Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 868415, 11 pages
http://dx.doi.org/10.1155/2014/868415

Research Article

Adaptive Sliding Mode Control of Chaos in Permanent Magnet Synchronous Motor via Fuzzy Neural Networks

Tat-Bao-Thien Nguyen,[1] Teh-Lu Liao,[1] and Jun-Juh Yan[2]

[1] *Department of Engineering Science, National Cheng Kung University, Tainan 701, Taiwan*
[2] *Department of Computer and Communication, Shu-Te University, Kaohsiung 824, Taiwan*

Correspondence should be addressed to Jun-Juh Yan; jjyan@mail.stu.edu.tw

Received 26 November 2013; Accepted 30 December 2013; Published 10 February 2014

Academic Editor: Lu Zhen

In this paper, based on fuzzy neural networks, we develop an adaptive sliding mode controller for chaos suppression and tracking control in a chaotic permanent magnet synchronous motor (PMSM) drive system. The proposed controller consists of two parts. The first is an adaptive sliding mode controller which employs a fuzzy neural network to estimate the unknown nonlinear models for constructing the sliding mode controller. The second is a compensational controller which adaptively compensates estimation errors. For stability analysis, the Lyapunov synthesis approach is used to ensure the stability of controlled systems. Finally, simulation results are provided to verify the validity and superiority of the proposed method.

1. Introduction

Nowadays, permanent magnet synchronous motors are extensively used in industrial applications because it possesses many advantageous merits. Due to high power to weight ratio, high torque to current ratio, fast response, high power factor, simple structure, and low maintaining cost, PMSMs were effectively applied to some fields of industry which require high performances [1–4]. Nevertheless, there are still numerous challenges in controlling a PMSM to get the superior performances, because it has highly nonlinear characteristics and chaotic motion.

The chaotic phenomenon in PMSM was comprehensively examined by Li et al. [5]. This study indicated that the chaotic oscillations occur when the system parameters lie in a certain region. Since the undesirable chaotic oscillations can break down the system stability or even cause the drive system to collapse, the chaos suppression and control in a PMSM have received much attention in the field of nonlinear control of electric motor. Until now, various control methods have been developed for chaos suppression and control in a PMSM, including nonlinear feedback control [6, 7], time delay feedback control [8–10], backstepping control [11, 12], sliding mode control [13], quasisliding mode control [14, 15], dynamic surface control [16], and adaptive control [17, 18]. However, shortcomings still exist in these methods. An exact mathematical model of a PMSM is necessary for these methods to calculate the control laws. This leads to difficulties in applying these control methods to a real-time system where the mathematical model might be dynamic and unknown due to parameter perturbations and noise disturbances. Moreover, time delay feedback control faces some problems when the control target is not an equilibrium point or located at unstable periodic orbit; determining the time delay is also difficult. In conventional sliding mode control, chattering often appears and it causes the heat loss in electrical power circuits and undesirable vibrations in mechanical systems leading to degrade the whole systems. Adaptive control can work well even when the parameters vary, but cannot solve the control problems when the mathematical model is deeply changed due to external noises.

In recent years, fuzzy logic and neural networks have exhibited the superior abilities in modeling and controlling the highly uncertain, ill-defined, and complex systems [19–22], especially in chaotic PMSM [23–25]. A fuzzy logic controller can incorporate the expert experience of a human

operator in the design of the controller in controlling a process whose input-output relationship is described by collection of fuzzy rules involving linguistic variables rather than a complicated dynamic model. On the other hand, neural networks have the potential for very complicated behavior. The strong learning abilities allow a neural network to generate input-output maps which can approximate any continuous function with the required degree of accuracy. These learning abilities equip neural networks to design controllers which do not depend on exact mathematical models. The combination of fuzzy logic system and neural networks is known as fuzzy neural networks [26, 27] in which a fuzzy logic system is expressed by a neural network. A fuzzy neural network can exploit the fuzzy inference of a fuzzy logic system and the learning abilities of a neural network. Then, the fuzzy neural networks become powerful and confident tools in controlling highly nonlinear and complex systems.

As the control methods mentioned above still have some weaknesses, it is necessary to develop a improved controller which can suppress chaos and obtain satisfied performance; even the mathematical model of PMSM is significantly varied due to parameter perturbations and external noise disturbances. In order to meet these requirements, based on a fuzzy neural network and incorporating the concept of sliding mode control, we successfully develop an adaptive sliding model control method. Since the developed controller is derived from sliding mode control, it can inherit the merits of sliding mode controller for controlling nonlinear systems. Moreover, the use of fuzzy neural networks gives the learning ability for the proposed controller to estimate unknown models existing in the system. These abilities allow the controller to operate effectively and robustly even with unknown system parameters of the PMSM. In contrast, many previous articles for chaos control of the PMSM depend on the mathematical model of PMSM; that is, an exact model of PMSM is necessary for designing controllers. This also implies that these controllers cannot work or work imprecisely when the system parameters or model of PMSM are not sufficiently known. Therefore, in comparison with previous articles, the proposed control shows the improvements in controlling chaotic PMSM. The developed controller cannot only suppress chaotic behaviors in a PMSM but also allow the motor speed to follow the desired trajectory, while the tracking error is led to zero despite of the existence of uncertainties. In addition, chattering phenomenon can be removed by choosing the suitable parameters for the designed controller. The robustness of the developed controller can give us the feasibility to realize the method in real-time system. Simulations results are provided to illustrate the effectiveness and robustness of the proposed controller.

The paper is organized as follows. In Section 2, the dynamics of a PMSM and the formulation of the chaos control problem are presented. The design of the adaptive sliding mode controller as well as the stability analysis is described in Section 3. In Section 4 the simulation results are displayed to verify the validity of the proposed method. Finally, the conclusion is given in Section 5.

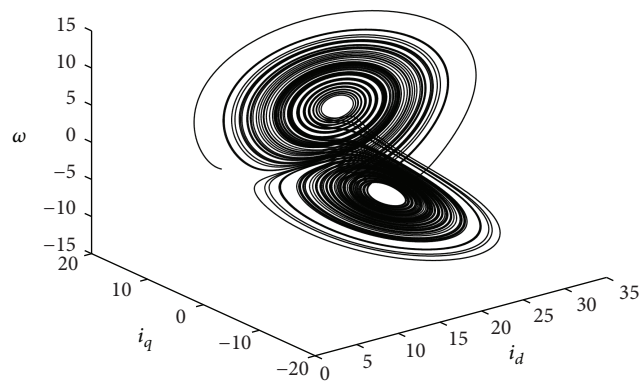

Figure 1: Chaotic motion in a PMSM with $\sigma = 5.45$ and $\gamma = 20$.

2. Problem Statement and Preliminaries

2.1. Mathematical Model of Chaotic PMSM. In dimensionless form, the mathematical model of a smooth-air-gap PMSM can be modeled as follows [5]:

$$\begin{aligned} \frac{d\omega}{dt} &= \sigma\left(i_q - \omega\right) + T, \\ \frac{di_q}{dt} &= -i_q - i_d\omega + \gamma\omega + u_q, \\ \frac{di_d}{dt} &= -i_d + i_q\omega + u_d, \end{aligned} \tag{1}$$

where i_d, i_q, and ω are state variables, which denote direct-quadrature currents and motor angular frequency, respectively. T_L, u_d, and u_q represent the load torque and direct-quadrature axis stator voltage components, respectively, while σ and γ are system parameters.

In system (1), after an operating period, the external inputs are set to zero, namely, $T_L = u_q = u_d = 0$. Then, the system in (1) becomes an unforced system as

$$\begin{aligned} \frac{d\omega}{dt} &= \sigma\left(i_q - \omega\right), \\ \frac{di_q}{dt} &= -i_q - i_d\omega + \gamma\omega, \\ \frac{di_d}{dt} &= -i_d + i_q\omega. \end{aligned} \tag{2}$$

The bifurcation and chaos phenomena of a PMSM drive system have been completely studied by Li et al. [5]. System (2) generates chaotic oscillations when the system parameters and initial condition are set as $\sigma = 5.45$, $\gamma = 20$, and $\lfloor \omega(0), i_q(0), i_d(0) \rfloor = [2, 1, 3]$. Figure 1 shows the typical chaotic motion of system (2). To make an overall inspection of the dynamical behavior of the PMSM, the bifurcation diagrams of the motor angular frequency ω versus the parameters σ and γ, respectively, are also plotted as shown in Figure 2. Since the chaotic oscillations in a PMSM can destroy the stability of drive system or lead the system to collapse, suppressing chaos, controlling speed, and ensuring

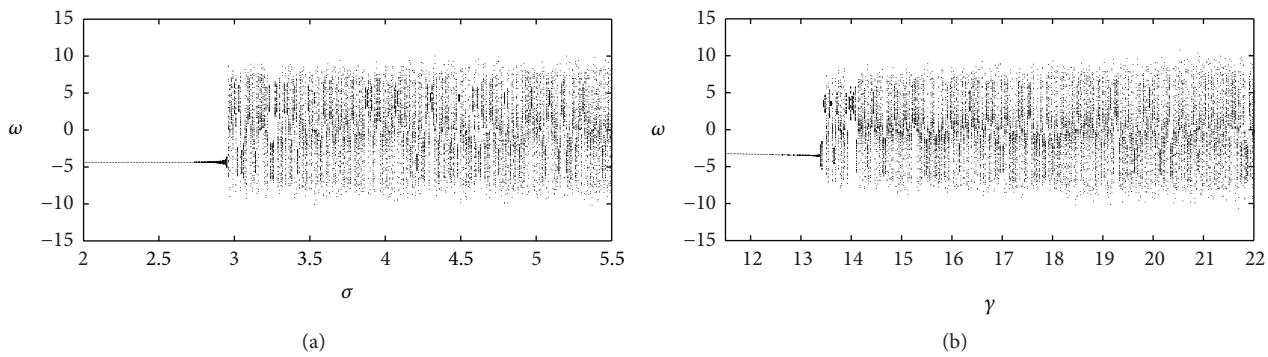

Figure 2: Bifurcation diagrams of ω versus (a) σ with $\gamma = 20$, (b) γ with $\sigma = 5.45$.

the robustness against uncertainties in a PMSM drive system are significantly necessary. In order to solve these problems, we propose the adaptive sliding mode control technique based on fuzzy neural networks.

2.2. Conventional Sliding Mode Control and Problem Statement. Let us consider the PMSM drive system as shown in (2). In order to control this system, we add a control signal u to the second differential equation as an adjustable variable which is desirable for real applications. And for simplicity, we introduce new notations as $x_1 = \omega$, $x_2 = i_q$, and $x_3 = i_d$. In this manner, the system in (2) with uncertainties can be rewritten as follows:

$$\begin{aligned} \dot{x}_1 &= \sigma(x_2 - x_1) + \Delta_1, \\ \dot{x}_2 &= -x_2 - x_1 x_3 + \gamma x_1 + \Delta_2 + u, \\ \dot{x}_3 &= -x_3 + x_1 x_2 + \Delta_3, \end{aligned} \tag{3}$$

where $\Delta_i \in R$, $i = 1, 2, 3$, are uncertainties applied to the PMSM due to parameter perturbation and external noise disturbances. σ and γ are unknown system parameters and located within the chaotic region [5].

Assumption 1. $\Delta_i \in R$, $i = 1, 2, 3$, are bounded functions; further Δ_3 is zero when $x_1 = x_2 = 0$.

For suppressing chaos and controlling speed in the PMSM, the system in (3) with output $y(t) = x_1$ can be expressed in the standard form of single-input-single-output (SISO) system as follows:

$$\dot{x} = f(x) + g(x)u, \tag{4a}$$

$$y = h(x), \tag{4b}$$

where

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \qquad f(x) = \begin{bmatrix} \sigma(x_2 - x_1) + \Delta_1 \\ -x_2 - x_1 x_3 + \gamma x_1 + \Delta_2 \\ -x_3 + x_1 x_2 + \Delta_3 \end{bmatrix},$$
$$g(x) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \qquad h(x) = x_1. \tag{5}$$

Taking the second order derivative of output $y(t)$ and the control signal u appearing in this expression, we can conclude that the SISO system in (4a) and (4b) has relative degree $r = 2$. Then using Lie derivative and letting $a(x) = L_f^2 h(x)$ and $b(x) = L_g L_f h(x)$, (4b) can be rewritten as

$$\ddot{y} = a(x) + b(x)u, \tag{6}$$

where

$$\begin{aligned} a(x) = L_f^2 h(x) &= \left(-\sigma + \frac{\partial \Delta_1}{\partial x_1}\right)(\sigma x_2 - \sigma x_1 + \Delta_1) \\ &\quad + \left(\sigma + \frac{\partial \Delta_1}{\partial x_2}\right)(-x_2 - x_1 x_3 + \gamma x_1 + \Delta_2) \\ &\quad + \frac{\partial \Delta_1}{\partial x_3}(-x_3 + x_1 x_3 + \Delta_3), \\ b(x) = L_g L_f h(x) &= \sigma + \frac{\partial \Delta_1}{\partial x_2}. \end{aligned} \tag{7}$$

In order to guarantee that the system in (4a) and (4b) is controllable for all $x \in R^3$ in our study, we need a following assumption.

Assumption 2. $b(x)$ is bounded from below by a positive constant $\underline{b}$; that is, $0 < \underline{b} \le b(x)$, for all $x \in R^3$.

The aim is to design a controller that can suppress chaos and allow the output $y(t) \in R$ to follow a given desired trajectory $y_d(t) \in R$.

Assumption 3. Desired trajectory $y_d(t)$ is smooth and bounded up to the 2nd order; $\dot{y}_d(t)$ and $\ddot{y}_d(t)$ are available for measurement.

Let $e(t) = y(t) - y_d(t)$ be tracking error; we define a switching surface $S(t)$ in the state space R^3 as

$$S(t) = \left(\frac{d}{dt} + \lambda\right) e(t) = \dot{e}(t) + \lambda e(t), \tag{8}$$

where λ is a positive constant. The equation $S(t) = 0$ represents a linear differential equation whose solution implies

that the tracking error $e(t)$ converges to zero with the time constant $1/\lambda$ [28]. Differentiating $S(t)$ with respect to time and using (6), we obtain

$$\begin{aligned}\dot{S}(t) &= \ddot{e}(t) + \lambda\dot{e}(t) \\ &= \ddot{y}(t) - \ddot{y}_d(t) + \lambda\dot{e}(t) \\ &= a(x) + b(x)u - \ddot{y}_d(t) + \lambda\dot{e}(t).\end{aligned} \tag{9}$$

Let $v(t)$ be a new input variable and it is defined by

$$v(t) = -\ddot{y}_d(t) + \lambda\dot{e}(t). \tag{10}$$

Then (9) with $v(t)$ defined in (10) can be rewritten as

$$\dot{S}(t) = a(x) + b(x)u + v(t). \tag{11}$$

In order to meet the control objective, the conventional sliding mode control law can be used as

$$u = \frac{1}{b(x)}\left(-a(x) - v(t) - kS(t)\right), \tag{12}$$

where k is a positive constant.

Substituting (12) into (11), one can get

$$\dot{S}(t) = -kS(t). \tag{13}$$

Equation (13) implies that both $S(t)$ and therefore $e(t)$ converge to zero exponentially fast.

Moreover, by setting $x_1 = x_2 = 0$ and using Assumption 1, the zero dynamics of the SISO system in (4a) and (4b) can be described as $\dot{x}_3 = -x_3 + \Delta_3 = -x_3$. Because the zero dynamics is stable, we can conclude that the system in (4a) and (4b) is a minimum phase system. Thus the state variable x_3 is also stable when both state variables, x_1 and x_2, are stable.

Since the uncertainties $\Delta_i \in R$, $i = 1, 2, 3$, and system parameters σ and γ are unknown, the function $a(x)$ and $b(x)$ cannot be known exactly. The control law in (12) can no longer be used to control the system. In order to solve this problem, we develop an adaptive sliding mode control method in which a neural network is employed to estimate $a(x)$ and $b(x)$ online.

2.3. Description of Fuzzy Neural Networks. In this section, we describe the structure of a fuzzy neural network which is used to estimate the unknown nonlinear functions $a(x)$ and $b(x)$.

Let us start with the fuzzy logic system. The basic structure of a fuzzy logic system consists of input fuzzification, fuzzy rule base, fuzzy inference engine, and output defuzzification. In our study, the input fuzzification is the process of mapping inputs, state variable x_1, x_2, and x_3, to membership values in the input universes of discourse. The fuzzy rule base is made of nine IF-THEN rules in which the ith rule is described in the form of

$$\begin{aligned}\mathrm{FR}^i : \text{IF } &\left(x_1 \text{ is } A_1^i \text{ AND } x_2 \text{ is } A_2^i \text{ AND } x_3 \text{ is } A_3^i\right) \\ \text{THEN } &\left(\hat{a} \text{ is } B_a^i \text{ AND } \hat{b} \text{ is } B_b^i\right),\end{aligned} \tag{14}$$

Table 1: Parameters of Gaussian functions.

Parameter	Value
m_j^1	−1
m_j^2	−0.75
m_j^3	−0.5
m_j^4	−0.25
m_j^5	0
m_j^6	0.25
m_j^7	0.5
m_j^8	0.75
m_j^9	1
n	0.2

where A_1^i, A_2^i, A_3^i, B_a^i, and B_b^i are fuzzy sets which are represented by the membership functions $\mu_{A_1^i}$, $\mu_{A_2^i}$, $\mu_{A_3^i}$, $\mu_{B_a^i}$, and $\mu_{B_b^i}$, respectively. $\hat{a}(x) \in R$ and $\hat{b}(x) \in R$ are outputs of the fuzzy logic system, which stand for the estimations of $a(x)$ and $b(x)$, respectively. $\mu_{B_a^i}$ and $\mu_{B_b^i}$ are fuzzy singletons, while $\mu_{A_1^i}$, $\mu_{A_2^i}$, and $\mu_{A_3^i}$ use Gaussian functions to calculate its values as the following form:

$$\mu_{A_j^i} = \exp\left[-\frac{\left(x_j - m_j^i\right)^2}{2n^2}\right], \tag{15}$$

where $i = 1, 2, \ldots, 9$ correspond with nine rules and $j = 1, 2, 3$ correspond with three state variables. As the state variables are normalized in a range of $[-1, 1]$, the parameters of the chosen Gaussian functions are given in the Table 1. The fuzzy inference engine performs as a process of mapping membership values from the input windows, through the fuzzy rule base, to the output window. The fuzzy inference engine employs product inference for mapping. The output defuzzification is the procedure of mapping from a set of inferred fuzzy signals contained within a fuzzy output window to a crisp signal. Based on center-average defuzzification techniques, the outputs of the fuzzy logic system can be expressed as follows:

$$\begin{aligned}\hat{a}(x) &= \frac{\sum_{i=1}^{9}\theta_{ai}\left(\prod_{j=1}^{3}\mu_{A_j^i}(x)\right)}{\sum_{i=1}^{9}\left(\prod_{j=1}^{3}\mu_{A_j^i}(x)\right)} = \theta_a^T\varphi(x), \\ \hat{b}(x) &= \frac{\sum_{i=1}^{9}\theta_{bi}\left(\prod_{j=1}^{3}\mu_{A_j^i}(x)\right)}{\sum_{i=1}^{9}\left(\prod_{j=1}^{3}\mu_{A_j^i}(x)\right)} = \theta_b^T\varphi(x),\end{aligned} \tag{16}$$

where $\theta_a^T = [\theta_{a1} \ \theta_{a2} \ \cdots \ \theta_{a9}]$ and $\theta_b^T = [\theta_{b1} \ \theta_{b2} \ \cdots \ \theta_{b9}]$ are weighting vectors adjusted according to the adaptive laws described in the next section. The fuzzy singletons $\mu_{B_a^i}$ and $\mu_{B_b^i}$, respectively, achieve maximum values at the points θ_{ai} and θ_{bi} with $i = 1, 2, \ldots, 9$; that is, $\mu_{B_a^i}(\theta_{ai}) = \mu_{B_b^i}(\theta_{bi}) = 1$.

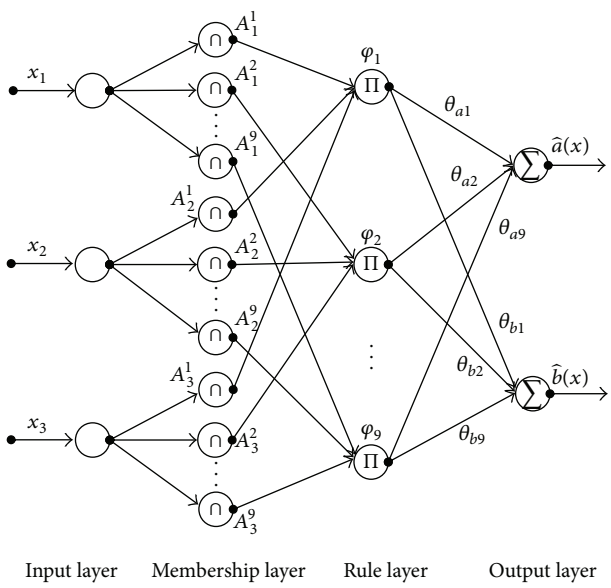

Figure 3: Structure of a fuzzy neural network.

$\varphi^T(x) = [\varphi_1(x) \ \ \varphi_2(x) \ \ \cdots \ \ \varphi_9(x)]$ is a fuzzy basic vector where each element $\varphi_i(x)$, $i = 1, 2, \ldots, 9$ is defined as

$$\varphi_i(x) = \frac{\prod_{j=1}^{3} \mu_{A_j^i}(x)}{\sum_{i=1}^{9} \left(\prod_{j=1}^{3} \mu_{A_j^i}(x) \right)}. \tag{17}$$

In order to exploit the fuzzy inference of a fuzzy logic system and the learning abilities of a neural network, a fuzzy logic system is expressed by a neural network which is known as a fuzzy neural network [26, 27]. By this way, the parameters in a fuzzy logic system can be found by a neural network through learning processes. As shown in Figure 3, the fuzzy neural network has four layers, including input layer, membership layer, rule layer, and output layer. There are three nodes in the input layer and each node is an input representing a state variable. The membership layer comprises twenty-seven nodes, each of which acts as a membership function and employs a Gaussian function to calculate the membership value. The rule layer has nine nodes, each node stands for an element $\varphi_i(x)$ of the fuzzy basis vector $\varphi(x)$ and performs a fuzzy rule. The links between the rule layer and the output layer are fully connected by weighting factors $\theta_{a1}, \theta_{a2}, \ldots, \theta_{a9}$ and $\theta_{b1}, \theta_{b2}, \ldots, \theta_{b9}$, which are the elements of weighting vector θ_a and θ_b, respectively. These factors are considered as parameters and adjusted in accordance with adaptive laws explained in the next section. In the output layer, two outputs represent the values of $\widehat{a}(x)$ and $\widehat{b}(x)$.

Therefore, the given fuzzy neural network has a fixed structure with four layers and nine fuzzy rules, while the parameter learning is governed by adaptive laws. This simple structure, as shown in Figure 3, allows the network to experience the low computational burden. For this reason, the cost of the system can be reduced and the controller can be implemented in real-time systems feasibly.

3. Design of Adaptive Sliding Mode Controller

When $a(x)$ and $b(x)$ in (7) cannot be determined exactly due to unknown parameters σ, γ and uncertainties Δ_i, $i = 1, 2, 3$, the conventional sliding mode controller in (12) cannot be used. In order to overcome this obstacle, we used a fuzzy neural network, as shown in Figure 3, to estimate $a(x)$ and $b(x)$ online. Then following the certainty equivalent approach, the adaptive sliding mode controller u_{asd}, which is modified from the conventional controller in (12), can be obtained as

$$u_{\text{asd}} = \frac{1}{\widehat{b}(x,t)} \left(-\widehat{a}(x,t) - v(t) - kS(t) \right), \tag{18}$$

where $\widehat{a}(x,t)$ and $\widehat{b}(x,t)$ are the online estimations of $a(x)$ and $b(x)$, respectively, and calculated by a fuzzy neural network as follows:

$$\begin{aligned} \widehat{a}(x,t) &= \theta_a^T(t)\,\varphi(x), \\ \widehat{b}(x,t) &= \theta_b^T(t)\,\varphi(x), \end{aligned} \tag{19}$$

where $\theta_a^T(t) = [\theta_{a1}(t) \ \ \theta_{a2}(t) \ \ \cdots \ \ \theta_{a9}(t)]$ and $\theta_b^T(t) = [\theta_{b1}(t) \ \ \theta_{b2}(t) \ \ \cdots \ \ \theta_{b9}(t)]$ are weighting vectors as depicted in the output layer of the neural network, while $\varphi^T(x) = [\varphi_1(x) \ \ \varphi_2(x) \ \ \cdots \ \ \varphi_9(x)]$ is the fuzzy basic vector of which each element $\varphi_i(x)$, $i = 1, 2, \ldots, 9$ is mentioned in (17). When the controller operates, the values of weighting vectors $\theta_a^T(t)$ and $\theta_b^T(t)$ are adjusted, so that $\widehat{a}(x,t)$ and $\widehat{b}(x,t)$ reach $a(x)$ and $b(x)$, respectively. The adaptive laws for $\theta_a^T(t)$ and $\theta_b^T(t)$ are chosen as follows:

$$\begin{aligned} \dot{\theta}_a(t) &= W_a^{-1} \varphi(x) S(t), \\ \dot{\theta}_b(t) &= W_b^{-1} \varphi(x) S(t) u_{\text{asd}}, \end{aligned} \tag{20}$$

where W_a and W_b are positive-definite weighting matrices.

In the adaptive mechanism, once $\widehat{a}(x,t)$ and $\widehat{b}(x,t)$, respectively, converge to $a(x)$ and $b(x)$, $\theta_a(t)$ and $\theta_b(t)$ reach their optimal values θ_a^* and θ_b^*, respectively. The achieved optimal weighting vectors θ_a^* and θ_b^* are defined by

$$\begin{aligned} \theta_a^* &= \arg\min_{\theta_a \in \Theta_a} \left\{ \sup_{x \in \Omega} \left| \theta_a^T(t)\,\varphi(x) - a(x) \right| \right\}, \\ \theta_b^* &= \arg\min_{\theta_b \in \Theta_b} \left\{ \sup_{x \in \Omega} \left| \theta_b^T(t)\,\varphi(x) - b(x) \right| \right\}, \end{aligned} \tag{21}$$

where Θ_a and Θ_b are sets of acceptable values of vector $\theta_a(t)$ and $\theta_b(t)$, respectively, and Ω is a compact set of state variable x.

In the ideal case, $\widehat{a}(x,t)$ and $\widehat{b}(x,t)$, respectively, approach to $a(x)$ and $b(x)$ when $\theta_a(t)$ and $\theta_b(t)$ approach to θ_a^* and θ_b^*, respectively. However, the estimations are carried out by a neural network which has a finite number of units in the hidden layer; the estimation errors are unable to avoid, namely, $\widehat{a}(x,t)$ and $\widehat{b}(x,t)$ cannot completely converge to $a(x)$ and $b(x)$ when $\theta_a(t)$ and $\theta_b(t)$ converge to θ_a^* and θ_b^*,

respectively. Let $\delta_a(x)$ and $\delta_b(x)$ be the estimation errors; then the exact models of $a(x)$ and $b(x)$ can be expressed by:

$$\begin{aligned} a(x) &= \theta_a^* \varphi(x) + \delta_a(x), \\ b(x) &= \theta_b^* \varphi(x) + \delta_b(x). \end{aligned} \tag{22}$$

We suppose that the estimation errors are bounded according to a following assumption.

Assumption 4. The estimation errors are bounded above by some known constants $\overline{\delta}_a > 0$ and $\overline{\delta}_b > 0$ over the compact set $\Omega \subset R^3$; that is,

$$\begin{aligned} \sup_{x\in\Omega} |\delta_a(x)| &\le \overline{\delta}_a, \\ \sup_{x\in\Omega} |\delta_b(x)| &\le \overline{\delta}_b. \end{aligned} \tag{23}$$

The differences between the estimation models and exact models can be computed as follows:

$$\begin{aligned} \widehat{a}(x,t) - a(x) &= \left(\theta_a(t) - \theta_a^*\right)^T \varphi(x) - \delta_a(x) \\ &= \widetilde{\theta}_a^T(t)\varphi(x) - \delta_a(x), \\ \widehat{b}(x,t) - b(x) &= \left(\theta_b(t) - \theta_b^*\right)^T \varphi(x) - \delta_b(x) \\ &= \widetilde{\theta}_b^T(t)\varphi(x) - \delta_b(x), \end{aligned} \tag{24}$$

where $\widetilde{\theta}_a(t) = \theta_a(t) - \theta_a^*$ and $\widetilde{\theta}_b(t) = \theta_b(t) - \theta_b^*$ are parameter errors.

Since the estimation errors exist, the stability of closed-loop system may be lost under only action of the adaptive sliding mode controller u_{asd}. In order to repress the undesirable effect of estimation errors and keep the system robust, a compensational controller u_{cc} is used as an additional controller. This controller is able to compensate the estimation errors and its formula is given as

$$u_{\text{cc}} = -\frac{1}{\underline{b}}\left(\overline{\delta}_a + \overline{\delta}_b |u_{\text{asd}}|\right) \operatorname{sgn}(S(t)). \tag{25}$$

Therefore, the whole controller u has two components; the first one is the adaptive sliding mode controller u_{asd} and the second one is compensational controller u_{cc}. The overall scheme of the controlled system is illustrated in Figure 4 and the total control signal is given as

$$\begin{aligned} u = u_{\text{asd}} + u_{\text{cc}} &= \frac{1}{\widehat{b}(x,t)}\left(-\widehat{a}(x,t) - v(t) - kS(t)\right) \\ &\quad - \frac{1}{\underline{b}}\left(\overline{\delta}_a + \overline{\delta}_b |u_{\text{asd}}|\right) \operatorname{sgn}(S(t)). \end{aligned} \tag{26}$$

Theorem 5. *Consider the system in* (3) *and the control law* (26) *with the adaptive laws* (20). *Assume that Assumptions 1–4 hold; then under the effect of the controller, chaos in the PMSM can be suppressed and its speed can track the desired trajectory successfully and the tracking error converges to zero asymptotically fast.*

Proof. Using (11) and (26), then taking some basic algebraic manipulations, one can obtain

$$\begin{aligned} \dot{S}(t) &= a(x) + b(x)u + v(t) \\ &= a(x) + v(t) + b(x)\left(u_{\text{asd}} + u_{\text{cc}}\right) \\ &= a(x) + v(t) + \widehat{b}(x,t)u_{\text{asd}} \\ &\quad + \left(b(x) - \widehat{b}(x,t)\right)u_{\text{asd}} + b(x)u_{\text{cc}}. \end{aligned} \tag{27}$$

Replacing u_{asd} in (27) by its expression in (18), (27) can be rewritten as

$$\begin{aligned} \dot{S}(t) &= a(x) + v(t) + \widehat{b}(x,t)u_{\text{asd}} \\ &\quad + \left(b(x) - \widehat{b}(x,t)\right)u_{\text{asd}} + b(x)u_{\text{cc}} \\ &= a(x) + v(t) + \left(-\widehat{a}(x,t) - v(t) - kS(t)\right) \\ &\quad + \left(b(x) - \widehat{b}(x,t)\right)u_{\text{asd}} + b(x)u_{\text{cc}} \\ &= -kS(t) + \left(a(x) - \widehat{a}(x,t)\right) \\ &\quad + \left(b(x) - \widehat{b}(x,t)\right)u_{\text{asd}} + b(x)u_{\text{cc}}. \end{aligned} \tag{28}$$

Substituting (24) into (28) yields

$$\begin{aligned} \dot{S}(t) &= -kS(t) + \left(a(x) - \widehat{a}(x,t)\right) \\ &\quad + \left(b(x) - \widehat{b}(x,t)\right)u_{\text{asd}} + b(x)u_{\text{cc}} \\ &= -kS(t) - \left(\widetilde{\theta}_a^T(t)\varphi(x) - \delta_a(x)\right) \\ &\quad - \left(\widetilde{\theta}_b^T(t)\varphi(x) - \delta_b(x)\right)u_{\text{asd}} + b(x)u_{\text{cc}}. \end{aligned} \tag{29}$$

Now we consider a Lyapunov function to study the stability of the system as follows:

$$V(t) = \frac{1}{2}S^2(t) + \frac{1}{2}\widetilde{\theta}_a^T(t)W_a\widetilde{\theta}_a(t) + \frac{1}{2}\widetilde{\theta}_b^T(t)W_b\widetilde{\theta}_b(t). \tag{30}$$

Taking the time derivative of $V(t)$ and noticing that $\dot{\widetilde{\theta}}_a = \dot{\theta}_a$, $\dot{\widetilde{\theta}}_b = \dot{\theta}_b$, one can get

$$\begin{aligned} \dot{V}(t) &= S(t)\dot{S}(t) + \frac{1}{2}\dot{\widetilde{\theta}}_a^T(t)W_a\widetilde{\theta}_a(t) + \frac{1}{2}\widetilde{\theta}_a^T(t)W_a\dot{\widetilde{\theta}}_a(t) \\ &\quad + \frac{1}{2}\dot{\widetilde{\theta}}_b^T(t)W_b\widetilde{\theta}_b(t) + \frac{1}{2}\widetilde{\theta}_b^T(t)W_b\dot{\widetilde{\theta}}_b(t) \\ &= S(t)\dot{S}(t) + \widetilde{\theta}_a^T(t)W_a\dot{\theta}_a(t) + \widetilde{\theta}_b^T(t)W_b\dot{\theta}_b(t). \end{aligned} \tag{31}$$

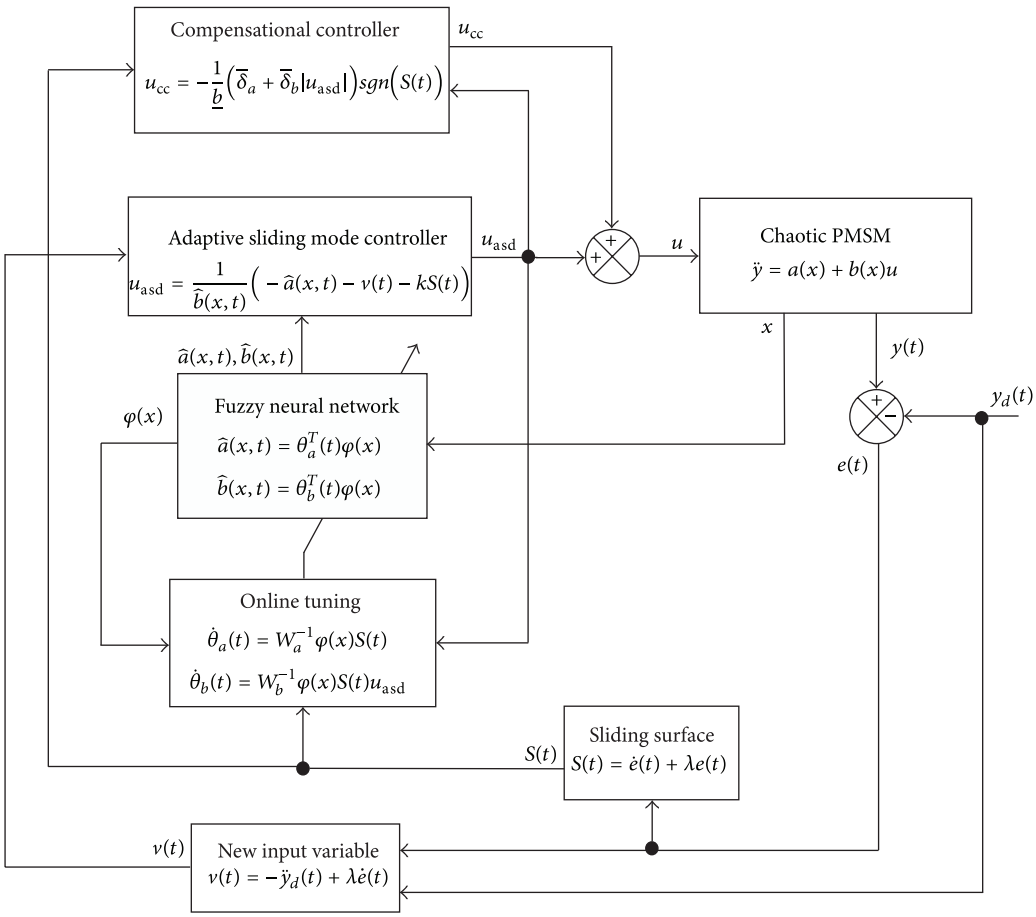

Figure 4: Overall scheme of controlled system.

Substituting (29) into (31), $\dot{V}(t)$ can be rewritten as

$$\begin{aligned}\dot{V}(t) &= S(t)\dot{S}(t) + \widetilde{\theta}_a^T(t)W_a\dot{\theta}_a(t) + \widetilde{\theta}_b^T(t)W_b\dot{\theta}_b(t)\\ &= -kS^2(t) - \left(\widetilde{\theta}_a^T(t)\varphi(x) - \delta_a(x)\right)S(t)\\ &\quad - \left(\widetilde{\theta}_b^T(t)\varphi(x) - \delta_b(x)\right)S(t)u_{asd}\\ &\quad + b(x)S(t)u_{cc} + \widetilde{\theta}_a^T(t)W_a\dot{\theta}_a(t) + \widetilde{\theta}_b^T(t)W_b\dot{\theta}_b(t)\\ &= -kS^2(t) + \widetilde{\theta}_a^T(t)\left(W_a\dot{\theta}_a(t) - \varphi(x)S(t)\right)\\ &\quad + \widetilde{\theta}_b^T(t)\left(W_b\dot{\theta}_b(t) - \varphi(x)S(t)u_{asd}\right)\\ &\quad + b(x)S(t)u_{cc} + S(t)\delta_a(x) + S(t)\delta_b(x)u_{asd}.\end{aligned} \tag{32}$$

Replacing $\dot{\theta}_a(t)$ and $\dot{\theta}_b(t)$ in (32) by their expression in adaptive laws (20) and (32) can be rewritten as

$$\begin{aligned}\dot{V}(t) &= -kS^2(t) + b(x)S(t)u_{cc} + S(t)\delta_a(x)\\ &\quad + S(t)\delta_b(x)u_{asd}\\ &\le -kS^2(t) + b(x)S(t)u_{cc}\\ &\quad + |S(t)|\left(|\delta_a(x)| + |\delta_b(x)||u_{asd}|\right)\\ &\le -kS^2(t) + b(x)S(t)u_{cc} + |S(t)|\left(\overline{\delta}_a + \overline{\delta}_b|u_{asd}|\right).\end{aligned} \tag{33}$$

Substituting the compensational controller in (25) into (33) and noticing that $\mathrm{sgn}(S(t))S(t) = |S(t)|$, one can obtain

$$\begin{aligned}\dot{V}(t) &\le -kS^2(t) - \frac{b(x)}{\underline{b}}\left(\overline{\delta}_a + \overline{\delta}_b|u_{asd}|\right)\mathrm{sgn}(S(t))S(t)\\ &\quad + |S(t)|\left(\overline{\delta}_a + \overline{\delta}_b|u_{asd}|\right)\\ &\le -kS^2(t) - \left(\frac{b(x)}{\underline{b}} - 1\right)\left(\overline{\delta}_a + \overline{\delta}_b|u_{asd}|\right)|S(t)| \le 0.\end{aligned} \tag{34}$$

From (30) and (34), we can find that $V(t) > 0$ and $\dot{V}(t) \le 0$. For these reasons, the close-loop controlled system is stable. Also, $S(t) \in L_\infty$, $\|\widetilde{\theta}_a(t)\| \in L_\infty$, and $\|\widetilde{\theta}_b(t)\| \in L_\infty$ can be determined.

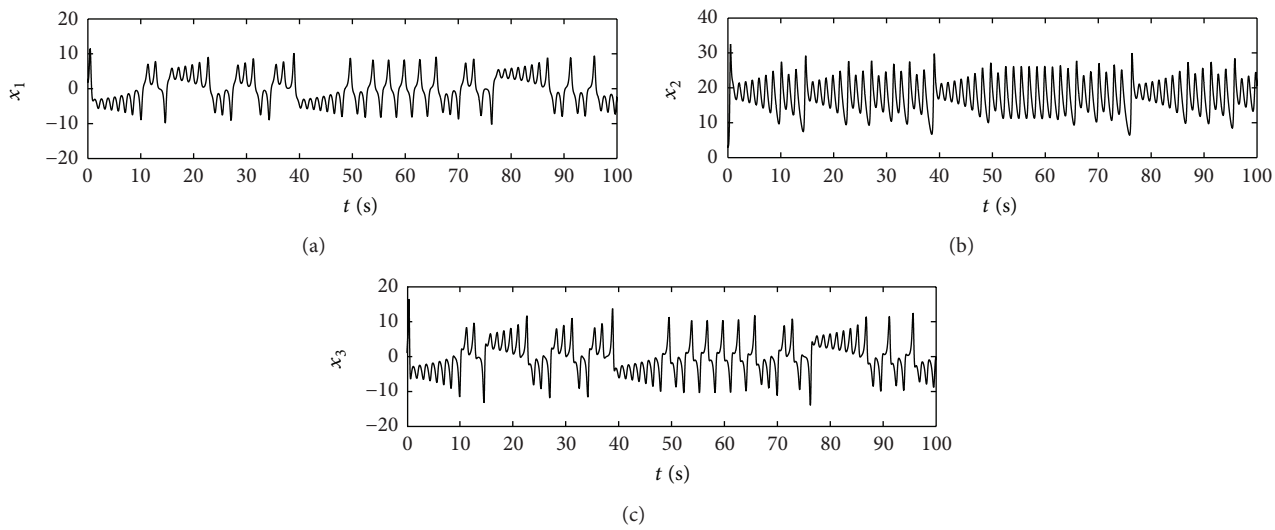

Figure 5: Chaotic oscillations of an uncontrolled PMSM.

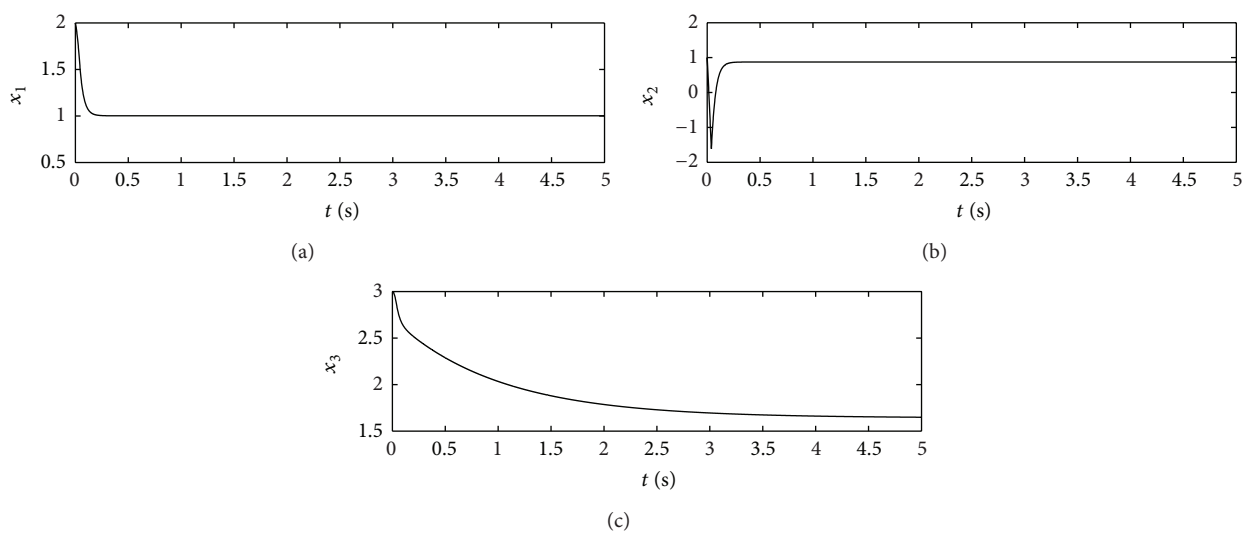

Figure 6: Chaos suppression under the controller action.

Further, from the inequality in (34), we have the following result:

$$\int_0^\infty kS^2(t)\,dt \le -\int_0^\infty \dot{V}(t)\,dt = V(0) - V(\infty) < \infty. \tag{35}$$

The inequality in (35) implies that $S(t) \in L_2$, leading to $S(t) \in L_2 \cap L_\infty$. On the other hand, because of (8), we can obtain $e(t) \in L_\infty$, $\dot{e}(t) \in L_\infty$, and $\dot{S}(t) \in L_\infty$. Then, incorporating Barbalat's lemma [28] yields $\lim_{t\to\infty} S(t) = 0$, so $\lim_{t\to\infty} e(t) = 0$. Therefore, the system stability is ensured and the perfect tracking performance is achieved. This proof is finished. □

4. Simulation Study

Here numerical simulations are carried out to verify the validity of the proposed method. The system parameters and initial conditions are kept the same as above; namely, $\sigma = 5.45$, $\gamma = 20$, and $[x_1(0), x_2(0), x_3(0)] = [2, 1, 3]$ are maintained.

First, the uncontrolled system is considered. The behavior of the system without the action of the controller is simulated over 100 seconds. As a result shown in Figure 5, all state variables experience chaotic oscillations separately. Then, for examining the ability of chaos suppression, we set the desired value $y_d(t) = 1$ and let the controller be operated since the beginning time. As displayed in Figure 6, the incipient chaos is quickly suppressed when the controller is active at the first of period time, and all state variables converge to constant values asymptotically fast.

Second, the proposed controller is employed to repress chaos and track the desired speed in a PMSM. The simulation is implemented with the presence of uncertainties and perturbation of system parameters. The simulation time is 40 s and the controller is turned on at time $t = 10$ s. The system

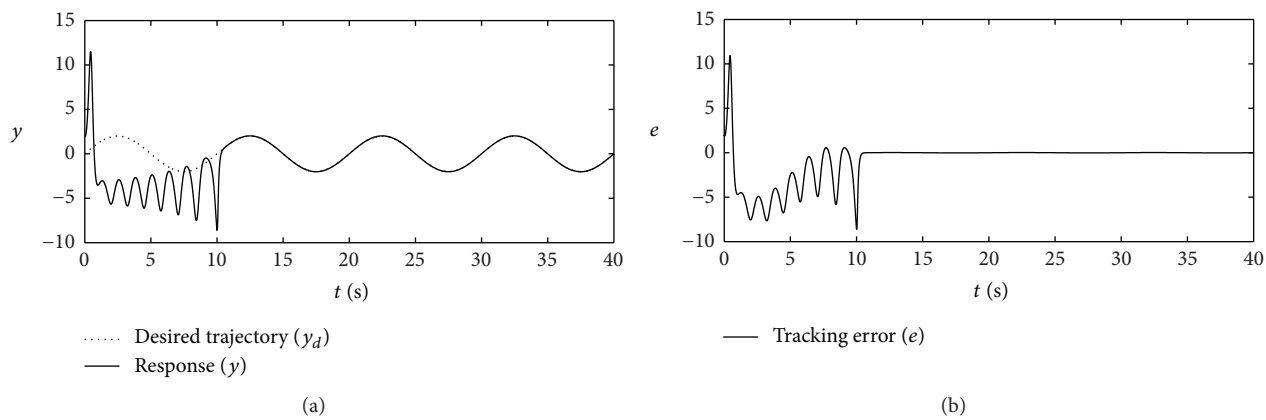

Figure 7: Speed tracking of the chaotic PMSM when the controller is turned on at time $t = 10$ s.

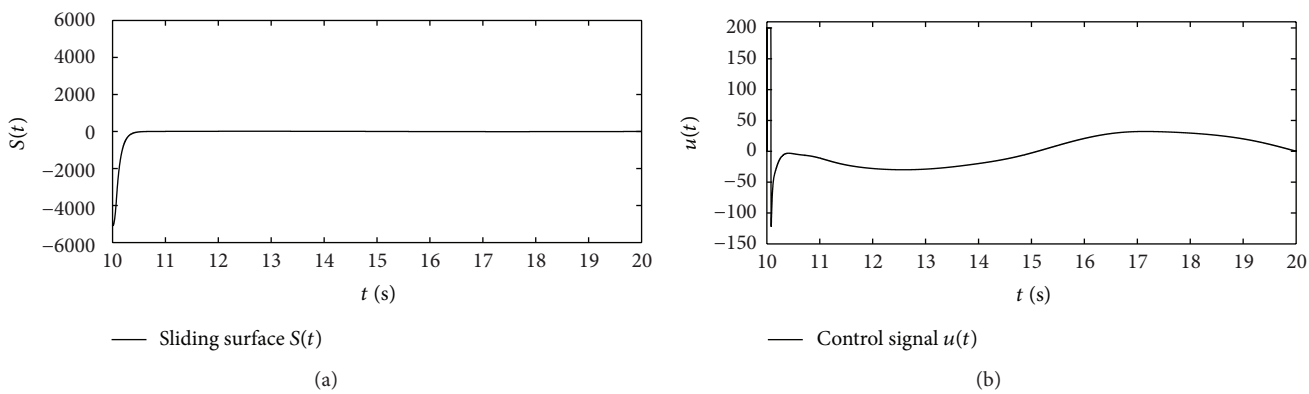

Figure 8: Sliding surface and control signal when the controller is active.

parameters are chosen in such a way that they can vary within the chaotic region [5]. One can choose $\sigma = 5.45 + 0.1\sin(x_1)$ and $\gamma = 20 + \cos(x_3)$ to meet the requirement for chaotic region. On the other hand, for satisfying Assumptions 1 and 2, the uncertainties can be chosen as $\Delta_1 = 1 + \cos(x_1 + x_3)$, $\Delta_2 = 1$, and $\Delta_3 = \sin(x_2)$. The desired trajectory $y_d(t) = 2\sin((\pi/5)t)$, which satisfies Assumption 3, is assigned for this simulation, while the control parameters are specified as follows:

$$\lambda = 587.9, \qquad k = 7048.6, \qquad \underline{b} = 1, \qquad \overline{\delta}_a = \overline{\delta}_b = 0.01,$$
$$W_a = 3368 * \text{eye}(9), \qquad W_b = 9569.7 * \text{eye}(9). \tag{36}$$

The results, as depicted in Figures 7–9, demonstrate that the chaotic oscillations are completely suppressed and the speed of PMSM perfectly follows the desired trajectory, while the tracking error asymptotically converges to zero when the controller is turned on at time $t = 10$ s. As displayed in Figure 7(a), the tracking performance is illustrated over the simulation time. The response $y(t)$, which is denoted by a solid line, nearly overlaps the desired trajectory $y_d(t) = 2\sin((\pi/5)t)$, which is represented by a dotted line, after the 10th second. Also, the tracking error is described in Figure 7(b), where the tracking error converges to zero asymptotically fast when the controller is turned on at time $t = 10$ s. In Figure 8, the sliding surface $S(t)$ and controller force $u(t)$ are shown in the period of the 10th second to the 20th second. After the controller starts, the value of switching surface converges to zero speedily. It is also noticeable that the chattering phenomenon, which is usually considered as a drawback of conventional sliding model control, does not appear in our design. On the other hand, the responses of all state variables are expressed in Figure 9 and they demonstrate that the chaotic motion in PMSM is suppressed quickly when the controller runs.

5. Conclusion

Based on fuzzy neural networks, the adaptive sliding mode control scheme cannot only completely suppress chaos but also successfully track the desired speed in an uncertain chaotic permanent magnet synchronous motor. By choosing the appropriate controller parameters, chattering phenomenon can be avoided instead of compromise in conventional sliding mode control. In addition, because the adaptive laws are derived from Lyapunov function, the system stability is guaranteed and perfect tracking performance is ensured even if the uncertainties affect the system. Numerical

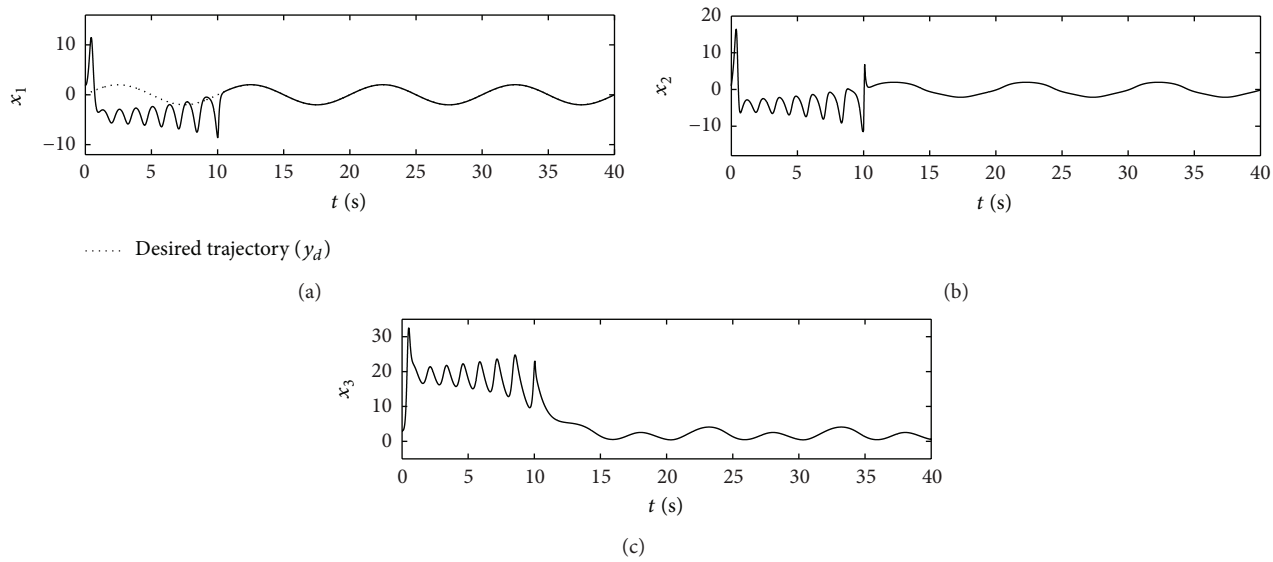

Figure 9: State responses of the chaotic PMSM when the controller is turned on at time $t = 10$ s.

simulations were realized to demonstrate the effectiveness and robustness of the proposed method.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This paper was supported by the National Science Council of Taiwan under Contracts NSC101-2221-E-006-190-MY2 and NSC102-2221-E-366-003.

References

[1] P. Pillay and R. Krishnan, "Control characteristics and speed controller design for a high performance permanent magnet synchronous motor drive," *IEEE Transactions on Power Electronics*, vol. 5, no. 2, pp. 151–159, 1989.

[2] A. Bara, C. Rusu, and S. Dale, "DSP Application on PMSM drive control for robot axis," in *Proceedings of the 13th WSEAS International Conference on Systems*, pp. 381–385, Rodos, Greece, July 2009.

[3] C.-H. Lin and C.-P. Lin, "The hybrid RFNN control for a PMSM drive electric scooter using rotor flux estimator," *Advances in Fuzzy Systems*, vol. 2012, Article ID 319828, 11 pages, 2012.

[4] C. M. Verrelli, "Synchronization of permanent magnet electric motors: new nonlinear advanced results," *Nonlinear Analysis: Real World Applications*, vol. 13, no. 1, pp. 395–409, 2012.

[5] Z. Li, J. B. Park, Y. H. Joo, B. Zhang, and G. Chen, "Bifurcations and chaos in a permanent-magnet synchronous motor," *IEEE Transactions on Circuits and Systems I*, vol. 49, no. 3, pp. 383–387, 2002.

[6] H. Ren and D. Liu, "Nonlinear feedback control of chaos in permanent magnet synchronous motor," *IEEE Transactions on Circuits and Systems II*, vol. 53, no. 1, pp. 45–50, 2006.

[7] D. Liu, H. Ren, and X. Liu, "Chaos control in permanent magnet synchronous motor," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '04)*, vol. 4, pp. IV-732–IV-735, Vancouver, Canada, May 2004.

[8] Y. Gao and K. T. Chau, "Chaotification of permanent-magnet synchronous motor drives using time-delay feedback," in *Proceedings of the 28th Annual Conference of the IEEE Industrial Electronics Society (IECON '02)*, vol. 1, pp. 762–766, Sevilla, Spain, November 2002.

[9] H. P. Ren and C. Z. Han, "Chaotifying control of permanent magnet synchronous motor," in *Proceedings of the CES/IEEE 5th International Power Electronics and Motion Control Conference (IPEMC '06)*, vol. 1, pp. 1–5, Shanghai, China, August 2006.

[10] Z. Meng, C. Sun, Y. An, J. Cao, and P. Gao, "Chaos anti-control of permanent magnet synchronous motor based on model matching," in *Proceedings of the International Conference on Electrical Machines and Systems (ICEMS '07)*, pp. 1748–1752, Seoul, Republic of Korea, October 2007.

[11] A. M. Harb, "Nonlinear chaos control in a permanent magnet reluctance machine," *Chaos, Solitons and Fractals*, vol. 19, no. 5, pp. 1217–1224, 2004.

[12] X. Ge and J. Huang, "Chaos control of permanent magnet synchronous motor," in *Proceedings of the 8th International Conference on Electrical Machines and Systems (ICEMS '05)*, vol. 1, pp. 484–488, Nanjing, China, September 2005.

[13] C. Ma, L. Wang, Z. Yin, J. Liu, and D. Chen, "Sliding mode control of chaos in the noise-perturbed permanent magnet synchronous motor with non-smooth air-gap," *Mining Science and Technology*, vol. 21, no. 6, pp. 835–838, 2011.

[14] C.-F. Huang, J.-S. Lin, T.-L. Liao, C.-Y. Chen, and J.-J. Yan, "Quasi-sliding mode control of chaos in permanent magnet synchronous motor," *Mathematical Problems in Engineering*, vol. 2011, Article ID 964240, 10 pages, 2011.

[15] C.-F. Huang, T.-L. Liao, C.-Y. Chen, and J.-J. Yan, "The design of quasi-sliding mode control for a permanent magnet synchronous motor with unmatched uncertainties," *Computers and Mathematics with Applications*, vol. 64, no. 5, pp. 1036–1043, 2012.

[16] D. Q. Wei, X. S. Luo, B. H. Wang, and J. Q. Fang, "Robust adaptive dynamic surface control of chaos in permanent magnet synchronous motor," *Physics Letters A*, vol. 363, no. 1-2, pp. 71–77, 2007.

[17] H. H. Choi, "Adaptive control of a chaotic permanent magnet synchronous motor," *Nonlinear Dynamics*, vol. 69, no. 3, pp. 1311–1322, 2012.

[18] D.-Q. Wei, B. Zhang, and X.-S. Luo, "Adaptive synchronization of chaos in permanent magnet synchronous motors based on passivity theory," *Chinese Physics B*, vol. 21, no. 3, Article ID 030504, 2012.

[19] F. Beaufays, Y. Abdel-Magid, and B. Widrow, "Application of neural networks to load-frequency control in power systems," *Neural Networks*, vol. 7, no. 1, pp. 183–194, 1994.

[20] X. P. Cheng and R. V. Patel, "Neural network based tracking control of a flexible macro-micro manipulator system," *Neural Networks*, vol. 16, no. 2, pp. 271–286, 2003.

[21] F.-H. Hsiao, "Optimal exponential synchronization of chaotic systems with multiple time delays via fuzzy control," *Abstract and Applied Analysis*, vol. 2013, Article ID 742821, 19 pages, 2013.

[22] S. Xing, Q. Zhang, and Y. Zhang, "Finite-time stability analysis and control for a class of stochastic singular biological economic systems based on T-S fuzzy model," *Abstract and Applied Analysis*, vol. 2013, Article ID 946491, 10 pages, 2013.

[23] J. Yu, H. Yu, B. Chen, J. Gao, and Y. Qin, "Direct adaptive neural control of chaos in the permanent magnet synchronous motor," *Nonlinear Dynamics*, vol. 70, no. 3, pp. 1879–1887, 2012.

[24] Y. Y. Hou, "Controlling chaos in permanent magnet synchronous motor control system via fuzzy guaranteed cost controller," *Abstract and Applied Analysis*, vol. 2012, Article ID 650863, 10 pages, 2012.

[25] T. B. T. Nguyen, T. L. Liao, and J. J. Yan, "Adaptive tracking control for an uncertain chaotic permanent magnet synchronous motor based on fuzzy neural networks," *Journal of Vibration and Control*, 2013.

[26] C. T. Lin and C. S. G. Lee, *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*, Prentice Hall, Upper Saddle River, NJ, USA, 1996.

[27] O. Omidvar and D. L. Elliott, *Neural Systems for Control*, Academic Press, New York, NY, USA, 1997.

[28] J. J. E. Slotine and W. Li, *Applied Nonlinear Control*, Pearson Education Taiwan, Taipei City, Taiwan, 2005.