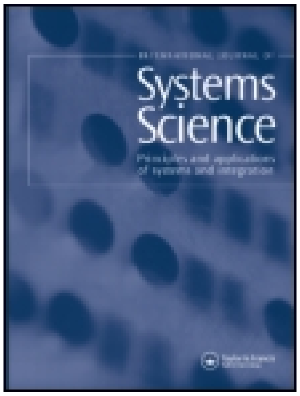

International Journal of Systems Science
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tsys20

Robust adaptive self-structuring fuzzy control design for nonaffine, nonlinear systems

Pin-Cheng Chen [a] , Chi-Hsu Wang [b] & Tsu-Tian Lee [a]
[a] Department of Electrical Engineering , National Taipei University of Technology , Sec. 3, Chung-hsiao E. Rd, Taipei 10608, Taiwan
[b] Department of Electrical and Control Engineering , National Chiao Tung University , Hsinchu 300, Taiwan
Published online: 05 Nov 2010.

International Journal of Systems Science
Vol. 42, No. 1, January 2011, 149–169

Taylor & Francis
Taylor & Francis Group

Robust adaptive self-structuring fuzzy control design for nonaffine, nonlinear systems

Pin-Cheng Chen[a]*, Chi-Hsu Wang[b] and Tsu-Tian Lee[a]

[a]*Department of Electrical Engineering, National Taipei University of Technology, Sec. 3, Chung-hsiao E. Rd, Taipei 10608, Taiwan;* [b]*Department of Electrical and Control Engineering, National Chiao Tung University, Hsinchu 300, Taiwan*

(*Received 7 September 2007; final version received 1 November 2009*)

In this article, a robust adaptive self-structuring fuzzy control (RASFC) scheme for the uncertain or ill-defined nonlinear, nonaffine systems is proposed. The RASFC scheme is composed of a robust adaptive controller and a self-structuring fuzzy controller. In the self-structuring fuzzy controller design, a novel self-structuring fuzzy system (SFS) is used to approximate the unknown plant nonlinearity, and the SFS can automatically grow and prune fuzzy rules to realise a compact fuzzy rule base. The robust adaptive controller is designed to achieve an L_2 tracking performance to stabilise the closed-loop system. This L_2 tracking performance can provide a clear expression of tracking error in terms of the sum of lumped uncertainty and external disturbance, which has not been shown in previous works. Finally, five examples are presented to show that the proposed RASFC scheme can achieve favourable tracking performance, yet heavy computational burden is relieved.

Keywords: adaptive control; robust control; fuzzy system; structure adaptation

1. Introduction

A fuzzy system (FS), which adopts human experience and human decision-making behaviour, has been widely recognised as a powerful tool in industrial control, commercial prediction, image processing applications, etc. (Terano, Asai, and Sugeno 1992; Castro 1995; Gil-Lafuente 2005). To build an FS, two different phases are to be carried out. The first is the structuring phase, which is used to construct the structure of the FS, and the second is the parameter phase, which is used to determine the parameters of the FS. Constructing the structure of the FS is mainly to determine the optimal partition of fuzzy sets and the minimum number of fuzzy rules to achieve favourable performance. The adjustments of the parameters involve the tuning of the consequences of the fuzzy rules, the centres, widths, slopes of membership functions, etc. Traditionally, these two phases are performed by human experts or experienced operators. However, consulting experts may be difficult and expert knowledge may be either unavailable or not helpful enough to achieve favourable performance. Having achieved many practical successes, fuzzy control (FC) using an FS has still not been viewed as rigorous because it lacks a systematic design procedure to determine proper membership functions with fuzzy rules, and the way to guarantee the global stability. Adaptive fuzzy control (AFC) has been extensively studied to tackle this problem (Wang 1994; Lin and Hsu 2002; Li and Tong 2003; Chatterjee and Watanabe 2005; Hsu and Lin 2005; Labiod, Boucherit, and Guerra 2005). The AFC can approximate the unknown system dynamics or ideal controller through learning in the Lyapunov sense, and thus the global stability can be guaranteed.

Although the control performances Wang (1994), Lin and Hsu (2002), Li and Tong (2003), Chatterjee and Watanabe (2005), Hsu and Lin (2005), Labiod et al. (2005) are acceptable, the structures of the FSs need to be predefined by a time-consuming trial-and-error process. Generally speaking, a more favourable performance requires more fuzzy rules, but this may lead to heavy computational burden. On the contrary, an FS with small fuzzy rule base may result in a poor approximation.

To solve the problem of structure determination, many researchers have focused their efforts on the self-structuring or self-evolving FSs, which have both parameter and structure adaptations. Some valuable results are obtained (Shann and Fu 1995; Pal and Pal 1999; Angelov and Filev 2004; Lin and Lin 2004; Meng and Chang 2004; Lin and Chen 2005; Juang and Tsao 2008). In Lin and Chen (2005), the structure learning phase aims at minimising the number of rules generated and the number of fuzzy sets in the universe of discourse. A structure learning algorithm is proposed based on fuzzy similarity measure and

*Corresponding author. Email: pcchen@ntut.edu.tw

ISSN 0020–7721 print/ISSN 1464–5319 online

DOI: 10.1080/00207720903494635
http://www.informaworld.com

fuzzy rules can be created from the training data. In Meng and Chang (2004), the structure identification is accomplished automatically based only on Q-learning, which is the most important category of reinforcement learning algorithm. The basic fuzzy rules are used as starting points to reduce the number of iterations used to find an optimal fuzzy controller. In Lin and Lin (2004), the firing strength of a rule is used as the degree measure to judge whether or not to simultaneously generate a new membership function for every input variable (or equivalently, to generate a new rule). Then, if the newly generated membership of the first input variable fails to pass the similarity checking, all new membership functions are abandoned. In Shann and Fu (1995), parameter and structure learning are performed sequentially for the proposed fuzzy neural network. That is, the fuzzy neural network is initially constructed to contain all possible fuzzy rules, and then after the parameter training is done, a pruning process is performed to delete redundant rules for obtaining a concise fuzzy rule base. Note that the initially constructed rule base contains incompatible rules, i.e. the rules with the same antecedent but different consequents. The rule pruning strategy is that if the centroid of a set of incompatible rules is in the support of a consequent (an output fuzzy set), the corresponding fuzzy rule is retained and all other incompatible rules are pruned. In Pal and Pal (1999), the authors modified the fuzzy neural network proposed in (Shann and Fu 1995) and proposed a rule pruning scheme that always produces a rule set without incompatible rules. In Juang and Tsao (2008), a self-evolving interval type-2 fuzzy neural network (SEIT2FNN) is proposed. An online clustering method is used to generate Takagi–Sugeno–Kang type fuzzy rules that flexibly partition the input space. A fuzzy set reduction method is proposed to avoid highly overlapping fuzzy sets. A gradient descent algorithm and rule-ordered Kalman filter algorithm are used to tune the antecedent and consequent parts of the fuzzy rules, respectively. In Angelov and Filev (2004), an evolving Takagi–Sugino (ETS) model is proposed. The potential of a new data point is used for the rule generation criterion and shown to be more reasonable than the criterion based on the distance to a rule centre because the spatial information and history are not ignored. Both SEIT2FNN and ETS can start learning without *a priori* information and only one data sample. However, although some achievements have been made in these works, there are still some problems that need to be solved. In Lin and Chen (2005), the performance of the proposed neural FS is acceptable, but the back propagation learning algorithm cannot guarantee the global stability. In Meng and Chang (2004), during the training process, prior knowledge of fuzzy rules is needed to keep safe operation of the controlled system with fast convergence speed of parameters. In Lin and Lin (2004), the simplified similarity checking to reduce the complexity of the algorithm may weaken the power of the checking itself. In (Shann and Fu 1995), because the connection weights of the network are unrestricted in sign, incompatible rules may be retained even when the rule pruning process is performed. This is contradictory to the basic design philosophy of FSs. Besides, the proposed sequential learning scheme is suitable for offline instead of online operation. In Pal and Pal (1999), although the fuzzy neural network in Shann and Fu (1995) is modified to guarantee a compatible rule base, the searching space for the connection weights is restricted to $R+$. This may harm the capability of the proposed network to lower the value of residual square error. In Juang and Tsao (2008), the consequent part parameters are determined by experience rather than by theoretical analysis. In addition, although the proposed fuzzy set reduction method to avoid the generation of highly overlapping fuzzy sets reduces the number of parameters, it does not remarkably release the computational burden. The common drawback in Shann and Fu (1995), Pal and Pal (1999), Angelov and Filev (2004), Lin and Lin (2004), Meng and Chang (2004), Lin and Chen (2005) and Juang and Tsao (2008) is that the structuring learning phase conducts either rule generation or rule reduction, instead of both.

Recently, control system design for nonlinear systems has attracted a lot of research interest. Many remarkable results have been obtained, including feedback linearisation (Isidori 1989), adaptive backstepping design (Krstic, Kanellakopoulos, and Kokotovic 1995), neural network control (Lewis, Jagannathan, and Yesildirek 1999), fuzzy logic control (Wang 1994) and fuzzy neural control (Leu, Lee, and Wang, 1999). Most of these works deal with the control problems of affine nonlinear systems, i.e. systems characterised by inputs appearing linearly in the system state equation. However, relatively few results are available for nonaffine, nonlinear systems where the control input appears in a nonlinear fashion (Ge and Wang 2002). In practice, there are many systems falling into this category, such as Van der Pol oscillator (Wang and Krstic 2000; Pourhiet, Correge, and Caruana 2003; Mahmoud and Farghaly 2004), magnetic servo levitation systems (Gutierrez and Ro 2005), aircraft flight control systems (Hunt and Meyer 1997) and biochemical process (Krstic, Kanellakopoulos, and Kokotovic 1992). Comparing to affine nonlinear systems, nonaffine, nonlinear systems are more complex and general, and we can say that affine systems can be viewed as a special kind of nonaffine systems (Ge and Zhang 2003). Thus, the control system design for nonaffine, nonlinear systems is not an easy task.

Reviewing some literature on nonaffine, nonlinear system control, we find some problems left to be addressed. In Labiod and Guerra (2007) and Park and Kim (2005), although system stability is guaranteed in the Lyapunov sense in Labiod and Guerra (2007), the unmeasurable term in the adaptive law needs to be approximated which will make the system stability questionable. Even if the system stability can be guaranteed, the tracking error is only ultimately uniformly bounded in Labiod and Guerra (2007). In Park and Kim (2005), the tracking error is uniformly asymptotically stable, but the robust controller to compensate the external disturbance causes the chattering of control input. Although the authors Park and Kim (2005) suggested some remedies to reduce the chattering, the tracking error may not be uniformly asymptotically stable due to these remedies.

To fix the drawbacks mentioned above, this article first proposes a novel SFS, which is used to approximate the unknown plant nonlinearity. The SFS considers both the growing and the pruning of fuzzy rules. In fact, it is possible that some rules are less or never fired throughout the operation of FS. These redundant rules, which make no meaningful contributions to the system output, are insignificant and thus should be removed to ease computational load. Second, a robust adaptive self-structuring fuzzy control (RASFC) scheme is proposed for a single-input and single-output (SISO) nonlinear, nonaffine system. Robust design is needed to guarantee the robust performance under of the RASFC scheme under system uncertainties and external disturbances. As well known, linear matrix inequality (LMI) techniques are widely used for robust design from 1990s (Boyd, Gahoui, Feron, and Balakrishnan 1994; Da, Cheng, and Tang 2000; Ramos, Alberto, and Bretas 2003). However, considering the complexity of LMI-based robust design procedure, in this article, a simple but powerful robust adaptive controller is merged into the control law to achieve L_2 tracking performance with a designed attenuation level. This L_2 tracking performance can provide a clear expression of tracking error in terms of the sum of lumped uncertainty and external disturbance, which has not been shown in previous works (Park and Kim 2005; Labiod and Guerra 2007). Moreover, all control parameters of the RASFC system are tuned online by the adaptive laws derived in the Lyapunov sense to achieve favourable fuzzy approximation. Finally, five examples are presented. For the purpose of interpreting the novel self-structuring algorithm, approximations of unknown nonlinear functions are performed in Examples 1 and 2 to illustrate the rule generation and pruning capabilities of the SFS. In Examples 3–5, tracking controls are provided to verify the effectiveness of the proposed RASFC scheme. To highlight the power of the proposed SFS, an adaptive FS with fixed number of rules and an SFS which can only automatically grow rules are also adopted in the Examples 3–5 for comparisons. Simulation results show that the proposed RASFC can achieve favourable tracking performance with a compact fuzzy rule base profited from the self-structuring algorithm. Comparing adaptive FS with fixed number of rules and SFS which can only grow rules, the proposed SFS with both rule growing and pruning capabilities can relieve computational load, yet maintain good tracking performance.

2. Problem formulation

Consider a SISO nonaffine, nonlinear system

$$x^{(n)} = f(\mathbf{x}, u) + d, \tag{1}$$

where $\mathbf{x} = [x \ \dot{x} \ \cdots \ x^{(n-1)}]^T$ is the measurable state vector of the system on a domain $\boldsymbol{\Omega}_\mathbf{x} \subset R^n$, $f(\mathbf{x}, u) : \boldsymbol{\Omega}_\mathbf{x} \times R \to R$ is the smooth unknown nonlinear function, u is the control input and d is the bounded external disturbance. Here, the single output is x. It should be noted that $f(\mathbf{x}, u)$ is an implicit function with respect to u. Feedback linearisation is performed by rewriting (1) as

$$x^{(n)} = cu + \Delta(\mathbf{x}, u) + d, \tag{2}$$

where c is a constant to be designed and $\Delta(\mathbf{x}, u) = f(\mathbf{x}, u) - cu$. Here, we assume that $\partial f(\mathbf{x}, u)/\partial u$ is nonzero for all $(\mathbf{x}, u) \in \boldsymbol{\Omega}_\mathbf{x} \times R$ with a known sign. Without losing generality, we further assume that (Calise, Hovakimyan, and Idan 2001; Hovakimyan Nardi, Calise, and Kim 2002; Park and Kim 2005)

$$\frac{\partial f(\mathbf{x}, u)}{\partial u} > 0 \tag{3}$$

for all $f(\mathbf{x}, u) \in \boldsymbol{\Omega}_\mathbf{x} \times R$. Note that for the nonaffine systems with property $\boldsymbol{\partial f(\mathbf{x}, u)\partial u < 0}$, the control scheme can be easily defined with minor modifications discussed in Section 4. The control objective is to develop a control scheme for the nonaffine, nonlinear system (1) so that the output trajectory x can track a given trajectory x_c closely. The tracking error is defined as

$$e = x_c - x. \tag{4}$$

If the system dynamics and the external disturbance are well-known, the ideal feedback controller can be determined as

$$u_{id}=\frac{1}{c}[u_{lc}-d-\Delta(\mathbf{x},u)], \tag{5}$$

where

$$u_{lc}=x_c^{(n)}+\mathbf{k}^T\mathbf{e} \tag{6}$$

with $\mathbf{e}=[e\ \dot{e}\ \cdots\ e^{(n-1)}]^T$ and $\mathbf{k}=[k_n\ k_{n-1}\ \cdots\ k_1]^T$. Applying (5) to (2) and using (4) yields the following error dynamics

$$e^{(n)}+k_1e^{(n-1)}+\cdots+k_ne=0. \tag{7}$$

If k_i, $i=1,2,\ldots,n$ are chosen so that all roots of the polynomial $H(s)\underline{\underline{\Delta}}\, s^n+k_1s^{n-1}+\cdots+k_n$ lie strictly in the open left half of the complex plane, then $\lim_{t\to\infty}e(t)=0$ can be implied for any initial conditions. However, since $\Delta(\mathbf{x},u)$ and the external disturbance d may be unknown or perturbed, the ideal feedback controller u_{id} in (5) cannot be implemented. Thus, to achieve the control objective, an SFS is designed to estimate the system uncertainty $\Delta(\mathbf{x},u)$ in (2).

3. Self-structuring fuzzy system

3.1. *Description of fuzzy system*

FSs are attractive candidates for the systems that are structurally difficult to model due to inherent nonlinearity and model complexities. Typically, an FS includes four well-known stages: a fuzzifier, a rule base, an inference engine and a defuzzifier. The rule base is the collection of fuzzy rules which characterise the simple input–output relation of the system. Note that the self-structuring algorithm introduced in this section is applicable to a multi-input and multi-output (MIMO) FS. However, without losing generality and to simplify the notation, a MISO FS is adopted to describe the algorithm. A MISO FS can be expressed as (Terano et al. 1992):

$$\begin{aligned}\text{Rule}_{i_1,i_2,\ldots,i_m}:\ &\text{IF } X_1 \text{ is } F_1^{i_1} \text{ and } X_2 \text{ is } F_2^{i_2} \text{ and} \ldots \text{and}\\ &X_m \text{ is } F_m^{i_m} \ldots \text{THEN } y \text{ is } \alpha_{i_1,i_2,\ldots,i_m}\end{aligned} \tag{8}$$

where X_j, $j=1,2,\ldots,m$ are input variables; y is output variable; $\alpha_{i_1,i_2,\ldots,i_m}$ is the crisp singleton consequent; and $F_j^{i_j}$ is the fuzzy sets characterised by the fuzzy membership function $F_j^{i_j}(X_j)$, with $i_j\in\{1,2,\ldots,N_j\}$ being the ordinal number of membership functions of X_j. Define a set $\boldsymbol{\Omega}$ which collects all possible fuzzy rules

$$\begin{aligned}\boldsymbol{\Omega}=\{&\text{Rule}_{i_1,i_2,\ldots,i_m}|i_1=1,2,\ldots,N_1;\ i_2=1,2,\ldots,N_2;\\ &i_m=1,2,\ldots,N_m\}\end{aligned} \tag{9}$$

The output of the FS can be expressed as (Terano et al. 1992):

$$y=\frac{\sum_{\text{Rule}_{i_1,i_2,\ldots,i_m}\in\boldsymbol{\Omega}_{\text{sub}}}\alpha_{i_1,i_2,\ldots,i_m}\left[\prod_{j=1}^m\mu_{F_j^{i_j}}(X_j)\right]}{\sum_{\text{Rule}_{i_1,i_2,\ldots,i_m}\in\boldsymbol{\Omega}_{\text{sub}}}\left[\prod_{j=1}^m\mu_{F_j^{i_j}}(X_j)\right]}, \tag{10}$$

where $\boldsymbol{\Omega}_{\text{sub}}\subseteq\boldsymbol{\Omega}$ is the rule base. From (10), the output of the FS can be represented as a linear combination of fuzzy basis functions defined as

$$\xi_{i_1,i_2,\ldots,i_m}=\frac{\prod_{j=1}^m\mu_{F_j^{i_j}}(X_j)}{\sum_{\text{Rule}_{i_1,i_2,\ldots,i_m}\in\boldsymbol{\Omega}_{\text{sub}}}\left[\prod_{j=1}^m\mu_{F_j^{i_j}}(X_j)\right]},\quad i_j\in\{1,2,\ldots,N_j\},\ j=1,2,\ldots,m. \tag{11}$$

That is, (10) can be rewritten as

$$y=\alpha^T\xi \tag{12}$$

where $\alpha\in R^{n\times1}$ collects singleton consequents $\alpha_{i_1,i_2,\ldots,i_m}$ of all rules in $\boldsymbol{\Omega}_{\text{sub}}$, $\xi\in R^{n\times1}$ collects $\xi_{i_1,i_2,\ldots,i_m}$ described in (11) and n is the number of the existing fuzzy rules. In this chapter, a Gaussian membership function is defined as

$$\mu_{F_j^{i_j}}(X_j,c_j^{i_j},\sigma_j^{i_j})=\exp\left\{-\frac{[X_j-c_j^{i_j}]^2}{\sigma_j^{i_j2}}\right\}, \tag{13}$$

where $c_j^{i_j}$ and $\sigma_j^{i_j}$ are the mean and standard deviation (SD) of the Gaussian function, respectively.

3.2. *Structure learning algorithm*

The developed self-structuring algorithm consists of two parts: growing and pruning of fuzzy rules. Effective membership functions in the input spaces can be generated and ineffective fuzzy rules can be pruned automatically by the self-structuring algorithm and thus, a concise rule base can be obtained. In order to construct the fuzzy rule base, every input space $S(X_j)$ is partitioned into several overlapping clusters to construct the fuzzy sets of X_j. It can happen that for some incoming X_j, the degree of belongings to all its fuzzy sets are quite small, i.e. $F_j^{i_j}(X_j)$, $i_j=1,2,\ldots,N_j$ are quite small, as depicted in Figure 1(a). This means that the input space $S(X_j)$ is not properly clustered. Hence, the fundamental concept of the growing of fuzz rules is developed to adjust the inappropriate clustering. Initially, create one initial fuzzy rule with the given initial state as

$$\begin{aligned}\text{Rule}_{1,1,\ldots,1}:\ &\text{IF } X_1 \text{ is } F_1^1 \text{ and } X_2 \text{ is } F_2^2 \text{ and} \ldots \text{and } X_m \text{ is}\\ &F_m^1 \text{ THEN } y \text{ is } \alpha_{1,1,\ldots,1},\end{aligned} \tag{14}$$

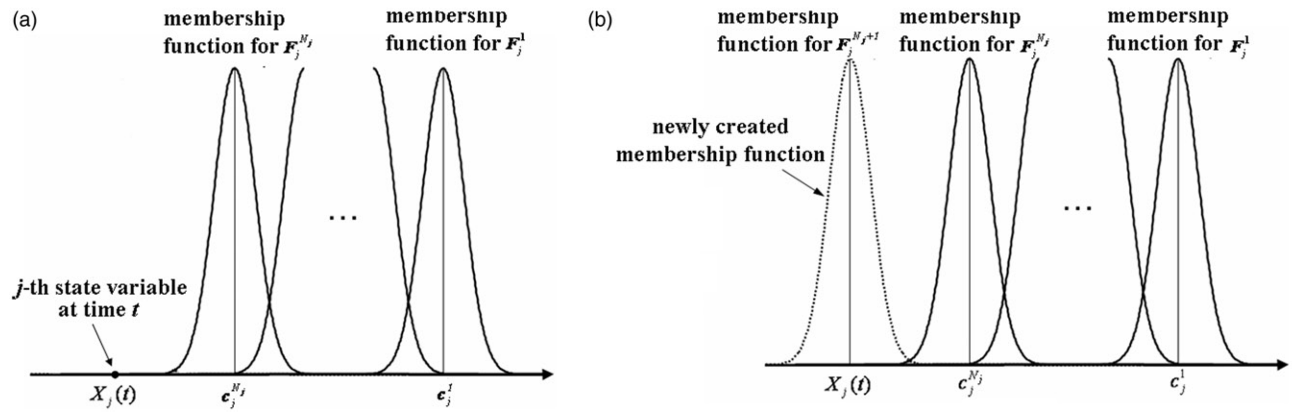

Figure 1. (a) The improper fuzzy clustering of input variable X_j; (b) The newly created membership function.

where the membership functions for $F_j^1, j = 1, 2, \ldots, m$, are defined with the initial input $X_j(0)$ as

$$\mu_{F_j^1}(X_j) = \exp\left\{-\frac{[X_j - X_j(0)]^2}{\sigma_j^{12}}\right\}. \tag{15}$$

The SFS will start operating from this single rule. Define the growing criterion as

$$\mu_j^{\max} < \Theta_g, \quad j = 1, 2, \ldots, m, \tag{16}$$

where $\mu_j^{\max} = \max_{i_j=1,2,\ldots,N_j} \mu_{F_j^{i_j}}(X_j)$ is the maximum membership function degree of X_j and $\Theta_g \in (0, 1)$ is a given threshold. If at some time t_g, the growing criterion (16) is satisfied for a new incoming datum, $X_j(t_g)$, $1 \le j \le m$, a new membership function is created, whose initial mean and SD are

$$c_j^{N_j+1} = X_j(t_g), \tag{17}$$

$$\sigma_j^{N_j+1} = q, \tag{18}$$

where $q > 0$ can be arbitrarily chosen and it will be tuned by the adaptive law introduced in later section. The created membership function is shown in Figure 1(b). For the case that one new membership function is created at some time, $N_1 \times \ldots \times N_{j-1} \times N_{j+1} \times \cdots \times N_m$ new fuzzy rules will be generated according to the new membership function as:

$$\begin{aligned}
&\text{Rule}_{1,\ldots,N_j+1,\ldots,1}: \text{IF } X_1 \text{ is } F_1^1 \ldots X_j \text{ is } F_j^{N_j+1} \ldots \text{ and } X_m \text{ is } F_m^1, \text{ THEN } y \text{ is } \alpha_{1,\ldots,N_j+1,\ldots,1}.\\
&\text{Rule}_{2,\ldots,N_j+1,\ldots,1}: \text{IF } X_1 \text{ is } F_1^2 \ldots X_j \text{ is } F_j^{N_j+1} \ldots \text{ and } X_m \text{ is } F_m^1, \text{ THEN } y \text{ is } \alpha_{2,\ldots,N_j+1,\ldots,1}.\\
&\text{Rule}_{N_1,\ldots,N_j+1,\ldots,N_m}: \text{IF } X_1 \text{ is } F_1^{N_1} \ldots X_j \text{ is } F_j^{N_j+1} \ldots \text{ and } X_m \text{ is } F_m^{N_m}, \text{ THEN } y \text{ is } \alpha_{N_1,\ldots,N_j+1,\ldots,N_m}.
\end{aligned} \tag{19}$$

For example, consider an FS ($m = 2$, $N_1 = 1$ and $N_2 = 2$) with the rule base:

$$\text{Rule}_{1,1}: \text{IF } X_1 \text{ is } F_1^1 \text{ and } X_2 \text{ is } F_2^1, \text{ THEN } y \text{ is } \alpha_{1,1}.$$
$$\text{Rule}_{1,2}: \text{IF } X_1 \text{ is } F_1^1 \text{ and } X_2 \text{ is } F_2^2, \text{ THEN } y \text{ is } \alpha_{1,2}.$$

Assume that the growing criterion for X_1 is satisfied at time t. Then, a new membership function

$$\mu_{F_1^2} = \exp\left\{-\frac{[X_1 - X_1(t)]^2}{\sigma_1^2}\right\} \tag{20}$$

is created, and two rules are grown according to the new membership function as

$$\begin{aligned}
&\text{Rule}_{2,1}: \text{IF } X_1 \text{ is } F_1^2 \text{ and } X_2 \text{ is } F_2^1, \text{ THEN } y \text{ is } \alpha_{2,1}.\\
&\text{Rule}_{2,2}: \text{IF } X_1 \text{ is } F_1^2 \text{ and } X_2 \text{ is } F_2^2, \text{ THEN } y \text{ is } \alpha_{2,2}.
\end{aligned} \tag{21}$$

It can be observed from (16) and (17) that the proposed rule growing strategy in nature has less chance to suffer from the problem of generating highly overlapping fuzzy sets.

An SFS with only a rule generation algorithm may suffer from the computational load or learning failure caused by an overly large rule base which includes both effective and redundant fuzzy rules. In the following, the strategy to prune redundant rules is developed to solve this problem. Recall that there are n existing fuzzy rules, and then express (12) as

$$y = \boldsymbol{\alpha}^T \boldsymbol{\xi} = \begin{bmatrix} \alpha_k & \boldsymbol{\alpha}_{rm} \end{bmatrix} \begin{bmatrix} \xi_k \\ \boldsymbol{\xi}_{rm} \end{bmatrix}, \tag{22}$$

where $\alpha_k \in R$ and $\alpha_{rm} \in R^{(n-1)1}$ represent the singleton consequent and the fuzzy basis function of the k-th fuzzy rule, respectively; $\boldsymbol{\alpha}_{rm} \in R^{(n-1)\times 1}$ and $\boldsymbol{\xi}_{rm} \in R^{(n-1)\times 1}$ represent the collections of the singleton consequents and the fuzzy basis functions of the rest of fuzzy rules, respectively. Thus, the contribution made by k-th rule on the output y can be defined as follows:

$$C_k = \frac{|y_k|}{\sum_{k=1}^{n} |y_k|}, \quad k = 1, 2, \ldots, n, \tag{23}$$

where $y_k = \alpha_k \xi_k$. Now, we are ready to introduce the significance index which can help us to decide whether or not to prune a fuzzy rule. The significance index is a measurement of the importance of every fuzzy rule. S_k, which represents the significance index of the k-th fuzzy rule, is updated as follows:

$$S_k = \begin{cases} S_k^{rc}\tau, & \text{if } C_k < \beta \\ S_k^{rc}, & \text{if } C_k \geq \beta \end{cases}, \quad k = 1, 2, \ldots, n, \tag{24}$$

where S_k^{rc} is the most recent S_k, $\tau \in (0, 1)$ is a decay constant and $\beta \in (0, 1)$ is a given constant. All S_k, $k = 1, 2, \ldots, n$, are initialised from ones. According to (18), if the contribution C_k is equal or larger than β, S_k keeps invariant; if C_k is smaller than β, S_k will be attenuated. An invariant significance implies that the associated rule is still important and should remain; a decaying significance index implies that the associated rule is becoming less and less important and thus should be pruned. The selection of τ will affect the rate of pruning the fuzzy rules. The smaller the τ is (or the larger the β is), the faster the significance index S_k decays, and thus the faster the ineffective fuzzy rules will be pruned. The pruning criterion of the k-th fuzzy rule is defined as follows based on this knowledge

$$S_k < \Theta_p, \quad k = 1, 2, \ldots, n, \tag{25}$$

where $\Theta_p \in (0, 1)$ is a selected threshold. If the pruning criterion is satisfied for S_k, the associated k-th rule is pruned.

Remark 1: It is a difficult task to determine the initial values of the singleton consequents of the newly generated fuzzy rules. Because an SFS is in general equipped with a parameter learning algorithm to automatically tune the parameters of the fuzzy rules, the initial values of the singleton consequents can simply be set as zeros. However, from (10), we can see that this will cause abrupt variation of the fuzzy output y and the performance of the SFS may deteriorate for a short period. This phenomenon can be observed in Figure 5(b). To fix this drawback, we maintain the approximation property of the SFS at the instant that new rules are generated. Assume that at some time t_g, an SFS has n fuzzy rules and the last h rules are just newly generated. Define y_p as the 'pseudo fuzzy output' of the original n–h rules if h new rules were not generated at t_g. The initial consequents of those new rules are chosen so that $y(t_g) = y_p$. Thus, we have

$$y(t_g) = \alpha_{new} \sum_{k=n-h+1}^{n} \xi_k + \sum_{k=1}^{n-h} \alpha_k \xi_k = y_p, \tag{26}$$

where $\alpha_{n-h+1} = \alpha_{n-h+2} = \cdots = \alpha_n = \alpha_{new}$. From (26), we can easily obtain

$$\alpha_{new} = \frac{y_p - \sum_{k=1}^{n-h} \alpha_k \xi_k}{\sum_{k=n-h+1}^{n} \xi_k}. \tag{27}$$

In this way, not only the bad effect caused by the abrupt variation can be mitigated, but also the future performance of the SFS can be improved by the h new rules.

Remark 2: While controlling, a membership function is possible to be pruned if all fuzzy rules associated with this membership function are pruned sequentially.

Remark 3: In the implementations of practical systems, if computational burden is the issue having highest priority, the threshold Θ_p can be chosen large enough so that more fuzzy rules are pruned. Hence, the computational burden will be substantially reduced at the expense of less favourable system performance.

Figure 2 shows the flowchart to summarise the self-structuring algorithm for the SFS. The growing and pruning effects during the control period will be illustrated in later sections with excellent result.

4. Design of RASFC

Now, we are ready for developing a RASFC for the unknown nonaffine, nonlinear systems. In the RASFC, an SFS is used to estimate the system uncertainty $\Delta(\mathbf{x}, u)$ in (2). The control law u in the RASFC system is designed as

$$u = \frac{1}{c}(u_{rac} - u_{fc}), \tag{28}$$

where u_{rac} is the robust adaptive controller to achieve a L_2 tracking performance with a designed attenuation level and u_{fc} is the self-structuring fuzzy controller to approximate unknown system dynamics $\Delta(\mathbf{x}, u)$. Substituting (28) into (2) and using (4) yields

$$\begin{aligned} e^{(n)} &= x_c^{(n)} - \left[u_{rac} - u_{fc} + \Delta(\mathbf{x}, u) + d\right] \\ &= x_c^{(n)} - u_{lc} - \left\{\left[\Delta(\mathbf{x}, u) - u_{fc}\right] + (u_{rac} - u_{lc}) + d\right\} \\ &= -\mathbf{k}^T e - \left\{\left[\Delta(\mathbf{x}, u) - u_{fc}\right] + (u_{rac} - u_{lc}) + d\right\}, \end{aligned} \tag{29}$$

or

$$\dot{\mathbf{e}} = \mathbf{A}\mathbf{e} - \mathbf{b}\left[\Delta(\mathbf{x}, u) - u_{fc} + (u_{rac} - u_{lc}) + d\right], \tag{30}$$

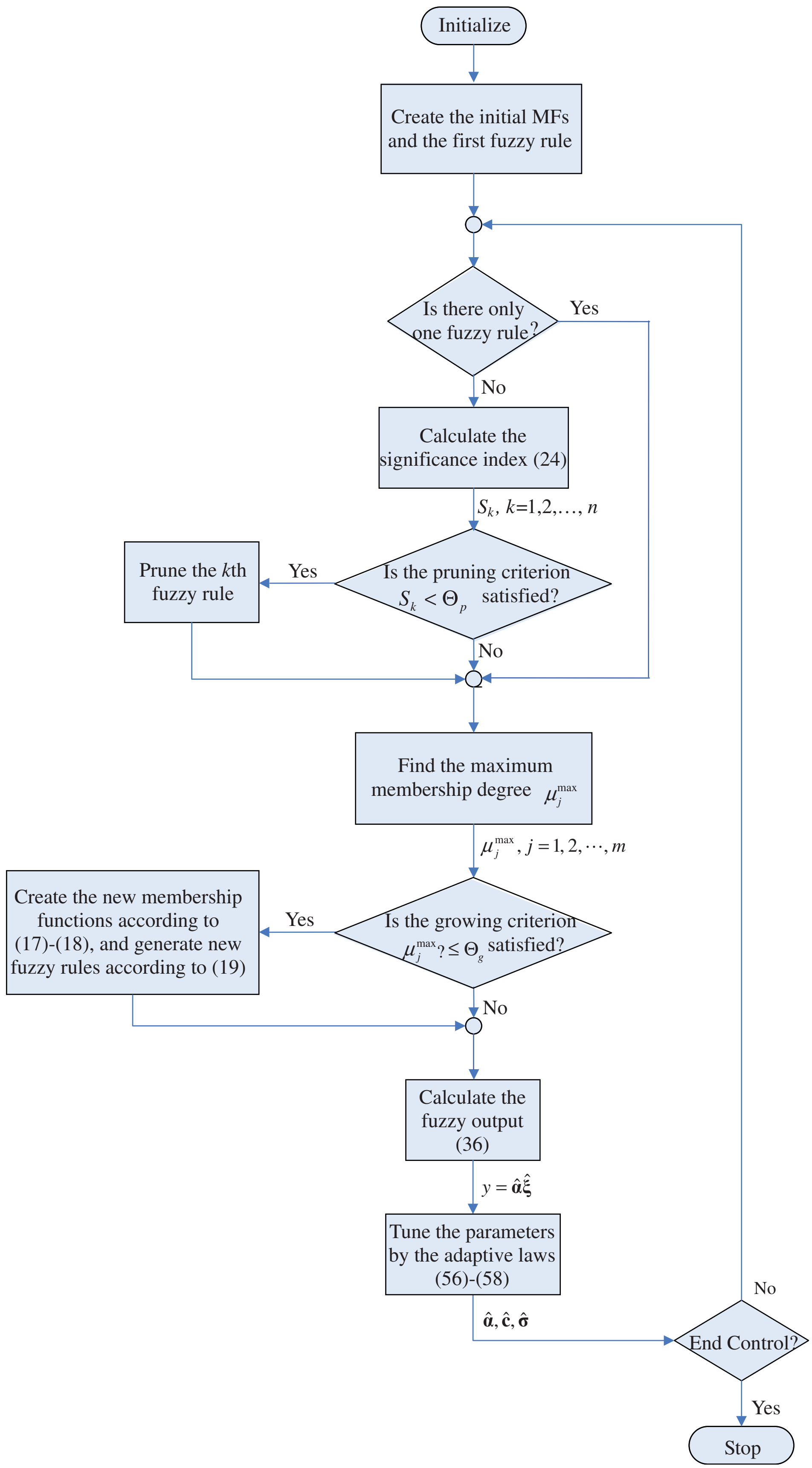

Figure 2. The flowchart of the self-structuring algorithm for the SFS.

where

$$\mathbf{A}=\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 1 \\ -k_n & -k_{n-1} & \cdots & \cdots & -k_1 \end{bmatrix} \quad \text{and}$$

$$\mathbf{b}=\begin{bmatrix} 0 & 0 & \cdots & 1 \end{bmatrix}^T.$$

4.1. *Fuzzy approximation*

The unknown nonlinear function $\Delta(\mathbf{x}, u)$ is approximated by an SFS with inputs $\mathbf{x}$ and u. In this way, the output of the SFS u_{fc} should be directly fed back to produce u, which is one of the input of the SFS. This kind of FS is called a recurrent FS, as depicted in Figure 3(a). However, a recurrent FS will lead to a fixed-point problem which must be solved at every time instant and thus imposes computational burden (Calise et al. 2001; Hovakimyan et al. 2002; Park and Kim 2005).

Thus, the following Lemma 1 is stated to avoid this problem (Calise et al. 2001; Hovakimyan et al. 2002; Park and Kim 2005).

Lemma 1: *Let the constant c satisfy the condition*

$$c > \frac{1}{2}\left(\frac{\partial f}{\partial u}\right). \tag{31}$$

Then, there exist a unique u_{fc}^* *which is a function of* $\mathbf{x}$ *and* u_{rac} *so that* $u_{fc}^*(\mathbf{x}, u_{rac})$ *satisfies*

$$\psi(\mathbf{x}, u_{rac}, u_{fc}^*) \overset{\Delta}{=} \Delta(\mathbf{x}, u_{rac}, u_{fc}^*) - u_{fc}^*(\mathbf{x}, u_{rac}) = 0, \tag{32}$$

for all $(\mathbf{x}, u_{rac}) \in \mathbf{\Omega}_\mathbf{x} \times R$.

The Proof of Lemma 1 can be found in Park and Kim (2005).

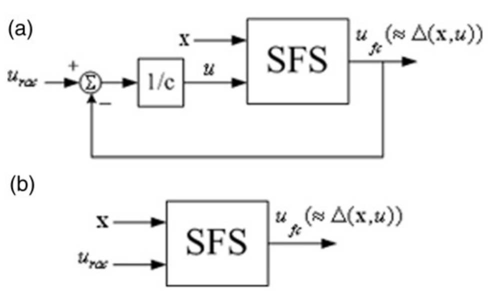

Figure 3. (a) The recurrent fuzzy system; (b) The static fuzzy system.

According to Lemma 1, the feedback path in Figure 3(a) can be removed. Consequently, a static FS in Figure 3(b) can be used to approximate $\Delta(\mathbf{x}, u)$, and thus we do not need to solve the fixed-point problem at every time instant. For the nonaffine systems with the property $\partial f(\mathbf{x}, u)/\partial u < 0$, Lemma 1 can be satisfied as well by simply modifying (31) as $c < 1/2(\partial f/\partial u)$.

Define the vectors $\mathbf{c}$ and $\boldsymbol{\sigma}$ as

$$\mathbf{c} = \begin{bmatrix} \mathbf{c}_1 & \mathbf{c}_2 & \cdots & \mathbf{c}_m \end{bmatrix}^T, \tag{33}$$

$$\boldsymbol{\sigma} = \begin{bmatrix} \boldsymbol{\sigma}_1 & \boldsymbol{\sigma}_2 & \cdots & \boldsymbol{\sigma}_m \end{bmatrix}^T, \tag{34}$$

where $\mathbf{c}_j = [c_j^1 \quad \cdots \quad c_j^{N_j}]$ and $\boldsymbol{\sigma}_j = [\sigma_j^1 \quad \cdots \quad \sigma_j^{N_j}]$ collect the means and SDs of the Gaussian membership functions of X_j, $j = 1, 2, \ldots, m$, respectively. Rewrite (12) in the vector form as

$$y = \boldsymbol{\alpha}^T \boldsymbol{\xi}(\mathbf{X}, \mathbf{c}, \boldsymbol{\sigma}) = \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_n \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_n \end{bmatrix}, \tag{35}$$

where $\mathbf{X} = [\mathbf{x} \quad u_{rac}]^T$ is the input vector. The output of the SFS used to approximate $\Delta(\mathbf{x}, u)$ is defined as

$$\mathbf{u}_{fc} = \hat{\boldsymbol{\alpha}}^T \boldsymbol{\xi}(\mathbf{X}, \hat{\mathbf{c}}, \hat{\boldsymbol{\sigma}}) = \hat{\boldsymbol{\alpha}}^T \hat{\boldsymbol{\xi}}, \tag{36}$$

where $\hat{\boldsymbol{\alpha}}$, $\hat{\mathbf{c}}$ and $\hat{\boldsymbol{\sigma}}$ are the estimation vectors of $\boldsymbol{\alpha}$, $\mathbf{c}$ and $\boldsymbol{\sigma}$ and $\hat{\boldsymbol{\xi}} = \boldsymbol{\xi}(\mathbf{X}, \hat{\mathbf{c}}, \hat{\boldsymbol{\sigma}})$. Define the optimal vectors $\boldsymbol{\alpha}^*$, $\mathbf{c}^*$ and $\boldsymbol{\sigma}^*$ as (Wang 1994):

$$(\boldsymbol{\alpha}^*, \mathbf{c}^*, \boldsymbol{\sigma}^*) = \underset{\hat{\boldsymbol{\alpha}} \in \mathbf{\Omega}_{\boldsymbol{\alpha}},\, \hat{\mathbf{c}} \in \mathbf{\Omega}_{\mathbf{c}},\, \hat{\boldsymbol{\sigma}} \in \mathbf{\Omega}_{\boldsymbol{\sigma}}}{\arg\min} \left[\sup_{\mathbf{X} \in \mathbf{\Omega}_\mathbf{x} \times R} \left| u_{fc}^*(\mathbf{X}) - u_{fc}(\mathbf{X}, \hat{\boldsymbol{\alpha}}, \hat{\mathbf{c}}, \hat{\boldsymbol{\sigma}}) \right| \right], \tag{37}$$

where

$$\mathbf{\Omega}_{\boldsymbol{\alpha}} = \{ \hat{\boldsymbol{\alpha}} : \|\hat{\boldsymbol{\alpha}}\| \le M_{\boldsymbol{\alpha}} \}, \tag{38}$$

$$\mathbf{\Omega}_{\mathbf{c}} = \{ \hat{\mathbf{c}} : \|\hat{\mathbf{c}}\| \le M_{\mathbf{c}} \}, \tag{39}$$

$$\mathbf{\Omega}_{\boldsymbol{\sigma}} = \{ \hat{\boldsymbol{\sigma}} : \|\hat{\boldsymbol{\sigma}}\| \le M_{\boldsymbol{\sigma}} \}, \tag{40}$$

And $M_{\boldsymbol{\alpha}}$, $M_{\mathbf{c}}$ and $M_{\boldsymbol{\sigma}}$ are positive constants specified by designers. The unknown nonlinear function $\Delta(\mathbf{x}, u)$ can be described as

$$\Delta = \boldsymbol{\alpha}^{*T} \boldsymbol{\xi}(\mathbf{X}, \mathbf{c}^*, \boldsymbol{\sigma}^*) + \omega = \boldsymbol{\alpha}^{*T} \boldsymbol{\xi}^* + \omega, \tag{41}$$

where $\boldsymbol{\xi}^* = \boldsymbol{\xi}(\mathbf{X}, \mathbf{c}^*, \boldsymbol{\sigma}^*)$ and ω denotes the approximation error bounded by $|\omega| \le \bar{\omega}$, in which $\bar{\omega}$ is a finite positive constant. Then, modelling error $\tilde{u}$ can be expressed as

$$\tilde{u} = \Delta - u_{fc} = \tilde{\boldsymbol{\alpha}}^T \hat{\boldsymbol{\xi}} + \hat{\boldsymbol{\alpha}}^T \tilde{\boldsymbol{\xi}} + \tilde{\boldsymbol{\alpha}}^T \tilde{\boldsymbol{\xi}} + \omega, \tag{42}$$

where $\tilde{\boldsymbol{\alpha}} = \boldsymbol{\alpha}^* - \hat{\boldsymbol{\alpha}}$ and $\tilde{\boldsymbol{\xi}} = \boldsymbol{\xi}^* - \hat{\boldsymbol{\xi}}$. In the following, some preliminaries will be made for adaptive online tuning of the parameters of fuzzy rules and thus favourable approximation performance can be achieved in the presence of unexpected disturbances. To achieve this goal, the Taylor linearisation technique is employed to transform the nonlinear fuzzy basis function into partially linear form as follows (Han, Su, and Stepanenko 2001; Hsu and Lin 2005):

$$\tilde{\boldsymbol{\xi}} = \begin{bmatrix} \tilde{\xi}_1 \\ \tilde{\xi}_2 \\ \vdots \\ \tilde{\xi}_n \end{bmatrix} = \begin{bmatrix} \frac{\xi_1}{\partial \mathbf{c}} \\ \frac{\xi_2}{\partial \mathbf{c}} \\ \vdots \\ \frac{\xi_n}{\partial \mathbf{c}} \end{bmatrix}\Bigg|_{\mathbf{c}=\hat{\mathbf{c}}} (\mathbf{c}^* - \hat{\mathbf{c}}) + \begin{bmatrix} \frac{\xi_1}{\partial \boldsymbol{\sigma}} \\ \frac{\xi_2}{\partial \boldsymbol{\sigma}} \\ \vdots \\ \frac{\xi_n}{\partial \boldsymbol{\sigma}} \end{bmatrix}\Bigg|_{\boldsymbol{\sigma}=\hat{\boldsymbol{\sigma}}} (\boldsymbol{\sigma}^* - \hat{\boldsymbol{\sigma}}) + o, \tag{43}$$

or

$$\tilde{\boldsymbol{\xi}} = \boldsymbol{\xi}_{\mathbf{c}}^T \tilde{\mathbf{c}} + \boldsymbol{\xi}_{\boldsymbol{\sigma}}^T \tilde{\boldsymbol{\sigma}} + o, \tag{44}$$

where $\boldsymbol{o}$ represents the higher order term, $\tilde{\mathbf{c}} = \mathbf{c}^* - \hat{\mathbf{c}}$, $\tilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma}^* - \hat{\boldsymbol{\sigma}}$ and

$$\boldsymbol{\xi}_{\mathbf{c}} = \begin{bmatrix} \frac{\partial \xi_1}{\partial \mathbf{c}} & \frac{\partial \xi_2}{\partial \mathbf{c}} & \cdots & \frac{\partial \xi_n}{\partial \mathbf{c}} \end{bmatrix}\Bigg|_{\mathbf{c}=\hat{\mathbf{c}}}, \tag{45}$$

$$\boldsymbol{\xi}_{\boldsymbol{\sigma}} = \begin{bmatrix} \frac{\partial \xi_1}{\partial \boldsymbol{\sigma}} & \frac{\partial \xi_2}{\partial \boldsymbol{\sigma}} & \cdots & \frac{\partial \xi_n}{\partial \boldsymbol{\sigma}} \end{bmatrix}\Bigg|_{\boldsymbol{\sigma}=\hat{\boldsymbol{\sigma}}}. \tag{46}$$

Substituting (44) into (42) yields

$$\begin{aligned} \tilde{u} &= \tilde{\boldsymbol{\alpha}}^T \hat{\boldsymbol{\xi}} + \hat{\boldsymbol{\alpha}}^T \boldsymbol{\xi}_{\mathbf{c}}^T \tilde{\mathbf{c}} + \hat{\boldsymbol{\alpha}}^T \boldsymbol{\xi}_{\boldsymbol{\sigma}}^T \tilde{\boldsymbol{\sigma}} + \varepsilon \\ &= \tilde{\boldsymbol{\alpha}}^T \hat{\boldsymbol{\xi}} + \tilde{\mathbf{c}}^T \boldsymbol{\xi}_{\mathbf{c}} \hat{\boldsymbol{\alpha}} + \tilde{\boldsymbol{\sigma}}^T \boldsymbol{\xi}_{\boldsymbol{\sigma}} \hat{\boldsymbol{\alpha}} + \varepsilon, \end{aligned} \tag{47}$$

where $\hat{\boldsymbol{\alpha}}^T \boldsymbol{\xi}_{\mathbf{c}}^T \tilde{\mathbf{c}} = \tilde{\mathbf{c}}^T \boldsymbol{\xi}_{\mathbf{c}} \hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\alpha}}^T \boldsymbol{\xi}_{\boldsymbol{\sigma}}^T \tilde{\boldsymbol{\sigma}} = \tilde{\boldsymbol{\sigma}}^T \boldsymbol{\xi}_{\boldsymbol{\sigma}} \hat{\boldsymbol{\alpha}}$ since they are scalars and $\boldsymbol{\varepsilon} = \tilde{\boldsymbol{\alpha}}^T \tilde{\boldsymbol{\xi}} + \hat{\boldsymbol{\alpha}}^T \boldsymbol{o} + \boldsymbol{\omega}$ is the lumped uncertainty. The higher order term $\boldsymbol{o}$ satisfies

$$\begin{aligned} \|\boldsymbol{o}\| &= \tilde{\boldsymbol{\xi}} - \boldsymbol{\xi}_{\mathbf{c}}^T \tilde{\mathbf{c}} + \boldsymbol{\xi}_{\boldsymbol{\sigma}}^T \tilde{\boldsymbol{\sigma}} \\ &\leq \left\|\tilde{\boldsymbol{\xi}}\right\| + \left\|\boldsymbol{\xi}_{\mathbf{c}}^T\right\| \|\tilde{\mathbf{c}}\| + \left\|\boldsymbol{\xi}_{\boldsymbol{\sigma}}^T\right\| \|\tilde{\boldsymbol{\sigma}}\| \\ &\leq b_0 + b_1 \|\tilde{\mathbf{c}}\| + b_2 \|\tilde{\boldsymbol{\sigma}}\|, \end{aligned} \tag{48}$$

where b_0, b_1 and b_2 are bounded positive constants satisfying $\|\tilde{\boldsymbol{\xi}}\| \leq b_0$, $\left\|\boldsymbol{\xi}_{\mathbf{c}}^T\right\| \leq b_1$ and $\left\|\boldsymbol{\xi}_{\boldsymbol{\sigma}}^T\right\| \leq b_2$. It is reasonable that b_0, b_1 and b_2 exist because Gaussian function and its derivative are always bounded by constants. Moreover, $\tilde{\boldsymbol{\alpha}}$, $\tilde{\mathbf{c}}$ and $\tilde{\boldsymbol{\sigma}}$ satisfy

$$\|\tilde{\boldsymbol{\alpha}}\| = \left\|\boldsymbol{\alpha}^* - \hat{\boldsymbol{\alpha}}\right\| \leq \|\boldsymbol{\alpha}^*\| + \left\|\hat{\boldsymbol{\alpha}}\right\| \leq M_{\boldsymbol{\alpha}} + \left\|\hat{\boldsymbol{\alpha}}\right\|, \tag{49}$$

$$\|\tilde{\mathbf{c}}\| = \left\|\mathbf{c}^* - \hat{\mathbf{c}}\right\| \leq \|\mathbf{c}^*\| + \left\|\hat{\mathbf{c}}\right\| \leq M_{\mathbf{c}} + \left\|\hat{\mathbf{c}}\right\|, \tag{50}$$

$$\|\tilde{\boldsymbol{\sigma}}\| = \left\|\boldsymbol{\sigma}^* - \hat{\boldsymbol{\sigma}}\right\| \leq \|\boldsymbol{\sigma}^*\| + \left\|\hat{\boldsymbol{\sigma}}\right\| \leq M_{\boldsymbol{\sigma}} + \left\|\hat{\boldsymbol{\sigma}}\right\|. \tag{51}$$

Thus, the lumped uncertainty ε satisfies

$$\begin{aligned} |\varepsilon| &= \left|\tilde{\boldsymbol{\alpha}}^T(\boldsymbol{\xi}_{\mathbf{c}}^T \tilde{\mathbf{c}} + \boldsymbol{\xi}_{\boldsymbol{\sigma}}^T \tilde{\boldsymbol{\sigma}} + \boldsymbol{o}) + \hat{\boldsymbol{\alpha}}^T \boldsymbol{o} + \omega\right| \\ &= \left|\tilde{\boldsymbol{\alpha}}^T \boldsymbol{\xi}_{\mathbf{c}}^T \tilde{\mathbf{c}} + \tilde{\boldsymbol{\alpha}}^T \boldsymbol{\xi}_{\boldsymbol{\sigma}}^T \tilde{\boldsymbol{\sigma}} + \boldsymbol{\alpha}^{*^T} \boldsymbol{o} + \omega\right| \\ &\leq b_1(M_{\boldsymbol{\alpha}} + \|\hat{\boldsymbol{\alpha}}\|)(M_{\mathbf{c}} + \|\hat{\mathbf{c}}\|) + b_2(M_{\boldsymbol{\alpha}} + \|\hat{\boldsymbol{\alpha}}\|)(M_{\boldsymbol{\sigma}} + \|\hat{\boldsymbol{\sigma}}\|) \\ &\quad + M_{\boldsymbol{\alpha}}[b_0 + b_1(M_{\mathbf{c}} + \|\hat{\mathbf{c}}\|) + b_2(M_{\boldsymbol{\sigma}} + \|\hat{\boldsymbol{\sigma}}\|)] + \bar{\omega} \\ &= [\Lambda_1\, \Lambda_2\, \Lambda_3\, \Lambda_4\, \Lambda_5\, \Lambda_6][1\, \|\hat{\boldsymbol{\alpha}}\|\, \|\hat{\mathbf{c}}\|\, \|\hat{\boldsymbol{\sigma}}\|\, \|\hat{\boldsymbol{\alpha}}\|\, \|\hat{\mathbf{c}}\|\, \|\hat{\boldsymbol{\alpha}}\|\, \|\hat{\boldsymbol{\sigma}}\|]^T \\ &= \boldsymbol{\Lambda}^T \boldsymbol{\Gamma}, \end{aligned} \tag{52}$$

where $\boldsymbol{\Lambda} = [\Lambda_1\, \Lambda_2\, \Lambda_3\, \Lambda_4\, \Lambda_5\, \Lambda_6]^T$, $\Lambda_1 = (b_0 + 2b_1 M_{\mathbf{c}} + 2b_2 M_{\boldsymbol{\sigma}}) M_{\boldsymbol{\alpha}} + \bar{\omega}$, $\Lambda_2 = b_1 M_{\mathbf{c}} + b_2 M_{\boldsymbol{\sigma}}$, $\Lambda_3 = 2b_1 M_{\boldsymbol{\alpha}}$, $\Lambda_4 = 2b_2 M_{\boldsymbol{\alpha}}$, $\Lambda_5 = b_1$, $\Lambda_6 = b_2$ and $\boldsymbol{\Gamma} = [1 \|\hat{\boldsymbol{\alpha}}\| \|\hat{\mathbf{c}}\| \|\hat{\boldsymbol{\sigma}}\| \|\hat{\boldsymbol{\alpha}}\| \|\hat{\mathbf{c}}\| \|\hat{\boldsymbol{\alpha}}\| \|\hat{\boldsymbol{\sigma}}\|]^T$. Since $\boldsymbol{\Lambda}$ is a bounded vector, if $\boldsymbol{\Gamma}$ is guaranteed to be bounded, the lumped uncertainty term ε is thus bounded. We can guarantee the boundness of $\boldsymbol{\Gamma}$ by Lemma 2 given in the next subsection.

4.2. *Parameter learning algorithm*

Substituting (47) into (30) yields

$$\dot{\mathbf{e}} = \mathbf{A}\mathbf{e} - \mathbf{b}[\tilde{\boldsymbol{\alpha}}^T \hat{\boldsymbol{\xi}} + \tilde{\mathbf{c}}^T \boldsymbol{\xi}_{\mathbf{c}} \hat{\boldsymbol{\alpha}} + \tilde{\boldsymbol{\sigma}}^T \boldsymbol{\xi}_{\boldsymbol{\sigma}} \hat{\boldsymbol{\alpha}} + \varepsilon + d + (u_{rac} - u_{lc})]. \tag{53}$$

Lemma 2 (Wang 1994)**:** *Suppose that the adaptive laws are chosen as* (56)–(58), *where* $\mathbf{Pr}(\cdot)$ *is the projection operator and the symmetric positive* $\boldsymbol{P}$ *satisfies the following Riccati-like equation*

$$\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} + \mathbf{Q} + \mathbf{P}\mathbf{b}\left(\frac{1}{\rho^2} - \frac{1}{\delta}\right)\mathbf{b}^T\mathbf{P} = 0, \tag{54}$$

where $\mathbf{Q}$ *is a positive definite symmetric matrix and* ρ *is an attenuation level which satisfies* $(1/\rho^2) - (1/\delta) \leq 0$. *If* $\hat{\boldsymbol{\alpha}}(0) \in \boldsymbol{\Omega}_{\boldsymbol{\alpha}}$, $\hat{\mathbf{c}}(0) \in \boldsymbol{\Omega}_{\mathbf{c}}$ *and* $\hat{\boldsymbol{\sigma}}(0) \in \boldsymbol{\Omega}_{\boldsymbol{\sigma}}$, *then* $\hat{\boldsymbol{\alpha}}(t) \in \boldsymbol{\Omega}_{\boldsymbol{\alpha}}$, $\hat{\mathbf{c}}(t) \in \boldsymbol{\Omega}_{\mathbf{c}}$ *and* $\hat{\boldsymbol{\sigma}}(t) \in \boldsymbol{\Omega}_{\boldsymbol{\sigma}}$ *for all* $t \geq 0$ *can be guaranteed.*

According to Lemma 2, $\boldsymbol{\Gamma}$ in (52) is bounded, and hence the lumped uncertainty ε is bounded. The following theorem shows the properties of the developed control system.

Theorem 1: *Suppose the assumption* (3) *holds. Consider a SISO nonaffine, nonlinear system* (1) *with the control law* (28), *where the self-structuring fuzzy controller is given as*

$$u_{fc} = \hat{\boldsymbol{\alpha}}^T \boldsymbol{\xi}(\mathbf{X}, \hat{\mathbf{c}}, \hat{\boldsymbol{\sigma}}). \tag{55}$$

The adaptive laws are chosen as (56)–(58):

$$\dot{\hat{\boldsymbol{\alpha}}} = -\dot{\tilde{\boldsymbol{\alpha}}} = \begin{cases} -\eta_{\alpha}\mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\xi}}, \\ \quad \text{if } \|\hat{\boldsymbol{\alpha}}\| < M_{\boldsymbol{\alpha}} \text{ or } \left(\|\hat{\boldsymbol{\alpha}}\| = M_{\boldsymbol{\alpha}} \text{ and } \mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\alpha}}^T\hat{\boldsymbol{\xi}} \geq 0\right) \\ \mathbf{Pr}(\eta_{\alpha}\mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\xi}}), \\ \quad \text{if } (\|\hat{\boldsymbol{\alpha}}\| = M_{\boldsymbol{\alpha}} \text{ and } \mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\alpha}}^T\hat{\boldsymbol{\xi}} < 0) \end{cases}, \tag{56}$$

where $\eta_{\boldsymbol{\alpha}}$ *is the positive learning rate and* $\mathbf{Pr}(\eta_{\alpha}\mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\xi}}) = -\eta_{\alpha}\mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\xi}} + \eta_{\alpha}\mathbf{e}^T\mathbf{Pb}\frac{\hat{\boldsymbol{\alpha}}^T\hat{\boldsymbol{\xi}}}{\|\hat{\boldsymbol{\alpha}}\|^2}\hat{\boldsymbol{\alpha}}$.

$$\dot{\hat{\mathbf{c}}} = -\dot{\tilde{\mathbf{c}}} = \begin{cases} -\eta_c\mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\xi}}_{\mathbf{c}}\hat{\boldsymbol{\alpha}}, \\ \quad \text{if } \|\hat{\mathbf{c}}\| < M_{\mathbf{c}} \text{ or } \left(\|\hat{\mathbf{c}}\| = M_{\mathbf{c}} \text{ and } \mathbf{e}^T\mathbf{Pb}\hat{\mathbf{c}}^T\boldsymbol{\xi}_{\mathbf{c}}\hat{\boldsymbol{\alpha}} \geq 0\right) \\ \mathbf{Pr}(\eta_c\mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\xi}}_{\mathbf{c}}\hat{\boldsymbol{\alpha}}), \\ \quad \text{if } \left(\|\hat{\mathbf{c}}\| = M_{\mathbf{c}} \text{ and } \mathbf{e}^T\mathbf{Pb}\hat{\mathbf{c}}^T\boldsymbol{\xi}_{\mathbf{c}}\hat{\boldsymbol{\alpha}} < 0\right) \end{cases}, \tag{57}$$

where $\eta_{\mathbf{c}}$ *is positive learning rate and*

$$\mathbf{Pr}(\eta_{\mathbf{c}}\mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\xi}}_{\mathbf{c}}\hat{\boldsymbol{\alpha}}) = -\eta_{\mathbf{c}}\mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\xi}}_{\mathbf{c}}\hat{\boldsymbol{\alpha}} + \eta_{\mathbf{c}}\mathbf{e}^T\mathbf{Pb}\frac{\hat{\mathbf{c}}^T\boldsymbol{\xi}_{\mathbf{c}}\hat{\boldsymbol{\alpha}}}{\|\hat{\mathbf{c}}\|^2}\hat{\mathbf{c}}.$$

$$\dot{\hat{\boldsymbol{\sigma}}} = -\dot{\tilde{\boldsymbol{\sigma}}} = \begin{cases} -\eta_{\boldsymbol{\sigma}}\mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\xi}}_{\boldsymbol{\sigma}}\hat{\boldsymbol{\alpha}}, \\ \quad \text{if } \|\hat{\boldsymbol{\sigma}}\| < M_{\boldsymbol{\sigma}} \text{ or } (\|\hat{\boldsymbol{\sigma}}\| = M_{\boldsymbol{\sigma}} \text{ and } \mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\sigma}}^T\boldsymbol{\xi}_{\boldsymbol{\sigma}}\hat{\boldsymbol{\alpha}} \geq 0) \\ \mathbf{Pr}(\eta_{\boldsymbol{\sigma}}\mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\xi}}_{\boldsymbol{\sigma}}\hat{\boldsymbol{\alpha}}), \\ \quad \text{if } \left(\|\hat{\boldsymbol{\sigma}}\| = M_{\boldsymbol{\sigma}} \text{ and } \mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\sigma}}^T\boldsymbol{\xi}_{\boldsymbol{\sigma}}\hat{\boldsymbol{\alpha}} < 0\right) \end{cases}, \tag{58}$$

where $\eta_{\boldsymbol{\sigma}}$ *is positive learning rate and* $\mathbf{Pr}(\eta_{\boldsymbol{\sigma}}\mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\xi}}_{\boldsymbol{\sigma}}\hat{\boldsymbol{\alpha}}) = -\eta_{\boldsymbol{\sigma}}\mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\xi}}_{\boldsymbol{\sigma}}\hat{\boldsymbol{\alpha}} + \eta_{\boldsymbol{\sigma}}\mathbf{e}^T\mathbf{Pb}\frac{\hat{\boldsymbol{\sigma}}^T\boldsymbol{\xi}_{\boldsymbol{\sigma}}\hat{\boldsymbol{\alpha}}}{\|\boldsymbol{\sigma}\|^2}\hat{\boldsymbol{\sigma}}$.

The robust adaptive controller is given as

$$u_{rac} = u_{lc} + \frac{1}{2\delta}\mathbf{b}^T\mathbf{Pe}. \tag{59}$$

Note that since $\mathbf{A}$ is designed to be stable in (30) and $\mathbf{Q}$ in (54) is a positive definite symmetric matrix, therefore $\mathbf{P}$ must be a positive definite symmetric matrix. Then, the RASFC system can guarantee the global stability and robustness of the closed-loop system and achieve the following L_2 criterion (Wang, Chan, Hsu, and Lee 2002a; Hsu, Lin, and Chen 2005)

$$\frac{1}{2}\int_0^T \mathbf{e}^T\mathbf{Qe}\,\mathrm{d}t \leq \frac{1}{2}\mathbf{e}(0)^T\mathbf{Pe}(0) + \frac{\tilde{\boldsymbol{\alpha}}^T(0)\tilde{\boldsymbol{\alpha}}(0)}{2\eta_{\alpha}} + \frac{\tilde{\mathbf{c}}^T(0)\tilde{\mathbf{c}}(0)}{2\eta_c} + \frac{\tilde{\boldsymbol{\sigma}}(0)^T\tilde{\boldsymbol{\sigma}}(0)}{2\eta_{\sigma}} + \frac{\rho^2}{2}\int_0^T(\varepsilon + d)^2\mathrm{d}t \tag{60}$$

for $0 \leq T < \infty$, where $\mathbf{e}(0)$, $\tilde{\boldsymbol{\alpha}}(0)$, $\tilde{\mathbf{c}}(0)$ and $\tilde{\boldsymbol{\sigma}}(0)$ are the initial values of $\mathbf{e}$, $\tilde{\boldsymbol{\alpha}}$, $\tilde{\mathbf{c}}$ and $\tilde{\boldsymbol{\sigma}}$, respectively.

Proof: Define the Lyapunov function candidate as

$$V = \frac{1}{2}\mathbf{e}^T\mathbf{Pe} + \frac{1}{2\eta_{\alpha}}\tilde{\boldsymbol{\alpha}}^T\tilde{\boldsymbol{\alpha}} + \frac{1}{2\eta_c}\tilde{\mathbf{c}}^T\tilde{\mathbf{c}} + \frac{1}{2\eta_{\sigma}}\tilde{\boldsymbol{\sigma}}^T\tilde{\boldsymbol{\sigma}}. \tag{61}$$

Differentiating (61) with respect to time and using (53) yields

$$\begin{aligned}\dot{V} &= \frac{1}{2}\mathbf{e}^T\mathbf{P}\dot{\mathbf{e}} + \frac{1}{2}\dot{\mathbf{e}}^T\mathbf{Pe} + \frac{1}{\eta_{\alpha}}\tilde{\boldsymbol{\alpha}}^T\dot{\tilde{\boldsymbol{\alpha}}} + \frac{1}{\eta_c}\tilde{\mathbf{c}}^T\dot{\tilde{\mathbf{c}}} + \frac{1}{\eta_{\sigma}}\tilde{\boldsymbol{\sigma}}^T\dot{\tilde{\boldsymbol{\sigma}}} \\ &= \frac{1}{2}\mathbf{e}^T(\mathbf{A}^T\mathbf{P} + \mathbf{PA})e - \mathbf{e}^T\mathbf{Pb}[\tilde{\boldsymbol{\alpha}}^T\hat{\boldsymbol{\xi}} + \tilde{\mathbf{c}}^T\boldsymbol{\xi}_{\mathbf{c}}\hat{\boldsymbol{\alpha}} + \tilde{\boldsymbol{\sigma}}^T\boldsymbol{\xi}_{\boldsymbol{\sigma}}\hat{\boldsymbol{\alpha}} \\ &\quad + \varepsilon + d + (u_{rac} - u_{lc})] \\ &\quad + \frac{1}{\eta_{\alpha}}\tilde{\boldsymbol{\alpha}}^T\dot{\tilde{\boldsymbol{\alpha}}} + \frac{1}{\eta_c}\tilde{\mathbf{c}}^T\dot{\tilde{\mathbf{c}}} + \frac{1}{\eta_c}\tilde{\boldsymbol{\sigma}}^T\dot{\tilde{\boldsymbol{\sigma}}}.\end{aligned} \tag{62}$$

Substituting (59) into (62), we obtain

$$\begin{aligned}\dot{V} &= \frac{1}{2}\mathbf{e}^T\left(\mathbf{A}^T\mathbf{P} + \mathbf{PA} - \frac{1}{\delta}\mathbf{Pbb}^T\mathbf{P}\right)\mathbf{e} \\ &\quad - \mathbf{e}^T\mathbf{Pb}(\varepsilon + d) - G_{\boldsymbol{\alpha}} - G_{\mathbf{c}} - G_{\boldsymbol{\sigma}}\end{aligned} \tag{63}$$

where $G_{\boldsymbol{\alpha}} = \tilde{\boldsymbol{\alpha}}^T(\mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\xi}} - \frac{\dot{\hat{\boldsymbol{\alpha}}}}{\eta_{\alpha}})$, $G_{\mathbf{c}} = \tilde{\mathbf{c}}^T(\mathbf{e}^T\mathbf{Pb}\xi_c\hat{\boldsymbol{\alpha}} - \frac{\dot{\hat{\mathbf{c}}}}{\eta_c})$ and $G_{\boldsymbol{\sigma}} = \tilde{\boldsymbol{\sigma}}^T(\mathbf{e}^T\mathbf{Pb}\xi_{\sigma}\hat{\boldsymbol{\alpha}} - \frac{\dot{\hat{\boldsymbol{\sigma}}}}{\eta_{\sigma}})$. By using (54), we can rewrite (63) as

$$\begin{aligned}\dot{V} &= \frac{1}{2}\mathbf{e}^T\left(-\mathbf{Q} - \frac{1}{\rho^2}\mathbf{Pbb}^T\mathbf{P}\right)\mathbf{e} - \mathbf{e}^T\mathbf{Pb}(\varepsilon + d) \\ &\quad - G_{\boldsymbol{\alpha}} - G_{\mathbf{c}} - G_{\boldsymbol{\sigma}} \\ &= -\frac{1}{2}\mathbf{e}^T\mathbf{Qe} - \frac{1}{2}\left[\frac{1}{\rho}\mathbf{b}^T\mathbf{Pe} + \rho(\varepsilon + d)\right]^2 + \frac{1}{2}\rho^2(\varepsilon + d)^2 \\ &\quad - G_{\boldsymbol{\alpha}} - G_{\mathbf{c}} - G_{\boldsymbol{\sigma}}.\end{aligned} \tag{64}$$

By using (56), we have $G_{\boldsymbol{\alpha}} = 0$ for $[\|\hat{\boldsymbol{\alpha}}\| \leq M_{\boldsymbol{\alpha}}$ or $(\|\hat{\boldsymbol{\alpha}}\| = M_{\boldsymbol{\alpha}}$ and $\mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\alpha}}^T\hat{\boldsymbol{\xi}} \geq 0)]$. For $[(\|\hat{\boldsymbol{\alpha}}\| = M_{\alpha}$ and $\mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\alpha}}^T\hat{\boldsymbol{\xi}} < 0)]$, we have

$$G_{\boldsymbol{\alpha}} = \eta_{\alpha}\mathbf{e}^T\mathbf{Pb}\frac{\tilde{\boldsymbol{\alpha}}^T\hat{\boldsymbol{\alpha}}}{\|\hat{\boldsymbol{\alpha}}\|^2}\hat{\boldsymbol{\alpha}}^T\hat{\boldsymbol{\xi}}. \tag{65}$$

Because $\boldsymbol{\alpha}^*$ belongs to the constraint set $\boldsymbol{\Omega}_{\alpha}$, we have $\|\hat{\boldsymbol{\alpha}}\| = M_{\boldsymbol{\alpha}} \geq \|\boldsymbol{\alpha}^*\|$. Using this fact, we obtain $\tilde{\boldsymbol{\alpha}}^T\hat{\boldsymbol{\alpha}} = \frac{1}{2}(\|\boldsymbol{\alpha}^*\|^2 - \|\hat{\boldsymbol{\alpha}}\|^2 - \|\tilde{\boldsymbol{\alpha}}\|^2) \leq 0$. Thus, (65) can be rewritten as

$$G_{\boldsymbol{\alpha}} = \frac{\eta_{\alpha}}{2}\mathbf{e}^T\mathbf{Pb}\frac{\left(\|\boldsymbol{\alpha}^*\|^2 - \|\hat{\boldsymbol{\alpha}}\|^2 - \|\tilde{\boldsymbol{\alpha}}\|^2\right)}{\|\hat{\boldsymbol{\alpha}}\|^2}\hat{\boldsymbol{\alpha}}^T\hat{\boldsymbol{\xi}} \geq 0. \tag{66}$$

Similarly, we have (67) and (68) by using (57) and (58), respectively.

$$G_{\mathbf{c}} = \begin{cases} 0 \\ \quad \text{if } \|\hat{\mathbf{c}}\| < M_{\mathbf{c}} \text{ or } \left(\|\hat{\mathbf{c}}\| = M_{\mathbf{c}} \text{ and } \mathbf{e}^T\mathbf{Pb}\hat{\mathbf{c}}^T\boldsymbol{\xi}_{\mathbf{c}}\hat{\boldsymbol{\alpha}} \geq 0\right) \\ \frac{\eta_{\mathbf{c}}}{2}\mathbf{e}^T\mathbf{Pb}\frac{\left(\|\mathbf{c}^*\|^2 - \|\hat{\mathbf{c}}\|^2 - \|\tilde{\mathbf{c}}\|^2\right)}{\|\hat{\mathbf{c}}\|^2}\hat{\mathbf{c}}^T\boldsymbol{\xi}_{\mathbf{c}}\hat{\boldsymbol{\alpha}} \geq 0 \\ \quad \text{if } \left(\|\hat{\mathbf{c}}\| = M_{\mathbf{c}} \text{ and } \mathbf{e}^T\mathbf{Pb}\hat{\mathbf{c}}^T\boldsymbol{\xi}_{\mathbf{c}}\hat{\boldsymbol{\alpha}} < 0\right), \end{cases} \tag{67}$$

$$G_{\boldsymbol{\sigma}} = \begin{cases} 0 \\ \quad \text{if } \|\hat{\boldsymbol{\sigma}}\| < M_{\boldsymbol{\sigma}} \text{ or } (\|\hat{\boldsymbol{\sigma}}\| = M_{\boldsymbol{\sigma}} \text{ and } \mathbf{e}^T\mathbf{P}\mathbf{b}\hat{\boldsymbol{\sigma}}^T\boldsymbol{\xi}_{\boldsymbol{\sigma}}\hat{\boldsymbol{\alpha}} \geq 0) \\ \frac{\eta_{\boldsymbol{\sigma}}}{2}\mathbf{e}^T\mathbf{P}\mathbf{b}\frac{\left(\|\boldsymbol{\sigma}^*\|^2 - \|\hat{\boldsymbol{\sigma}}\|^2 - \|\tilde{\boldsymbol{\sigma}}\|^2\right)}{\|\hat{\boldsymbol{\sigma}}\|^2}\hat{\boldsymbol{\sigma}}^T\boldsymbol{\xi}_{\boldsymbol{\sigma}}\hat{\boldsymbol{\alpha}} \geq 0 \\ \quad \text{if } (\|\hat{\boldsymbol{\sigma}}\| = M_{\boldsymbol{\sigma}} \text{ and } \mathbf{e}^T\mathbf{P}\mathbf{b}\hat{\boldsymbol{\sigma}}^T\boldsymbol{\xi}_{\boldsymbol{\sigma}}\hat{\boldsymbol{\alpha}} < 0). \end{cases} \tag{68}$$

Consequently, for any possible condition in (56)–(58), $G_{\boldsymbol{\alpha}} \geq 0$, $G_{\mathbf{c}} \geq 0$ and $G_{\boldsymbol{\sigma}} \geq 0$ are satisfied. Thus, we can rewrite (64) as

$$\dot{V} \leq -\frac{1}{2}\mathbf{e}^T\mathbf{Q}\mathbf{e} + \frac{1}{2}\rho^2(\varepsilon + d)^2. \tag{69}$$

Assume that there exists a finite constant γ so that (Wang, Lin, Lee, and Liu 2002b)

$$\int_0^T (\varepsilon + d)^2 \mathrm{d}t \leq \gamma, \quad \forall T \in [0, \infty), \tag{70}$$

i.e. $(\varepsilon + d) \in L_2[0, T]$, $\forall T \in [0, \infty)$. Integrating both sides of the inequality (69) yields

$$V(T) - V(0) \leq -\frac{1}{2}\int_0^T \mathbf{e}^T\mathbf{Q}\mathbf{e}\,\mathrm{d}t + \frac{\rho^2}{2}\int_0^T (\varepsilon + d)^2 \mathrm{d}t, \quad 0 \leq T < \infty. \tag{71}$$

Since $V(T) \geq 0$, the following L_2 criterion can be obtained.

$$\frac{1}{2}\int_0^T \mathbf{e}^T\mathbf{Q}\mathbf{e}\,\mathrm{d}t \leq V(0) + \frac{\rho^2}{2}\int_0^T (\varepsilon + d)^2 \mathrm{d}t, \quad 0 \leq T < \infty. \tag{72}$$

Substituting (61) into (72), we have the L_2 criterion shown in (60). This completes the proof. □

From (72), we can see that because $V(0)$ is finite, the effect of lumped uncertainty and external disturbance on tracking error can be eliminated as small as possible by choosing an arbitrarily small attenuation level ρ. In other words, a smaller ρ results in smaller tracking error, which implies better tracking performance. The following Theorem 2 will present an explicit formulation of tracking error.

Theorem 2: *The tracking error $\|\mathbf{e}\|$ can be expressed in terms of the sum of lumped uncertainty and external disturbance as*

$$\|\mathbf{e}\| \leq \sqrt{\frac{2V(0) + \rho^2\gamma}{\lambda_{\min}(\mathbf{P})}}. \tag{73}$$

Proof: From (71), with the knowledge $\int_0^T \mathbf{e}^T\mathbf{Q}\mathbf{e}\,\mathrm{d}t \geq 0$ and assumption (70), we have

$$2V(T) \leq 2V(0) + \rho^2\gamma, \quad 0 \leq T < \infty. \tag{74}$$

From (61), it is obvious that $\mathbf{e}^T\mathbf{P}\mathbf{e} \leq 2V$ for any V. Because $\mathbf{P}$ is a positive definite symmetric matrix, we have

$$\lambda_{\min}(\mathbf{P})\|\mathbf{e}\|^2 = \lambda_{\min}(\mathbf{P})\mathbf{e}^T\mathbf{e} \leq \mathbf{e}^T\mathbf{P}\mathbf{e}, \tag{75}$$

where $\lambda_{\min}(\mathbf{P})$ is the minimum eigenvalue of $\mathbf{P}$. Thus, we obtain

$$\lambda_{\min}(\mathbf{P})\|\mathbf{e}\|^2 \leq \mathbf{e}^T\mathbf{P}\mathbf{e} \leq 2V(T) \leq 2V(0) + \rho^2\gamma \tag{76}$$

from (74)–(75). Therefore, (76) can be rearranged to yield the following important formula:

$$\|\mathbf{e}\| \leq \sqrt{\frac{2V(0) + \rho^2\gamma}{\lambda_{\min}(\mathbf{P})}}, \tag{77}$$

which explicitly describes the tracking error $\|\mathbf{e}\|$ in terms of the sum of lumped uncertainty and external disturbance. □

If initial state $V(0) = 0$, tracking error $\|\mathbf{e}\|$ can be made arbitrarily small by choosing adequate ρ. Unlike the results in Park and Kim (2005) and Labiod and Guerra (2007), (77) is very crucial to show that the proposed RASFC will provide the closed-loop stability rigorously in the Lyapunov sense.

Remark 4: Consider an SISO nonlinear affine system

$$x^{(n)} = F(\mathbf{x}) + G(\mathbf{x})u + d, \tag{78}$$

where $\mathbf{x} = [x \quad \dot{x} \quad \cdots \quad x^{(n-1)}]^T$ is the state vector of the system, $F(\mathbf{x})$ and $G(\mathbf{x})$ are unknown nonlinear mapping, u is the control input of the system and d is a bounded external disturbance. By letting $f(\mathbf{x}, u) = F(\mathbf{x}) + G(\mathbf{x})u$, we can easily find that the nonlinear affine system (78) can be viewed as a special case of nonaffine, nonlinear system (1). Thus, the proposed RASFC scheme can be directly applied to such a nonlinear affine system when necessary assumptions hold. The overall RASFC can be shown in Figure 4.

5. Simulation results

In this section, the simulations are performed using MATLAB under Windows XP. Five examples are presented. Approximations of unknown nonlinear functions are shown in Examples 1 and 2 to reveal the growing and pruning capabilities of the proposed self-structuring algorithm, respectively. Examples 3–5 are used to examine the applicability and effectiveness of the proposed RASFC system for nonaffine, nonlinear control problems. For comparison purpose, two cases are performed in Examples 3–5, respectively. Cases 3a, 4a and 5a show the effectiveness of the SFS

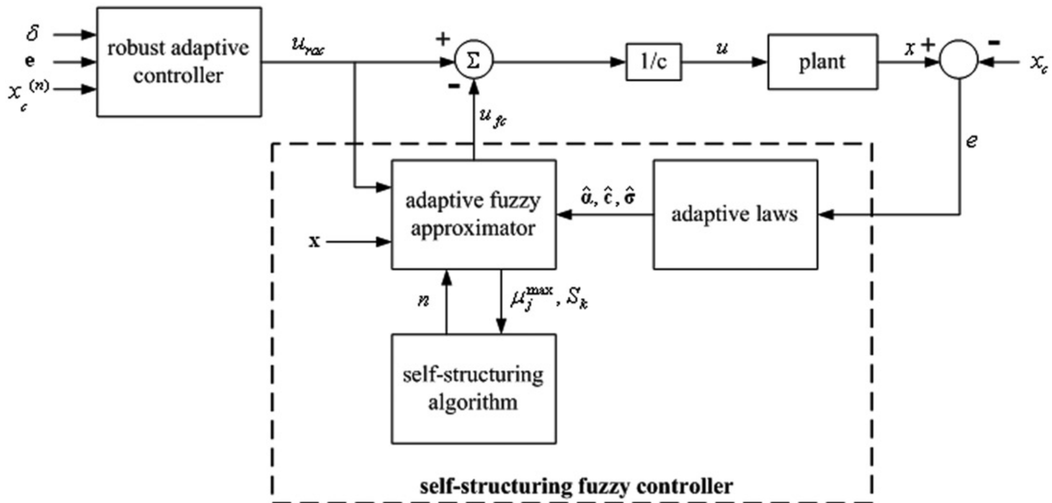

Figure 4. The block diagram of RASFC for nonaffine nonlinear systems.

with both rules growing and pruning capabilities. In Case 3b, an adaptive FS with fixed number of rules is adapted, and the parameters of the FS are also tuned by adaptive laws (56)–(58). In Cases 4b and 5b, only the growing of fuzzy rules by SFS is considered. It can be easily shown that the following examples of nonaffine system control satisfy $\partial f(\mathbf{x}, u)/\partial u > 0$. It should be emphasised that the development of the RASFC does not need to know the exact system dynamics of the controlled systems.

Example 1: Consider the following nonaffine, nonlinear system (Ge, Hang, and Zhang 1999):

$$\dot{x}_1 = x_2,$$

$$\dot{x}_2 = x_1^2 + 0.15u^3 + 0.1(1 + x_2^2)u + \sin(0.1u). \qquad (79)$$

In tracking control, the SFS is used to approximate an unknown function $\Delta(\mathbf{x}, u) = x_1^2 + 0.15u^3 + 0.1(1 + x_2^2)u + \sin(0.1u) - cu$. To illustrate the rule growing capability of the self-structuring algorithm, the approximation is performed under three conditions as shown in Table 1. Figure 5(a)–(c) shows the approximation results of Condition 1a, 1b and 1c, respectively, Figure 5(d) shows the absolute value of the modelling error, $|\tilde{u}|$ and Figure 5(e) shows the number of fuzzy rules. The approximation performances under Conditions 1a and 1b are better than that under Condition 1a after $t \geq 5$. In Figure 5(b), the abrupt variations are marked by circles. These abrupt variations are obviously caused by the rule generation so that the approximation performance is affected for a short period. In Figure 5(c), this phenomenon is mitigated by using (27) discussed in Remark 1. From Figure 5(d), we can see the approximation performance under Condition 1c is the best among three conditions.

Table 1. Three conditions in Example 1.

Desired trajectory of tracking control: $x_c = \sin(1.5t)$

	Number of rules	Consequents of newly generated fuzzy rules
Condition 1a	Fixed (4 rules)	
Condition 1b	$t < 5$: the same 4 rules in Condition 1a are used $t \geqq 5$: rule growing is operated	Initialised from zeros
Condition 1c	$t < 5$: the same 4 rules in Condition 1a are used $t \geqq 5$: rule growing is operated	Initialised according to (27)

Example 2: A third-order Chua's chaotic circuit is a simple electronic system that consists of one linear resistor (R_c), two capacitors (C_1, C_2), one inductor (L) and one nonlinear resistor (η). It has been shown to own very rich nonlinear dynamics such as chaos and bifurcations. The dynamic equations of Chua's circuit are written as (Wang et al. 2002b; Hsu, Chen, and Lee 2007)

$$\begin{aligned}\dot{v}_{C_1} &= \frac{1}{C_1}\left[\frac{1}{R}(v_{C_2} - v_{C_1}) - \eta(v_{C_2})\right]\\ \dot{v}_{C_2} &= \frac{1}{C_2}\left[\frac{1}{R}(v_{C_1} - v_{C_2}) + i_L\right]\\ \dot{i}_L &= \frac{1}{L}(-v_{C_1} - R_0 i_L),\end{aligned} \qquad (80)$$

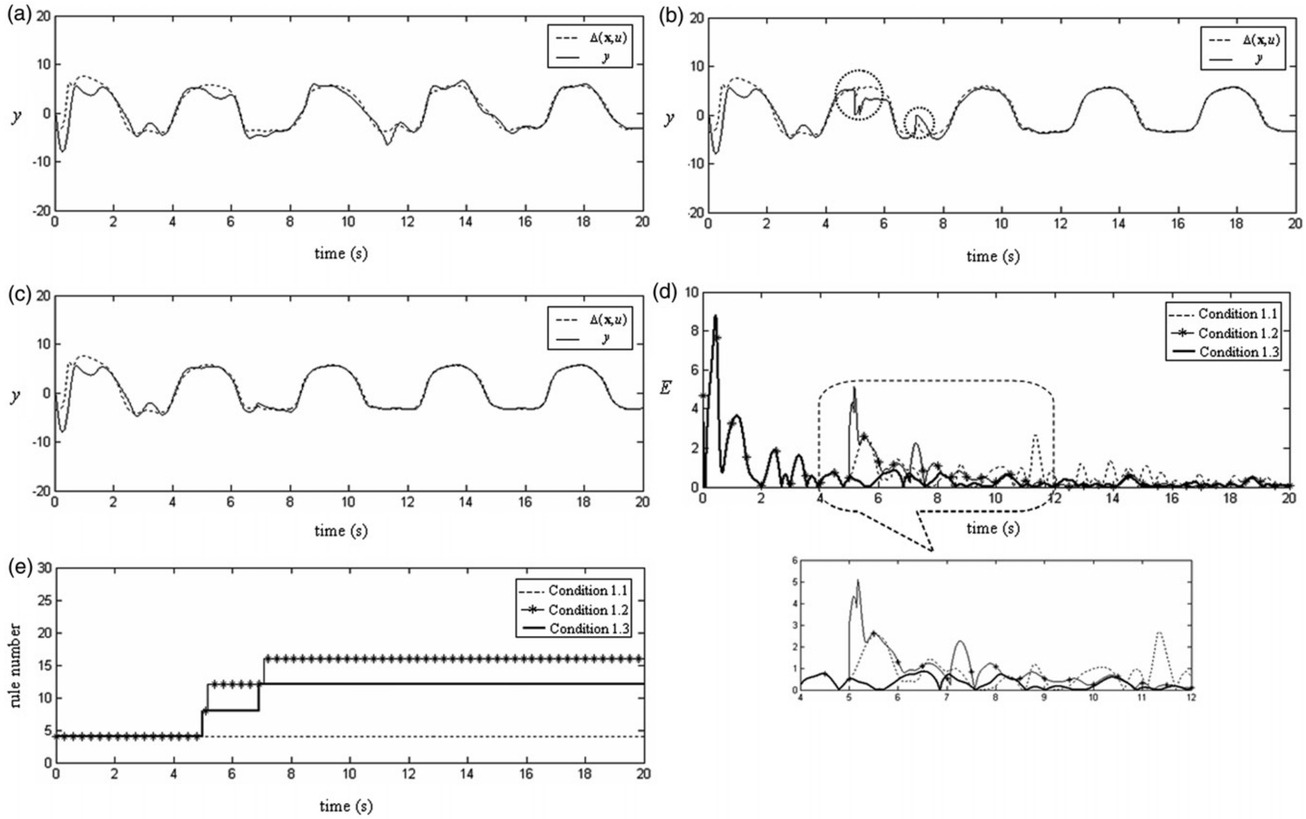

Figure 5. Approximation results in Example 1.

where the voltages v_{C_1}, v_{C_2} and current i_L are state variables, R_0 is a constant, and η denotes the nonlinear resistor, which is a function of the voltage across the two terminals of C_1. Here, ϕ is defined as a cubic function as

$$\phi = \lambda_1 v_{C_1} + \lambda_2 v_{C_1}^3 \quad (\lambda_1 < 0, \lambda_2 > 0). \tag{81}$$

The state equations in (80) are not in the standard canonical form. Therefore, a linear transformation is needed to transform them into the form of (14). Then, the dynamic equations of transformed Chua's circuit can be rewritten as

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= x_3, \\ \dot{x}_3 &= F + u, \\ y &= x_1, \end{aligned} \tag{82}$$

where $\mathbf{x} = [x_1 \quad x_2 \quad x_3]^T$ is the state vector of the system which is assumed to be available; the system dynamic function

$$F = \frac{14}{1805}x_1 - \frac{168}{9025}x_2 + \frac{1}{38}x_3 - \frac{2}{45}\left(\frac{28}{361}x_1 + \frac{7}{95}x_2 + x_3\right) \tag{83}$$

Table 2. Two conditions in Example 2.

Desired trajectory of tracking control: $x_c = 1.5\sin(t)$	
	Rule number
Condition 2a	Fixed (40 rules)
Condition 2b	$t \geqq 0$: rule pruning is operated

and u is the control input. The reference signal is $y_r(t) = 1.5\sin(t)$. In tracking control, the SFS is used to approximate an unknown function $\Delta(\mathbf{x}, u) = F + u - cu$. To illustrate the rule pruning of the self-structuring algorithm, the approximation is performed under two conditions as shown in Table 2. Figure 6(a)–(b) shows the approximation results. Figure 6(c) shows the approximation error E. Figure 6(d) shows the number of fuzzy rules. Taking the last pruned rule for example, we record the contribution and significance index of the rule pruned at $t = 2.28$ in Figure 6(e). Figure 6(a)–(c) shows that the approximation performances of Conditions 2a and 2b are both quit well. However, the convergence speed of approximation error E under

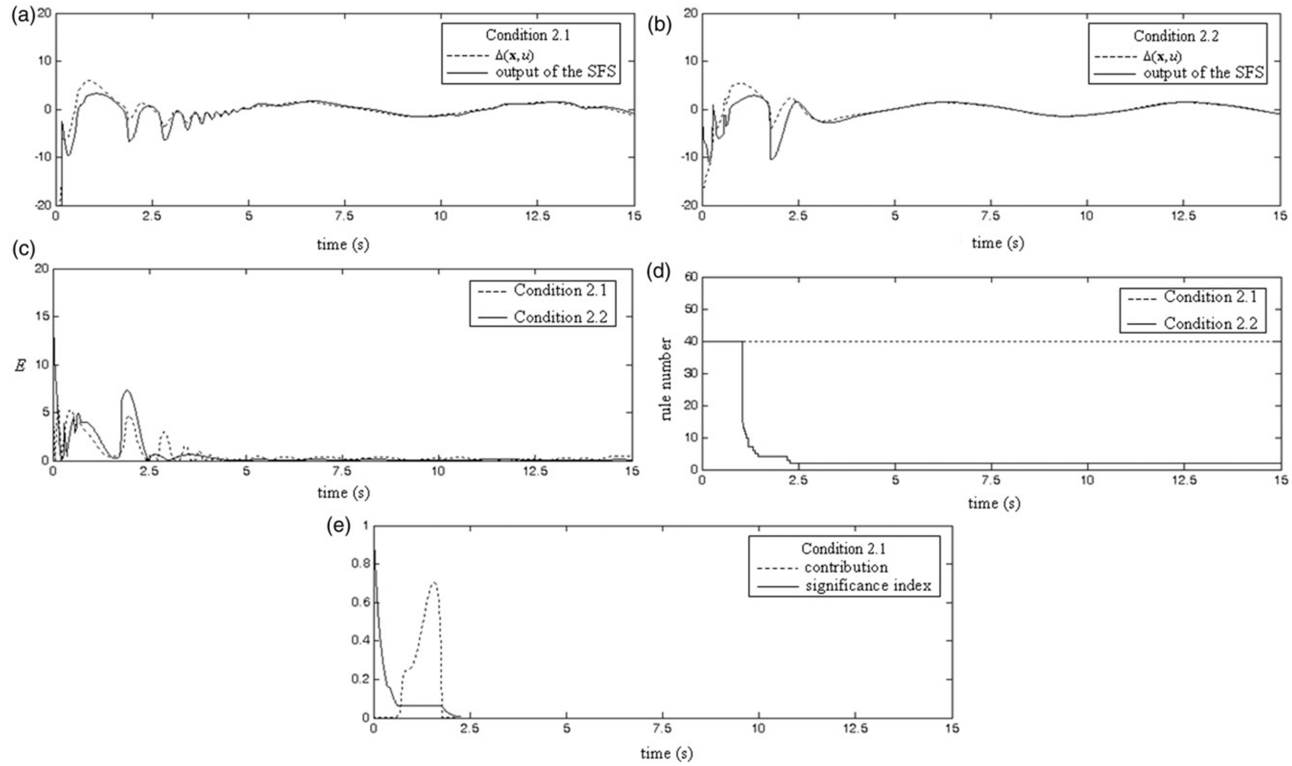

Figure 6. Approximation results in Example 2.

Condition 2b is faster than that of Condition 2a. This shows that the parameter training of a large number of fuzzy rules slow down the approximation convergence, and the pruned rules under Condition 2b are redundant and ineffective to the approximation performance. In Figure 6(e), we show the contribution and significance index of a certain rule pruned at $t=2.28$. When the contribution calculated by (17) is smaller than a given constant $\beta=0.005$, the significance index (18) decays with decay constant $\tau=0.99$. Once the significance index is smaller than the pruning threshold $\Theta_p=0.005$ at $t=2.28$, this rule is insignificant thereafter and thus pruned to ease computational load.

Example 3: Consider the following nonaffine, nonlinear system (Leu, Wang, and Lee 2005):

$$\begin{aligned}\dot{x}_1 &= x_2,\\ \dot{x}_2 &= 0.2(1+e^{x_1x_2})[(2+\sin(x_2)](u+e^u-1)+d,\end{aligned} \tag{84}$$

where d is a square wave with amplitude ± 3.0 and period 5 s. The desired trajectory is $x_d(t)=\sin(0.5t)+\cos(t)$. The initial sates are chosen as $\mathbf{x}(0)=\begin{bmatrix}x_1(0) & x_2(0)\end{bmatrix}=\begin{bmatrix}0 & 0\end{bmatrix}^T$. The learning rates are selected as $\eta_{\boldsymbol{\alpha}}=120$ and $\eta_{\mathbf{c}}=\eta_{\boldsymbol{\sigma}}=1$. The thresholds for growing and pruning criteria in Case 3a are selected as $\Theta_g=0.1$ and $\Theta_p=0.01$, respectively. These parameters are chosen through some trials to achieve favourable transient control performance. For a choice of $\mathbf{Q}=2\mathbf{I}$, $\mathbf{K}=[2 \quad 1]^T$ and $\rho^2=\delta$, we solve the Riccati-like equation shown in (62) and obtain a positive definite symmetric matrix $\mathbf{P}$:

$$\mathbf{P}=\begin{bmatrix}3.5 & 0.5\\ 0.5 & 1.5\end{bmatrix}. \tag{85}$$

The simulation results for Cases 3a and 3b are shown in Figures 7 and 8, respectively. The tracking responses of state x_1 are shown in Figures 7(a) and 8(a), the tracking responses of state x_2 are shown in Figures 7(b) and 8(b), the associated control inputs are shown Figures 7(c) and 8(c) and the numbers of fuzzy rules at every iteration are shown in Figures 7(d) and 8(d). From Figure 7(a)–(b) to Figure 8(a)–(b), we can see that the tracking performance in Case 3a is better than that in Case 3b under the external disturbance. In Figure 7(d), the maximum number of rules is 7; in Figure 8(d), the number of rules is 4. Table 3 shows the comparison between the two cases, where N_a represents the accumulated sum of computed rules and t_e denotes the total execution time during the simulation. The proposed self-structuring algorithm can relieve the heavy computational burden caused by 25,423 redundant rules (42.37% of the N_a in Case 3b) and the t_e in Case 3a is nearly one-half times faster than that in Case 3b.

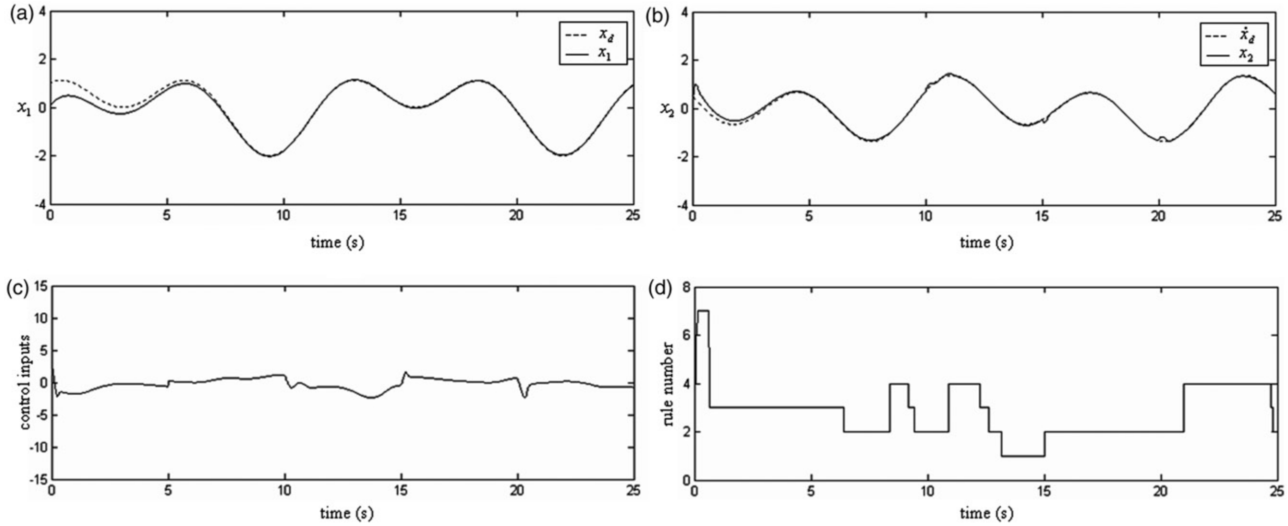

Figure 7. Simulation results of Case 3a in Example 3.

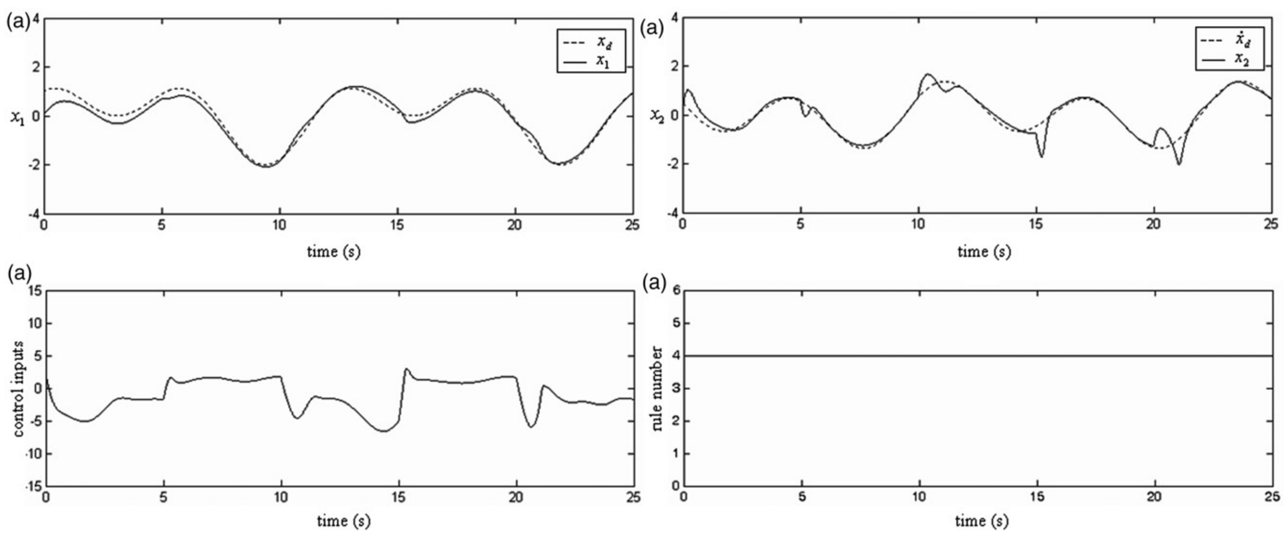

Figure 8. Simulation results of Case 3b in Example 3.

Table 3. Comparison between two cases in Example 3.

1.25×10^4 iterations	Case 3a	Case 3b
Maximum number of rules at any time instant	7	4 (fixed)
Accumulated sum of rule number, N_a	34,577	60,000
Total execution time, t_e(s)	12.88	18.14

Example 4: The Van der Pol oscillator is the main model of self-oscillatory system with two-dimensional phase space (Wang and Krstic 2000; Pourhiet et al. 2003; Mahmoud and Farghaly 2004). The oscillator and its extensions have been implemented in various types of electrical circuits. The nonaffine second-order Van der Pol oscillator with nonlinear damping is described as (Karimi, Menhaj, and Saboori 2006)

$$\dot{x}_1 = x_2,$$
$$\dot{x}_2 = -x_1 + x_2 + u + (x_1^2 + x_2^2)\left(\frac{1+e^{-u}}{1-e^{-u}}\right) - x_1^2 x_2 + d, \tag{86}$$

where d is a white noise with power 2 which occurs after $t \geq 15$. The desired trajectory is $x_d(t) = \sin(t) + \cos(0.5t)$ and the initial state is $\mathbf{x}(0) = \begin{bmatrix} x_1(0) & x_2(0) \end{bmatrix} = \begin{bmatrix} 0.6 & 0.5 \end{bmatrix}^T$. All other parameter

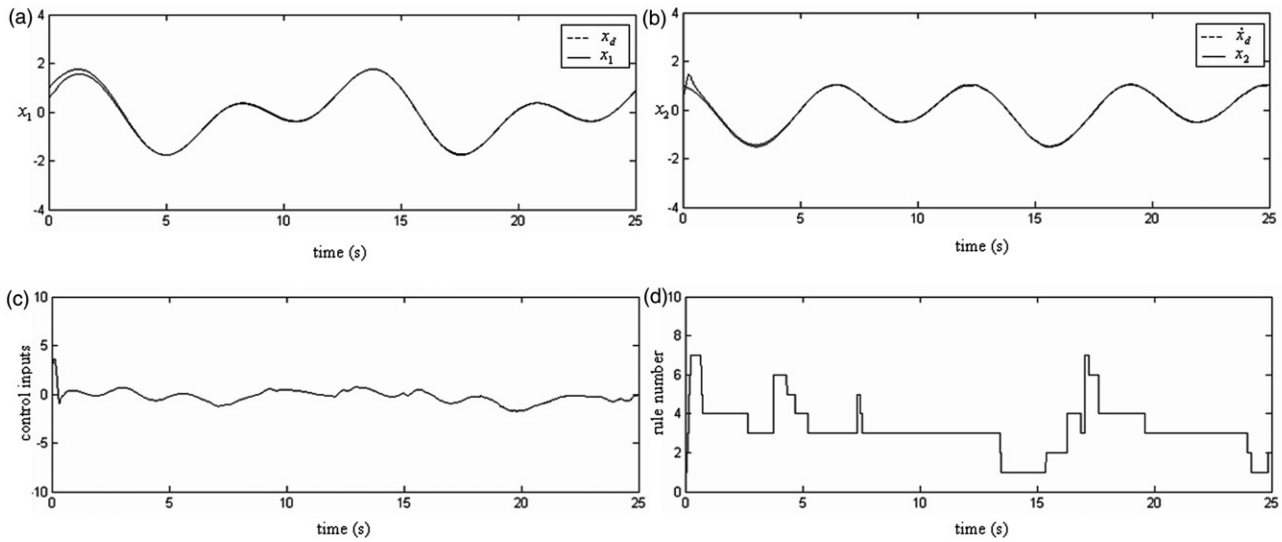

Figure 9. Simulation results of Case 4a in Example 4.

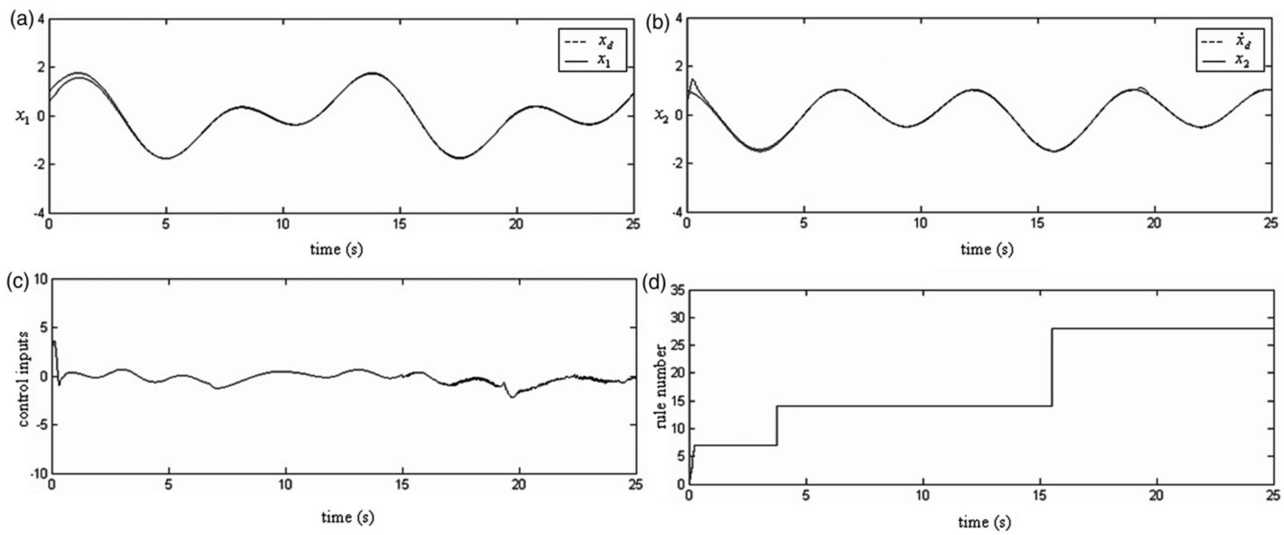

Figure 10. Simulation results of Case 4b in Example 4.

settings are chosen the same as those in Example 3. The simulation results for Cases 4a and 4b are shown in Figures 9 and 10, respectively. The tracking responses of state x_1 are shown in Figures 9(a) and 10(a), the tracking responses of state x_2 are shown in Figures 9(b) and 10(b), the associated control inputs are shown Figures 9(c) and 10(c) and the number of fuzzy rules at every iteration are shown in Figures 9(d) and 10(d). From the simulation results, we can see that the proposed RASFC scheme in Case 4a can achieve the same favourable tracking performance as that in Case 4b even if an external disturbance suddenly occurs. In Figure 9(d), rule growing plays the major role in SFS within $0 \leq t < 0.25$ and thus, the rule number is increased from one to produce a suitable control effort to suppress the tracking error. For $t > 0.25$, to reduce tracking error, the pruning of unnecessary rules will be activated in SFS and thus the number of rules decreases gradually. After a large external disturbance occurs at $t \geq 15$, the rule number apparently increases to eliminate the effect caused by the disturbance. When tracking error is again suppressed to a small level, the rule pruning effect will be activated again. In Figure 10(d), the number of rules increases very rapidly from the beginning to the end of control. Throughout the control process, the maximum number of rules is 7

Table 4. Comparison between two cases in Example 4.

1.25×10^4 iterations	Case 4a	Case 4b
Maximum number of rules at any time instant	7	28
Accumulated sum of computed fuzzy rules, N_a	39,973	227,650
Total execution time, t_e(s)	12.72	64.89

in Case 4a and 28 in Case 4b. Table 4 shows the comparison between two cases. From Table 4, it is obvious that our proposed self-structuring algorithm can relieve the heavy computational burden caused by 187,677 redundant rules (82.44% of N_a in Case 4b) and t_e in Case 4a is over five times faster than that in Case 4b. It can be imagined that the relief of computational load caused by the redundant rules will become more and more remarkable as the control period continues.

Example 5: Figure 11 shows a single-link manipulator with flexible joint and negligible dumping (Spong and Vidyasagar 1989). The dynamics can be described as

$$I\ddot{q}_1 + MgL\sin q_1 + K(q_1 - q_2) = 0, \\ J\ddot{q}_2 - k(q_1 - q_2) = u, \tag{87}$$

where q_1 and q_2 are the angular positions of the link and the motor, respectively, I and J are the moments of inertia, K is the stiffness constant, M is the total mass, L is the distance and u is the input torque. This system can be transformed into a fourth-order canonical form (5-2) through a global diffeomorphism (Khalil 2002) as

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= x_3, \\ \dot{x}_3 &= x_4, \\ \dot{x}_4 &= -\left(\frac{MgL}{I}\cos x_1 + \frac{K}{I} + \frac{K}{J}\right)x_3 \\ &\quad + \frac{MgL}{I}\left(x_2^2 - \frac{K}{J}\right)\sin x_1 + \frac{K}{IJ}u. \end{aligned} \tag{88}$$

To perform the simulation, the parameters are adopted as $MgL = 1$, $I = 0.008$, $J = 0.005$ and $k = 0.3$ (Corless and Zenieh 1995). The control object is to regulate to zero the angular positions and velocities of the manipulator. The initial states are chosen as $\mathbf{x}(0) = [x_1(0)\ x_2(0)\ x_3(0)\ x_4(0)] = [0.15\ 0.2\ 0.15\ 0.2]^T$. The learning rates are selected as $\eta_{\boldsymbol{\alpha}} = 120$ and $\eta_{\mathbf{c}} = \eta_{\boldsymbol{\sigma}} = 100$. The thresholds for growing and pruning criteria in Case 5a are selected as $\Theta_g = 0.0005$ and $\Theta_p = 0.01$, respectively. These parameters are chosen through some trials to achieve favourable transient control performance. For a choice of $\mathrm{Q} = 10\mathrm{I}$,

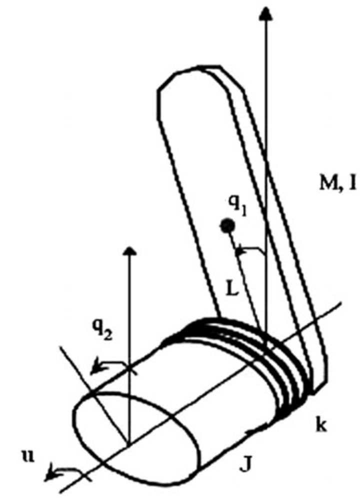

Figure 11. The single-link manipulator with flexible joint.

$\mathbf{K} = [9\ \ 28\ \ 38\ \ 4]^T$ and $\rho^2 = \delta$, we solve the Riccati-like equation shown in (62) and obtain a positive definite symmetric matrix $\mathbf{P}$:

$$\mathbf{P} = \begin{bmatrix} 52.3438 & 41.0156 & 14.1406 & 1.25 \\ 41.0156 & 76.9141 & 37.1094 & 1.2109 \\ 14.1406 & 37.1094 & 32.5586 & 1.5039 \\ 1.25 & 1.2109 & 1.5039 & 0.7227 \end{bmatrix}. \tag{89}$$

The simulation results for Cases 5a and 5b are shown in Figures 12 and 13, respectively. The tracking responses of the system states are shown in Figures 12(a)–(d) and 13(a)–(d), the associated control inputs are shown Figures 12(e) and 13(e) and the number of fuzzy rules at every iteration are shown in Figures 12(f) and 13(f). From the simulation results, we can see that that the proposed RASFC scheme in Case 5a can achieve the same favourable tracking performance as that in Case 5b. In Figure 12(f), we can see the number of rules rapidly increases from the beginning of regulation and then gradually decreases as the system states are regulated within the small neighbourhoods of zero. Throughout the control process, the maximum numbers of rules are 18 in Case 5a and 21 in Case 5b. Table 5 shows the comparison between two cases. Table 5 shows that heavy computational burden caused by 238,497 redundant rules (91.76% of N_a in Case 5b) is released and t_e in Case 5a is over five times faster than that in Case 5b.

It is shown from the simulation results that the proposed RASFC scheme can achieve satisfactory tracking performance for even high-order SISO non-affine and affine, nonlinear systems and in the mean while, release heavy computational burden. It is worth noting that in Examples 3–5, the control is started with only one fuzzy rule and thereafter a compact rule base

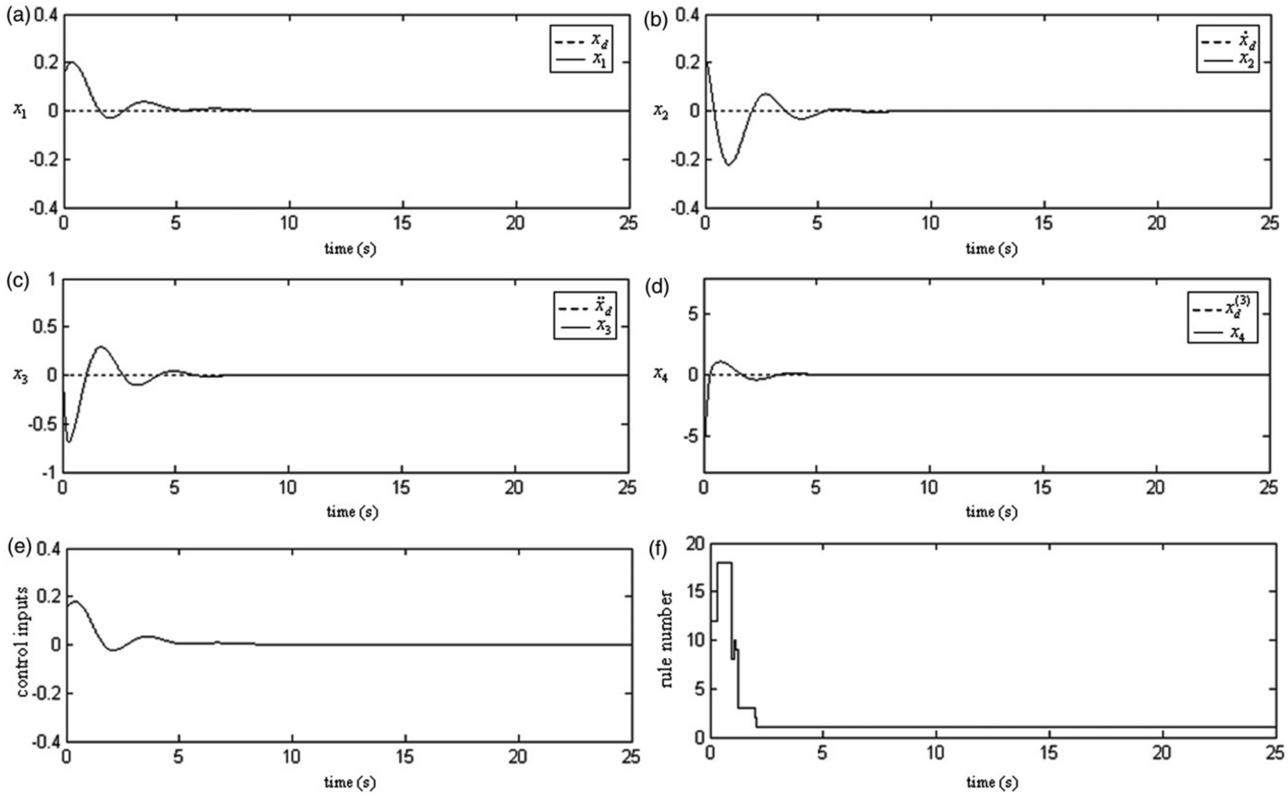

Figure 12. Simulation results of Case 5a in Example 5.

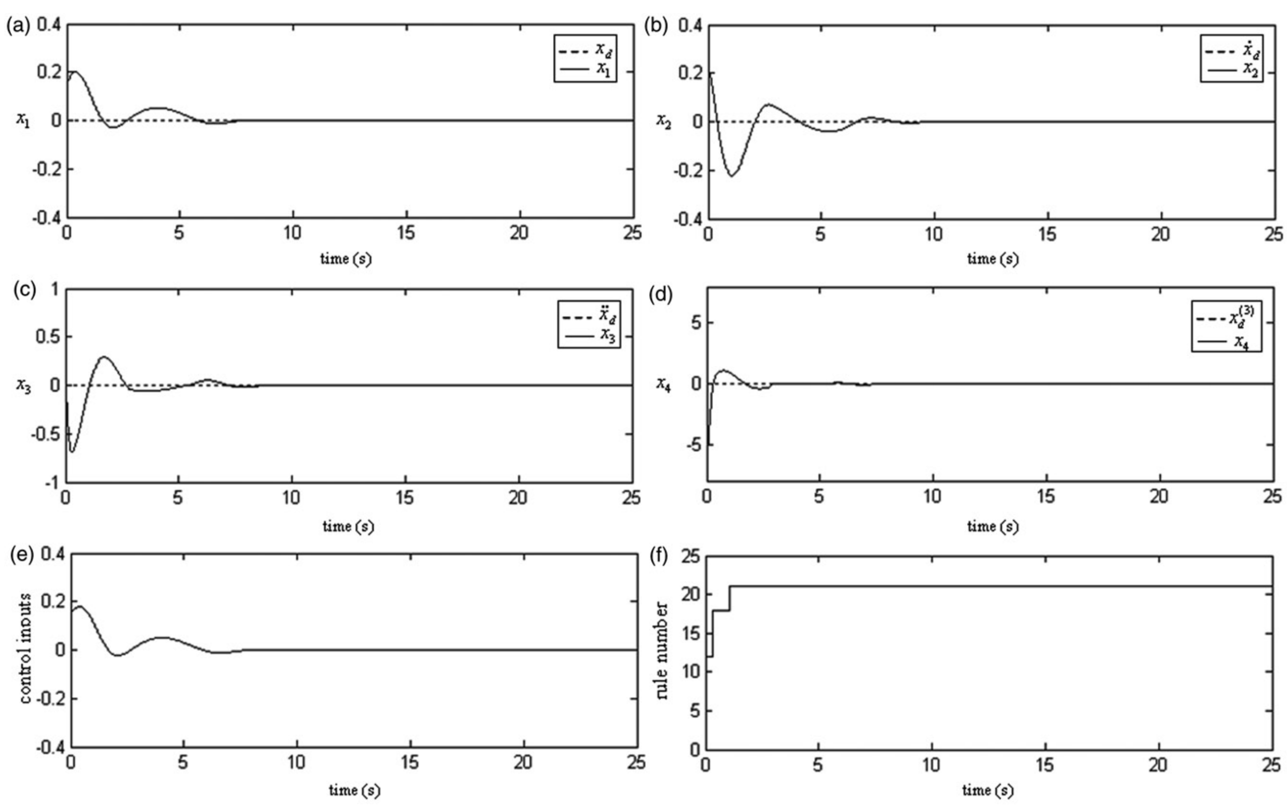

Figure 13. Simulation results of Case 5b in Example 5.

Table 5. Comparison between two cases in Example 5.

1.25×10^4 iterations	Case 5a	Case 5b
Maximum number of rules at any time instant	18	21
Accumulated sum of computed fuzzy rules, N_a	21,403	259,900
Total execution time, t_e(s)	9.08	45.56

is constructed automatically without human knowledge. Examples 4 and 5 show that without the rule pruning, fuzzy rules will grow to a very large number and thus lead to unacceptable computing burden.

6. Conclusion

Structure determination is a difficult task for practical implementations of FSs. More specifically, choosing the number of fuzzy rules, inherently involving fuzzy partitioning of input and output spaces, can greatly affect the performance of FSs. In this article, the proposed SFS can manage fuzzy rule base by automatic rule generation and pruning. The problems of determining the fuzzy partitions of input spaces and the number of fuzzy rules are solved simultaneously. The provided systematic method can cope with the tradeoff between the approximation accuracy and computational load of FS. New rules are generated according to the newly added membership functions to adjust the improper fuzzy clustering of the input spaces. Historically, insignificant rules with negligible contributions toward the output of FS will be removed. Comparing with the aforementioned self-evolving fuzzy/fuzzy neural systems developed in Angelov and Filev (2004) and Juang and Tsao (2008), the SFS proposed in this article has some valuable features: (1) the consequents of the newly generated rules are designed to maintain the approximation property of the SFS; (2) the rule growing strategy in nature has less chance to suffer from the problem of generating highly overlapping fuzzy sets, and hence remove the need of any fuzzy set reduction method and (3) the rule pruning strategy indeed lowers the computation load.

Further, a RASFC scheme for the uncertain or ill-defined nonlinear, nonaffine systems is proposed. Some adaptive laws for online tuning the parameters of fuzzy rules are derived in the Lyapunov sense to realise favourable fuzzy approximation. As shown in this article, the RASFC can achieve a L_2 tracking performance with arbitrarily attenuation level. This L_2 tracking performance can provide a clear expression of tracking error in terms of the sum of lumped uncertainty and external disturbance, which has not been shown in previous articles. Several examples are illustrated to show that the RASFC can achieve favourable tracking performance in the presence of external disturbance, yet heavy computational burden is relieved.

Acknowledgement

The authors appreciate the partial financial support from the National Science Council of Republic of China under grant NSC 92-2213-E-009-009.

Notes on contributors

Pin-Cheng Chen received his BS degree in Electrical and Control Engineering from the Chiao Tung University, Hsinchu, Taiwan in 1999, MS degree in Electronic Engineering from the Fu-Jen Catholic University, Taipei, Taiwan in 2003 and PhD degree in Electrical and Control Engineering from the Chiao Tung University, Taiwan in 2008. His research interests include neural networks, fuzzy logic systems, adaptive control, intelligent control and DSP design.

Chi-Hsu Wang received his BS degree in Control Engineering from National Chiao Tung University, Hsinchu, Taiwan, MS degree in Computer Science from the National Tsing Hua University, Hsinchu, Taiwan and PhD degree in Electrical and Computer Engineering from the University of Wisconsin, Madison, in 1976, 1978 and 1986, respectively. He was appointed as Associate Professor in 1986 and Professor in 1990 in the Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan. He is currently a Professor in the Department of Electrical Engineering, National Chiao Tung University, Hsinchu, Taiwan. His current research interests and publications are in the areas of digital control, fuzzy neural network, intelligent control, adaptive control and robotics. Dr Wang is an IEEE Fellow. He is currently serving as an associate editor of *IEEE Transactions on Systems, Man, and Cybernetics, Part B* and as a member of Board of Governors and Webmaster of IEEE Systems, Man, and Cybernetics Society.

Tsu-Tian Lee received his BS degree in Control Engineering from the National Chiao Tung University (NCTU), Hsinchu, Taiwan, in 1970 and his MS and PhD degrees in Electrical Engineering from the University of Oklahoma, Norman, in 1972 and 1975, respectively. In 1975, he was an Associate Professor with the Department of Control Engineering, NCTU, where he became a Professor and the Chairman in 1978. In 1981, he became a Professor and the Director of the Institute of control Engineering, NCTU. In 1986, he was a Visiting Professor with the University of Kentucky, Lexington, where he became a Full Professor of Electrical Engineering in 1987. In 1990, he was a Professor and the Chairman of the Department of Electrical Engineering, National Taiwan University of Science and Technology (NTUST), Taipei, where he became a Professor and the Dean of the Office of Research and Development in 1998. Since 2000, he has been with the Department of Electrical and Control Engineering, NCTU, where he is currently a Chair Professor. Since 2004, he has been with the Department of Electrical Engineering, National Taipei University of Technology, where he is currently the President. Prof. Lee is a Fellow of the Institution of Electrical Engineers and the New York Academy of Sciences. He received the Distinguished Research Award from the National Science Council, ROC from 1991 to 1998, the Academic Achievement Award in Engineering and Applied Science from the Ministry of Education, ROC in 1997, the National Endowed Chair from the Ministry of Education, ROC in 2003, the TECO Science and Technology Award from the TECO Technology Foundation in 2003 and the Norbert Wiener Award from the IEEE Systems, Man and Cybernetics (SMC) Society in 2009. He has been a member of the technical program and advisory committees of many IEEE-sponsored international conferences. He is currently the Vice President of Membership, a member of the Board of Governors and the Newsletter Editor of the IEEE Systems, Man, and Cybernetics Society.

References

Angelov, P.P., and Fliev, P. (2004), 'An Approach to Online Identification of Takagi-Sugeno Fuzzy Models', *IEEE Transactions Systems, Man and Cybernetics B*, 34, 484–498.

Boyd, S., Gahoui, L.E., Feron, E., and Balakrishnan, V. (1994), *Linear Matrix Inequalities in System and Control Theory*, Philadelphia, PA: SIAM.

Calise, A., Hovakimyan, N., and Idan, M. (2001), 'Adaptive Output Feedback Control of Nonlinear Systems Using Neural Networks', *Automatica*, 37, 1201–1211.

Castro, J.L. (1995), 'Fuzzy Logic Controllers are Universal Approximators', *IEEE Transactions Systems, Man and Cybernetics*, 25, 629–635.

Chatterjee, A., and Watanabe, K. (2005), 'An Adaptive Fuzzy Strategy for Motion Control of Robot Manipulators', *Soft Computing*, 9, 185–193.

Corless M., and Zenieh S. (1995), 'A New Control Design Methodology for Robotic Manipulators with Flexible Joints', in *Proceedings of the American Control Conference*, Seattle, WA, pp. 4316–4320.

Da, X.L., Cheng, C., and Tang, B. (2000), 'A Linear Matrix Inequality Approach for Robust Control of Systems with Delayed States', *European Journal of Operational Research*, 124, 332–341.

Ge, S.S., Hang, C.C., and Zhang, T. (1999), 'Adaptive Neural Network Control of Nonlinear Systems by State and Output Feedback', *IEEE Transactions Systems, Man and Cybernetics*, 129, 818–828.

Ge, S.S., and Wang, C. (2002), 'Adaptive NN Control of Uncertain Nonlinear Pure-feedback Systems', *Automatica*, 38, 671–682.

Ge, S.S., and Zhang, J. (2003), 'Neural-Network of Nonaffine Nonlinear Systems with Zero Dynamics by State and Output Feedback', *IEEE Transactions Neural Networks*, 14, 900–918.

Gil-Lafuente, A.M. (2005), *Fuzzy Logic in Financial Analysis*, Berlin, Heidelberg, New York: Springer.

Gutierrez, H.M., and Ro, P.I. (2005), 'Magnetic Servo Levitation by Sliding-mode of Nonaffine Systems Control with Algebraic Input Invertibility', *IEEE Transactions Industrial Electronics*, 52, 1449–1455.

Han, H., Su, C.Y., and Stepanenko, Y. (2001), 'Adaptive Control of a Class of Nonlinear Systems with Nonlinearly Parameterized Fuzzy Approximators', *IEEE Transactions Fuzzy Systems*, 9, 315–323.

Hovakimyan, N., Nardi, F., Calise, A., and Kim, N. (2002), 'Adaptive Output Feedback Control of Uncertain Nonlinear Systems Using Single-Hidden-Layer Neural Networks', *IEEE Transactions Neural Networks*, 13, 1420–1431.

Hsu, C.F., Chen, G.M., and Lee, T.T. (2007), 'Robust Intelligent Tracking Control with PID-type Learning Algorithm', *Neurocomputing*, 71, 234–243.

Hsu, C.F., and Lin, C.M. (2005), 'Fuzzy-Identification-Based Adaptive Controller Design Via Backstepping Approach', *Fuzzy Sets Systems*, 151, 43–57.

Hsu, C.F., Lin, C.M., and Chen, T.Y. (2005), 'Neural-Network-Identification-based Adaptive Control of Wing Rock Motion', *IEE Proceedings Control Theory and Applications*, 152, 65–71.

Hunt, L.R., and Meyer, G. (1997), 'Stable Inversion for Nonlinear Systems', *Automatica*, 33, 1549–1554.

Isidori, A. (1989), *Nonlinear Control System* (2nd ed.), Berlin, Germany: Springer-Verlag.

Juang, C.H., and Tsao, Y.W. (2008), 'A Self-evolving Interval Type-2 Fuzzy Neural Network with Online Structure and Parameter Learning', *IEEE Transactions Fuzzy Systems*, 16, 1411–1424.

Karimi, B., Menhaj, M.B., and Saboori, I. (2006), 'Robust Adaptive Control of Nonaffine Nonlinear Systems Using Radial Basis Function Neural Networks', *Proceedings IEEE Conference on Industrial Electronics*, 495–500.

Khalil, H.K. (2002), *Nonlinear Systems*, Upper Saddle River, NJ: Prentice Hall.

Krstic, M., Kanellakopoulos, I., and Kokotovic, P.V. (1992), 'Adaptive Nonlinear Control Without Overparameterization', *Systems and Control Letters*, 19, 177–185.

Krstic, M., Kanellakopoulos, I., and Kokotovic, P.V. (1995), *Nonlinear and Adaptive Control Design*, New York: Wiley.

Labiod, S., Boucherit, M.S., and Guerra, T.M. (2005), 'Adaptive Fuzzy Control of a Class of MIMO Nonlinear Systems', *Fuzzy Sets Systems*, 151, 59–77.

Labiod, S., and Guerra, T.M. (2007), 'Adaptive Fuzzy Control of a Class of SISO Nonaffine Nonlinear Systems', *Fuzzy Sets Systems*, 158, 1126–1137.

Leu, Y.G., Lee, T.T., and Wang, W.Y. (1999), 'Observer-Based Adaptive Fuzzy-Neural Control for Unknown Nonlinear Dynamical System', *IEEE Transactions Systems, Man and Cybernetics B*, 29, 583–591.

Leu, Y.G., Wang, W.Y., and Lee, T.T. (2005), 'Observer-Based Direct Adaptive Fuzzy-Neural Control for Nonaffine Nonlinear Systems', *IEEE Transactions Neural Networks*, 16, 853–861.

Lewis, F.L., Jagannathan, S., and Yesildirek, A. (1999), *Neural Network Control of Robot Manipulators and Nonlinear Systems*, London, UK: Taylor & Francis.

Li, H.X., and Tong, S.C. (2003), 'A Hybrid Adaptive Fuzzy Control for a Class of Nonlinear MIMO Systems', *IEEE Transactions Fuzzy Systems*, 11, 24–34.

Lin, C.J., and Chen, C.H. (2005), 'A Self-Constructing Compensatory Neural Fuzzy System and its Applications', *Mathematical and Computer Modelling*, 42, 339–351.

Lin, C.M., and Hsu, C.F. (2002), 'Guidance Law Design by Adaptive Fuzzy Sliding-mode Control', *Guidance, Control, and Dynamics*, 25, 248–256.

Lin, F.J., and Lin, C.H. (2004), 'A Permanent-Magnet Synchronous Motor Servo Drive Using Self-constructing Fuzzy Neural Network Controller', *IEEE Transactions Energy Conversion*, 19, 66–72.

Mahmoud, G.M., and Farghaly, A.A. (2004), 'Chaos Control of Chaotic Limit Cycles of a Real Complex Van Der Pol Oscillators', *Chaos, Solitons and Fractals*, 21, 915–924.

Meng, J.E., and Chang, D. (2004), 'Online Tuning of Fuzzy Inference Systems Using Dynamic Fuzzy Q-learning', *IEEE Transactions Systems, Man and Cybernetics*. *B*, 34, 1478–1489.

Park, J.H., and Kim, S.H. (2005), 'Direct Adaptive Self-structuring Fuzzy Controller for Nonaffine Nonlinear System', *IEE Proceedings Control Theory*, 153, 429–445.

Pal, N.R., and Pal, T. (1999), 'On Rule Pruning Using Fuzzy Neural Networks', *Fuzzy Sets Systems*, 106, 335–347.

Pourhiet, A.L., Correge, M., and Caruana, D. (2003), 'Control of Self-oscillating Systems', *IEE Proceedings Control Theory*, 150, 599–610.

Ramos, R.A., Alberto, L.F.C., and Bretas, N.G. (2003), 'Linear Matrix Inequality Based Controller Design with Feedback Linearisation: Application to Power Systems', *IEE Proceedings Control Theory and Applications*, 150, 551–556.

Shann, J.J., and Fu, H.C. (1995), 'A Fuzzy Neural Network for Rule Acquiring on Fuzzy Control Systems', *Fuzzy Sets Systems*, 71, 345–357.

Spong, M.W., and Vidyasagar, M. (1989), *Robot Dynamics and Control*, New York: Wiley.

Terano, T., Asai, K., and Sugeno, M. (1992), *Fuzzy Systems Theory and its Applications*, Boston: Academic Press.

Wang, L.X. (1994), *Adaptive Fuzzy Systems and Control – Design and Stability Analysis*, Englewood Cliffs, NJ: Prentice-Hall.

Wang, W.Y., Chan, M.L., Hsu, C.C., and Lee, T.T. (2002a), '*H Tracking-based Sliding Mode Control for Uncertain Nonlinear Systems Via an Adaptive Fuzzy-neural Approach', *IEEE Transactions Systems, Man and Cybernetics B*, 32, 483–492.

Wang, H.H., and Krstic, M. (2000), 'Extremum Seeking for Limit Cycle Minimization', *IEEE Transactions on Automatic control*, 45, 2432–2437.

Wang, C.H., Lin, T.C., Lee, T.T., and Liu, H.L. (2002b), 'Adaptive Hybrid Intelligent Control for Uncertain Nonlinear Dynamical Systems', *IEEE Transactions Systems, Man and Cybernetics B*, 32, 583–597.