

Research Article

Tuning of a TS Fuzzy Output Regulator Using the Steepest Descent Approach and ANFIS

Ricardo Tapia-Herrera, Jesús Alberto Meda-Campaña, Samuel Alcántara-Montes, Tonatiuh Hernández-Cortés, and Lizbeth Salgado-Conrado

Instituto Politécnico Nacional, SEPI-ESIME Zacatenco, Avenue IPN S/N, 07738 México, DF, Mexico

Correspondence should be addressed to Jesús Alberto Meda-Campaña; jmedac@ipn.mx

Received 15 March 2013; Accepted 27 May 2013

Academic Editor: Qingsong Xu

Copyright © 2013 Ricardo Tapia-Herrera 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 exact output regulation problem for Takagi-Sugeno (TS) fuzzy models, designed from linear local subsystems, may have a solution if input matrices are the same for every local linear subsystem. Unfortunately, such a condition is difficult to accomplish in general. Therefore, in this work, an adaptive network-based fuzzy inference system (ANFIS) is integrated into the fuzzy controller in order to obtain the optimal fuzzy membership functions yielding adequate combination of the local regulators such that the output regulation error in steady-state is reduced, avoiding in this way the aforementioned condition. In comparison with the steepest descent method employed for tuning fuzzy controllers, ANFIS approximates the mappings between local regulators with membership functions which are not necessary known functions as Gaussian bell (gbell), sigmoidal, and triangular membership functions. Due to the structure of the fuzzy controller, Levenberg-Marquardt method is employed during the training of ANFIS.

1. Introduction

A fundamental problem in dynamic systems is to control a plant in order to have its outputs tracking some reference signals produced by an exosystem (external generator), while the stability property of the closed-loop system is guaranteed. In this regard, the output regulation problem [1] consists of finding the control law capable of

- (a) stabilizing the closed-loop system when the plant is not influenced by the exosystem,
- (b) taking the tracking error asymptotically to zero when the plant is under the action of the exosystem.

One of the first works on output regulation theory presented in the literature was developed by Francis and Wonham in 1975 [2]. This paper obtains structural criteria necessary to synthesize multivariable linear regulators with structural stability to small perturbations. The main contribution of this work is that structural stability requires feedback from the regulate variable, and these structural features are called “internal model principle.” Later in 1977 [3], the linear regulation theory for tracking a reference is proposed by Francis,

by finding that the solution for the output regulation of an autonomous linear system subject to perturbations and reference signals can be obtained from a system of linear matrix equations.

Isidori and Byrnes [1] extend the results obtained by Francis to the general case in which the plant and/or the exosystem are nonlinear. They also show that Francis equations are a particular case of nonlinear output regulation equations. In addition, they also show, as in linear case, that if the plant is exponentially stabilizable via output feedback, the solution for the nonlinear case requires an error feedback regulator. Unfortunately, the so-called Francis-Isidori-Byrnes (FIB) equations used to solve the nonlinear output regulation problem involve a set of nonlinear partial differential equations which in many cases are too difficult, or even impossible to solve in a practical way (see Figure 1).

On the other hand, in the real world, there exists a wide range of problems that involve the analysis of uncertain and inaccurate information, and that is the main reason for the “soft computing.” The term soft computing, according to Zadeh, is “A collection of methodologies that aim to exploit the tolerance for imprecision and uncertainty to achieve

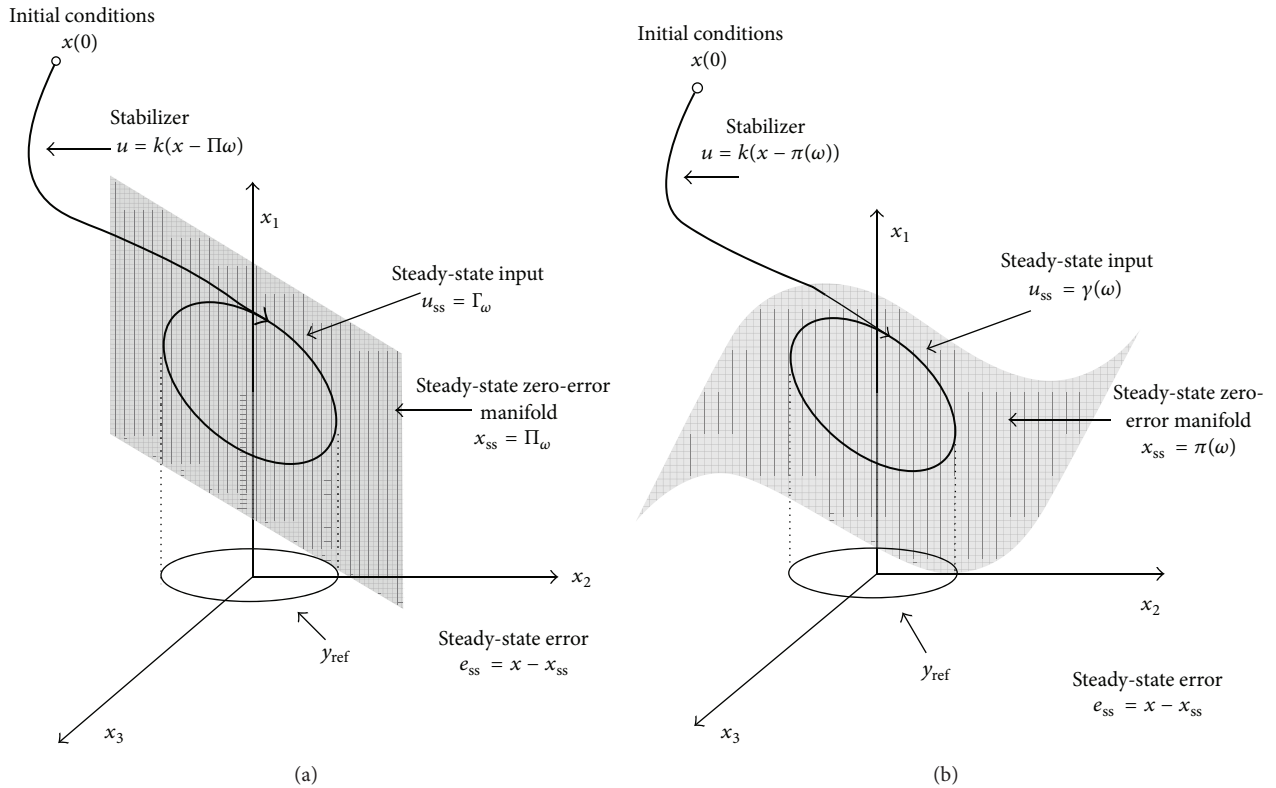


FIGURE 1: Output regulation scheme for (a) linear systems and (b) nonlinear systems [4].

tractability, robustness, and low solution cost.” Its principal constituents are fuzzy logic, neurocomputing, and probabilistic reasoning [5]. Each of these approaches has features that make them suitable to solve different problems. Fuzzy systems are characterized by their ability to handle and use inaccurate approximate reasoning, while neural networks have the property of learning. Many of these tools have been used with efficiency to solve complex real world problems as diagnosis, estimation, control, and autonomy of systems.

In that sense, some techniques have been developed to characterize nonlinear systems by means of linear local subsystems [5]. One of these approaches is the well-known Takagi Sugeno (TS) fuzzy modeling. This technique allows describing the nonlinear dynamics by means of a suitable “blending” of linear subsystems, each of them corresponding to different operation points. Basically, the “combination” is performed by a weighted summation of linear local subsystems. Thus, local controllers can be designed for each subsystem, obtaining the aggregate controller by using the same membership functions of the TS fuzzy plant.

As mentioned previously, the set of differential equations derived by Isidori and Byrnes in nonlinear regulation is difficult to solve, representing an interesting opportunity to apply soft computing: in particular neural networks and fuzzy logic.

For instance, in [6], an iterative algorithm with the structure of a recurrent neural network is presented. The network is introduced with the purpose of dealing with model errors caused by the estimation of parameters of an off-line

training and the possible dynamic changes; an online learning approach providing the necessary parameters for the network is also used. In this sense, the network can capture the uncertainties of the system, and the regulator adjusts the control action in order to guarantee the regulation condition. That is an attempt to combine the identification capability of neural networks and the stability properties of the control theory. As a result, the control structure learns the system dynamics and ensures the asymptotic convergence of the error. Subsequently, this method was applied at the control scheme of a solar plant [7]. Zhang and Wang [8] propose an approximation in power series form, wherein a neural network carries out the synthesis and self-tuning of controller in real time. This method ensures regulation properties, namely, stability and asymptotic tracking.

Later in [9], the fuzzy output regulation is formulated, in which a nonlinear system described by a Takagi-Sugeno fuzzy model follows a reference signal generated by a fuzzy exosystem. It was also shown that the fuzzy regulation problem can be solved by means of local linear regulator only if the following conditions are satisfied.

- (i) The zero error manifold $\pi(\omega)$ is equal for every local linear subsystem (see Figure 1).
- (ii) Input matrices B_i are equal for every local linear subsystem, that is, $(B_1 = B_2 = \dots = B_i)$.

Unfortunately, the aforementioned conditions are not satisfied in a great number of systems. A proposal to overcome this problem is presented in [5, 10], wherein the fuzzy

regulator is designed on the overall fuzzy model instead of local linear subsystems. The advantage of this approach is the global convergence of the tracking error in the interest region, in contrast to local properties of classical method of nonlinear control. Moreover, the solution of the fuzzy regulation problem is given in terms of the “dynamical Francis equations,” and a very practical way to solve such a dynamical equation is also given. However, a disadvantage of this approach is that exact fuzzy controller is far more complex than the regulator designed from the fuzzy summation of local linear regulators, in general. For that reason, in the present work, an approach based on the local design of linear regulators is proposed, but considering that the nonlinearity of the controller may be different from the nonlinearity of the plant. In other words, it will be assumed that the membership functions of the fuzzy plant and the membership functions of fuzzy regulator are different. Therefore, the appropriate membership functions for the fuzzy regulators will be obtained by means of an adaptive network-based fuzzy inference system (ANFIS) and the steepest descent approach.

A hybrid intelligent system combines two or more soft computing techniques [10]. Such combination allows taking advantage of each tool, improving the behavior of the overall system. According to [11, 12], the following combinations can be found in the literature:

- (i) fuzzy controllers tuned by neural networks,
- (ii) fuzzy controller tuned by evolutionary computation,
- (iii) neural networks synthesized by evolutionary computation,
- (iv) neural networks controlled by fuzzy logic,
- (v) evolutionary computation controlled by fuzzy logic.

The Artificial Neural Networks (ANN) have shown great efficiency in different fields as identification, classification, or recognition patterns. On the other hand, a system based on fuzzy logic has the capability of working with uncertain information and taking decisions based on rules previously established by an expert system. A neurofuzzy system takes the inference under cognitive uncertainty from fuzzy logic and from ANNs their learning capability, adaption, parallelism, and generalization [13, 14].

The performance of a fuzzy controller is extensively linked to the shape of membership functions that represent linguistic variables [15], so as the binary operators [16]. These parameters conform the structure of fuzzy controller and generally are defined by an expert to trial and error [17]. ANN brings the possibility of avoiding the trial and error procedure [18–20]. Therefore, ANN gives to fuzzy controller the capability of learning, which means that fuzzy system can be tuned by itself [21]. Besides, the neural approach can not only be used to adjust the parameters of the fuzzy controller but also an ANN can be used as a fuzzy controller, in the sense that the fuzzy controller will have the structure of a neural network [22]. For instance, Moreno-Velo develops Xfuzzy which is a program where several proposed approaches are applied to learning process on neurofuzzy systems. Xfuzzy integrates a wide variety of supervised learning algorithms [23].

In [24, 25], the authors present methodologies that reduce the time of training for neurofuzzy systems with the aim of using them on real-time applications.

(Adaptive Neural-Fuzzy Network Inference System) (ANFIS) proposed by Jang [26] builds a mapping between input-output pairs, using IF-THEN rules. ANFIS employs hybrid training, combining backpropagation and the least-squares estimator, and it has been used for modeling nonlinear functions, identification of nonlinear components in control systems, and for prediction of chaotic series.

So, the main contribution of the present work is to develop an approach based on ANFIS that is capable of approximating the appropriate membership functions, such that tracking error for the overall fuzzy system is reduced when the fuzzy controller is obtained from the fuzzy summation of local linear regulators, while avoiding the restriction of proposing a priori form of the membership functions, which is the main disadvantage of the steepest descent approach.

The rest of the paper is organized as follows. In Section 2, a review of basic concepts of fuzzy output regulation is presented. In Section 3, an overview of ANFIS and an introductory example using the steepest descent in fuzzy regulation are given. In Section 4, some cases of study are solved and their respective simulation results are depicted. Finally, in Section 5, some conclusions are drawn.

2. The Output Regulation Problem

From a given linear system

$$\begin{aligned}\dot{x} &= Ax + Bu, \\ y &= Cx\end{aligned}\quad (1)$$

and an exosystem

$$\begin{aligned}\dot{\omega} &= S\omega, \\ y_{\text{ref}} &= Q\omega,\end{aligned}\quad (2)$$

it is desired to design a controller that stabilizes asymptotically the system (1) when the exosystem (2) is disconnected, that is, $(\omega = 0)$, and achieving $e(t) \rightarrow 0$ as $t \rightarrow \infty$ when (1) is under the influence of (2). According to [26], the controller is $u = K(x - \Pi\omega) + \Gamma\omega$ and the tracking error is $e_{\text{ss}} = x - x_{\text{ss}}$, where $x_{\text{ss}} = \Pi\omega$

$$e_{\text{ss}} = x - \Pi\omega \quad \therefore \dot{e}_{\text{ss}} = \dot{x} - \Pi\dot{\omega} \longrightarrow \dot{x} = \dot{e}_{\text{ss}} + \Pi\dot{\omega}. \quad (3)$$

By substituting the previous equations in (1), one gets

$$\begin{aligned}\dot{e}_{\text{ss}} &= Ae_{\text{ss}} + A\Pi\omega + BK e_{\text{ss}} + BK\Pi\omega \\ &\quad - BK\Pi\omega + B\Gamma\omega - \Pi\dot{\omega}.\end{aligned}\quad (4)$$

But K has to be designed such that $(A + BK)$ is asymptotically stable. Therefore, $e(t) \rightarrow 0$ as $t \rightarrow \infty$ if and only if

$$\begin{aligned}\Pi S &= A\Pi + B\Gamma, \\ 0 &= C\Pi - Q.\end{aligned}\quad (5)$$

The set of (5) is called Francis equations. Dimensions of each matrix are

$$\begin{aligned} A &\in \mathbb{R}^{n \times n}, & B &\in \mathbb{R}^{n \times m}, \\ C &\in \mathbb{R}^{m \times n}, & \Pi &\in \mathbb{R}^{n \times r}, \\ S &\in \mathbb{R}^{r \times r}, & \Gamma &\in \mathbb{R}^{m \times r}, \\ P &\in \mathbb{R}^{n \times r}, & Q &\in \mathbb{R}^{m \times r}. \end{aligned} \quad (6)$$

According to Isidori and Byrnes, the extension of the regulation problem to the nonlinear case is

$$\begin{aligned} \dot{x} &= f(x, \omega, u), \\ y &= h(x), \\ \dot{\omega} &= s(\omega), \\ y_{\text{ref}} &= q(\omega). \end{aligned} \quad (7)$$

Thus, the control signal becomes $u = k[x - \pi(\omega)] + \gamma(\omega)$. Then, similar to the linear case, the error in steady-state is $\dot{e}_{\text{ss}} = \dot{x} - \dot{x}_{\text{ss}}$ with $x_{\text{ss}} = \Pi\omega$:

$$e_{\text{ss}} = f(x, \omega, u) - \frac{d}{dt}[\pi(\omega)], \quad (8)$$

$$e_{\text{ss}} = f(x, \omega, u) - \frac{\partial \pi}{\partial \omega} \dot{\omega}, \quad (9)$$

$$e_{\text{ss}} = f(x, \omega, u) - \frac{\partial \pi}{\partial \omega} s(\omega), \quad (10)$$

$$\begin{aligned} \frac{\partial \pi}{\partial \omega} s(\omega) &= f(\pi(\omega), \omega, \gamma(\omega)), \\ 0 &= h(\pi(\omega)) - q(\omega). \end{aligned} \quad (11)$$

The set (11) is called Francis-Isidori-Byrnes (FIB) equations [1], subjected to

$$\begin{aligned} f(0, 0, 0) &= 0, \\ h(0) &= 0, \\ \pi(0) &= 0, \\ \gamma(0) &= 0, \\ S(0) &= 0, \\ q(0) &= 0. \end{aligned} \quad (12)$$

2.1. Exact Fuzzy Output Fuzzy Regulation. The fuzzy model proposed by Takagi and Sugeno is a fuzzy system described by the rules of the type IF-THEN, where the consequent parts include the local dynamics represented by linear subsystems [27], that is.

Local linear subsystems:

$$\begin{aligned} \dot{x}(t) &= A_i x(t) + B_i u(t), & y(t) &= C_i x(t) \\ i &= 1, 2, \dots, r. \end{aligned} \quad (13)$$

Therefore, the nonlinear dynamics can be approximated by the following TS model:

$$\begin{aligned} \dot{x} &= \sum_{i=1}^{r_1} h_i^1 \{A_i x + B_i u\} + \sum_{i=1}^{r_1} h_i^1 P_i, \\ y &= \sum_{i=1}^{r_1} h_i^1 C_i x. \end{aligned} \quad (14)$$

In a similar way, the exosystem can be approximated by a TS fuzzy system of r_2 rules as follows:

$$\begin{aligned} \dot{\omega} &= \sum_{i=1}^{r_2} h_i^2 S_i \omega, \\ y_{\text{ref}} &= \sum_{i=1}^{r_2} h_i^2 Q_i \omega. \end{aligned} \quad (15)$$

Note that the membership functions of the plant are not necessarily the same as the exosystem functions. Fuzzy plant (14) and fuzzy exosystem (15) can be rewritten as

$$\begin{aligned} \dot{x} &= \tilde{A}(t)x(t) + \tilde{B}(t)u(t), \\ y &= \tilde{C}(t)x(t), \\ \dot{\omega} &= \tilde{S}(t)\omega(t), \\ y_{\text{ref}} &= \tilde{Q}(t)\omega(t). \end{aligned} \quad (16)$$

And the desired fuzzy controller is

$$u(t) = \sum_{i=1}^{r_1} h_i^1(z_1(t)) K_i [x(t) - \pi(\omega(t))] + \gamma(\omega(t)). \quad (17)$$

With this representation, the steady-state manifold is $\pi(\omega(t)) = \tilde{\Pi}(t)\omega(t)$, while the steady-state input is $\gamma(\omega(t)) = \tilde{\Gamma}(t)\omega(t)$ [9, 28]. $\tilde{\Pi}(t)$ and $\tilde{\Gamma}(t)$ can be obtained by differentiating the steady-state error:

$$\dot{e}_{\text{ss}} = \dot{x}(t) - \dot{\tilde{\Pi}}(t)\omega(t) - \tilde{\Pi}(t)\dot{\omega}(t). \quad (18)$$

In steady-state, one gets

$$\dot{\tilde{\Pi}}(t) + \tilde{\Pi}(t)\tilde{S}(t) = \tilde{A}(t)\tilde{\Pi}(t) + \tilde{B}(t)\tilde{\Gamma}(t) + \tilde{P}(t). \quad (19)$$

Solving (19) in an analytical way may be very difficult. But its numerical solution is given in [4] through a very practical way. However, the resulting controller is very complex, in general.

3. Overview of Adaptive-Network-Bases Fuzzy Inference System (ANFIS)

In Figure 2, an Adaptive-Network-Based Fuzzy Inference System network is shown. ANFIS is a system that employs the structure of fuzzy Takagi-Sugeno-Kang systems (TSK) and learning algorithms of neural networks with the purpose of

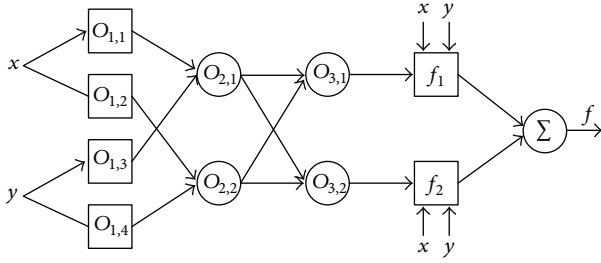


FIGURE 2: ANFIS architecture for 2 input variables.

adjusting the consequent and antecedent parameters of the fuzzy system.

Basically, ANFIS architecture is formed by five layers, each of them is described in the following.

Layer 1. Fuzzification process, variables x and y are the inputs to membership functions $O_{1,i}$. Generally, gbell functions are used in this stage, for example.

Rule 1. If x is $O_{1,1}$ and y is $O_{1,3}$, then $f_1 = p_1x + q_1y + r_1$.

Rule 2. If x is $O_{1,2}$ and y is $O_{1,4}$, then $f_2 = p_2x + q_2y + r_2$.

Consider

$$m_{A,j} = O_{1,j}(x) = \frac{1}{1 + ((x - c_i)/a_i)^2 b_j} \quad j = 1, 2 \quad (20)$$

or

$$m_{B,j} = O_{1,j}(y) = \frac{1}{1 + ((y - c_i)/a_i)^2 b_j} \quad j = 3, 4. \quad (21)$$

Parameters a_i , b_j , c_i are called premise parameters, where these premise parameters can be modified (Figure 3).

Layer 2. Nodes denoted with Π . In these nodes, the inferences between fuzzy sets are performed. The T -norm is used for firing strengths:

$$O_{2,j} = w_j = m_{A_j}(x) \cdot m_{B_j}(x), \quad j = 1, 2. \quad (22)$$

Layer 3. Normalization of firing strengths

$$O_{3,j} = \bar{w}_j = \frac{w_j}{w_1 + w_2}. \quad (23)$$

Layer 4. The consequent part of fuzzy system is in nodes O_i^4 , and it is given by a set of functions $f_i = p_i x + q_i y + r_i$, and they are activated according to the normalize firing strengths:

$$O_i^4 = \bar{w}_j f_j = \bar{w}_j (p_j x + q_j y + r_j). \quad (24)$$

Layer 5. Summation of input signals is obtained as

$$O_1^5 = \sum_j O_j^4 = \sum_j \bar{w}_j f_j. \quad (25)$$

3.1. Case of Study. Consider the following TS fuzzy system:

$$\begin{aligned} \dot{x}(t) &= \sum_{i=1}^2 h_i(x_1(t)) [A_i x(t) + B_i u(t)], \\ \dot{\omega}(t) &= S \omega(t), \\ e(t) &= \sum_{i=1}^2 h_i(x_1(t)) C_i x(t) - Q \omega(t), \end{aligned} \quad (26)$$

where

$$\begin{aligned} A_1 &= \begin{bmatrix} 0 & 1 \\ 2 & 0 \end{bmatrix}, & A_2 &= \begin{bmatrix} 0 & 1 \\ 3 & 0 \end{bmatrix}, \\ B_1 &= \begin{bmatrix} 0 \\ 2 \end{bmatrix}, & B_2 &= \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \end{aligned} \quad (27)$$

$$C_1 = C_2 = Q = [1 \ 0], \quad S = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix},$$

with the control

$$\begin{aligned} u(t) &= \underbrace{\sum_{i=1}^2 h_i(x_i(t)) K_i [x(t) - \Pi_i \omega(t)]}_{\text{fuzzy stabilizer}} \\ &\quad + \underbrace{\sum_{i=1}^2 h_i(x_i(t)) \Gamma_i \omega(t)}_{\text{fuzzy regulator}}. \end{aligned} \quad (28)$$

It is important to remark that the fuzzy regulator included in (28) is obtained by using the same membership functions of the plant.

Observe that in this example, x_1 should track ω_1 . This can be easily deduced from matrices C_1 , C_2 , and Q . Π_i and Γ_i are obtained from the solution of (5) for each of the two rules. Thus, a fuzzy regulator is designed on the basis of two local linear systems. The membership functions h_i are proposed as triangular ones satisfying $h_1 + h_2 = 1$.

The fuzzy steady-state manifold is

$$\begin{aligned} \pi(\omega(t)) &= \Pi_1 \omega(t) = \Pi_2 \omega(t) \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \omega_1(t) \\ \omega_2(t) \end{bmatrix} = \Pi \omega(t), \end{aligned} \quad (29)$$

while the steady-state input is

$$\begin{aligned} \gamma(\omega(t)) &= \sum_{i=1}^2 h_i(\omega_1(t)) \Gamma_i \omega(t) \\ &= \left[-\frac{3}{2} h_1(\omega_1(t)) - 4 h_2(\omega_1(t)) \right] \begin{bmatrix} \omega_1(t) \\ \omega_2(t) \end{bmatrix}. \end{aligned} \quad (30)$$

And the K_i gains are obtained from LMI approach [29] in order to ensure global stability:

$$\begin{aligned} K_1 &= [-1.61 \quad -0.35], \\ K_2 &= [-4.22 \quad -0.71]. \end{aligned} \quad (31)$$

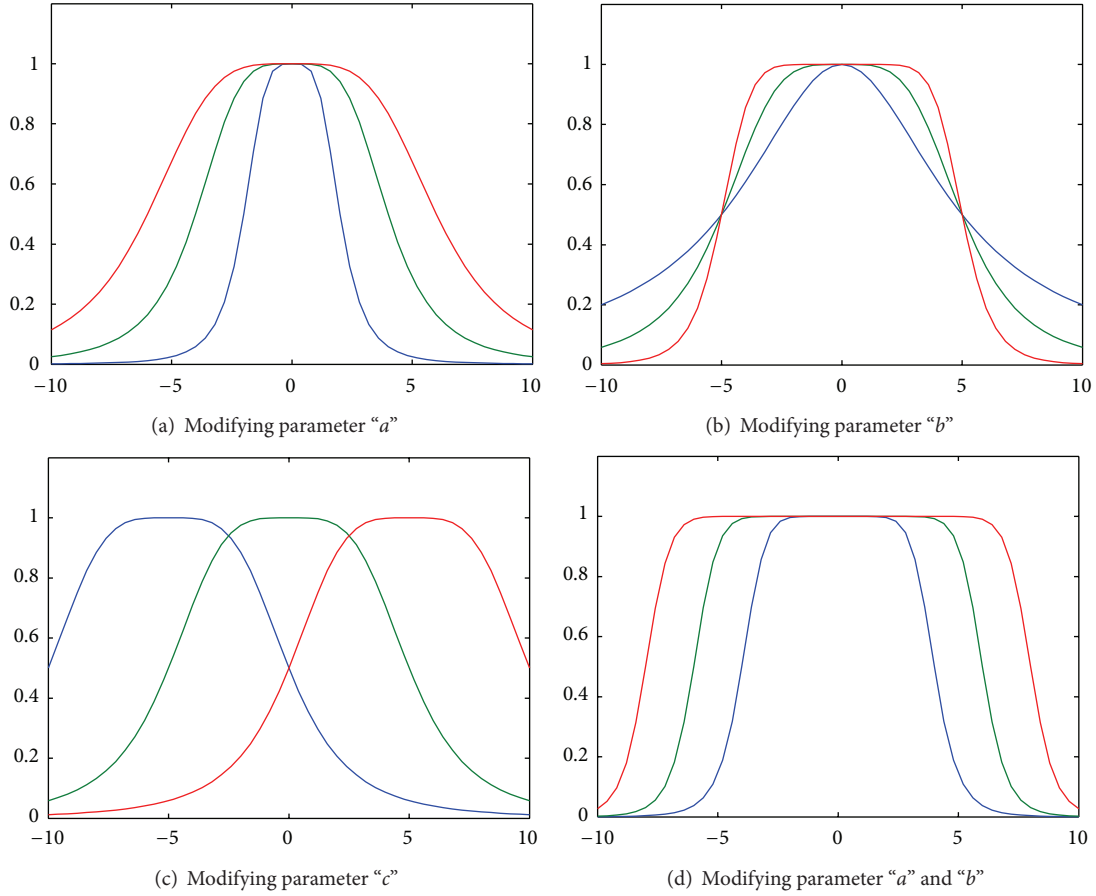


FIGURE 3: Shapes of gbell function with modified parameters.

From the matrices considered in the system (26), it is clear that B_1 and B_2 are different from each other. Therefore, an asymptotical error cannot be expected when the fuzzy regulator is constructed by the fuzzy summation of the local linear regulators using the same membership functions of the plant [28]. So, Figure 4 shows the behavior of state variable x_1 of fuzzy plant and state ω_1 of exosystem, while the tracking error is given in Figure 5.

In order to overcome the problem of controller (28), a decoupling between the membership functions of fuzzy stabilizer and fuzzy regulator is proposed, that is,

$$\begin{aligned}
 u(t) = & \underbrace{\sum_{i=1}^2 h_i(x_i(t)) K_i [x(t) - \Pi\omega(t)]}_{\text{Fuzzy Stabilizer}} \\
 & + \underbrace{\sum_{i=1}^2 \mu_i(x_i(t)) \Gamma_i \omega(t)}_{\text{Fuzzy Regulator}}.
 \end{aligned} \quad (32)$$

Observe that in this new controller, the membership functions of the regulation part are not the same as that considered in the fuzzy plant. In the following, these membership functions μ_i will be estimated by means of the Steepest Descent approach and ANFIS.

According to [5], a fuzzy system with membership functions of type gbell is a universal approximator and consequently gbell functions are used to approximate fuzzy mapping in local regulators, subjected to condition $\mu_2 = 1 - \mu_1$, where

$$\mu_1 = f(x, a, b, c) = \frac{1}{1 + |(x - c)/a|^{2b}}. \quad (33)$$

The parameters a , b , and c determine the shape of gbell function. As a consequence, the main goal is to estimate a , b , and c such that the resulting membership functions μ_1 and μ_2 improve the performance of the fuzzy controller (32). One of the most useful methods for tuning task found in the literature is the "steepest descent" method:

$$\begin{aligned}
 a_i(k+1) &= a_i(k) - \eta_{a_i} \frac{\partial E(a, b, c)}{\partial a} \quad i = 1, 2, \\
 b_i(k+1) &= b_i(k) - \eta_{b_i} \frac{\partial E(a, b, c)}{\partial b_i} \quad i = 1, 2, \\
 c_i(k+1) &= c_i(k) - \eta_{c_i} \frac{\partial E(a, b, c)}{\partial c_i} \quad i = 1, 2.
 \end{aligned} \quad (34)$$

It can be readily observed that expressions (34) depend on error $e_{ee}(k) = x(k) - \Pi\omega(k)$, where $x(k)$ and $\omega(k)$ represent

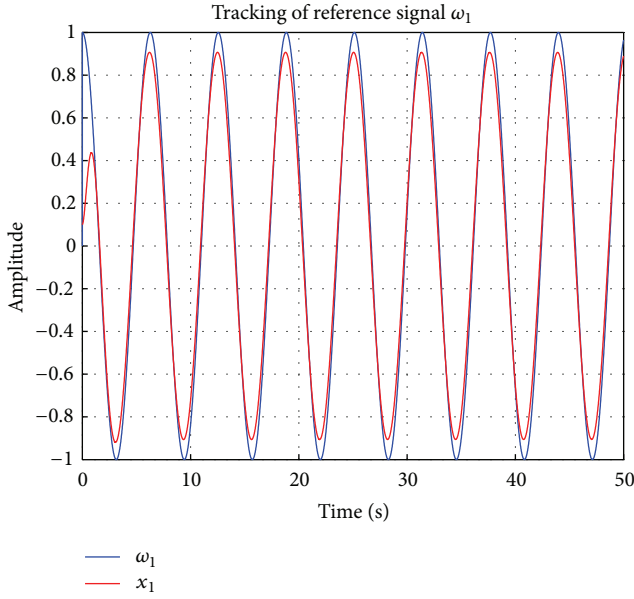


FIGURE 4: x_1 versus ω_1 when membership functions of the plant are used to fuzzily combine the local regulators.

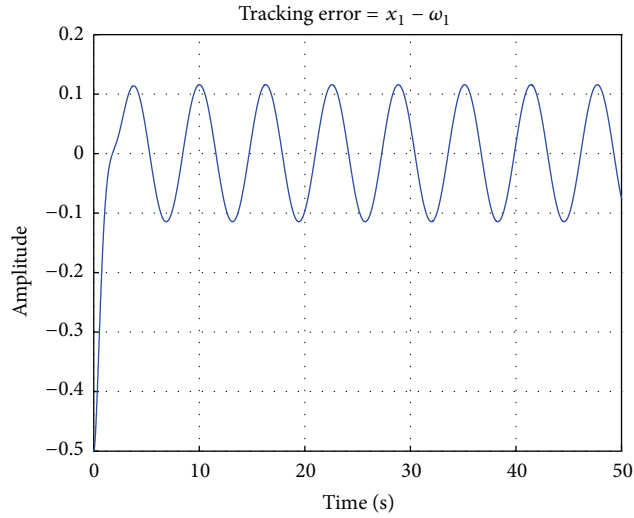


FIGURE 5: Tracking error when the membership functions of the plant are used to fuzzily combine the local regulators.

the k th training input-output pair for the fuzzy plant and the exosystem, respectively.

The “steepest descent” approach must reduce the error on the direction of the parameters of the controller. Therefore, (34) need to be adapted such that they consider the evolution of the states of the plant and exosystem (see Figure 6).

In order to obtain the training pairs, the plant is discretized using the approximation $\dot{x} \approx (x_k - x_{k-1})/T$. Therefore, the following discrete plant can be readily obtained:

$$x_k = \begin{bmatrix} (h_1 A_1 + h_2 A_2) x_{k-1} + (h_1 B_1 + h_2 B_2) \end{bmatrix}$$

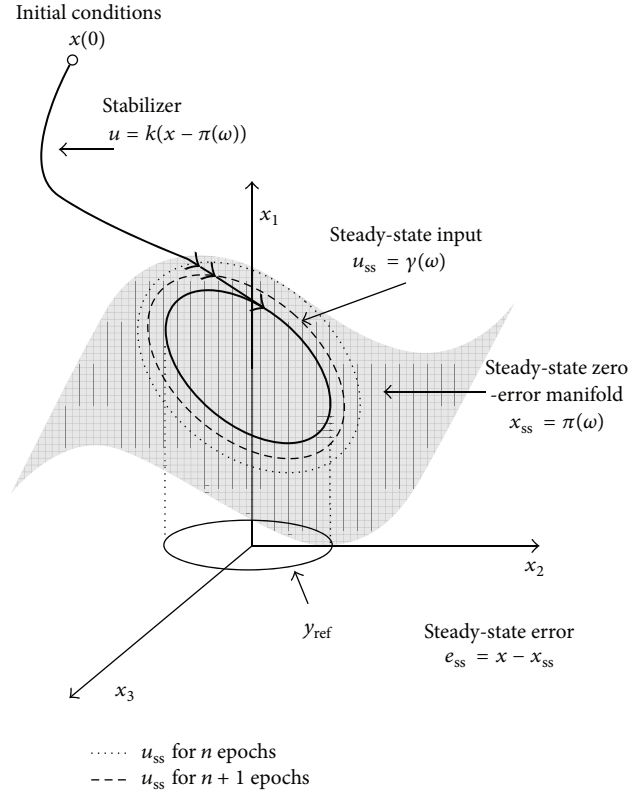


FIGURE 6: Scheme of the iterative regulation.

$$\begin{aligned} & \times \begin{bmatrix} h_1 K_1 x_{k-1} + h_2 K_2 x_{k-1} - h_1 K_1 \Pi \omega_k \\ -h_2 K_2 \Pi \omega_k + \mu_1 \Gamma_1 \omega_k + \mu_2 \Gamma_2 \omega_k \end{bmatrix} T \\ & + x_{k-1}. \end{aligned} \quad (35)$$

And the mean squared error is

$$e_{ss} = x - \Pi \omega \longrightarrow e_{ss}(k) = x_k - \Pi \omega_k \implies E_{ss} = \frac{1}{2} (x_k - \Pi \omega_k)^2. \quad (36)$$

The current value of state x_k is an explicit function of μ_1 and μ_2 , and this implies that x_k depends on parameters a , b , and c . Therefore, the steepest descent method should minimize the error function with respect to independent variables [30–33] (see Figure 7), that is,

$$\begin{aligned} \frac{\partial E}{\partial a_1} &= (x_k - \Pi \omega_k)^T [(h_1 B_1 + h_2 B_2) (\Gamma_1 - \Gamma_2) \omega_k] T \frac{\partial \mu_1}{\partial a_1}, \\ \frac{\partial E}{\partial b_1} &= (x_k - \Pi \omega_k)^T [(h_1 B_1 + h_2 B_2) (\Gamma_1 - \Gamma_2) \omega_k] T \frac{\partial \mu_1}{\partial b_1}, \\ \frac{\partial E}{\partial c_1} &= (x_k - \Pi \omega_k)^T [(h_1 B_1 + h_2 B_2) (\Gamma_1 - \Gamma_2) \omega_k] T \frac{\partial \mu_1}{\partial c_1}. \end{aligned} \quad (37)$$

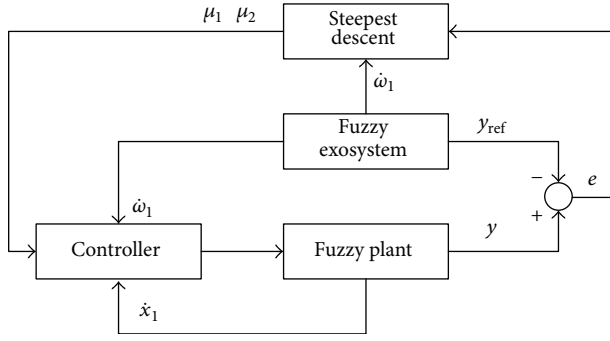


FIGURE 7: Training of the fuzzy controller.

Recalling that partial derivatives for a generalized bell (gbell) membership function are [34]

$$\frac{\partial \mu_1}{\partial a_1} = \frac{2b}{a} \mu_1 (1 - \mu_1),$$

$$\frac{\partial \mu_1}{\partial b_1} = \begin{cases} -\ln \left| \frac{\omega_{k-1} - c}{a} \right| \mu_1 (1 - \mu_1), & \text{if } \omega_{k-1} \neq c, \\ 0, & \text{if } \omega_{k-1} = c, \end{cases} \quad (38)$$

$$\frac{\partial \mu_1}{\partial c_1} = \begin{cases} \frac{2b}{\omega_{k-1} - c} \mu_1 (1 - \mu_1), & \text{if } \omega_{k-1} \neq c, \\ 0, & \text{if } \omega_{k-1} = c. \end{cases}$$

The parameters that adjust the membership functions μ_i are updated by the steepest descent method according to the following training law in the antecedent part:

$$a_1(k+1) = a_1(k) - \eta_a (x_k - \Pi \omega_k)^T \times [(h_1 B_1 + h_2 B_2) (\Gamma_1 - \Gamma_2) \omega_k] T \frac{\partial \mu_1}{\partial a_1},$$

$$b_1(k+1) = b_1(k) - \eta_b (x_k - \Pi \omega_k)^T \times [(h_1 B_1 + h_2 B_2) (\Gamma_1 - \Gamma_2) \omega_k] T \frac{\partial \mu_1}{\partial b_1}, \quad (39)$$

$$c_1(k+1) = c_1(k) - \eta_c (x_k - \Pi \omega_k)^T \times [(h_1 B_1 + h_2 B_2) (\Gamma_1 - \Gamma_2) \omega_k] T \frac{\partial \mu_1}{\partial c_1}.$$

The simulation results are shown in Figure 8, where state x_1 tracks to ω_1 . From Figure 9, it is observed that the error is minimized considerably in comparison with the linear controller (28) (Figure 5). When the regulator is fully trained, the reference tracking is achieved in a very acceptable way, which is depicted in Figures 10 and 11.

But according to [1], if the regulation problem has a solution, then it is unique. In fuzzy regulation, this means that the membership functions of the fuzzy regulator cannot have arbitrary shapes, and in many cases, their shape is unknown. Obviously, the steepest descent approach is not applicable if the shape of the membership functions is not given a priori.

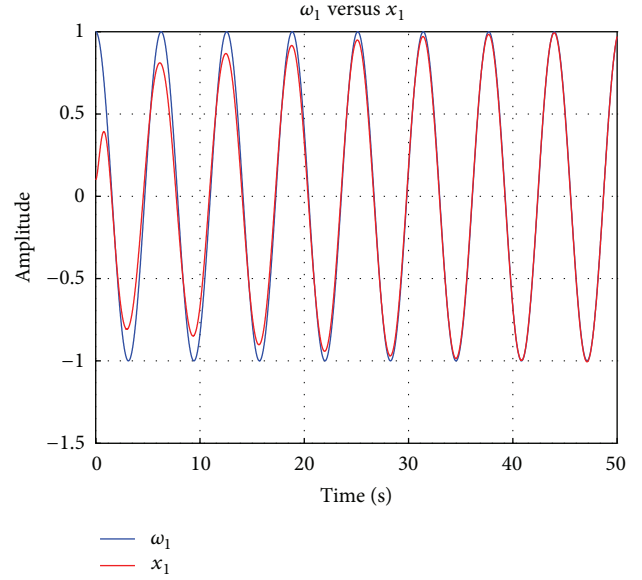
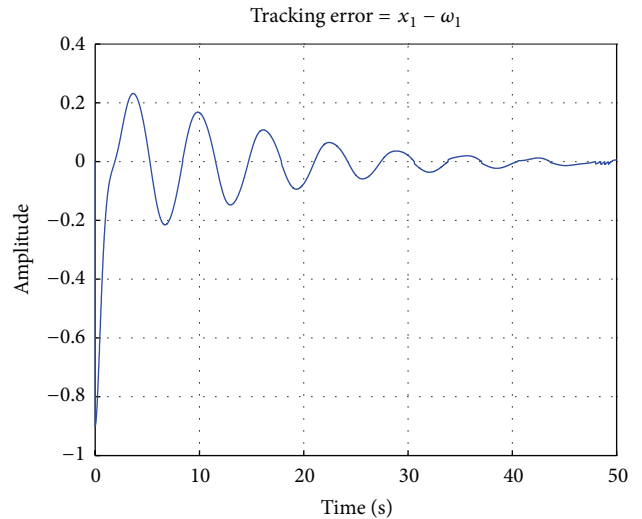
FIGURE 8: x_1 versus ω_1 during the learning scheme.

FIGURE 9: Tracking error during the learning scheme.

In the following, this disadvantage will be overcome by considering ANFIS.

3.2. ANFIS in Output Fuzzy Regulation. Considering (26), when h_1 is equal to 1, the dynamics of the overall fuzzy system are described by the first local linear subsystem and the exact output regulator is obtained from Γ_1 . In the same way, if h_2 is equal to 1, the exact output regulator is obtained directly from Γ_2 . With this in mind, the next fuzzy implications are obtained:

$$\text{IF } h_1 \text{ is 1, THEN } \mu_1 = 1 \rightarrow u(t) = h_1 K_1 [x(t) - \Pi \omega(t)] + \mu_1 \Gamma_1 \omega(t),$$

$$\text{IF } h_2 \text{ is 1, THEN } \mu_2 = 1 \rightarrow u(t) = h_2 K_2 [x(t) - \Pi \omega(t)] + \mu_2 \Gamma_2 \omega(t).$$

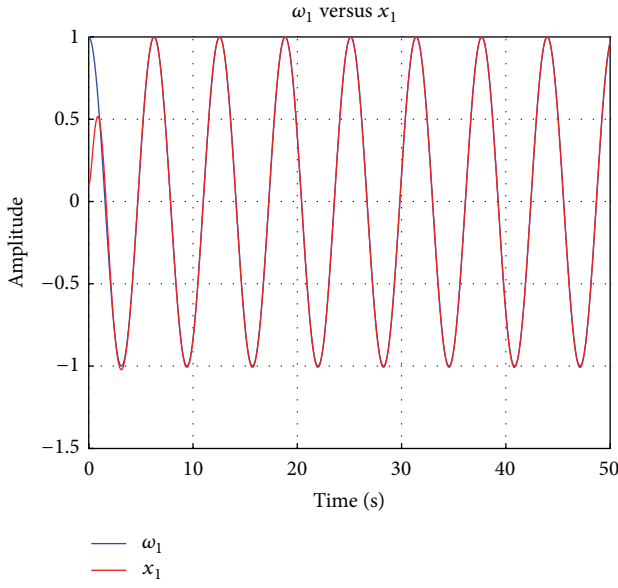
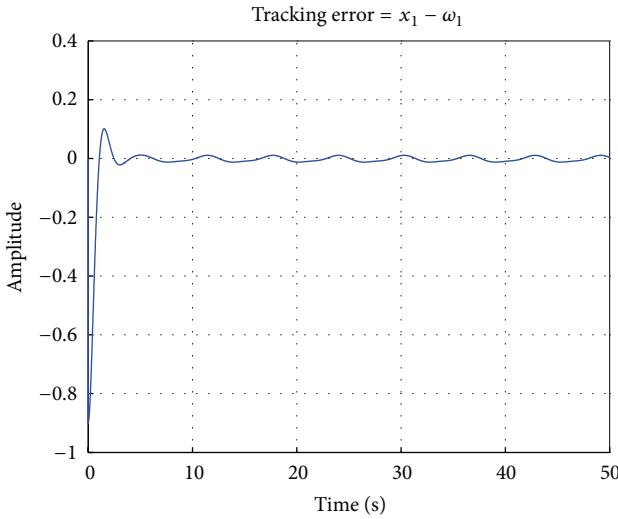

 FIGURE 10: x_1 versus ω_1 when the controller has been fully trained.


FIGURE 11: Tracking error when the controller has been fully trained.

A set of fuzzy rules can be established, taking into account that the regulation problem is exactly solved by local linear regulator, that is,

$$\text{If } h_1 \text{ is 1 then } \mu_1 = 1 \Rightarrow u_{\text{reg}}(t) = \Gamma_1,$$

$$\text{If } h_2 \text{ is 1 then } \mu_2 = 1 \Rightarrow u_{\text{reg}}(t) = \Gamma_2.$$

In the fuzzy rules given previously, it is supposed that μ_i is a function of ω_1 , while μ_2 can be obtained from $\mu_2 = 1 - \mu_1$. Then, ω_1 is chosen as the input variable in ANFIS. Figure 12 shows the ANFIS architecture with one input.

Layer 1. For this example, gbell membership functions are chosen, and 8 nodes are distributed uniformly (see Figure 13) according to [35]

$$\begin{aligned} a_1 &= 4, & b_1 &= 2, & c_1 &= -30, \\ a_2 &= 4, & b_2 &= 2, & c_2 &= -21, \\ a_3 &= 4, & b_3 &= 2, & c_3 &= -12, \\ a_4 &= 4, & b_4 &= 2, & c_4 &= -3, \\ a_5 &= 4, & b_5 &= 2, & c_5 &= 3, \\ a_6 &= 4, & b_6 &= 2, & c_6 &= 12, \\ a_7 &= 4, & b_7 &= 2, & c_7 &= 21, \\ a_8 &= 4, & b_8 &= 2, & c_8 &= 30, \end{aligned}$$

$$\mu_{A_j}(h_1) = \frac{1}{1 + ((h_1 - c_j)/a_j)^2 b_j} \quad j = 1, 2, \dots, 8; \quad 0 \leq h_1 \leq 1. \quad (40)$$

Layer 2. The weight of each function corresponds to its own membership function because only one input variable is considered:

$$\begin{aligned} w_1 &= \mu_{A1}, & w_2 &= \mu_{A2}, \\ w_3 &= \mu_{A3}, & w_4 &= \mu_{A4}, \\ w_5 &= \mu_{A5}, & w_6 &= \mu_{A6}, \\ w_7 &= \mu_{A7}, & w_8 &= \mu_{A8}. \end{aligned} \quad (41)$$

Layer 3. Weight normalizing:

$$\bar{w}_i = \frac{w_i}{w_1 + w_2 + w_3 + w_4 + w_5 + w_6 + w_7 + w_8}, \quad i = 1, 2, \dots, 8. \quad (42)$$

Layer 4. Takagi-Sugeno coefficients:

$$f_i = \bar{w}_i (p_i h_1 + r_i) \quad i = 1, 2, \dots, 8. \quad (43)$$

Layer 5.

$$f = f_1 + f_2 + f_3 + f_4 + f_5 + f_6 + f_7 + f_8. \quad (44)$$

3.3. Adjustment of Input Layer Parameters. Thus, the goal is to generate the adequate membership functions for the fuzzy regulator by means of ANFIS regulators, but these values depend on the reference signal. The proposed scheme is depicted in Figure 14.

As discussed in Section 3.1, the steepest descent method adjusts parameters a_i , b_i , c_i . Then, the chain rule used for the error function $e_{\text{ce}} = x_1 - \omega_1$ depends on parameters a_i , b_i , c_i , that is,

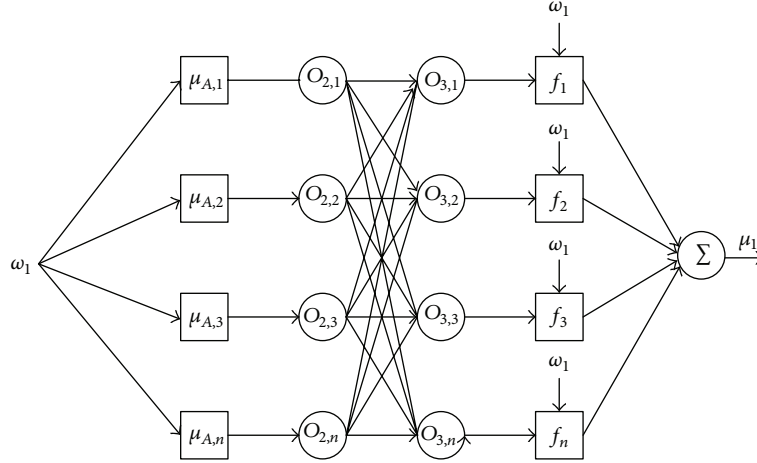


FIGURE 12: ANFIS architecture for one input variable.

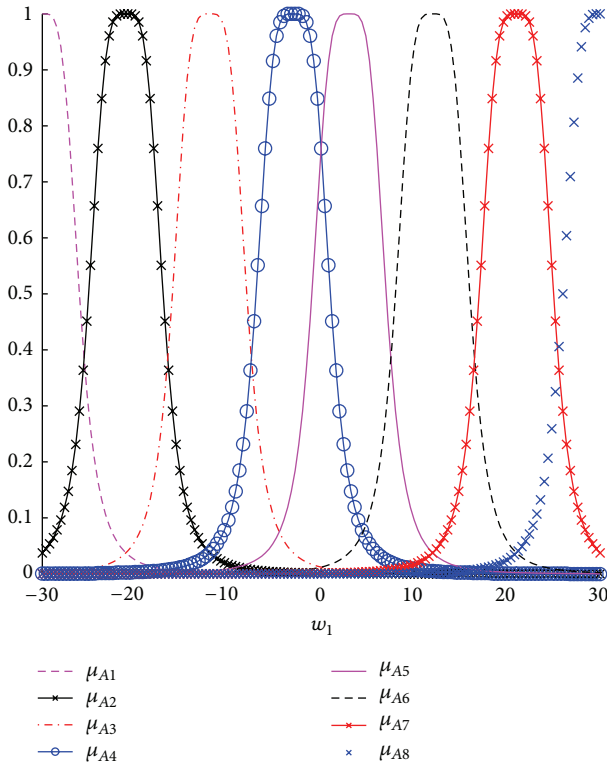


FIGURE 13: Distribution of the membership functions for input layer.

$$\begin{aligned}\frac{\partial E}{\partial a_i} &= \frac{\partial E}{\partial x_1} \frac{\partial x_1}{\partial \mu_1} \frac{\partial \mu_1}{\partial f_i} \frac{\partial f_i}{\partial w_i} \frac{\partial w_i}{\partial \mu_{A_i}} \frac{\partial \mu_{A_i}}{\partial a_i}, \\ \frac{\partial E}{\partial b_i} &= \frac{\partial E}{\partial x_1} \frac{\partial x_1}{\partial \mu_1} \frac{\partial \mu_1}{\partial f_i} \frac{\partial f_i}{\partial w_i} \frac{\partial w_i}{\partial \mu_{A_i}} \frac{\partial \mu_{A_i}}{\partial b_i}, \\ \frac{\partial E}{\partial c_i} &= \frac{\partial E}{\partial x_1} \frac{\partial x_1}{\partial \mu_1} \frac{\partial \mu_1}{\partial f_i} \frac{\partial f_i}{\partial w_i} \frac{\partial w_i}{\partial \mu_{A_i}} \frac{\partial \mu_{A_i}}{\partial c_i}, \\ \frac{\partial E}{\partial x_1} &= (x_1 - \omega_1),\end{aligned}$$

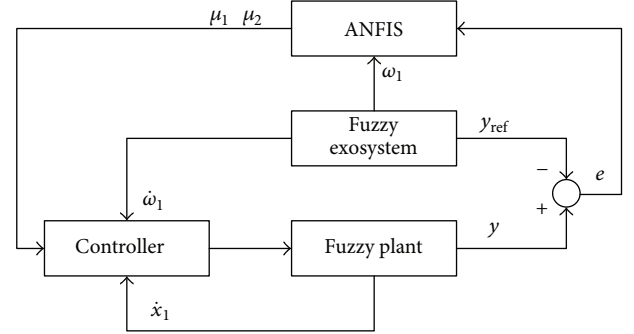


FIGURE 14: Rule tuning including ANFIS.

$$\frac{\partial x_1}{\partial \mu_1} = [(\Gamma_1 - \Gamma_2) h_1] T,$$

$$\frac{\partial \mu_1}{\partial F_i} = 1,$$

$$\frac{\partial F_i}{\partial w_i} = \frac{w_i}{\sum w_i} (p_i x + r_i) = \frac{p_i x + r_i - f}{\sum w_i},$$

$$\frac{\partial w_i}{\partial \mu_{A_i}} = 1.$$

(45)

The terms $\partial \mu_{A_i} / \partial a_1$, $\partial \mu_{A_i} / \partial b_1$, and $\partial \mu_{A_i} / \partial c_1$ are calculated from (38), while

$$\begin{aligned}\frac{\partial E}{\partial a_i} &= (x_1 - \omega_1) [(\Gamma_1 - \Gamma_2) h_1] T \left(\frac{p_i h_1 + r_i - f}{\sum w_i} \right) \left(\frac{\partial \mu_{A_i}}{\partial a_i} \right), \\ \frac{\partial E}{\partial b_i} &= (x_1 - \omega_1) [(\Gamma_1 - \Gamma_2) h_1] T \left(\frac{p_i h_1 + r_i - f}{\sum w_i} \right) \left(\frac{\partial \mu_{A_i}}{\partial b_i} \right), \\ \frac{\partial E}{\partial c_i} &= (x_1 - \omega_1) [(\Gamma_1 - \Gamma_2) h_1] T \left(\frac{p_i h_1 + r_i - f}{\sum w_i} \right) \left(\frac{\partial \mu_{A_i}}{\partial c_i} \right).\end{aligned}$$

(46)

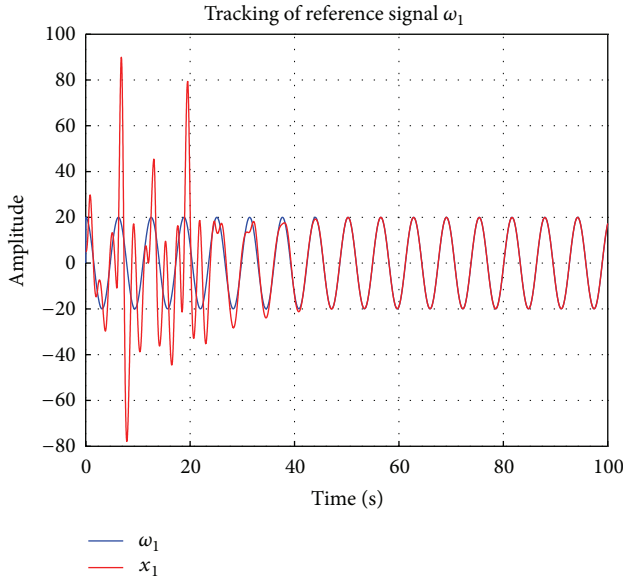


FIGURE 15: x_1 versus ω_1 when ANFIS is included in the fuzzy controller.

Each one of the n functions that are proposed in the input layer of ANFIS is fitted by using the previous expressions.

3.4. ANFIS Learning Using Levenberg-Marquardt Approach. The Levenberg-Marquardt (L-M) method is an effective nonlinear least-squares approach to nonlinear regression problems [36]:

$$\Theta_{k+1} = \Theta_k - (J^T J + \lambda I)^{-1} g_h, \quad (47)$$

where $g_h \equiv (1/2)g$; λ is a nonnegative number.

It has been proved that the L-M method reduces the root mean squared error further than hybrid learning method. In this work, the mean square error is used, and the gradient takes into account T - S parameters of the fourth layer. Therefore, the following training rules can be obtained:

$$\begin{aligned} \frac{\partial E}{\partial p_i} &= \frac{\partial E}{\partial x_1} \frac{\partial x_1}{\partial \mu_1} \frac{\partial \mu_1}{\partial f_i} \frac{\partial f_i}{\partial p_i} \quad i = 1, \dots, n, \\ \frac{\partial E}{\partial r_i} &= \frac{\partial E}{\partial x_1} \frac{\partial x_1}{\partial \mu_1} \frac{\partial \mu_1}{\partial f_i} \frac{\partial f_i}{\partial r_i} \quad i = 1, \dots, n. \end{aligned} \quad (48)$$

4. Numerical Results after Applying ANFIS

From Figures 15 and 16, it can be observed that the fuzzy tracking is improved and that the tracking error is smaller than the one obtained through the steepest descent approach.

Figure 17 shows membership function μ_1 obtained from different methods: (1) analytically computed, (2) membership functions for fuzzy regulator are the same than ones of fuzzy plant ($\mu_1 = h_1$), and (3) ANFIS estimation. Note that the membership function generated by ANFIS tends to the real one (analytically computed). It is important to recall that the $\mu_1 + \mu_2 = 1$, and therefore the missing membership function, μ_2 , can be easily obtained.

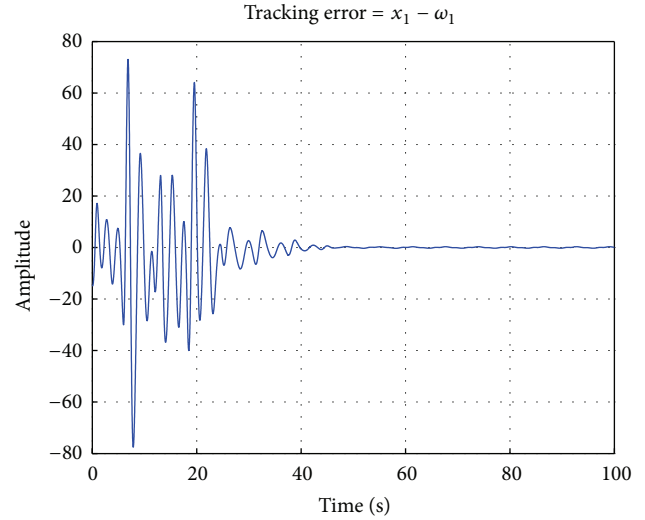


FIGURE 16: Tracking error when ANFIS is included in the fuzzy controller.

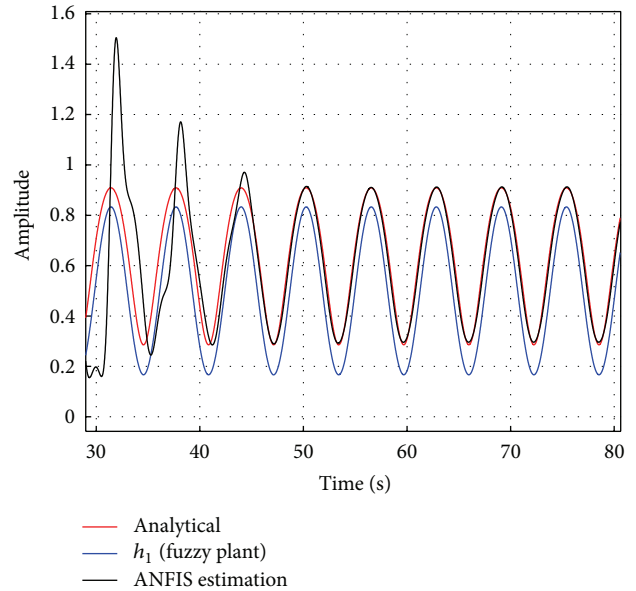


FIGURE 17: Membership function μ_1 obtained from different methods.

The control signal appears in Figure 18.

Case 1. In some applications, it is desired that external reference signal has variable frequency, and thus a new exosystem is proposed:

$$S = \begin{bmatrix} 0 & 3 \\ -3 & 0 \end{bmatrix}. \quad (49)$$

The local steady-state manifolds can be obtained from

$$\begin{aligned} \Pi_1 = \Pi_2 &= \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix}, \\ \Gamma_1 &= \begin{bmatrix} -\frac{11}{2} & 0 \end{bmatrix}, \quad \Gamma_2 = \begin{bmatrix} -13 & 0 \end{bmatrix}. \end{aligned} \quad (50)$$

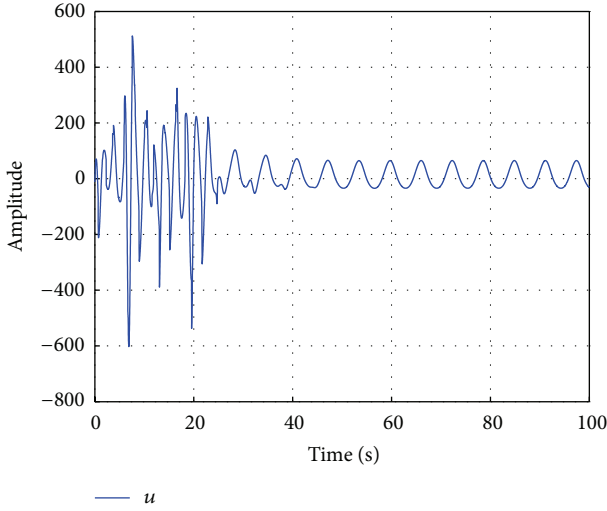


FIGURE 18: Control signal when ANFIS is included in the fuzzy controller.

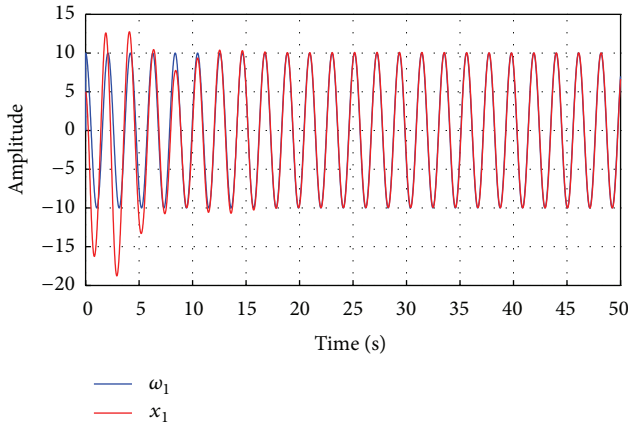


FIGURE 19: x_1 versus ω_1 for Case 1.

Figure 19 shows the system behavior during the training stage. As expected, the fuzzy plant tracks asymptotically the exosystem.

Case 2. A practical example is the inverted pendulum (Figure 20), whose nonlinear model is given by [29]

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= \frac{g \sin(x_1(t)) - amlx_2^2(t) \sin(2x_1(t)) / 2 - a \cos(x_1(t)) u(t)}{4l/3 - aml \cos^2(x_1(t))}, \end{aligned} \quad (51)$$

where the mass of the cart is $M = 8$ kg, the mass of the pendulum is $m = 2$ kg, the length of the pendulum is $2l = 1$ m, the gravity is $g = 9.81$ m/s², and $a = 1/(m + M)$.

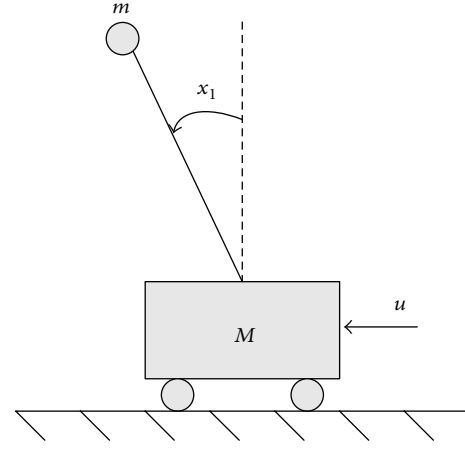


FIGURE 20: Inverted pendulum on a cart.

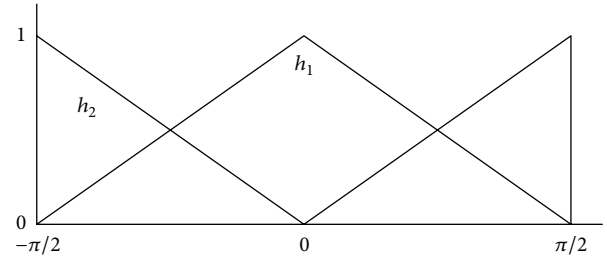


FIGURE 21: Membership functions for the two-rule model [29].

This nonlinear model can be approximated by a TS fuzzy model of two rules with

$$\begin{aligned} A_1 &= \begin{bmatrix} 0 & 1 \\ 17.31176 & 0 \end{bmatrix}, & A_2 &= \begin{bmatrix} 0 & 1 \\ 9.36957 & 0 \end{bmatrix}, \\ B_1 &= \begin{bmatrix} 0 \\ -0.17647 \end{bmatrix}, & B_2 &= \begin{bmatrix} 0 \\ -0.00523 \end{bmatrix}, \end{aligned} \quad (52)$$

$$S = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \quad K_1 = [-120.6667 \quad -22.6667],$$

$$K_2 = [-2551.6 \quad -764],$$

and membership functions as those depicted in Figure 21.

The regulation condition is not satisfied when the membership functions of the plant are used to combine the local regulators because matrices B_1 and B_2 are different (Figure 22).

On the other hand, from Figures 23 and 24, it can be observed that the tracking is improved, that is, the tracking error is considerably reduced when ANFIS is used to obtain the adequate membership.

5. Conclusions

In this paper, it has been shown that the nonlinearity of the plant is not necessarily the same nonlinearity needed by the regulator to ensure reference tracking. In other words, the membership functions for fuzzy regulator and for the fuzzy

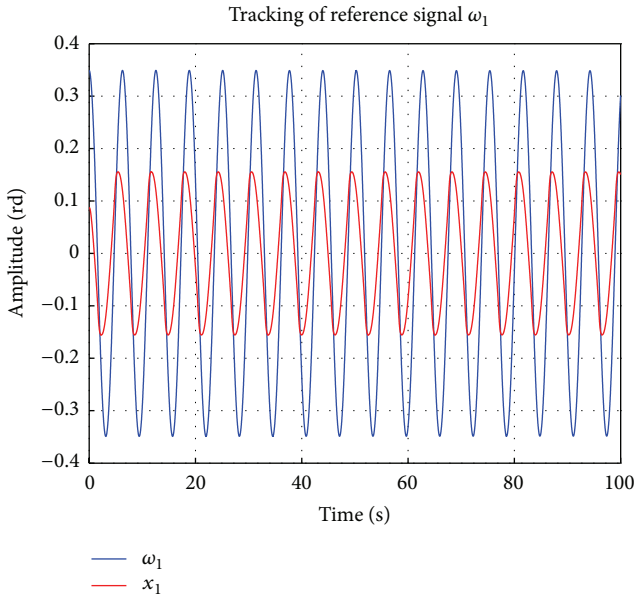


FIGURE 22: x_1 versus ω_1 when the regulator uses the same membership functions of the plant.

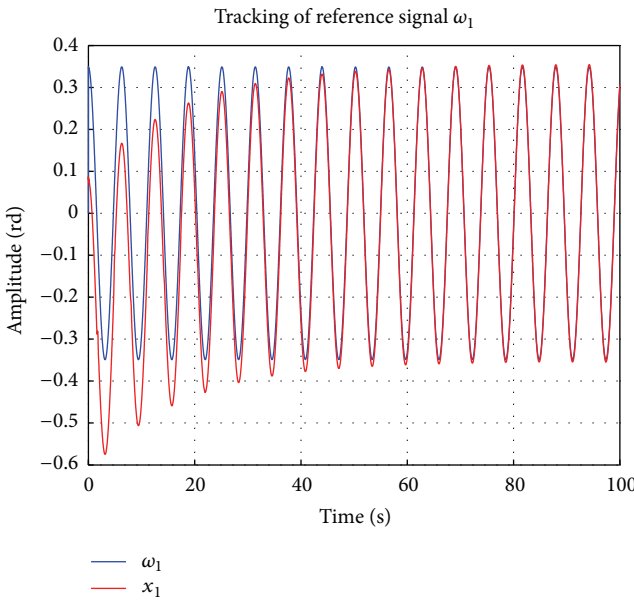


FIGURE 23: x_1 versus ω_1 when ANFIS is included in the fuzzy controller.

plant are not the same, in general. In that sense, the steepest descent and ANFIS approaches have been used to estimate the adequate membership functions of fuzzy regulator in order to reduce the tracking error when the overall fuzzy controller is obtained from the fuzzy summation of local controllers.

It is also important to mention that although the Steepest Descent can be used to approximate the fuzzy membership functions, the approach can be trapped in a local minimum or local maximum. On the other hand, ANFIS due the hybrid training avoids this problem.

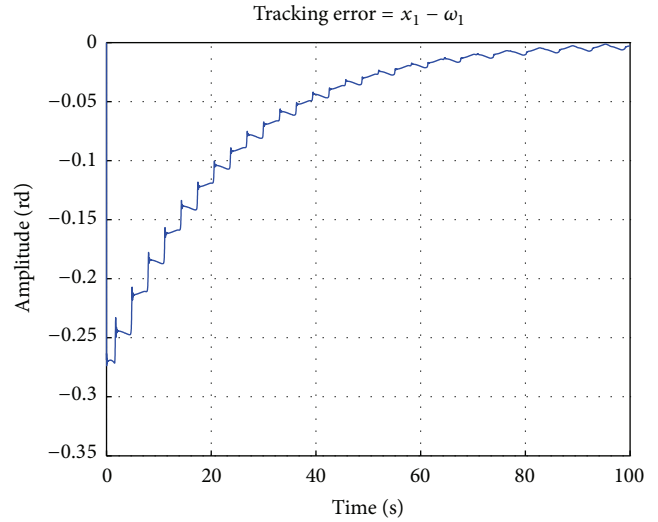


FIGURE 24: Tracking error when ANFIS is included in the fuzzy controller.

On the other hand, the use of ANFIS avoids the restriction of proposing a priori form of the membership functions, which is another disadvantage of the Steepest Descent approach.

Finally, it can be concluded that a suitable approximation of the nonlinearity of the fuzzy regulator is obtained when ANFIS is used as a tuning tool during the designing of fuzzy regulators.

Acknowledgments

This work was supported by CONACYT through scholarship SNI, by IPN through scholarships PIFI, COFAA, and EDI, and research Project 20130760.

References

- [1] A. Isidori and C. I. Byrnes, "Output regulation of nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 35, no. 2, pp. 131–140, 1990.
- [2] B. A. Francis and W. M. Wonham, "The internal model principle for linear multivariable regulators," *Applied Mathematics & Optimization*, vol. 2, no. 2, pp. 170–194, 1975.
- [3] B. A. Francis, "The linear multivariable regulator problem," *SIAM Journal on Control and Optimization*, vol. 15, no. 3, pp. 486–505, 1977.
- [4] J. A. Meda-Campaña, J. C. Gómez-Mancilla, and B. Castillo-Toledo, "Exact output regulation for nonlinear systems described by Takagi-Sugeno fuzzy models," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 2, pp. 235–247, 2012.
- [5] L. A. Zadeh, "Soft computing and fuzzy logic," *IEEE Software*, vol. 11, no. 6, pp. 48–56, 1994.
- [6] J. Henriques, B. Castillo, A. Titli, P. Gil, and A. Dourado, "Learning and output regulation with recurrent neural network," in *Proceedings of the 4th Portuguese Conference on Automatic Control*, pp. 324–329, 2000.
- [7] J. Henriques, P. Gil, A. Cardoso, P. Carvalho, and A. Dourado, "Adaptive neural output regulation control of a solar power

- plant," *Control Engineering Practice*, vol. 18, no. 10, pp. 1183–1196, 2010.
- [8] Y. Zhang and J. Wang, "Recurrent neural networks for nonlinear output regulation," *Automatica*, vol. 37, no. 8, pp. 1161–1173, 2001.
- [9] B. Castillo-Toledo, J. A. Meda-Campaña, and A. Titli, "A fuzzy output regulator for Takagi-Sugeno fuzzy models," in *Proceedings of the IEEE International Symposium on Intelligent Control*, pp. 310–315, Houston, Tex, USA, October 2003.
- [10] R. Fullér, *Introduction to Neuro-Fuzzy Systems*, Advances in Soft Computing, Springer, New York, NY, USA, 2000.
- [11] P. P. Bonissone, Y. U. O. Chen, K. Goebel, and P. S. Khedkar, "Hybrid soft computing systems: industrial and commercial applications," *Proceedings of the IEEE*, vol. 87, no. 9, pp. 1641–1667, 1999.
- [12] A. Saad, "An overview of hybrid soft computing techniques for classifier design and feature selection," in *Proceedings of the 8th International Conference on Hybrid Intelligent Systems (HIS '08)*, pp. 579–583, Barcelona, Spain, September 2008.
- [13] H. Nomura, I. Hayashi, and N. Wakami, "A learning method of fuzzy inference rules by descent method," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, pp. 203–210, March 1992.
- [14] D. Nauck and R. Kruse, "A fuzzy neural network learning fuzzy control rules and membership functions by fuzzy error backpropagation," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 1, pp. 1022–1027, 1993.
- [15] W. Textor, S. Wessel, and K. U. Hffgen, "Learning fuzzy rules from artificial neural nets," in *Proceedings of the Computer Systems and Software Engineering*, pp. 121–126, 1992.
- [16] T. Miyoshi, S. Tano, Y. Kato, and T. Arnouuld, "Operator tuning in fuzzy production rules using neural networks," in *Proceedings of the 2nd IEEE International Conference on Fuzzy Systems*, vol. 1, pp. 641–646, San Francisco, Calif, USA, 1993.
- [17] K. Nishimori, S. Hirakawa, K. Fujimura et al., "Automatic tuning method of membership functions in simulation of driving control of a model car," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 3, pp. 2937–2940, October 1993.
- [18] W. Li, "Optimization of a fuzzy controller using neural network," in *IEEE World Congress on Computational Intelligence, Proceedings of the 3rd IEEE Conference on Fuzzy Systems*, vol. 1, pp. 223–227, Orlando, Fla, USA, June 1994.
- [19] S. K. Halgamuge and M. Glesner, "Neural networks in designing fuzzy systems for real world applications," *Fuzzy Sets and Systems*, vol. 65, no. 1, pp. 1–12, 1994.
- [20] J. Sitte and S. Geva, "Design of fuzzy controllers with local response neurons," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 2021–2026, October 1994.
- [21] D. van Cleave and K. S. Rattan, "Tuning of fuzzy logic controller using neural network," in *Proceedings of the IEEE National Aerospace and Electronics Conference (NAECON '00)*, pp. 305–312, October 2000.
- [22] C. K. Lin and S. D. Wang, "Robust self-tuning rotated fuzzy basis function controller for robot arms," *IEEE Proceedings on Control Theory and Applications*, vol. 144, no. 4, pp. 293–298, 1997.
- [23] F. J. Moreno-Velo, I. Baturone, R. Senhadji, and S. Sánchez-Solano, "Tuning complex fuzzy systems by supervised learning algorithms," in *Proceedings of the 12th IEEE International Conference on Fuzzy Systems (FUZZ '03)*, vol. 1, pp. 226–231, May 2003.
- [24] S. Chopra, R. Mitra, and V. Kumar, "Neural network tuned fuzzy controller for MIMO system," *International Journal of Electrical and Computer Engineering*, vol. 2, no. 5, pp. 371–378, 2007.
- [25] R. Jain, N. Sivakumaran, and T. K. Radhakrishnan, "Design of self tuning fuzzy controllers for nonlinear systems," *Expert Systems with Applications*, vol. 38, no. 4, pp. 4466–4476, 2011.
- [26] J. R. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.
- [27] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 15, no. 1, pp. 116–132, 1985.
- [28] J. A. Meda-Campaña, J. C. Gómez-Mancilla, and B. Castillo-Toledo, "On the exact output regulation for Takagi-Sugeno fuzzy systems," in *Proceedings of the 8th IEEE International Conference on Control and Automation (ICCA '10)*, pp. 417–422, Xiamen, China, June 2010.
- [29] K. Tanaka and H. Wang, *Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach*, John Wiley & Sons, New York, NY, USA, 2001.
- [30] E. J. Haug, J. S. Arora, and K. Matsui, "A steepest-descent method for optimization of mechanical systems," *Journal of Optimization Theory and Applications*, vol. 19, no. 3, pp. 401–424, 1976.
- [31] X. Wang, "Method of steepest descent and its applications," *IEEE Microwave and Wireless Components Letters*, vol. 12, pp. 24–26, 2008.
- [32] M. Bartholomew-Biggs, "The steepest descent method," in *Non-linear Optimization With Engineering Applications*, vol. 19, pp. 75–82, Springer, Boston, Mass, USA, 2008.
- [33] G. Teschke and C. Borries, "Accelerated projected steepest descent method for nonlinear inverse problems with sparsity constraints," *Inverse Problems*, vol. 26, no. 2, Article ID 025007, 2010.
- [34] R. Jang, *Neuro-Fuzzy and Soft Computing*, Prentice Hall, New York, NY, USA, 1997.
- [35] M. A. Denai, F. Palis, and A. Zeghib, "ANFIS based modelling and control of non-linear systems: a tutorial," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '04)*, pp. 3433–3438, October 2004.
- [36] J. R. Jang and E. Mizutani, "Levenberg-Marquardt method for ANFIS learning," in *Proceedings of the Biennial Conference of the North American Fuzzy Information Processing Society—NAFIPS*, vol. 1, pp. 87–91, June 1996.



Hindawi

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

