(t)} \left\{ \sum_{i=1}^{n-1} c_i x_{i+1} - \sum_{i=1}^n a_i(t) x_i - f \right\} - f \end{aligned} \quad (7)$$

Therefore, under ideal sliding motion, the system described by (1) can be reduced to

$$\begin{aligned} \dot{x}_i(t) &= x_{i+1}(t) \quad i = 1, \dots, n-1 \\ \dot{x}_n(t) &= -\sum_{i=1}^{n-1} c_i x_{i+1} = -c_1 x_2 - c_2 x_3 - \dots - c_{n-1} x_n \end{aligned} \quad (8)$$

The eigenvalues of the system can be set arbitrarily by choosing the value of c_i , $i = 1, \dots, n-1$. If the sliding surface satisfies the Hurwitz stability, which implies that $\lim_{t \rightarrow \infty} x_1(t) = 0$, the main objective is achieved. Since the system parameters are not exactly known, the optimal equivalent control cannot be implemented. Our purpose is to design a VSC controller using CMAC-based learning approach to approximate the optimal equivalent control u^* .

Let the control function $u(\underline{x}|\underline{w})$ be decomposed into

$$u(\underline{x}|\underline{w}) = u_N(\underline{x}|\underline{w}) + u_s(\underline{x}) \quad (9)$$

where $u_N(\underline{x}|\underline{w})$ is a CMAC neural network output, and $u_s(\underline{x})$, a stabilizer control, will be determined later.

Substituting (9) into (1), we have

$$\begin{aligned} \dot{x}_i(t) &= x_{i+1}(t) \quad i = 1, \dots, n-1 \\ \dot{x}_n(t) &= -\sum_{i=1}^n a_i(t) x_i - b(t) [u_N(\underline{x}|\underline{w}) + u_s(\underline{x})] - f \end{aligned} \quad (10)$$

Now subtracting and adding $b(t)u^*$ to (10) and after some straightforward manipulation we obtain the error equation governing the closed-loop system:

$$\begin{aligned} \dot{x}_i(t) &= x_{i+1}(t) \quad i = 1, \dots, n-1 \\ \dot{x}_n(t) &= -\sum_{i=1}^{n-1} c_i x_{i+1} + \sum_{i=1}^n a_i(t) x_i + f - \sum_{i=1}^n a_i(t) x_i + b(t) [u^* - u_N(\underline{x}|\underline{w}) - u_s(\underline{x})] - f \end{aligned} \quad (11)$$

i.e.

$$\begin{cases} \dot{x}_i(t) = x_{i+1}(t) & i = 1, \dots, n-1 \\ \dot{x}_n(t) = -\sum_{i=1}^{n-1} c_i x_{i+1} + b(t) [u^* - u_N(\underline{x}|\underline{w}) - u_s(\underline{x})] \end{cases} \quad (12)$$

Define a standard Lyapunov function is

$$V = \frac{1}{2} \sigma^2 \quad (13)$$

Then

$$\dot{V} = \sigma \dot{\sigma} \quad (14)$$

Taking the time derivative of σ , and using the error equation (12), we have

$$\dot{\sigma} = \sum_{i=1}^n c_i \dot{x}_i = \dot{x}_n + \sum_{i=1}^{n-1} c_i x_{i+1} \quad (15)$$

$$= -\sum_{i=1}^{n-1} c_i x_{i+1} + b(t) [u^* - u_N(\underline{x}|\underline{w}) - u_s(\underline{x})] + \sum_{i=1}^{n-1} c_i x_{i+1}$$

$$= b(t) [u^* - u_N(\underline{x}|\underline{w}) - u_s(\underline{x})] \quad (16)$$

$\Rightarrow$

$$\dot{\sigma} \sigma = \sigma \bullet b(t) [u^* - u_N(\underline{x}|\underline{w}) - u_s(\underline{x})] \quad (17)$$

$$= \sigma \bullet b(t) [u^* - u_N(\underline{x}|\underline{w})] - \sigma \bullet b(t) \bullet u_s(\underline{x})$$

$$\leq |\sigma| \bullet b(t) [|u^*| + |u_N(\underline{x}|\underline{w})|] - \sigma \bullet b(t) \bullet u_s(\underline{x})$$

$$= |\sigma| \cdot b(t) \left(\frac{1}{b(t)} \left| \sum_{i=1}^{n-1} c_i x_{i+1} - \sum_{i=1}^n a_i(t) x_i - f \right| + |u_N(x)| \right) - \sigma \cdot b(t) \cdot u_s(x) \\ \leq |\sigma| \cdot b(t) \left(\frac{1}{b(t)} \left| \sum_{i=1}^{n-1} c_i x_{i+1} \right| + \left| \sum_{i=1}^n a_i(t) x_i \right| + |f| + |u_N(x)| \right) - \sigma \cdot b(t) \cdot u_s(x) \quad (18)$$

Our task now is to design u_s such that $\dot{V} \leq 0$. In order to do so, we need the following assumption:
Assumption 1: We can determine the parameter bounds such that

$$|a_i(t)| \leq a_i^U, \quad i=1, \dots, n, \quad 0 < b_L \leq b(t) \quad \text{and} \quad |f| \leq f^U$$

Therefore, we construct the u_s as follows:

$$u_s = \text{sat}\left(\frac{\sigma}{\varepsilon}\right) \cdot \left\{ |u_N| + \frac{1}{b_L} \left(\left| \sum_{i=1}^{n-1} c_i x_{i+1} \right| + \left| \sum_{i=1}^n a_i^U x_i \right| + f^U \right) \right\} \quad (19)$$

where $\text{sat}\left(\frac{\sigma}{\varepsilon}\right)$ is a saturation function defined as follows.

$$\text{sat}\left(\frac{\sigma}{\varepsilon}\right) = \begin{cases} \frac{\sigma}{\varepsilon} & \left| \frac{\sigma}{\varepsilon} \right| \geq 1 \\ \text{sgn}\left(\frac{\sigma}{\varepsilon}\right) & \left| \frac{\sigma}{\varepsilon} \right| < 1 \end{cases}, \quad \varepsilon \text{ is a small positive constant} \quad (20)$$

Substituting (19), (20) and (6) into (18), and considering the

$\left| \frac{\sigma}{\varepsilon} \right| \geq 1$ case, we have

$$\begin{aligned} \dot{\sigma} &\leq |\sigma| \cdot b(t) \left(\frac{1}{b(t)} \left| \sum_{i=1}^{n-1} c_i x_{i+1} \right| + \left| \sum_{i=1}^n a_i(t) x_i \right| + |f| + |u_N| \right) \\ &\quad - \sigma \cdot b(t) \cdot \text{sgn}\left(\frac{\sigma}{\varepsilon}\right) \cdot \left\{ |u_N| + \frac{1}{b_L} \left(\left| \sum_{i=1}^{n-1} c_i x_{i+1} \right| + \left| \sum_{i=1}^n a_i^U x_i \right| + f^U \right) \right\} \\ &= |\sigma| \cdot b(t) \left(\frac{1}{b(t)} \left| \sum_{i=1}^{n-1} c_i x_{i+1} \right| + \left| \sum_{i=1}^n a_i(t) x_i \right| + |f| + |u_N| \right) \\ &\quad - |\sigma| \cdot b(t) \cdot \left\{ |u_N| + \frac{1}{b_L} \left(\left| \sum_{i=1}^{n-1} c_i x_{i+1} \right| + \left| \sum_{i=1}^n a_i^U x_i \right| + f^U \right) \right\} \\ &= |\sigma| \cdot b(t) \left(\frac{1}{b(t)} \left| \sum_{i=1}^{n-1} c_i x_{i+1} \right| + \left| \sum_{i=1}^n a_i(t) x_i \right| + |f| + |u_N| \right) \\ &\quad - \frac{1}{b_L} \left(\left| \sum_{i=1}^{n-1} c_i x_{i+1} \right| + \left| \sum_{i=1}^n a_i^U x_i \right| + f^U \right) \\ &\leq |\sigma| \cdot b(t) \left(\frac{1}{b(t)} \left| \sum_{i=1}^{n-1} c_i x_{i+1} \right| + \left| \sum_{i=1}^n a_i(t) x_i \right| + |f| \right) - \frac{1}{b_L} \left(\left| \sum_{i=1}^{n-1} c_i x_{i+1} \right| + \left| \sum_{i=1}^n a_i^U x_i \right| + f^U \right) \\ &\leq 0 \end{aligned} \quad (21)$$

Therefore, using the stabilizer control u_s , we always have

$\left| \frac{\sigma}{\varepsilon} \right| < 1$. Choose a suitable ε value the sliding surface function $|\sigma|$ will be less than ε . Since sliding surface function σ satisfies the Hurwitz stability condition, the boundedness of σ implies the boundedness of error state x . The bounded range can be found in [18]. If we

replace the $\text{sat}\left(\frac{\sigma}{\varepsilon}\right)$ term of u_s as $\text{sgn}(\sigma)$, then we can

guarantee the error x converging to zero. But u_s is a discontinuous function and is proportional to the parameter upper bound, which is usually very large. Using a saturation function can smooth the control signal. Since u_s is used to force the system trajectory to move along the sliding surface, a small σ value implies the u_N part is well behaved. Large stabilizer control signals are not necessary.

3.2 Learning process

The CMAC network is used to learn the optimal equivalent control signal u^* . If the CMAC learned the u^* , then the output trajectory will stay on the sliding surface and σ is equal to zero, insofar as we can let the sliding surface be the teacher of the CMAC. At each control cycle, the desired sliding surface determines the desired state of the system x_d for the next control cycle. The desired next state is sent to the CMAC network to produce u_N . u_N is assumed to be the estimation of the real equivalent control signal required to achieve the ideal sliding mode control. It adds the output of the $u_s(x)$ to form the controller signal u for the robot actuator drivers.

At the end of each control cycle a training step is executed. The observed sliding surface of the system during the previous control cycle is used to modify the weighting values. Generally the following simple steepest-descent update rule is used in most CMAC-based learning systems [11][17-18].

$$w_i = w_i + \beta \frac{(\sigma_d - \sigma)}{A^*} \quad (22)$$

In this equation, $\sigma_d = 0$ is the desired sliding surface, σ the actual output, A^* the number of association memory cells, and β is the learning factor.

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 sliding surface function σ_d that satisfies the Hurwitz stable condition.
- Step 2. Specify the design parameters M_x, M_u and M_w based on practical constraints.
- Step 3. Transform the sliding surface to the desired error states for the next control cycle $\sigma_d(t) \Rightarrow x_d(t)$

- Step 4. Send the desired next error states $x_d(t)$ to the CMAC network
- Step 5. Perform a series of mappings to transform the input values to the CMAC output $u_N(t)$
- Step 6. Calculate the stabilizer control signal $u_s(t)$
- Step 7. 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 control signal $u(t) = u_N(t) + u_s(t)$
- Step 8. Send $u(t)$ to plant to produce the actual output states $x_o(t)$
- Step 9. Calculate the values of the actual sliding surface function $\sigma(t)$
- Step 10. Update the fired weights using equation (22)
- $$w_i = w_i + \beta \frac{(\sigma_d - \sigma)}{A^*}$$
- Step 12. If $\|x_o(t)\| \leq \varepsilon_1$, stop. (ε_1 is a small positive value.) Otherwise, go to step 3.

4. Numerical example

4.1 Uncertain DC servo system

Considering a robot system driven by DC motors of Fig. 2, the open loop step response curve was shown in Fig. 3. From the response curves, the link 1 model can be identified as following:

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= 0x_1 - (33 \pm 5)x_2 - (300 \pm 50)u \end{aligned} \quad (23)$$

where $x_1 = \theta_d - \theta$. θ_d is the desired end-effector output and θ is the actual output. The simulation model for demonstrating the effect of the proposed scheme is described as following.

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -(33 + 5\sin t) \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} - \begin{bmatrix} 0 \\ 300 + 50\sin t \end{bmatrix} u(t) \quad (24)$$

4.2 Selection of sliding surface

As shown in Fig.4, we chose two segments sliding surface [15] and the sliding function described as follows:

$$\sigma_1 = -x_1 - \frac{1}{2a_{\max}} x_2^2 + x_0 \quad (25)$$

$$\sigma_2 = c_1 x_1 + x_2 \quad (26)$$

where $a_{\max}$ represents the maximum acceleration signal of error output and c_1 is the slope of deceleration segment. The segment σ_1 is used to steer the transient state to move along a predefined surface. The derivation of control law is same as segment σ_2 .

4.3 Simulation results and discussions

The simulation results are shown in Fig 5-6. The associated parameters are listed in Table1.

Table 1. Parameter value for figure 5-6

Para.	Quan. level	A^*	Mem. size	$a_{\max}$	c_1	M_x	M_u	M_w	ε	β
Fig. No.										
Fig. 5	15	5	1024	40	10	5	10	2	0.9	0.9
Fig. 6	15	5	1024	30	20	5	10	2	0.9	0.9

To examine the robustness property of the control scheme, we use a different sliding surface and keep the same controller structure to demonstrate the possibility of the proposed method. It is clear that the proposed approach can give an almost accurate servo tracking response to unknown parameters system (Fig. 5(b)(c)). Figs. 5(d) and 6(d) show the waveforms of control functions, and we see that the CMAC outputs are dominant gradually. The CMAC network can learn the approximated equivalent control on line. It is noted that the initial values of all the memory locations have small non-zero values. Since the initial state is located on the sliding surface, i.e. $\sigma = 0$, a zero initial value will lead to the system being unable to self-start.

5. Conclusion

In this paper, a new variable structure control scheme is proposed to solve the unknown parameter problems of VSC. A CMAC network is used to learn the equivalent control of VSC for the unknown system model. A stabilizer controller based on Lyapunov synthesis approach is designed to guarantee the stability of the proposed scheme in the sense that all signals involved are uniformly bounded. This non-parametric scheme doesn't need the system model to design the controller parameters. Using the generalization property of CMAC, the new control scheme can learn the approximated equivalent control signal on line and yield a robust sliding mode control. Applying this scheme to the simplified robot manipulator system, it can produce a variable structure control signal for different performance requirement or parameter bounds. The simulation results demonstrate the success of the proposed scheme to robot control with unknown parameters.

6. Reference

1. 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.
2. 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.
3. K. S. Yeung, Y. P. Chen, "A new controller design for manipulator using the theory of variable structure systems", *IEEE Trans. Automation Control*, vol. 33, no. 2, pp. 200-206, 1988.
4. C. C. Kung, C. C. Liao, "Fuzzy-sliding mode

- controller design for tracking control of non-linear system", *Proc. of American Control Conf.*, pp. 180-184, 1994.
5. S. Bentalba, A. E. Haijaji, A. Rachid, "Fuzzy sliding mode controller of mobile robot", *IEEE Conf. on Decision and Control*, pp.4264-4265, 1998
 6. B. Yoo, W. Ham, "Adaptive fuzzy sliding mode control of nonlinear system", *IEEE Trans. on Fuzzy Sytem*, vol. 6,no. 2, pp. 315-321, 1998.
 7. S. Kawamura, F. Miyazaki, S. Arimoto, "Realization of robot motion based on a learning method", *IEEE Trans. Syst., Man, Cybern.*, vol.18, no. 1, pp. 126-134, 1988.
 8. S. R. Oh, Z. Bien, Il Hong Suh, "A model algorithmic learning method for continuous-path Control of a robot manipulator," *Robotica*, vol. 8, pp. 31-36, 1990.
 9. W. T. Miller, R. P. Hewes, F. H. Glanz, and L. G. Kraft. "Real-time dynamical control of an industrial manipulator using a neural-network-based learning controller", *IEEE Trans. Robotics Automat.*, vol. 6, no.1, pp. 1-9, 1990.
 10. C. P. Bottura, M. C. M. Teixeira, M. J. Borden, "Stabilization of an inverted pendulum with CMAC in a variable structure controller", *Proc. of American Control Conf.*, San Diego, California, pp. 1042-1046, 1999.
 11. J. S. Albus, "A new approach to manipulator control: the cerebeller model articulation controller (CMAC)" , *Trans. ASME J. Dynam., Syst., Meas., and Contr.*, vol. 97, pp.220-227, 1975
 12. 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.
 13. K. S. Fu, R.C. Gonzalez, C.S.G. Lee," ROBOTICS: control, sensing, vision, and intelligence", McGRAW-Hill, 1987.
 14. L. X. Wang, "Stable adaptive fuzzy control of nonlinear system", *IEEE Trans. on Fuzzy Sytem*, vol. 1,no. 2, pp. 146-155, 1993.
 15. 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.
 16. A. Davari, and Z. Zhang, "Application of the three-segments variable structure systems", *Proc. Am. Control Conf.*, 1, pp. 62-63, 1991.
 17. M. Sznaier,"Robust control of dynamic systems using neuromorphic controller: A CMAC approach", *IEEE Conf. on Decision and Control*, pp.2710-2715, 1992.
 18. 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.

Fig. 1 Block diagram of proposed control scheme

Fig. 2. Photo of robot manipulator system

Fig. 3 Open loop step responses for different input levels

Fig. 4 Two segments sliding surface

Fig. 5(a) Weight value of memory cells for CMAC network

Fig. 6(a) Weight value of memory cells for CMAC network

Fig. 5(b) Desired and actual output state x1

Fig. 6(b) Desired and actual output state x1

Fig. 5(c) Desired and actual output state x2

Fig. 6(c) Desired and actual output state x2

Fig. 5(d) CMAC output and the u_s output

Fig. 6(d) CMAC output and the u_s output

Fig. 5(e) Phase plane plot of desired and actual output states

Fig. 6(e) Phase plane plot of desired and actual output states