

## Research Article

# Improved Adaptive Sliding Mode Control for a Class of Uncertain Nonlinear Systems Subjected to Input Nonlinearity via Fuzzy Neural Networks

Tat-Bao-Thien Nguyen,<sup>1</sup> Teh-Lu Liao,<sup>1</sup> and Jun-Juh Yan<sup>2</sup>

<sup>1</sup>Department of Engineering Science, National Cheng Kung University, Tainan 701, Taiwan

<sup>2</sup>Department of Computer and Communication, Shu-Te University, Kaohsiung 824, Taiwan

Correspondence should be addressed to Jun-Juh Yan; [jjyan@mail.stu.edu.tw](mailto:jjyan@mail.stu.edu.tw)

Received 8 September 2014; Revised 28 December 2014; Accepted 29 December 2014

Academic Editor: Cheng Shao

Copyright © 2015 Tat-Bao-Thien Nguyen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The paper presents an improved adaptive sliding mode control method based on fuzzy neural networks for a class of nonlinear systems subjected to input nonlinearity with unknown model dynamics. The control scheme consists of the modified adaptive and the compensation controllers. The modified adaptive controller online approximates the unknown model dynamics and input nonlinearity and then constructs the sliding mode control law, while the compensation controller takes into account the approximation errors and keeps the system robust. Based on Lyapunov stability theorem, the proposed method can guarantee the asymptotic convergence to zero of the tracking error and provide the robust stability for the closed-loop system. In addition, due to the modification in controller design, the singularity problem that usually appears in indirect adaptive control techniques based on fuzzy/neural approximations is completely eliminated. Finally, the simulation results performed on an inverted pendulum system demonstrate the advanced functions and feasibility of the proposed adaptive control approach.

## 1. Introduction

Due to the wide existence of nonlinear systems in many fields of engineering, the controller design for nonlinear systems still received much attention from many researchers. The early control techniques were developed for nonlinear systems and presented their good performances [1–6]. The fundamental ideas of these control techniques are to transform a nonlinear dynamic system into a linear one through state feedback mechanism and then apply the existing methods developed for linear systems. Although good performances can be obtained with these control techniques, the major deficiency remains. The controller design largely relies on the exact cancellation of nonlinear terms or restricts to conditions in that the unknown parameters of nonlinear systems are assumed to appear linearly. This leads to awful performances of the controllers when the existing uncertainties or nonlinear terms of the nonlinear systems are completely unknown. In addition, all control

methods above are carried out with an ideal assumption of linear input. Nevertheless, in practical conditions, there exist nonlinearities in the control input because of physical limitations. The existence of the nonlinear input may lead to degradation or even make the system unstable [7].

Nowadays, fuzzy logic and neural networks are found to be powerful tools for modeling and controlling highly uncertain, nonlinear, and complex systems due to their abilities of universal approximation [8–15]. Although fuzzy logic and neural networks have universal approximation abilities, some differences exist between them. The fuzzy logic has characteristics of linguistic information and logic control, while neural networks possess characteristics of learning, parallelism, and fault-tolerance. The combination of fuzzy logic and neural networks, known as fuzzy neural networks, which incorporate the advantages of fuzzy inference and neurolearning, was developed and has presented advanced functions in modeling and controlling nonlinear systems [15–22]. Based on universal approximation theorem, fuzzy

logic and neural networks have been developed and incorporated into adaptive control techniques. In such techniques, Lyapunov approach is used to analyze the system stability and obtain the adaptive laws as well. Conceptually, there are two distinct approaches to design the adaptive controllers: direct and indirect adaptive control methods. In the direct control method, a fuzzy logic system or neural networks are employed to simulate the action of the ideal controller and the parameters are directly adjusted to meet the control objective [15, 19, 23–31]. In contrast, the indirect control method uses a fuzzy logic system or neural networks to approximate the unknown nonlinear terms of model dynamics and then synthesizes control laws based on these approximations [15, 20, 21, 32–40]. In the indirect control method [15, 21, 32–40], the authors considered the single-input single-output (SISO) nonlinear systems in the form,  $y^{(r)} = f(\mathbf{x}) + g(\mathbf{x})u$ , where  $\mathbf{x} \in \mathfrak{R}^r$  is the overall state vector,  $u \in \mathfrak{R}$  is the control input, and  $y \in \mathfrak{R}$  is the system output.  $f(\mathbf{x}) \in \mathfrak{R}$  and  $g(\mathbf{x}) \in \mathfrak{R}$  are unknown nonlinear functions. In order to meet the control objectives, they developed the indirect adaptive controllers which are in the form  $u = (1/\hat{g}(\mathbf{x}, \theta_g))(v(t) - \hat{f}(\mathbf{x}, \theta_f))$ , where  $v(t) \in \mathfrak{R}$  is a new input transforming the nonlinear system into the linear one.  $\hat{g}(\mathbf{x}, \theta_g) \in \mathfrak{R}$  and  $\hat{f}(\mathbf{x}, \theta_f) \in \mathfrak{R}$  represent the parameterized approximations of the actual nonlinear functions,  $f(\mathbf{x}) \in \mathfrak{R}$  and  $g(\mathbf{x}) \in \mathfrak{R}$ , respectively. The weighting vectors,  $\theta_g(t)$  and  $\theta_f(t)$ , which vary according to adaptive laws, are adjusted parameters of the approximations. Since the approximations  $\hat{g}(\mathbf{x}, \theta_g)$  and  $\hat{f}(\mathbf{x}, \theta_f)$  are calculated by a fuzzy logic system or neural networks, it is well known that these approximations cannot be guaranteed to be bounded away from zero for all time  $t$ . In other words,  $\hat{g}(\mathbf{x}, \theta_g)$  may tend to zero or may be close to zero in some points in time. Such situations lead to very large control signals which may cause the controlled systems to lose their controllability or even damage the whole systems. This problem was known as a singularity problem which usually appears in indirect adaptive control method based on fuzzy/neural approximations.

Since the input nonlinearity and the singularity problem may cause negative effects on controlled systems, it is necessary to develop a new indirect adaptive control method which can bypass the singularity problem even with the existence of input nonlinearity. Therefore, by incorporating both advantages of fuzzy neural networks and sliding mode control technique, we developed a new indirect adaptive control method for a class of uncertain nonlinear systems with input nonlinearity. The proposed controller uses fuzzy neural networks to approximate the unknown nonlinear terms of model dynamics and then synthesizes the sliding mode controller. Due to the novel modifications in design of the proposed controller especially, the denominator in the adaptive control law is guaranteed to be away from zero, and, therefore, the singularity problem is completely avoided. Moreover, in order to treat the undesired effects of approximation errors, a robust compensation is added to the controller to ensure the stability of the controlled system and force the tracking errors to converge to zero as well. In contrast, many previous works dealing with indirect adaptive

fuzzy control [15, 21, 32–40] still have a weakness in that the controllers may face to the singularity problem for some cases. When the controlled systems fall into the singularity problem, these controllers may produce very large control signals. These situations may lead the systems to lose their controllability or cause the serious damage to the whole systems. Moreover, these control methods are only proper when the inputs are assumed to be linear ideally. This problem induces the controllers to the restriction of applications because the control inputs may appear nonlinearly due to the physical limitations of some components in many physical systems. For this condition, the linear input may not be suitable to use or show the awful performance. Therefore, in comparison with previous methods, the proposed control method shows the improvements in controller design in that the singularity problem is completely solved. In addition, the proposed controller can show the advanced tracking performance even the controlled system is under the influence of the input nonlinearity. With the proposed controller, the output of the system is forced to follow the desired trajectory successfully and the tracking error converges to zero asymptotically. Finally, the simulations are carried out to illustrate the effectiveness and robustness of the proposed controller.

The rest of this paper is organized as follows. The conventional sliding mode control and problem statement are presented in Section 2. The design of fuzzy neural networks is addressed in Section 3, and the design of the adaptive controller is described in Section 4. In Section 5, simulation results are given to confirm the validity of the proposed method. Finally, the conclusion is given in Section 6.

## 2. Problem Statement and Sliding Mode Control Design

Considering the  $n$ th order SISO nonlinear system and assuming that the control input is nonlinearly perturbed due to physical limitations, the dynamic equations can be expressed in normal form as follows:

$$\begin{aligned} \dot{x}_1(t) &= x_2(t), \\ \dot{x}_2(t) &= x_3(t), \\ &\vdots \\ \dot{x}_n(t) &= f(\mathbf{x}) + g(\mathbf{x})\phi(u(t)), \\ y(t) &= x_1(t), \end{aligned} \tag{1}$$

where  $\mathbf{x} = [x_1(t) \ x_2(t) \ \cdots \ x_n(t)]^T \in \mathfrak{R}^n$  is the overall state vector of the nonlinear system which is assumed to be available for measurement and  $f(\mathbf{x}) \in \mathfrak{R}$ ,  $g(\mathbf{x}) \in \mathfrak{R}$  are completely unknown smooth functions.  $u(t) \in \mathfrak{R}$  is the scalar control input, while  $y(t) \in \mathfrak{R}$  is the scalar system output.  $\phi(u(t))$  is a continuous nonlinear function and inside the sector  $[\beta_1 \ \beta_2]$ ; that is,

$$\beta_1 u^2 \leq u\phi(u) \leq \beta_2 u^2, \tag{2}$$

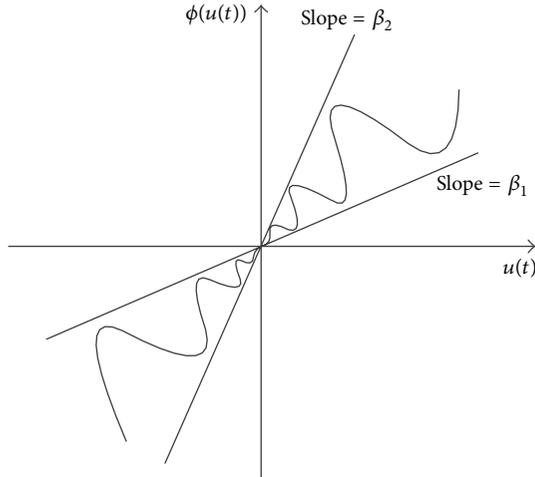


FIGURE 1: The scalar nonlinear function  $\phi(u(t))$  inside the sector  $[\beta_1 \ \beta_2]$ .

where  $\beta_1$  and  $\beta_2$  are positive constants and  $\phi(0) = 0$ . The scalar nonlinear function  $\phi(u(t))$  is illustrated in Figure 1.

Obviously according to (2), there always exists a function  $g_u(u(t)) \in \mathfrak{R}$  inside the sector  $[\beta_1 \ \beta_2]$  satisfying  $\phi(u(t)) = g_u(u(t))u(t)$  and  $\beta_1 u^2 \leq g_u(u(t))u^2 \leq \beta_2 u^2$ . The objective of this paper is to design the control law  $u(t)$  such that the output  $y(t)$  can successfully track a given desired trajectory  $y_d(t) \in \mathfrak{R}$ , which is a known smooth function. Before designing the controller to meet the control objective, we first rewrite the dynamic equations in (1) as follows:

$$\begin{aligned} \dot{x}_1(t) &= x_2(t), \\ \dot{x}_2(t) &= x_3(t), \\ &\vdots \\ \dot{x}_n(t) &= f(\mathbf{x}) + G(\mathbf{x}, u(t))u(t), \\ y(t) &= x_1(t), \end{aligned} \quad (3)$$

where  $G(\mathbf{x}, u(t)) = g(\mathbf{x})g_u(u(t))$ .

Now we define the tracking error  $e(t) \in \mathfrak{R}$  and the sliding surface  $S(t) \in \mathfrak{R}$ , which describes the tracking error dynamics as follows:

$$e(t) = y(t) - y_d(t), \quad (4)$$

$$S(t) = \left( \frac{d}{dt} + \lambda \right)^{n-1} e(t), \quad (5)$$

where  $\lambda$  is a positive designed constant. The surface  $S(t) = 0$  corresponds to a linear differential equation of which the solution denotes that the tracking error  $e(t)$  converges to zero with time  $(n-1)/\lambda$  [6]. Taking the time derivative of  $S(t)$ , we can get

$$\dot{S}(t) = e^{(n)}(t) + k_{n-1}e^{(n-1)}(t) + \dots + k_1\dot{e}(t), \quad (6)$$

where  $k_{n-1}, \dots, k_1$  represent coefficients in the Hurwitz polynomial which expanded from (5). Here the notation  $(\cdot)^{(n)}$  denotes the derivative  $d^n(\cdot)/dt^n$ .

From (3) and (4), and noticing that  $y^{(n)}(t) = x_1^{(n)}(t) = \dot{x}_n$ , we can rewrite (6) as

$$\begin{aligned} \dot{S}(t) &= y^{(n)}(t) - y_d^{(n)}(t) \\ &\quad + (k_{n-1}e^{(n-1)}(t) + \dots + k_1\dot{e}(t)) \\ &= x_1^{(n)}(t) - y_d^{(n)}(t) \\ &\quad + (k_{n-1}e^{(n-1)}(t) + \dots + k_1\dot{e}(t)) \\ &= f(\mathbf{x}) + G(\mathbf{x}, u(t))u(t) - y_d^{(n)}(t) \\ &\quad + (k_{n-1}e^{(n-1)}(t) + \dots + k_1\dot{e}(t)). \end{aligned} \quad (7)$$

We define a new input variable as

$$v(t) = -y_d^{(n)}(t) + k_{n-1}e^{(n-1)}(t) + \dots + k_1\dot{e}(t). \quad (8)$$

Then  $\dot{S}(t)$  in (7) can be expressed in the compact form as

$$\dot{S}(t) = f(\mathbf{x}) + G(\mathbf{x}, u(t))u(t) + v(t). \quad (9)$$

If  $f(\mathbf{x})$  and  $G(\mathbf{x}, u(t))$  in (9) are known, in order to meet the control objective, the control law based on ideal sliding mode control can be used as

$$u(t) = \frac{1}{G(\mathbf{x}, u(t))} (-f(\mathbf{x}) - v(t) - \eta S(t)), \quad (10)$$

where  $\eta$  is a positive designed constant,  $G(\mathbf{x}, u(t)) \neq 0$  must be satisfied to make the control law in (10) proper and ensure the controllability of the system. However, in this paper, we consider that  $f(\mathbf{x})$  and  $G(\mathbf{x}, u(t))$  are completely unknown, so this controllability condition is modified for stability analysis in the next sections as the following assumption.

*Assumption 1.*  $G(\mathbf{x}, u(t))$  is bounded from below and above by some known positive constants  $\underline{g}$  and  $\bar{g}$ ; that is,  $0 < \underline{g} \leq G(\mathbf{x}, u(t)) \leq \bar{g}, \forall \mathbf{x} \in \mathfrak{R}^n$ .

Substituting (10) into (9) yields

$$\dot{S}(t) = -\eta S(t). \quad (11)$$

The equation in (11) implies that  $S(t)$  converges to zero exponentially fast; therefore, the tracking error  $e(t)$  converges to zero exponentially fast.

However, in fact,  $f(\mathbf{x})$  and  $G(\mathbf{x}, u(t))$  are unknown, the ideal control law in (10) can no longer be used. In order to overcome this problem, we use a fuzzy neural network to approximate both  $f(\mathbf{x})$  and  $G(\mathbf{x}, u(t))$ .

### 3. Design of Fuzzy Neural Networks

The basic configuration of a fuzzy logic system comprises four principal components: fuzzification, rule base, fuzzy

inference, and defuzzification. In the fuzzification process, the inputs,  $n$  state variables  $x_1, x_2, \dots, x_n \in \mathfrak{R}$ , are mapped to membership values in the input universes of discourse. The rule base holds a set of antecedent-consequent linguistic rules (IF-THEN rules) that quantify the knowledge that human experts have amassed about solving particular problems. Let  $m$  be the number of IF-THEN rules. Then the  $i$ th rule is described in the form of

$$\begin{aligned} &\text{IF } (x_1 \text{ is } A_1^i \text{ AND } x_2 \text{ is } A_2^i \text{ AND } \dots \text{ AND } x_n \text{ is } A_n^i) \\ &\text{THEN } (\hat{f} \text{ is } B_f^i \text{ AND } \hat{G} \text{ is } B_g^i), \end{aligned} \quad (12)$$

where  $A_1^i, A_2^i, \dots, A_n^i$ ,  $B_f^i$  and  $B_g^i$  are fuzzy sets that correspond to the membership functions  $\mu_{A_1^i}, \mu_{A_2^i}, \dots, \mu_{A_n^i}, \mu_{B_f^i}$  and  $\mu_{B_g^i}$ , respectively.  $\hat{f}(\mathbf{x}, t) \in \mathfrak{R}$  and  $\hat{G}(\mathbf{x}, t) \in \mathfrak{R}$ , which stand for the approximations of  $f(\mathbf{x})$  and  $G(\mathbf{x}, u(t))$ , respectively, are the outputs of the fuzzy logic system.  $\mu_{B_f^i}$  and  $\mu_{B_g^i}$  are fuzzy singletons, while  $\mu_{A_1^i}, \mu_{A_2^i}, \dots, \mu_{A_n^i}$  use Gaussian functions to calculate the membership values according to the following equation:

$$\mu_{A_j^i}(x_j) = \exp \left[ -\frac{(x_j - p_j^i)^2}{2q^2} \right], \quad (13)$$

where  $i = 1, 2, \dots, m$  correspond with  $m$  rules and  $j = 1, 2, \dots, n$  correspond with  $n$  state variables.  $p_j^i$  and  $q$  are the parameters of Gaussian functions. The fuzzy inference engine, which uses the product inference for mapping, performs as a process of mapping membership values from the input windows through the fuzzy rule base to the output window. The engine makes successive decisions about which rules are most relevant to the current situation and applies the actions indicated by these rules. The output defuzzification is the procedure of mapping from a set of inferred fuzzy signals contained within a fuzzy output window to a crisp numeric values as control actions. Using the center-average defuzzification techniques, the outputs of the fuzzy logic system can be calculated as

$$\begin{aligned} \hat{f}(\mathbf{x}, t) &= \frac{\sum_{i=1}^m \theta_{fi}(t) \left( \prod_{j=1}^n \mu_{A_j^i}(x_j) \right)}{\sum_{i=1}^m \left( \prod_{j=1}^n \mu_{A_j^i}(x_j) \right)} = \theta_f^T(t) \varphi(\mathbf{x}), \\ \hat{G}(\mathbf{x}, t) &= \frac{\sum_{i=1}^m \theta_{gi}(t) \left( \prod_{j=1}^n \mu_{A_j^i}(x_j) \right)}{\sum_{i=1}^m \left( \prod_{j=1}^n \mu_{A_j^i}(x_j) \right)} = \theta_g^T(t) \varphi(\mathbf{x}), \end{aligned} \quad (14)$$

where  $\theta_f^T(t) = [\theta_{f1}(t) \theta_{f2}(t) \dots \theta_{fm}(t)]$  and  $\theta_g^T(t) = [\theta_{g1}(t) \theta_{g2}(t) \dots \theta_{gm}(t)]$  are weighting vectors that are online tuned according to the adaptive laws described in the next section to meet the control objective. The fuzzy singletons  $\mu_{B_f^i}$  and  $\mu_{B_g^i}$  reach their maximum values at points  $\theta_{fi}$  and  $\theta_{gi}$  with  $i = 1, 2, \dots, m$ , respectively; that is,  $\mu_{B_f^i}(\theta_{fi}) = \mu_{B_g^i}(\theta_{gi}) = 1$ . The fuzzy basic vector

$\varphi^T(\mathbf{x}) = [\varphi_1(\mathbf{x}) \varphi_2(\mathbf{x}) \dots \varphi_m(\mathbf{x})]$  has the element  $\varphi_i(\mathbf{x})$ ,  $i = 1, 2, \dots, m$ , defined by

$$\varphi_i(\mathbf{x}) = \frac{\prod_{j=1}^n \mu_{A_j^i}(x_j)}{\sum_{i=1}^m \left( \prod_{j=1}^n \mu_{A_j^i}(x_j) \right)}. \quad (15)$$

A fuzzy logic system, which can reason with imprecise information through the fuzzy inference, is good at explaining its actions but it cannot automatically acquire the rules it uses to make those actions. On the other hand, a neural network is good at recognizing patterns but it is not good at explaining how it reaches its decisions. These limitations have been a central driving force behind the creation of hybrid systems called fuzzy neural networks [9, 12, 13]. A fuzzy neural network can combine the human-like reasoning style of a fuzzy system with the learning and connectionist structure of a neural network. In this manner, the parameters in a fuzzy logic system can be found by a neural network through learning processes. Figure 2 shows the structure of the fuzzy neural network with four layers: input layer, membership layer, rule layer, and output layer. Nodes in the input layer are input nodes that represent input linguistic variables. In this context, the inputs are  $n$  state variables and their values are directly transmitted to the membership layer. The membership layer has  $n \times m$  nodes of which each unit performs a membership function to an input and uses a Gaussian function to calculate the membership value. The rule layer has  $m$  nodes of which each node stands for an element  $\varphi_i(\mathbf{x})$  of the fuzzy basis vector  $\varphi(\mathbf{x})$  and performs a fuzzy rule. Thus, all nodes of rule layer form the fuzzy rule set. The links between the rule layer and the output layer express the weighting factors,  $\theta_{f1}, \theta_{f2}, \dots, \theta_{fm}$  and  $\theta_{g1}, \theta_{g2}, \dots, \theta_{gm}$ , which are the elements of the weighting vectors,  $\theta_f$  and  $\theta_g$ , respectively. These factors are the parameters and are adjusted by designed adaptive laws explained in the next section. In the output layer, nodes represent the output linguistic variables. Two nodes in the output layer, as depicted in Figure 2, act for the values of  $\hat{f}(\mathbf{x}, t)$  and  $\hat{G}(\mathbf{x}, t)$ .

Therefore, the fuzzy neural network has four layers with  $n$  inputs and  $m$  fuzzy rules. The inputs correspond to the state variables of the system, so the number of inputs is chosen so that it equals the order (number of state variables) of the system. The parameters of the membership functions and the number of fuzzy rules are significantly relevant to the approximation accuracy of the network. These parameters must be defined so that they can appropriately cover all possible working area of state variables. For the number of fuzzy rules, in general, the more rules the network has, the more accuracy the approximation can get. However, a great number of rules lead to the complication of the designed system and the increase of the system cost. Thus, the arrangements of parameters of membership functions and the number of fuzzy rules depend on the technical experts who have much knowledge about specific systems.

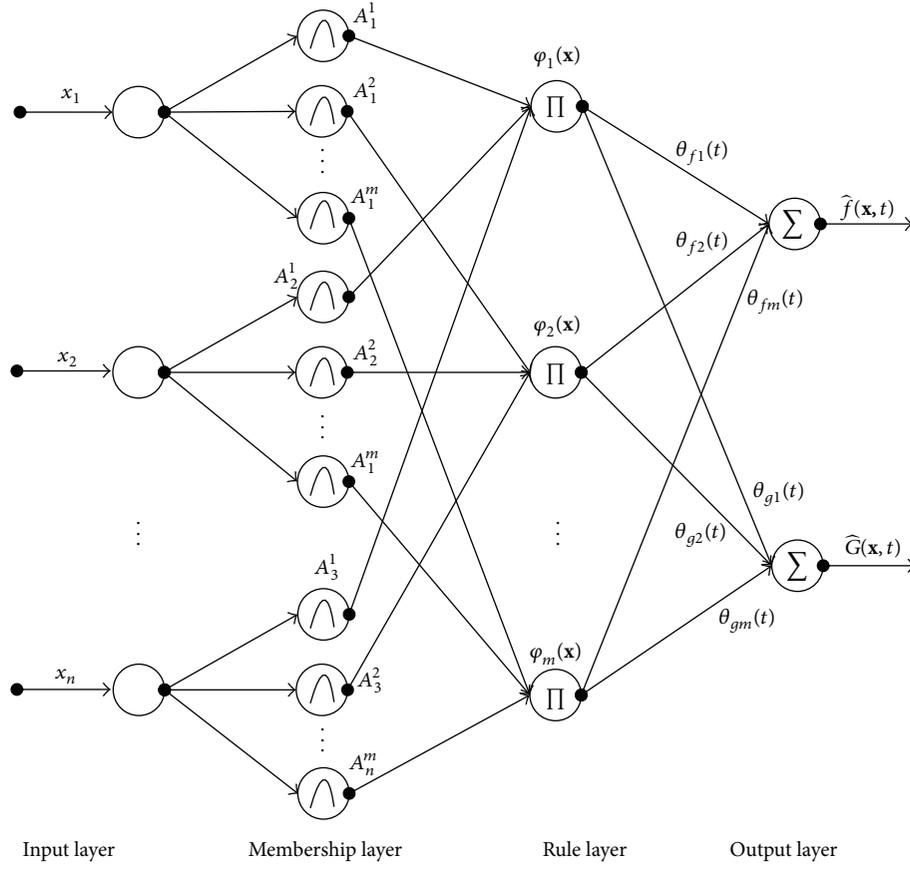


FIGURE 2: The structure of a fuzzy neural network.

#### 4. Design of Adaptive Controller

Since  $f(\mathbf{x})$  and  $G(\mathbf{x}, u(t))$  are completely unknown, the ideal sliding mode control law in (10) cannot be determined. This problem leads to the uselessness of the ideal control law in (10). In order to take care of this problem, a fuzzy neural network, as shown in Figure 2, is used to approximate  $f(\mathbf{x})$  and  $G(\mathbf{x}, u(t))$  online. Then incorporating the certainty equivalent approach, the adaptive controller  $u_{ac}(t)$  inspired from the ideal sliding mode control law, can be obtained as

$$u_{ac}(t) = \frac{1}{\hat{G}(\mathbf{x}, t)} \left( -\hat{f}(\mathbf{x}, t) - v(t) - \eta S(t) \right), \quad (16)$$

where  $\hat{f}(\mathbf{x}, t)$  and  $\hat{G}(\mathbf{x}, t)$  are the online approximations of  $f(\mathbf{x})$  and  $G(\mathbf{x}, u(t))$ , respectively. The values of  $\hat{f}(\mathbf{x}, t)$  and  $\hat{G}(\mathbf{x}, t)$  are calculated by a fuzzy neural network as described in (14).

However,  $\hat{G}(\mathbf{x}, t)$  may tend to be zero or be close to zero in some point in time during the operation time, especially in initial phase. This leads to very large control signal values, which may damage the whole system. This situation is known as a singularity problem. In order to avoid the singularity problem, we modify the control law in that  $\hat{G}(\mathbf{x}, t)$  is replaced with  $\hat{G}'(\mathbf{x}, t)$ , and the fundamental idea for operation of  $\hat{G}'(\mathbf{x}, t)$  is described as follows. First we consider the value

of  $\hat{G}(\mathbf{x}, t)$  during the operation time. With any initial value of  $\hat{G}(\mathbf{x}, t)$  satisfying  $\hat{G}(\mathbf{x}, 0) \geq \underline{g}$ , we consider whether  $\hat{G}(\mathbf{x}, t)$  reaches the lower bound  $\underline{g}$ . If so,  $\hat{G}(\mathbf{x}, t)$  may have a tendency in that its value is close to zero, and the control signal will be very large, leading to lose the controllability. Thus, when the  $\hat{G}(\mathbf{x}, t)$  reaches the lower bound  $\underline{g}$  we stop the update law for  $\hat{G}(\mathbf{x}, t)$  and forcedly assign  $\underline{g}$  to  $\hat{G}'(\mathbf{x}, t)$ . In contrast, we assign  $\hat{G}(\mathbf{x}, t)$  to  $\hat{G}'(\mathbf{x}, t)$  elsewhere. By this way, the singularity problem can be surely avoided. Therefore, the adaptive controller  $u_{ac}(t)$  in (16) is replaced with the modified adaptive controller  $u'_{ac}(t)$  as

$$u'_{ac}(t) = \frac{1}{\hat{G}'(\mathbf{x}, t)} \left( -\hat{f}(\mathbf{x}, t) - v(t) - \eta S(t) \right), \quad (17)$$

where  $\hat{f}(\mathbf{x}, t)$  and  $\hat{G}'(\mathbf{x}, t)$  are calculated via the fuzzy neural network as follows:

$$\hat{f}(\mathbf{x}, t) = \theta_f^T(t) \varphi(\mathbf{x}),$$

$$\hat{G}'(\mathbf{x}, t) = \begin{cases} \underline{g} & \text{if } \hat{G}(\mathbf{x}, t) \leq \underline{g}, \\ \hat{G}(\mathbf{x}, t) = \theta_g^T(t) \varphi(\mathbf{x}) & \text{elsewhere.} \end{cases} \quad (18)$$

When the controller works, the values of the weighting vectors,  $\theta_f(t)$  and  $\theta_g(t)$ , are adjusted so that  $\hat{f}(\mathbf{x}, t)$  and  $\hat{G}(\mathbf{x}, t)$

reach  $f(\mathbf{x}, t)$  and  $G(\mathbf{x}, u(t))$ , respectively. The adaptive laws for  $\theta_f(t)$  and  $\theta_g(t)$  are chosen as

$$\dot{\theta}_f(t) = \mathbf{W}_f^{-1} \varphi(\mathbf{x}) S(t), \quad (19)$$

$$\dot{\theta}_g(t) = \begin{cases} 0 & \text{if } \widehat{G}(\mathbf{x}, t) \leq \underline{g}, \\ \mathbf{W}_g^{-1} \varphi(\mathbf{x}) S(t) u'_{ac}(t) & \text{elsewhere,} \end{cases} \quad (20)$$

where  $\mathbf{W}_f \in \mathfrak{R}^{m \times m}$  and  $\mathbf{W}_g \in \mathfrak{R}^{m \times m}$  are positive-defined weighting matrices. These matrices govern the speed of adaptation. Now we use the notation  $\|\cdot\|$  to express the  $L^2$ -norm of a vector. If  $\|\mathbf{W}_f\|$  and  $\|\mathbf{W}_g\|$  are large,  $\|\mathbf{W}_f^{-1}\|$  and  $\|\mathbf{W}_g^{-1}\|$  are small, leading to low speed of adaptation. On the contrary, the small values of  $\|\mathbf{W}_f\|$  and  $\|\mathbf{W}_g\|$  imply that  $\|\mathbf{W}_f^{-1}\|$  and  $\|\mathbf{W}_g^{-1}\|$  are large, leading to high speed of adaptation. Nevertheless, the high adaptive speed has a drawback in that the controlled system is very sensitive to external uncertainties. This may cause the system to lose its controllability.

In the adaptive mechanism,  $\widehat{f}(\mathbf{x}, t)$  and  $\widehat{G}(\mathbf{x}, t)$  are adjusted so that they converge to  $f(\mathbf{x})$  and  $G(\mathbf{x}, u(t))$ , respectively. In this situation,  $\theta_f(t)$  and  $\theta_g(t)$  achieve their optimal values,  $\theta_f^*$  and  $\theta_g^*$ , respectively. Notice that these optimal values,  $\theta_f^*$  and  $\theta_g^*$ , are artificial constant quantities which are introduced only for analytical purpose, and they are not used in implementation. The optimal values,  $\theta_f^* \in \mathfrak{R}^m$  and  $\theta_g^* \in \mathfrak{R}^m$ , are then defined by

$$\theta_f^* = \arg \min_{\theta_f \in \Theta_f} \left\{ \sup_{\mathbf{x} \in \Omega} \left| \theta_f^T(t) \varphi(\mathbf{x}) - f(\mathbf{x}) \right| \right\}, \quad (21)$$

$$\theta_g^* = \arg \min_{\theta_g \in \Theta_g} \left\{ \sup_{\mathbf{x} \in \Omega} \left| \theta_g^T(t) \varphi(\mathbf{x}) - g(\mathbf{x}) \right| \right\},$$

where  $\Theta_f \subset \mathfrak{R}^m$  and  $\Theta_g \subset \mathfrak{R}^m$  are sets of the acceptable values of weighting vector,  $\theta_f(t)$  and  $\theta_g(t)$ , respectively.  $\Omega \subset \mathfrak{R}^n$  is a compact set of the state variables  $x_1, x_2, \dots, x_n$ . In this paper, we assume that the compact set  $\Omega$  is large enough so that the state variables  $x_1, x_2, \dots, x_n$  remain within  $\Omega$  under closed-loop control and the designed fuzzy neural network does not violate the universal approximation property on  $\Omega$ .

In the ideal case of approximation, when  $\widehat{f}(\mathbf{x}, t)$  and  $\widehat{G}(\mathbf{x}, t)$  reach  $f(\mathbf{x})$  and  $G(\mathbf{x}, u(t))$ , respectively,  $\theta_f(t)$  and  $\theta_g(t)$  reach  $\theta_f^*$  and  $\theta_g^*$ , respectively. However, the designed fuzzy neural network which has a finite number of units in the hidden layer is utilized to approximate  $f(\mathbf{x})$  and  $G(\mathbf{x}, u(t))$ , so the approximation errors appear and affect the controlled system. Due to these errors,  $\widehat{f}(\mathbf{x}, t)$  and  $\widehat{G}(\mathbf{x}, t)$  cannot converge to  $f(\mathbf{x})$  and  $G(\mathbf{x}, u(t))$  exactly even though  $\theta_f(t)$  and  $\theta_g(t)$  completely converge to  $\theta_f^*$  and  $\theta_g^*$ , respectively. Let  $\delta_f(\mathbf{x})$  and  $\delta_g(\mathbf{x})$  be the approximation errors, then the exact models of  $f(\mathbf{x})$  and  $G(\mathbf{x}, u(t))$  can be expressed as follows:

$$\begin{aligned} f(\mathbf{x}) &= \theta_f^* \varphi(\mathbf{x}) + \delta_f(\mathbf{x}), \\ G(\mathbf{x}, u(t)) &= \theta_g^* \varphi(\mathbf{x}) + \delta_g(\mathbf{x}). \end{aligned} \quad (22)$$

*Assumption 2.* The approximation errors are bounded by some known constants  $\bar{\delta}_f > 0$  and  $\bar{\delta}_g > 0$  over the compact set  $\Omega \subset \mathfrak{R}^n$  as follows:

$$\begin{aligned} \sup_{\mathbf{x} \in \Omega} |\delta_f(\mathbf{x})| &\leq \bar{\delta}_f, \\ \sup_{\mathbf{x} \in \Omega} |\delta_g(\mathbf{x})| &\leq \bar{\delta}_g. \end{aligned} \quad (23)$$

Now the different quantities between the approximation models and exact models can be calculated as

$$\begin{aligned} \widehat{f}(\mathbf{x}, t) - f(\mathbf{x}) &= (\theta_f(t) - \theta_f^*)^T \varphi(\mathbf{x}) - \delta_f(\mathbf{x}) \\ &= \bar{\theta}_f^T(t) \varphi(\mathbf{x}) - \delta_f(\mathbf{x}), \end{aligned} \quad (24)$$

$$\begin{aligned} \widehat{G}(\mathbf{x}, t) - G(\mathbf{x}, u(t)) &= (\theta_g(t) - \theta_g^*)^T \varphi(\mathbf{x}) - \delta_g(\mathbf{x}) \\ &= \bar{\theta}_g^T(t) \varphi(\mathbf{x}) - \delta_g(\mathbf{x}), \end{aligned}$$

where  $\bar{\theta}_f(t) = \theta_f(t) - \theta_f^*$  and  $\bar{\theta}_g(t) = \theta_g(t) - \theta_g^*$  are parameter errors.

Because the approximation errors exist and affect the controlled system, the modified adaptive controller  $u'_{ac}(t)$  may be difficult to ensure the stability of the closed-loop controlled system alone. In order to suppress the undesirable effects of the approximation errors, a compensation controller  $u_{cc}(t)$  is developed and added to the controlled system. This controller is able to compensate the approximation error effects and keep the close-loop system robust. The compensation controller  $u_{cc}(t)$  is designed according to the following equation:

$$u_{cc}(t) = -\frac{1}{\underline{g}} \left( \bar{\delta}_f + D(\widehat{G}(\mathbf{x}, t)) \left| u'_{ac}(t) \right| \right) \text{sgn}(S(t)), \quad (25)$$

where the switching function  $D(\widehat{G}(\mathbf{x}, t))$  is defined as

$$D(\widehat{G}(\mathbf{x}, t)) = \begin{cases} \bar{g} - \underline{g} & \text{if } \widehat{G}(\mathbf{x}, t) \leq \underline{g}, \\ \bar{\delta}_g & \text{elsewhere.} \end{cases} \quad (26)$$

Therefore, there are two controllers working together to force the controlled system to match the control objective: the modified adaptive controller  $u'_{ac}(t)$  and the compensation controller  $u_{cc}(t)$ . The overall scheme of the controlled system

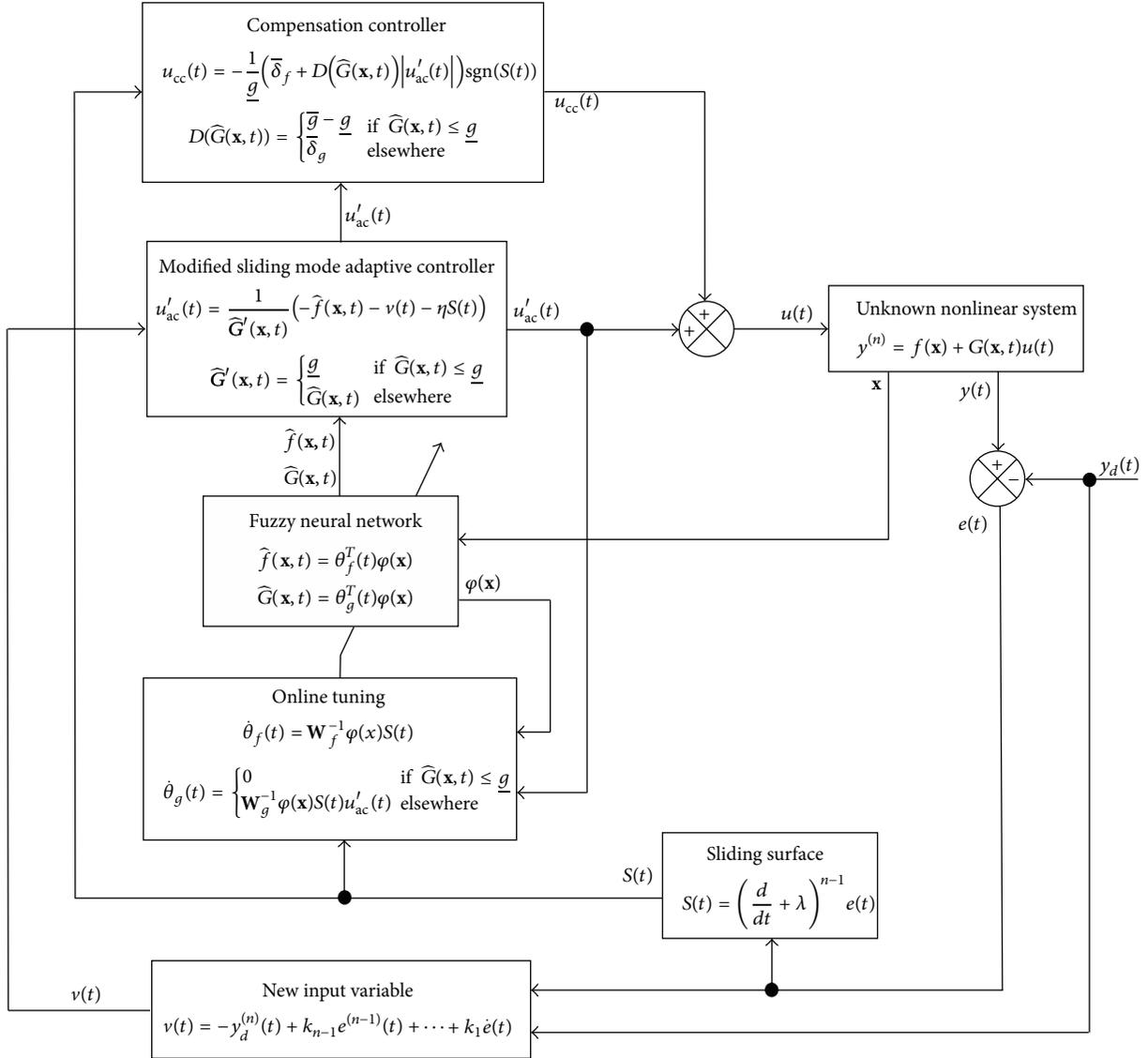


FIGURE 3: Overall control scheme for an unknown nonlinear system.

is illustrated in Figure 3. The total controller  $u(t)$  is the sum of these two controllers and its formula is given as

$$\begin{aligned}
 u(t) &= u'_{ac}(t) + u_{cc}(t) \\
 &= u'_{ac}(t) - \frac{1}{\underline{g}} \left( \bar{\delta}_f + D(\widehat{G}(\mathbf{x}, t)) |u'_{ac}(t)| \right) \text{sgn}(S(t)) \\
 &= \frac{1}{\widehat{G}'(\mathbf{x}, t)} \left( -\widehat{f}(\mathbf{x}, t) - v(t) - \eta S(t) \right) - \frac{\bar{\delta}_f}{\underline{g}} \text{sgn}(S(t)) \\
 &\quad - \frac{D(\widehat{G}(\mathbf{x}, t))}{\underline{g}} \\
 &\quad \cdot \left| \frac{1}{\widehat{G}'(\mathbf{x}, t)} \left( -\widehat{f}(\mathbf{x}, t) - v(t) - \eta S(t) \right) \right| \text{sgn}(S(t)). \tag{27}
 \end{aligned}$$

**Theorem 3.** Consider the unknown nonlinear system in (1) and suppose that Assumptions 1 and 2 are satisfied. Then the controller (27) with the designed adaptive laws (19) and (20) can guarantee that the system output tracks the desired trajectory successfully and the tracking error converges to zero asymptotically fast.

*Proof.* From (9) and (27), we take some basic algebraic manipulations and obtain

$$\begin{aligned}
 \dot{S}(t) &= f(\mathbf{x}) + G(\mathbf{x}, u(t))u(t) + v(t) \\
 &= f(\mathbf{x}) + v(t) + G(\mathbf{x}, u(t)) \left( u'_{ac}(t) + u_{cc}(t) \right) \\
 &= f(\mathbf{x}) + v(t) + \widehat{G}'(\mathbf{x}, t) u'_{ac}(t) \\
 &\quad + \left( G(\mathbf{x}, u(t)) - \widehat{G}'(\mathbf{x}, t) \right) u'_{ac}(t) + G(\mathbf{x}, u(t)) u_{cc}(t). \tag{28}
 \end{aligned}$$

Replacing  $u'_{ac}(t)$  in (28) with its expression in (17), we can rewrite (28) as

$$\begin{aligned} \dot{S}(t) &= f(\mathbf{x}) + v(t) + (-\hat{f}(\mathbf{x}, t) - v(t) - \eta S(t)) \\ &\quad + (G(\mathbf{x}, u(t)) - \widehat{G}'(\mathbf{x}, t)) u'_{ac}(t) + G(\mathbf{x}, u(t)) u_{cc}(t) \\ &= -\eta S(t) + (f(\mathbf{x}) - \hat{f}(\mathbf{x}, t)) \\ &\quad + (G(\mathbf{x}, u(t)) - \widehat{G}'(\mathbf{x}, t)) u'_{ac}(t) + G(\mathbf{x}, u(t)) u_{cc}(t). \end{aligned} \quad (29)$$

Then, using (18) and (24), (29) can be rewritten as

$$\begin{aligned} \dot{S}(t) &= \begin{cases} -\eta S(t) + (f(\mathbf{x}) - \hat{f}(\mathbf{x}, t)) \\ \quad + (G(\mathbf{x}, u(t)) - \underline{g}) u'_{ac}(t) \\ \quad + G(\mathbf{x}, u(t)) u_{cc}(t) & \text{if } \widehat{G}(\mathbf{x}, t) \leq \underline{g}, \\ -\eta S(t) + (f(\mathbf{x}) - \hat{f}(\mathbf{x}, t)) \\ \quad + (G(\mathbf{x}, u(t)) - \widehat{G}'(\mathbf{x}, t)) u'_{ac}(t) \\ \quad + G(\mathbf{x}, u(t)) u_{cc}(t) & \text{elsewhere,} \end{cases} \\ &= \begin{cases} -\eta S(t) - (\bar{\theta}_f^T(t) \varphi(\mathbf{x}) - \delta_f(\mathbf{x})) \\ \quad + (G(\mathbf{x}, u(t)) - \underline{g}) u'_{ac}(t) \\ \quad + G(\mathbf{x}, u(t)) u_{cc}(t) & \text{if } \widehat{G}(\mathbf{x}, t) \leq \underline{g}, \\ -\eta S(t) - (\bar{\theta}_f^T(t) \varphi(\mathbf{x}) - \delta_f(\mathbf{x})) \\ \quad - (\bar{\theta}_g^T(t) \varphi(\mathbf{x}) - \delta_g(\mathbf{x})) u'_{ac}(t) \\ \quad + G(\mathbf{x}, u(t)) u_{cc}(t) & \text{elsewhere.} \end{cases} \end{aligned} \quad (30)$$

We define a Lyapunov-like function  $V(\mathbf{x}, t)$  for stability analysis as follows:

$$V(\mathbf{x}, t) = \frac{1}{2} S^2(t) + \frac{1}{2} \bar{\theta}_f^T(t) \mathbf{W}_f \bar{\theta}_f(t) + \frac{1}{2} \bar{\theta}_g^T(t) \mathbf{W}_g \bar{\theta}_g(t). \quad (31)$$

Taking the time derivative of  $V(\mathbf{x}, t)$  with the fact,  $\dot{\bar{\theta}}_f = \dot{\theta}_f$  and  $\dot{\bar{\theta}}_g = \dot{\theta}_g$ , and using (20), we can obtain

$$\begin{aligned} \dot{V}(\mathbf{x}, t) &= S(t) \dot{S}(t) + \frac{1}{2} \dot{\bar{\theta}}_f^T(t) \mathbf{W}_f \bar{\theta}_f(t) + \frac{1}{2} \bar{\theta}_f^T(t) \mathbf{W}_f \dot{\bar{\theta}}_f(t) \\ &\quad + \frac{1}{2} \dot{\bar{\theta}}_g^T(t) \mathbf{W}_g \bar{\theta}_g(t) + \frac{1}{2} \bar{\theta}_g^T(t) \mathbf{W}_g \dot{\bar{\theta}}_g(t) \\ &= \begin{cases} S(t) \dot{S}(t) + \bar{\theta}_f^T(t) \mathbf{W}_f \dot{\bar{\theta}}_f(t) & \text{if } \widehat{G}(\mathbf{x}, t) \leq \underline{g}, \\ S(t) \dot{S}(t) + \bar{\theta}_f^T(t) \mathbf{W}_f \dot{\bar{\theta}}_f(t) \\ \quad + \bar{\theta}_g^T(t) \mathbf{W}_g \dot{\bar{\theta}}_g(t) & \text{elsewhere.} \end{cases} \end{aligned} \quad (32)$$

Substituting (30) into (32), then (32) can be expressed as

$$\begin{aligned} \dot{V}(\mathbf{x}, t) &= \begin{cases} -\eta S^2(t) - (\bar{\theta}_f^T(t) \varphi(\mathbf{x}) - \delta_f(\mathbf{x})) S(t) \\ \quad + (G(\mathbf{x}, u(t)) - \underline{g}) S(t) u'_{ac}(t) \\ \quad + G(\mathbf{x}, u(t)) S(t) u_{cc}(t) \\ \quad + \bar{\theta}_f^T(t) \mathbf{W}_f \dot{\bar{\theta}}_f(t) & \text{if } \widehat{G}(\mathbf{x}, t) \leq \underline{g}, \\ -\eta S^2(t) - (\bar{\theta}_f^T(t) \varphi(\mathbf{x}) - \delta_f(\mathbf{x})) S(t) \\ \quad - (\bar{\theta}_g^T(t) \varphi(\mathbf{x}) - \delta_g(\mathbf{x})) \\ \quad \cdot S(t) u'_{ac}(t) \\ \quad + G(\mathbf{x}, u(t)) S(t) u_{cc}(t) \\ \quad + \bar{\theta}_f^T(t) \mathbf{W}_f \dot{\bar{\theta}}_f(t) + \bar{\theta}_g^T(t) \mathbf{W}_g \dot{\bar{\theta}}_g(t) & \text{elsewhere,} \end{cases} \\ &= \begin{cases} -\eta S^2(t) \\ \quad + \bar{\theta}_f^T(t) (\mathbf{W}_f \dot{\bar{\theta}}_f(t) - \varphi(\mathbf{x}) S(t)) \\ \quad + (G(\mathbf{x}, u(t)) - \underline{g}) S(t) u'_{ac}(t) \\ \quad + G(\mathbf{x}, u(t)) S(t) u_{cc}(t) \\ \quad + S(t) \delta_f(\mathbf{x}) & \text{if } \widehat{G}(\mathbf{x}, t) \leq \underline{g}, \\ -\eta S^2(t) + \bar{\theta}_f^T(t) \\ \quad \cdot (\mathbf{W}_f \dot{\bar{\theta}}_f(t) - \varphi(\mathbf{x}) S(t)) + \bar{\theta}_g^T(t) \\ \quad \cdot (\mathbf{W}_g \dot{\bar{\theta}}_g(t) - \varphi(\mathbf{x}) S(t) u'_{ac}(t)) \\ \quad + G(\mathbf{x}, u(t)) S(t) u_{cc}(t) \\ \quad + S(t) \delta_f(\mathbf{x}) + S(t) \delta_g(\mathbf{x}) u'_{ac}(t) & \text{elsewhere.} \end{cases} \end{aligned} \quad (33)$$

Applying the adaptive laws in (19) and (20) to (33), (33) can be rewritten as

$$\begin{aligned} \dot{V}(\mathbf{x}, t) &= \begin{cases} -\eta S^2(t) + (G(\mathbf{x}, u(t)) - \underline{g}) S(t) u'_{ac}(t) \\ \quad + G(\mathbf{x}, u(t)) S(t) u_{cc}(t) \\ \quad + S(t) \delta_f(\mathbf{x}) & \text{if } \widehat{G}(\mathbf{x}, t) \leq \underline{g}, \\ -\eta S^2(t) + G(\mathbf{x}, u(t)) S(t) u_{cc}(t) \\ \quad + S(t) \delta_f(\mathbf{x}) + S(t) \delta_g(\mathbf{x}) u'_{ac}(t) & \text{elsewhere,} \end{cases} \\ &\leq \begin{cases} -\eta S^2(t) + G(\mathbf{x}, u(t)) S(t) u_{cc}(t) \\ \quad + |S(t)| \\ \quad \cdot (|\delta_f(\mathbf{x})| + |G(\mathbf{x}, u(t)) - \underline{g}| |u'_{ac}(t)|) & \text{if } \widehat{G}(\mathbf{x}, t) \leq \underline{g}, \\ -\eta S^2(t) + G(\mathbf{x}, u(t)) S(t) u_{cc}(t) \\ \quad + |S(t)| (|\delta_f(\mathbf{x})| + |\delta_g(\mathbf{x})| |u'_{ac}(t)|) & \text{elsewhere,} \end{cases} \end{aligned}$$

$$\begin{aligned}
 & \leq \begin{cases} -\eta S^2(t) + G(\mathbf{x}, u(t)) S(t) u_{cc}(t) \\ \quad + |S(t)| (\bar{\delta}_f + (\bar{g} - \underline{g}) |u'_{ac}(t)|) & \text{if } \widehat{G}(\mathbf{x}, t) \leq \underline{g}, \\ -\eta S^2(t) + G(\mathbf{x}, u(t)) S(t) u_{cc}(t) \\ \quad + |S(t)| (\bar{\delta}_f + \bar{\delta}_g |u'_{ac}(t)|) & \text{elsewhere,} \end{cases} \\
 & \leq -\eta S^2(t) + G(\mathbf{x}, u(t)) S(t) u_{cc}(t) \\
 & \quad + |S(t)| (\bar{\delta}_f + D(\widehat{G}(\mathbf{x}, t)) |u'_{ac}(t)|), \tag{34}
 \end{aligned}$$

where the switching function  $D(\widehat{G}(\mathbf{x}, t))$  is defined in (26).

Replacing the compensation controller  $u_{cc}(t)$  in (34) with its expression in (25), and noticing that  $\text{sgn}(S(t))S(t) = |S(t)|$ , we can get

$$\begin{aligned}
 \dot{V}(\mathbf{x}, t) & \leq -\eta S^2(t) - \frac{G(\mathbf{x}, u(t))}{\underline{g}} \\
 & \quad \cdot (\bar{\delta}_f + D(\widehat{G}(\mathbf{x}, t)) |u'_{ac}(t)|) \text{sgn}(S(t)) S(t) \\
 & \quad + |S(t)| (\bar{\delta}_f + D(\widehat{G}(\mathbf{x}, t)) |u'_{ac}(t)|) \\
 & \leq -\eta S^2(t) - \left( \frac{G(\mathbf{x}, u(t))}{\underline{g}} - 1 \right) \\
 & \quad \cdot (\bar{\delta}_f + D(\widehat{G}(\mathbf{x}, t)) |u'_{ac}(t)|) |S(t)| \\
 & \leq -\eta S^2(t) \leq 0. \tag{35}
 \end{aligned}$$

From (32) and (35), we can get  $V(\mathbf{x}, t) > 0$  and  $\dot{V}(\mathbf{x}, t) \leq 0$ . Hence, the close-loop controlled system is stable under the effect of the controller. Also, we can obtain  $S(t) \in L_\infty$ ,  $\|\bar{\theta}_f(t)\| \in L_\infty$  and  $\|\bar{\theta}_g(t)\| \in L_\infty$ . These reveal that the tracking error and the adjusted parameters are bounded.

In addition, from the inequality in (35), we have the following inequality:

$$\begin{aligned}
 \int_0^\infty \eta S^2(t) dt & \leq - \int_0^\infty \dot{V}(\mathbf{x}, t) dt \\
 & = V(\mathbf{x}, 0) - V(\mathbf{x}, \infty) < \infty. \tag{36}
 \end{aligned}$$

The inequality in (36) implies that  $S(t) \in L_2$ , and incorporating  $S(t) \in L_\infty$  leads to  $S(t) \in L_2 \cap L_\infty$ . On the other hand, using (5),  $e(t) \in L_\infty$ ,  $\dot{e}(t) \in L_\infty$ ,  $\ddot{e}(t) \in L_\infty$ , and  $\dot{S}(t) \in L_\infty$  can be determined. Then, using Barbalat's lemma [6], we can get  $\lim_{t \rightarrow \infty} S(t) = 0$  leading to  $\lim_{t \rightarrow \infty} e(t) = 0$ . Therefore, the perfect tracking performance is achieved and the system stability is ensured, finishing the proof.  $\square$

## 5. Illustrative Example

This section presents the simulation results of the proposed control method for a class of unknown nonlinear dynamical system to illustrate the stability and effectiveness of the control algorithm.

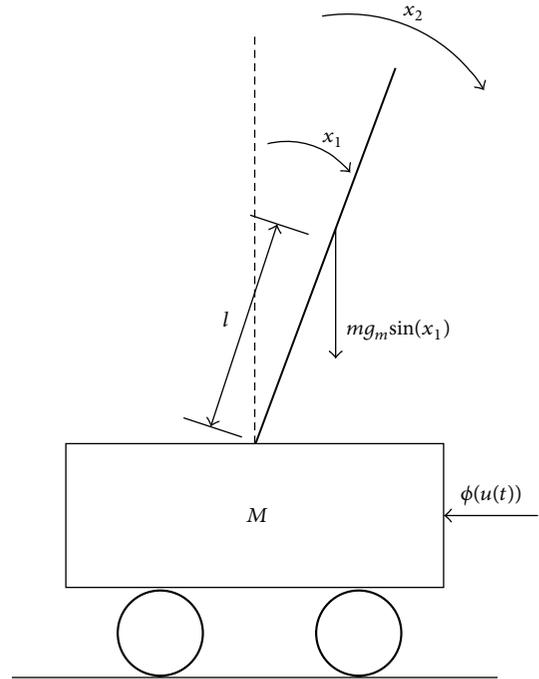


FIGURE 4: The inverted pendulum system with input nonlinearity.

Let us consider the inverted pendulum system with input nonlinearity as shown in Figure 4. If  $x_1$  is the angle of the pendulum with respect to the vertical line and  $x_2$  denotes the angular velocity, the dynamic equations governing the inverted pendulum system are given as [33]

$$\begin{aligned}
 \dot{x}_1 & = x_2, \\
 \dot{x}_2 & = f(\mathbf{x}) + g(\mathbf{x}) \phi(u(t)), \\
 y & = x_1, \tag{37}
 \end{aligned}$$

where

$$\begin{aligned}
 f(\mathbf{x}) & = \frac{mlx_2 \sin x_1 \cos x_1 - (M+m)g_m \sin x_1}{ml \cos^2 x_1 - (4/3)l(M+m)}, \\
 g(\mathbf{x}) & = \frac{-\cos x_1}{ml \cos^2 x_1 - (4/3)l(M+m)}. \tag{38}
 \end{aligned}$$

$\mathbf{x} = [x_1 \ x_2]^T$  is the state vector, while  $y = x_1$  is the output of the system.  $M$  is the mass of cart,  $m$  is the mass of rod,  $l$  is the half length of the rod,  $g_m$  is the gravitational acceleration, and  $\phi(u(t)) = g_u(u(t))u(t)$  is the nonlinear control input. In this example, it is assumed that  $g_u(u(t)) = 1 + 0.5 \sin(u(t))$ ,  $M = 1$  kg,  $m = 0.1$  kg,  $l = 0.5$  m, and  $g_m = 9.81$  m/s<sup>2</sup>. Let  $G(\mathbf{x}, u(t)) = g_u(u(t))g(\mathbf{x})$ ; the system in (37) can be rewritten as follows:

$$\begin{aligned}
 \dot{x}_1 & = x_2, \\
 \dot{x}_2 & = f(\mathbf{x}) + G(\mathbf{x}, u(t)) u(t), \\
 y & = x_1, \tag{39}
 \end{aligned}$$

where

$$G(\mathbf{x}, u(t)) = g_u(u(t))g(\mathbf{x}) = \frac{-\cos x_1(1 + 0.5 \sin(u(t)))}{ml \cos^2 x_1 - (4/3)l(M + m)}. \quad (40)$$

With the given model parameters, it is easy to find that  $G(\mathbf{x}, u(t)) > 0$  when  $|x_1| < \pi/2$ . Let  $y_d(t) = (\pi/30) \sin(t)$  be the desired trajectory; then the control objective is to design a controller  $u(t)$  such that the output  $y = x_1$  follows the desired trajectory  $y_d(t) = (\pi/30) \sin(t)$  as close as possible.

Since the inverted pendulum is the second order system, a fuzzy neural network with 2 inputs,  $x_1$  and  $x_2$ , and 2 outputs,  $\hat{f}(\mathbf{x}, t)$  and  $\hat{G}(\mathbf{x}, t)$ , is designed. Nine fuzzy rules are used to construct the fuzzy inference of the network. When the inputs are normalized in a range  $[-1 \ 1]$ , the membership functions with  $j = 1, 2$  are chosen as follows:

$$\begin{aligned} \mu_{A_j^1}(x) &= \exp\left[-\frac{(x+1)^2}{2 * 0.2^2}\right], \\ \mu_{A_j^2}(x) &= \exp\left[-\frac{(x+0.75)^2}{2 * 0.2^2}\right], \\ \mu_{A_j^3}(x) &= \exp\left[-\frac{(x+0.5)^2}{2 * 0.2^2}\right], \\ \mu_{A_j^4}(x) &= \exp\left[-\frac{(x+0.25)^2}{2 * 0.2^2}\right], \\ \mu_{A_j^5}(x) &= \exp\left[-\frac{(x+0)^2}{2 * 0.2^2}\right], \\ \mu_{A_j^6}(x) &= \exp\left[-\frac{(x-0.25)^2}{2 * 0.2^2}\right], \\ \mu_{A_j^7}(x) &= \exp\left[-\frac{(x-0.5)^2}{2 * 0.2^2}\right], \\ \mu_{A_j^8}(x) &= \exp\left[-\frac{(x-0.75)^2}{2 * 0.2^2}\right], \\ \mu_{A_j^9}(x) &= \exp\left[-\frac{(x-1)^2}{2 * 0.2^2}\right]. \end{aligned} \quad (41)$$

On the other hand, the designed parameters for modified adaptive controller are chosen as  $\eta = 50$ ,  $\lambda = 60$ ,  $\mathbf{W}_f = \mathbf{W}_g = 20I$ , where  $I$  is a 9-by-9 unit matrix. The designed parameters for compensation controller are arranged as  $\bar{\delta}_f = 0.05$ ,  $\bar{\delta}_g = 0.05$ ,  $\bar{g} = 1.5$ , and  $\bar{\gamma} = 2$ . In the numerical simulations, the fourth-order Runge-Kutta is used to solve the problem with time step size of 0.001 second. The initial condition  $\mathbf{x}(0) = [0.2 \ 0.2]^T$  is chosen for the simulation. Also, the initial values of adjusted parameters of the neural network are selected to be  $\theta_{fi}(0) = 1$ ,  $\theta_{gi}(0) = 2$ ,  $i = 1, 2, \dots, 9$ . Figure 5 shows the tracking performance in that the output,  $y = x_1$ , follows the desired trajectory,  $y_d(t) = (\pi/30) \sin(t)$ , successfully. In Figure 6, the tracking error which asymptotically converges

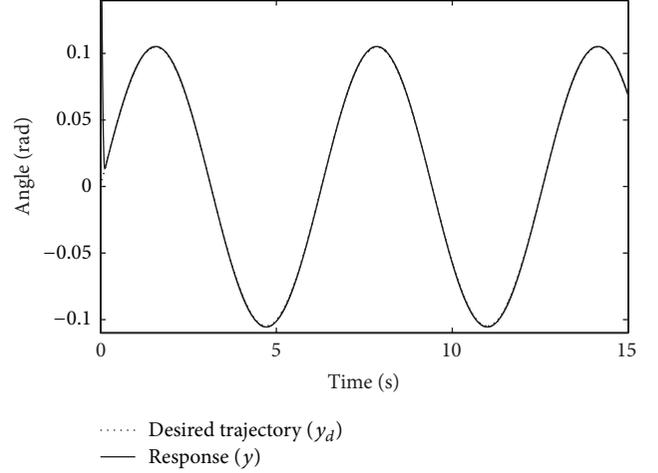


FIGURE 5: The tracking performance.

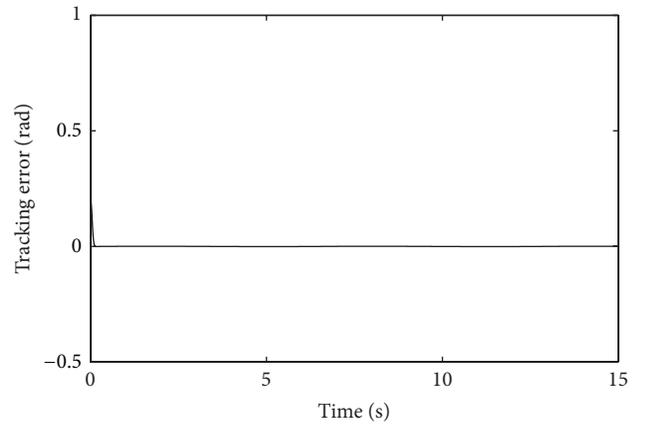


FIGURE 6: The tracking error  $e(t)$ .

to zero under the effect of the proposed controller is depicted. Also, the value of the sliding surface which tends to zero is displayed in Figure 7. Figure 8 especially illustrates that the value of function  $\hat{G}(\mathbf{x}, t)$  is always more than or equal to  $\bar{g} = 1.5$  during the period of simulation. This demonstrates that the singularity problem can be completely avoided with the proposed method. The control input  $u(t)$  and nonlinear control input  $\phi(u(t))$  during the simulation period are shown in Figures 9 and 10, respectively.

Furthermore, the proposed control method is compared with the observer-based adaptive fuzzy control (O AFC) developed in [33] to demonstrate the advantage of our control approach. The designed parameters for the controller and the observer are maintained as provided in [33]. These parameters are given as follows:

$$\begin{aligned} K_c^T &= [100 \ 0], & K_0^T &= [40 \ 700], \\ P_1 &= \begin{bmatrix} 51 & 0.05 \\ 0.05 & 0.504 \end{bmatrix}, & P_2 &= \begin{bmatrix} 74 & -5 \\ -5 & 0.46 \end{bmatrix}, \\ \gamma_1 &= 70, & \gamma_2 &= 70, & \rho &= 0.01. \end{aligned} \quad (42)$$

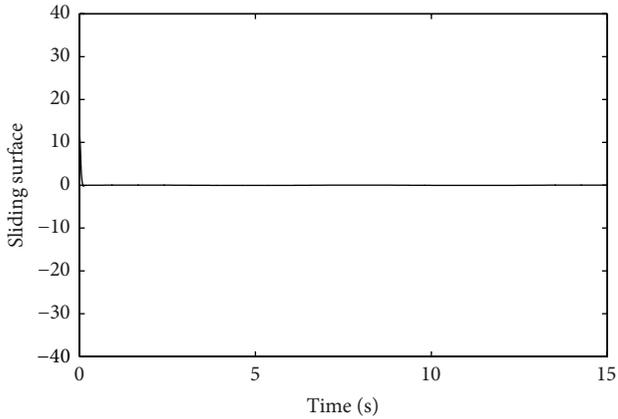


FIGURE 7: The sliding surface  $S(t)$ .

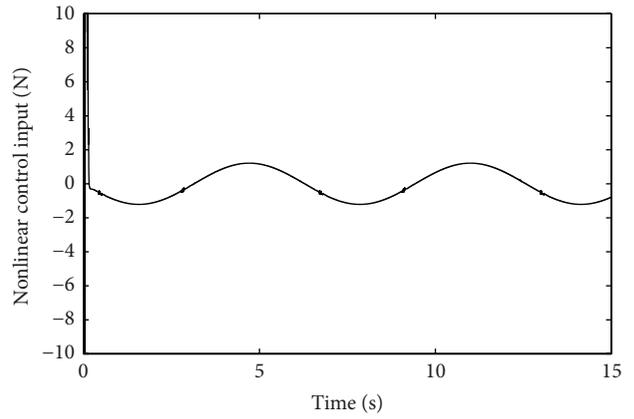


FIGURE 10: The trajectory of the nonlinear control input  $\phi(u(t))$ .

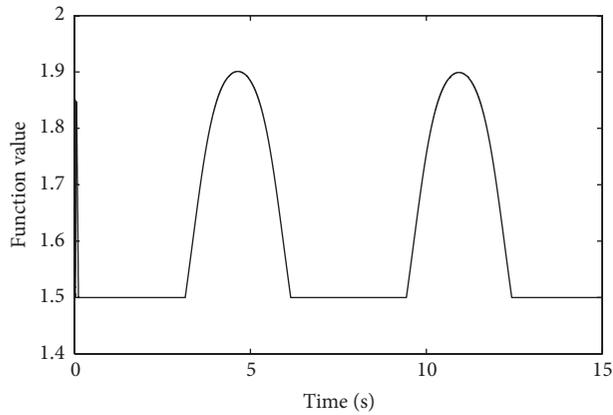


FIGURE 8: The value of function  $\widehat{G}(x, t)$  during the simulation time.

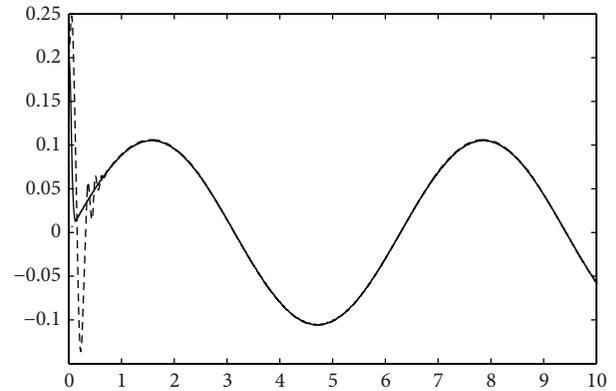


FIGURE 11: The tracking performances of the proposed method and O AFC.

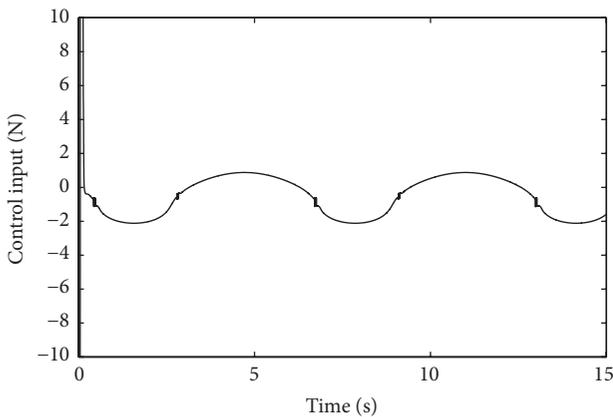


FIGURE 9: The trajectory of the control input  $u(t)$ .

Both simulations of the proposed method and O AFC are performed in the case of input nonlinearity. The tracking performances of the proposed method and O AFC are depicted together in Figure 11. The proposed method, denoted by a solid line, shows better tracking performance than O AFC. The proposed method especially demonstrates the absolute advantage in the settling time. In Figure 12, both tracking

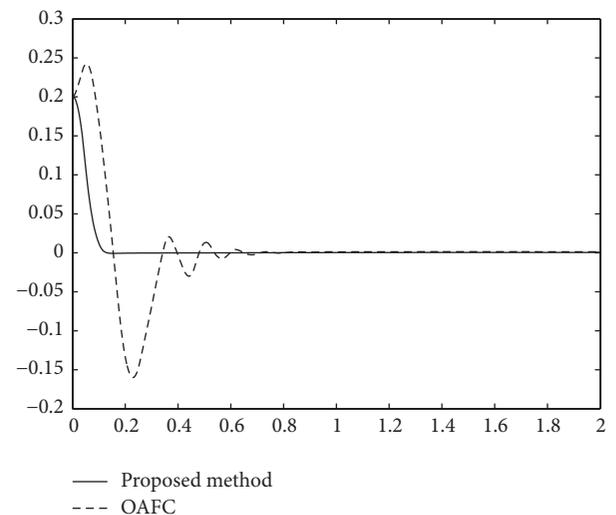


FIGURE 12: The tracking errors of the proposed method and O AFC.

errors of these methods are displayed. The tracking error of the proposed control method significantly converges to zero faster than OAFC. Also, the tracking error value of the proposed method is less than OAFC even in the steady state.

## 6. Conclusion

In this paper, we have proposed the advanced adaptive control approach for a class of SISO nonlinear systems subjected to nonlinear inputs. The proposed control scheme has two parts where the adaptive fuzzy controller simulates the ideal feedback control law, while the compensation controller reduces the effects of the approximation errors and keeps the system robust. With the improvement in the controller design, the proposed controller not only meets the control objective even with nonlinear inputs but also surely avoids the singularity problem that may be a serious drawback in the indirect adaptive control techniques based on fuzzy/neural networks approximations. Finally, numerical simulations were executed to verify the validity of the proposed method.

## Conflict of Interests

The authors, Tat-Bao-Thien Nguyen, Teh-Lu Liao, and Jun-Juh Yan, declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

The authors gratefully acknowledge the support of National Science Council of Taiwan through Grants NSC101-2221-E-006-190-MY2 and NSC102-2221-E-366-003.

## References

- [1] M. Krstić, I. Kanellakopoulos, and P. V. Kokotović, *Nonlinear and Adaptive Control Design*, John Wiley & Sons, New York, NY, USA, 1995.
- [2] H. K. Khalil, *Nonlinear Systems*, Prentice Hall PTR, Englewood Cliffs, NJ, USA, 2002.
- [3] D. Lalili, A. Mellit, N. Lourci, B. Medjahed, and E. M. Berkouk, "Input output feedback linearization control and variable step size MPPT algorithm of a grid-connected photovoltaic inverter," *Renewable Energy*, vol. 36, no. 12, pp. 3282–3291, 2011.
- [4] S. Mohammed, P. Poinet, P. Fraisse, and D. Guiraud, "Toward lower limbs movement restoration with input-output feedback linearization and model predictive control through functional electrical stimulation," *Control Engineering Practice*, vol. 20, no. 2, pp. 182–195, 2012.
- [5] J. Seo, R. Venugopal, and J.-P. Kenné, "Feedback linearization based control of a rotational hydraulic drive," *Control Engineering Practice*, vol. 15, no. 12, pp. 1495–1507, 2007.
- [6] J. J. E. Slotine and W. Li, *Applied Nonlinear Control*, Pearson Education Taiwan, Taipei, Taiwan, 2005.
- [7] J.-J. Yan, "Sliding mode control design for uncertain time-delay systems subjected to a class of nonlinear inputs," *International Journal of Robust and Nonlinear Control*, vol. 13, no. 6, pp. 519–532, 2003.
- [8] O. Castillo, J. R. Castro, P. Melin, and A. Rodriguez-Diaz, "Universal approximation of a class of interval type-2 fuzzy neural networks in nonlinear identification," *Advances in Fuzzy Systems*, vol. 2013, Article ID 136214, 16 pages, 2013.
- [9] D. Driankov and R. Palm, *Advances in Fuzzy Control*, Physica, Heidelberg, Germany, 2013.
- [10] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.
- [11] Y. Li, T. Li, and X. Jing, "Indirect adaptive fuzzy control for input and output constrained nonlinear systems using a barrier Lyapunov function," *International Journal of Adaptive Control and Signal Processing*, vol. 28, no. 2, pp. 184–199, 2014.
- [12] C.-T. Lin and C. S. G. Lee, *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1996.
- [13] O. Omidvar and D. L. Elliott, *Neural Systems for Control*, Academic Press, New York, NY, USA, 1997.
- [14] L.-X. Wang and J. M. Mendel, "Fuzzy basis functions, universal approximation, and orthogonal least-squares learning," *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp. 807–814, 1992.
- [15] L. X. Wang, *A Course in Fuzzy Systems and Control*, Prentice Hall, New York, NY, USA, 1997.
- [16] C. L. P. Chen, Y.-J. Liu, and G.-X. Wen, "Fuzzy neural network-based adaptive control for a class of uncertain nonlinear stochastic systems," *IEEE Transactions on Cybernetics*, vol. 44, no. 5, pp. 583–593, 2014.
- [17] M. Jamshidi, N. Vadiiee, and T. J. Ross, *Fuzzy Logic and Control: Software and Hardware Applications*, Prentice Hall PTR, 1993.
- [18] Y.-G. Leu, T.-T. Lee, and W.-Y. Wang, "On-line tuning of fuzzy-neural network for adaptive control of nonlinear dynamical systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 27, no. 6, pp. 1034–1043, 1997.
- [19] 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*, vol. 21, no. 3, pp. 580–590, 2015.
- [20] T.-B.-T. Nguyen, T.-L. Liao, H.-H. Kuo, and J.-J. Yan, "An improved adaptive tracking controller of permanent magnet synchronous motor," *Abstract and Applied Analysis*, vol. 2014, Article ID 987308, 12 pages, 2014.
- [21] T.-B. Nguyen, T.-L. Liao, and J.-J. Yan, "Adaptive sliding mode control of chaos in permanent magnet synchronous motor via fuzzy neural networks," *Mathematical Problems in Engineering*, vol. 2014, Article ID 868415, 11 pages, 2014.
- [22] C.-H. Wang, W.-Y. Wang, T.-T. Lee, and P.-S. Tseng, "Fuzzy B-spline membership function (BMF) and its applications in fuzzy-neural control," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 25, no. 5, pp. 841–851, 1995.
- [23] M. Bahita and K. Belarbi, "Radial basis function controller of a class of nonlinear systems using mamdani type as a fuzzy estimator," *Procedia Engineering*, vol. 41, pp. 501–509, 2012.
- [24] Y.-C. Chang, "Adaptive fuzzy-based tracking control for nonlinear SISO systems via VSS and H<sub>∞</sub> approaches," *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 2, pp. 278–292, 2001.
- [25] M. Chemachema, "Output feedback direct adaptive neural network control for uncertain SISO nonlinear systems using a fuzzy estimator of the control error," *Neural Networks*, vol. 36, pp. 25–34, 2012.
- [26] T.-C. Lin, "Based on interval type-2 fuzzy-neural network direct adaptive sliding mode control for SISO nonlinear systems," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 12, pp. 4084–4099, 2010.

- [27] Z.-C. Qiu, B. Wang, X.-M. Zhang, and J.-D. Han, "Direct adaptive fuzzy control of a translating piezoelectric flexible manipulator driven by a pneumatic rodless cylinder," *Mechanical Systems and Signal Processing*, vol. 36, no. 2, pp. 290–316, 2013.
- [28] J. T. Spooner and K. M. Passino, "Stable adaptive control using fuzzy systems and neural networks," *IEEE Transactions on Fuzzy Systems*, vol. 4, no. 3, pp. 339–359, 1996.
- [29] S. Tong and H.-X. Li, "Direct adaptive fuzzy output tracking control of nonlinear systems," *Fuzzy Sets and Systems*, vol. 128, no. 1, pp. 107–115, 2002.
- [30] M. J. Zhang, C. Shao, and S. K. Li, "Self-adaptive fuzzy control with unknown control gain function," *Journal of Dalian University of Technology*, vol. 54, no. 2, pp. 257–261, 2014.
- [31] M. Zhang, C. Shao, and S. Li, "Observer-based robust direct adaptive fuzzy control," in *Proceedings of the IEEE International Conference on Vehicular Electronics and Safety (ICVES '13)*, pp. 248–252, Dongguan, China, July 2013.
- [32] A. Boulkroune, N. Bounar, M. M'saad, and M. Farza, "Indirect adaptive fuzzy control scheme based on observer for nonlinear systems: a novel SPR-filter approach," *Neurocomputing*, vol. 135, pp. 378–387, 2014.
- [33] S. Tong, H.-X. Li, and W. Wang, "Observer-based adaptive fuzzy control for SISO nonlinear systems," *Fuzzy Sets and Systems*, vol. 148, no. 3, pp. 355–376, 2004.
- [34] S. Wang, L. Hou, L. Dong, and H. Xiao, "Adaptive fuzzy sliding mode control of uncertain nonlinear SISO systems," *Procedia Engineering*, vol. 24, pp. 33–37, 2011.
- [35] O. Cerman and P. Hušek, "Adaptive fuzzy sliding mode control for electro-hydraulic servo mechanism," *Expert Systems with Applications*, vol. 39, no. 11, pp. 10269–10277, 2012.
- [36] E. Nechadi, M. N. Harmas, A. Hamzaoui, and N. Essounbouli, "A new robust adaptive fuzzy sliding mode power system stabilizer," *International Journal of Electrical Power and Energy Systems*, vol. 42, no. 1, pp. 1–7, 2012.
- [37] W.-S. Yu and C.-C. Weng, "An observer-based adaptive neural network tracking control of robotic systems," *Applied Soft Computing Journal*, vol. 13, no. 12, pp. 4645–4658, 2013.
- [38] A. Poursamad and A. H. Davaie-Markazi, "Robust adaptive fuzzy control of unknown chaotic systems," *Applied Soft Computing*, vol. 9, no. 3, pp. 970–976, 2009.
- [39] S. Tong, T. Wang, and J. T. Tang, "Fuzzy adaptive output tracking control of nonlinear systems," *Fuzzy Sets and Systems*, vol. 111, no. 2, pp. 169–182, 2000.
- [40] M. R. Soltanpour, B. Zolfaghari, M. Soltani, and M. H. Khooban, "Fuzzy sliding mode control design for a class of nonlinear systems with structured and unstructured uncertainties," *International Journal of Innovative Computing, Information and Control*, vol. 9, no. 7, pp. 2713–2726, 2013.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

