Applied Soft Computing 9 (2009) 1367–1376

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A novel data reduction method for Takagi–Sugeno fuzzy system design based on statistical design of experiment

Yadollah Farzaneh *, Ali Akbarzadeh Tootoonchi

Department of Mechanical Engineering, Ferdowsi University of Mashhad, Mashhad, Iran

ARTICLE INFO

Article history:
Received 11 February 2008
Received in revised form 14 May 2009
Accepted 16 June 2009
Available online 25 June 2009

Keywords:
T–S system
Design of experiments
Factorial design
Data reduction
Simple design
Benchmark problems

ABSTRACT

This paper introduces a simple, systematic and effective method for designing Takagi–Sugeno (T–S) fuzzy systems utilizing a significantly smaller training data set versus existing methods. Creating proper training data is usually not an easy task and requires spending considerable time and resources. The proposed method first uses a three-level factorial design to partition the output space. Next the least square technique is used to estimate each of the partitioned output spaces. The membership functions are introduced with only three variables (min, max and number of membership functions). Fuzzy rules are generated with respect to the partitioned output surfaces and the membership functions. The proposed method is applied to two benchmark problems, controlling an inverted pendulum as well as modeling a nonlinear function. In the case of the inverted pendulum simulation results demonstrate significant improvement. In the case of nonlinear function modeling we demonstrated sufficient accuracy with only 9 training data, which represents 98% reduction in the number of training data compare to other method. Additionally, the proposed method offers extremely low computation time allowing it to be used with adaptive type systems.

1. Introduction

The Takagi–Sugeno (T–S) fuzzy systems [1] are a well-known landmark in the history of fuzzy theory and control. During the past years they have become a powerful practical engineering tool for modeling and control of uncertain nonlinear and complex systems [2–5]. It has also been applied to a variety of industrial applications [6–8] as well as complex robotics applications such as biped [9,10], snake [11] and fish robots [12]. The underlying T–S fuzzy system is an interpolation method which partitions the input space into fuzzy areas. Each area is approximated by a simple local model (often a linear model). The global model is obtained by interpolation between the different local models. This model permits the approximation of a strongly nonlinear function by a simple structure and a limited number of rules. The consequents of the fuzzy rules are expressed as analytic functions. The choice of the function depends on its practical applications.

Despite the many advantages of T–S system, its design significantly hinders its application [13,14]. Carrying out design of T–S system is difficult because the explicit structure of T–S system is generally unknown, and also due to their inherent nonlinear nature. Many efforts have been made to enhance systematic and simple design of T–S systems. In [15] the premise and consequent identification are separately performed using fuzzy *c*-means and the orthogonal least squares method, respectively. Jang [16] considers the T–S models as fuzzy-neural networks and neural-type algorithms are used for model learning. Yen et al. [17] developed several approaches that attempt to reduce the number of fuzzy rules by assessing their degrees of importance using singular value decomposition (SVD). They start with an oversized rule base and then remove redundant or less important fuzzy rules. Jin [18] proposed similarity-based approaches by merging a pair of similar fuzzy sets and fuzzy rules, respectively, at a time. Recently, considerable number of methods use genetic algorithms to build fuzzy Sugeno models. For instance Du and Zhang [2] proposed a new encoding scheme for learning the T–S fuzzy model from data by using genetic algorithms.

An important step in designing the T–S system is creating the training data. Most of the methods mentioned above require a large number of training data for designing the T–S system. However, data generation is not always an easy task. It can require excessive time, resources and it can be costly. It is clear that a method which requires fewer numbers of data and less computation time as well as being simpler to apply is more desirable. With respect to the author's knowledge only a few works have dealt with reducing the number of training data. Most researchers are focused on simplifying the T–S system by reducing the number of rules with similarity measure

* Corresponding author. Tel.: +98 511 8815100.
E-mail addresses: ya_farzaneh@yahoo.com (Y. Farzaneh), ali_akbarzadeh_t@yahoo.com (A. Akbarzadeh Tootoonchi).

1568-4946/$ – see front matter
doi:10.1016/j.asoc.2009.06.005

[18–20]. They assume that sufficient data is available and attempt to simplify the system after design is completed.

The primary objective of this research is to develop an efficient and simple method for designing T–S system requiring reduced number of training data and less computation time. To do this, factorial design commonly used in design of experiments (DOE) [21] is used for modeling the output space. Factorial design is one of the most powerful methods used in design of experiments. It is used for modeling the behavior of an unknown system with a reduced number of data and experiment. Factorial designs study the effects of two or more variables and are widely used in modeling and analysis of various complex processes [22–24]. In general, factorial designs are very efficient type of experiments [21]. In a factorial design, all possible combinations of the levels of the factors are investigated in each experimental trial. In this paper, factorial design is used to minimize the training data required for design of the T–S system. Next, the overall structure of the T–S system is defined with respect to the framework of the factorial design. This provides a simple and efficient method for designing the T–S system. The newly designed T–S system is applied to modeling of a nonlinear function as well as controlling a complex system namely an inverted pendulum.

The rest of this paper is organized as follows. Section 2 provides the necessary background information on the Takagi–Sugeno fuzzy system as well as factorial design. The main contribution of the paper is presented in Section 3. It describes the proposed systematic design of the T–S system. Section 4 provides two examples which demonstrate the applicability of the method. Finally, concluding remarks relating the overall study is drawn in the last section.

2. Preliminaries

2.1. Takagi–Sugeno fuzzy system

A static T–S fuzzy model as well as T–S fuzzy controller are described by a set of fuzzy "IF ... THEN" rules. A generic T–S rule can be written as follows:

$$\begin{aligned} &R_i : \text{IF}\, x_1 \text{ is } A_{i1} \text{ AND } x_2 \text{ is } A_{i2} \ldots \text{ AND } x_r \text{ is } A_{ir}, \\ &\text{THEN}\, y_i = f_i(x_1, x_2, \ldots, x_r), \quad i = 1, 2, \ldots, n_R. \end{aligned} \tag{1}$$

where A_{i1}, A_{i2},..., A_{ir} are fuzzy sets in the antecedent, while y_i is a crisp function in the consequent. y_i is usually a polynomial function of input variables. However, it can be any function as long as it can appropriately describe the output of the model within the fuzzy region specified by the antecedent of the rule. When y_i is a first-order polynomial, as in this paper, the resulting fuzzy inference system is called a first-order Sugeno fuzzy model [1].

$$y_i = a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{ir}x_r + b_i, \quad i = 1, 2, \ldots, n_R. \tag{2}$$

where a_{i1}, a_{i2},..., a_{ir} and b_i are parameters which should be identified. The consequents of the T–S system are hyperplanes (r-dimensional linear subspaces) in $\Re^{r+1}$, whereas the if-part of the rule partitions the input space and determines the validity of the n_R locally linear model for different regions of the antecedent space. Since each rule has a crisp output, the overall output of the T–S system could be obtained via weighted average formula (Eq. (3)).

$$y = \frac{\sum_{i=1}^{n_R} y_i.w_i}{\sum_{i=1}^{n_R} w_i}, \quad w_i = \prod_{i=1}^{r} \mu_{A_i}(x_i), n_R \text{ is equal to the number of rules} \tag{3}$$

What remains to complete the description of T–S system is a method to estimate parameters a_{i1}, a_{i2},..., a_{ir} and b_i of the model shown in Eq. (2). In the next section design of experiment methodology, specifically factorial design, is utilized to provide a systematic and an efficient way to obtain input–output data as well as to provide estimates for the model parameters.

2.2. Statistical design of experiment—factorial design

2.2.1. Experimental design—obtaining data

Experimental design is a critically important tool in the engineering world for improving the performance of products and manufacturing processes. Statistical design of experiments refers to the process of planning the experiment so that appropriate data that can be analyzed by statistical methods is collected, resulting in valid and objective conclusions. There are many different design types available. Among these designs, factorial design is perhaps the most widely used in experiments involving several variables (factors) where it is necessary to study the joint effect of the factors on one or more output responses. Factorial designs have two great advantages, they provide information that is not readily available from other methods and they use experimental material very efficiently. Factorial experiments identify the numbers of factors and the number of levels of each factor. For example if there are a levels of factor A and b levels of factor B, each experiment run contains all ab treatment combinations. The effect of a factor is defined to be the change in output response produced by the change in the level of the factor. This is frequently called the main effect because it refers to the primary factors of the interest in the experiments. Least square method can be used to model output response. The steps required to obtain the input–output as well as to provide estimates for the model parameters are as follows:

Step 1: Identify the critical output response variables.
Step 2: Identify the critical input variables (factors) affecting output response.
Step 3: Identify the strategy of experimentation. Factorial design is selected.
Step 4: Determine where to set the levels (low-intermediate-high) of the critical input variables so that the desired output response is obtained.
Step 5: Randomize the order of experiments. This will balance out the effect of any nuisance variable, noise, which may influence the observed output response.
Step 6: Conduct the required experiments outlined in previous step to get input–output data.
Step 7: Develop mathematical model for the output response surface.

In this study, 3^k factorial design is used; that is, a factorial arrangement with k factors each at three levels. Therefore, we need to determine where to set the levels (low-intermediate-high) of the critical input variables (factors) so that the desired output response is obtained. A usual notation for the three levels, called coded values, is −1, 0, +1. This notation facilitates fitting a regression model relating the response to the factor levels. For example, the coded design matrix for a factorial design with two factors, 3^2, is shown in Table 1.

The levels for each factor are chosen in a way to produce a balance design, every columns has equal number of +1, −1 and 0. For a system with two input variables these levels partition the input space into four orthogonal partitions as shown in Fig. 1.

Choosing the input variables to produce orthogonal partitions has few advantages. It simplifies the design and minimizes the variance of the coefficients of the first-order model.

2.2.2. Model development

Once input–output data are obtained, regression analysis may be used to fit the model. The relationship between the dependent y

Table 1
Treatment combinations in a 3^2 design.

Run	Input variable 1	Input variable 2	Output
1	−1	−1	y_1
2	−1	0	y_2
3	−1	+1	y_3
4	0	−1	y_4
5	0	0	y_5
6	0	+1	y_6
7	+1	−1	y_7
8	+1	0	y_8
9	+1	+1	y_9

(output response) and input variables ($x_1, x_2, x_3, \ldots$) may be known exactly of the form

$$y = f(x_1, x_2, x_3, \ldots) + \varepsilon \tag{4}$$

where ε represents the model error, measurement error and other variations. f is a first- or second-order polynomial which is the empirical response surface model. The successful application of regression relies on the identification of a suitable approximation for f. This will generally be a first-order model of the form

$$f = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k \tag{5}$$

The β terms in Eq. (5) are called the regression coefficients. These terms comprise the unknown parameter set which can be estimated by collecting experimental data. The seven experimental design steps outlined in Section 2.2.1 may be used to obtain these data. The method of least squares is typically used to estimate, β's, the regression coefficients. Using least squares method, we can write the model equation (Eq. (5)) in terms of the obtained experimental data.

$$\begin{aligned} y_i &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_k x_{ik} + \varepsilon_i \\ &= \beta_0 + \sum_{j=1}^{k} \beta_j x_{ij} + \varepsilon_i; \quad i = 1, 2, \ldots, n. \end{aligned} \tag{6}$$

where x_{ij} denotes the ith level of variable x_j and y_i is a observed response. The method of least squares chooses the β's in Eq. (6) so that the sum of the squares of the errors, ε_i, is minimized. The least

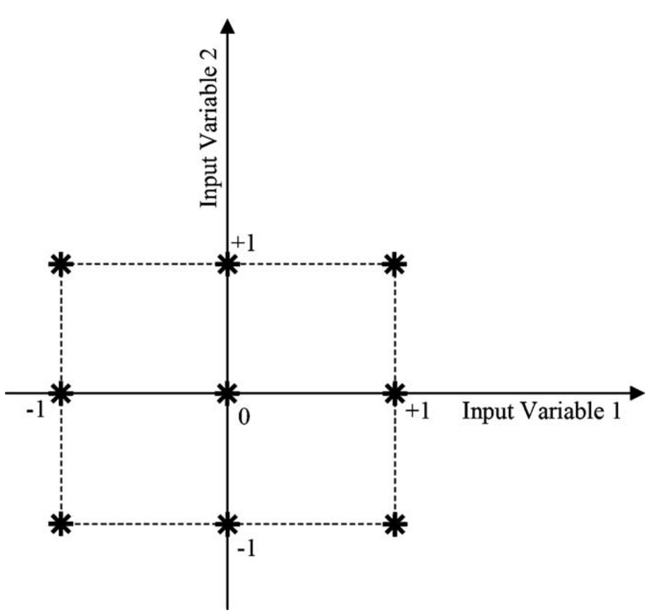

Fig. 1. Input space in the three-level (3^2) factorial design.

squares function is

$$L = \sum_{i=1}^{n} \varepsilon_i^2 = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{k} \beta_j x_{ij} \right)^2 \tag{7}$$

The function L is to be minimized with respect to $\beta_0, \beta_1, \ldots, \beta_k$. The gradian minimization method is used to produce $k + 1$ least squares normal equations for each unknown regression coefficient.

It is simpler to represent the model in matrix form.

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \varepsilon \tag{8}$$

where

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1k} \\ 1 & x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nk} \end{bmatrix}. \\ \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{bmatrix}, \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_0 \\ \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{bmatrix} \tag{9}$$

We wish to find the vector of least squares estimators, $\hat{\boldsymbol{\beta}}$, that minimizes

$$L = \sum_{i=1}^{n} \varepsilon_i^2 = \varepsilon'\varepsilon = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \tag{10}$$

The least squares estimators must satisfy

$$\frac{\partial L}{\partial \boldsymbol{\beta}} = -2\mathbf{X}'\mathbf{y} + 2\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}} = 0 \tag{11}$$

Eq. (11) is the matrix form of the least squares normal equations. Thus, the least squares estimator of β's can be shown in the following equation [21]:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \tag{12}$$

Therefore, the fitted regression model is expressed as,

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}} \tag{13}$$

where $\hat{\mathbf{y}}$ is the predicted response of the model.

2.2.3. Orthogonal design

It is very important for a linear model that the effect of regression coefficients to be independent of each other. The implication is that the roles of the variables can be assessed independent of each other. The covariance (*Cov*) is a measure of the linear association between two terms. Therefore, if the $Cov(\beta_i, \beta_j) = 0$ then the regression coefficients are independent. So, we should choose a design which admits statistically independent estimates of effects. Orthogonal design is a design with this interesting property [25]. A design is orthogonal if the inner product of the columns of X are equal to zero ($\sum x_{ip} x_{iq} = 0$). This implies that the off-diagonal elements of the coded (X′ X) matrix are all zero as well as the levels of the two corresponding variables are linearly independent.

The goal of orthogonal design is to perform a minimum number of tests but acquire the most valuable information of the considered problem [26]. As a result, well-balanced subsets of level combinations will be chosen. Therefore, the orthogonal design provides an economic method for studying the effect of process variables on process response. The factorial design discussed in previous section insures orthogonality for obtaining data. As shown in Table 1, the design columns are all balanced vertically, i.e., there are an equal number of +1 and −1 values in each column. That is the sum of all the numbers in each column is zero. Furthermore, these columns are also horizontally balanced for each level of each column. These balancing properties result in design orthogonality which allows us to estimate the effects of each factor independently of the others.

3. Proposed method

This section investigates the general procedure for nonlinear modeling as well as controlling systems using the proposed method. In order to better describe the method, first a simple model having two inputs and one output is selected. It is worth noting that the proposed method can be generalized to higher dimensional systems. The general steps are defined as following.

3.1. Establish the upper and lower margins for the inputs

The assignment process to define the upper and lower margins can be intuitive or it can be based on some algorithmic or logical operation. It is however, usually derived through understanding and prior knowledge about the system. For example, if temperature is used as an input variable to define the range of human comfort we get one range, and if temperature is used to define the range of safe operating temperature for a steam turbine we get another range. If prior knowledge about the system is not available the designer may need to conduct a series of baseline experiments to help establish these margins [21]. This may be viewed as an algorithmic or logical approach.

3.2. Defining fuzzy membership functions

The choice of membership functions will help define output surface which itself is made of combining multiple surfaces. In order to insure smooth transition among these surfaces, the number of membership function must be even, same type and input domain must be equally divided (Fig. 2). It must be noted that the method is not sensitive to the number of membership functions. These requirements will be more discussed in following sections.

3.3. Training data

Having an appropriate set of input–output training data is one of the most important factors in designing T–S systems. This data should explain the behavior of the unknown system. However, creating data is not an easy task and requires spending excessive time and resources. The main goal of the proposed method is to design a suitable system which uses the least number of training data. A three-level factorial design is used to obtain the input–output data. If n indicates the number of inputs, the total number of required data is computed by Eq. (14).

$$\text{Total number of data} = 3^n \tag{14}$$

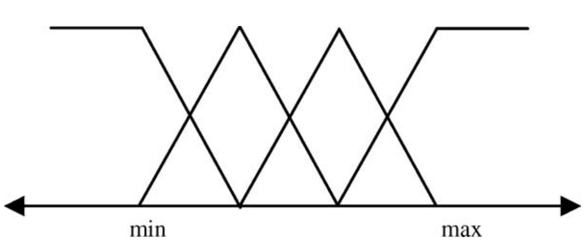

Fig. 2. Membership functions.

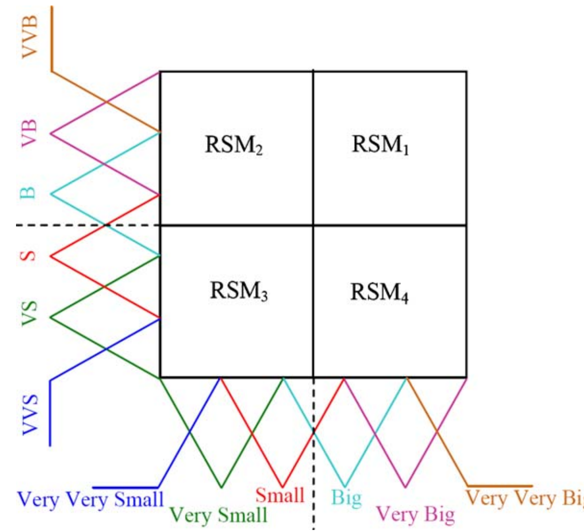

Fig. 3. Relationship between inputs and output.

Each input has three levels (low, medium and high). For a system with two inputs, the input space is divided as shown in Fig. 1. The star points in Fig. 1, indicate the location in input space where experiments should be conducted in order to collect the output response values.

3.4. Consequents part of the fuzzy rules

Least square is used to construct the consequents part of the fuzzy rules in T–S fuzzy system. The input domain is divided into four sections. Each section is represented by a first-order surface (RSM_1–RSM_4). Using least square method, each surface is formed with the four output data obtained by performing the physical experiments. Alternatively, the values for the output may be obtained from previous history or some expert knowledge about the system. If x_1 and x_2 are first and second input variables and a_{ij} are constant parameters, then the four response surfaces model (RSM) are defined by Eq. (15).

$$\begin{aligned}
\text{RSM}_1 &= y_1 = a_{11}x_1 + a_{12}x_2 + a_{13}\\
\text{RSM}_2 &= y_2 = a_{21}x_1 + a_{22}x_2 + a_{23}\\
\text{RSM}_3 &= y_3 = a_{31}x_1 + a_{32}x_2 + a_{33}\\
\text{RSM}_4 &= y_4 = a_{41}x_1 + a_{42}x_2 + a_{43}
\end{aligned} \tag{15}$$

Fig. 3 will clearly illustrate the relationship between inputs and output.

3.5. Fuzzy rules and deffuzification

Assuming six membership functions for each input, as in Fig. 3, the fuzzy rules for this system are given in Table 2.

Finally, the weighted average method is employed to deffuzzify the output variable (Eq. (3)).

4. Illustrative examples

In this section, we demonstrate the effectiveness of our approach by showing results of two simulations, controlling the inverted pendulum around its unstable equilibrium point as well as modeling a nonlinear function. These two experiments have been performed using a PC with 2.67-GHz CPU and 512-MB RAM memory.

4.1. Inverted pendulum

The inverted pendulum is a highly nonlinear and unstable system. It is therefore often used as a benchmark for verifying the

Table 2
Fuzzy rules.

X1	X2					
	VVS	VS	S	B	VB	VVB
VVS	RSM_3	RSM_3	RSM_3	RSM_2	RSM_2	RSM_2
VS	RSM_3	RSM_3	RSM_3	RSM_2	RSM_2	RSM_2
S	RSM_3	RSM_3	RSM_3	RSM_2	RSM_2	RSM_2
B	RSM_4	RSM_4	RSM_4	RSM_1	RSM_1	RSM_1
VB	RSM_4	RSM_4	RSM_4	RSM_1	RSM_1	RSM_1
VVB	RSM_4	RSM_4	RSM_4	RSM_1	RSM_1	RSM_1

performance and effectiveness of new control methods. The system consists of an inverted pole hinged on a cart which is free to move in the horizontal direction. The inverted pendulum inherently has two equilibria, one of which is stable while the other is unstable. The stable equilibrium corresponds to a state in which the pendulum is pointing downwards. In the absence of any control force, the system will naturally return to this state. The unstable equilibrium corresponds to a state in which the pendulum points strictly upwards and, thus, requires a control force to maintain this position. The basic control objective of the inverted pendulum problem is to maintain the unstable equilibrium position when the pendulum initially starts with some nonzero angle from the vertical position. An inverted pendulum is shown in Fig. 4. The nonlinear dynamic equations are given by Eq. (16) [27].

$$\begin{aligned}&(M+m)\ddot{x}+ml\ddot{\theta}\cos\theta-ml\dot{\theta}^2\sin\theta=F\\&ml\ddot{x}\cos\theta+4/3(ml^2\ddot{\theta})-mgl\sin\theta=0\end{aligned} \tag{16}$$

where θ (deg) and $\dot{\theta}$ (deg/s) are the angular displacement and angular velocity of the pole, g (acceleration due to the gravity) is 9.8 m/s^2, M (mass of the cart) is 1 kg, m (mass of the pole) is 0.1 kg, l (half length of the pole) is 0.5m and F is the application force in *Newton* which is the required force to bring the pole into equilibrium position. The seven experimental design steps outlined in Section 2.2.1 may be used to obtain input–output data.

Step 1: Force is the critical output of the controller.
Step 2: θ and $\dot{\theta}$ are the critical input variables (factors) affecting the required force.
Step 3: Factorial design is selected as the experiment strategy.
Step 4: The low-intermediate-high levels can be chosen by the designer according to feasible domains of input variables or from some prior knowledge [28]. In this example, the levels of θ and $\dot{\theta}$ are varied within (−20, 0, 20) and (−70, 0, 70) respectively.
Step 5 & 6: If no prior knowledge about system performance is available, then this step requires performing the actual experiment to obtain output data. The order in which the experiments are conducted must be randomized. However, in this example, the governing equations of motion are available therefore the outputs are obtained by conducting simulations. Results are shown in Table 3.
Step 7: Four linear surfaces defined in Eq. (15) can now be formed through the use of least square method. Results are shown in Eq. (17). The order of numbering is inspired by Fig. 3.

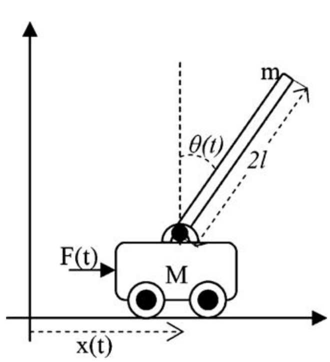

Fig. 4. Inverted pendulum.

$$\begin{aligned}&RSm1=17.1887\theta+4.9111\dot{\theta}+1.5\\&RSm2=34.3775\theta+4.9111\dot{\theta}+1.5\\&RSm3=17.1887\theta+4.9111\dot{\theta}-1.5\\&RSm4=34.3775\theta+4.9111\dot{\theta}-1.5\end{aligned} \tag{17}$$

The seven steps completed thus far, allowed us to derive the consequence part of the fuzzy rules. Next, membership functions and fuzzy rules must be defined in order to design the fuzzy controller. θ (deg) and $\dot{\theta}$ (deg/s) are the controller inputs and are varied within [−20, 20] and [−70, 70] respectively. For each input variable, six membership functions are assumed and are divided equally between the two limits (Fig. 5).

The fuzzy rules are listed in Table 2. The overall response of the system is shown in Fig. 6.

Performance of the controller under various initial conditions is evaluated. The output responses are plotted in Fig. 7.

Fig. 7 indicates that the controller performs well even with a large deviation ($-71.4^\circ \le \theta \le 67.9^\circ$) from the equilibrium point. Next, we compare our proposed controller with other recent fuzzy controllers. Roopaei et al. [29] proposed a novel adaptive fuzzy sliding mode control (AFSMC) methodology based on the integration of sliding mode control (SMC) and adaptive fuzzy control (AFC). They evaluate the performance of the proposed AFSMC for the inverted pendulum problem with the same parameters as used in our study. The comparison between AFSMC [29], classical SMC [29] and our proposed controller are performed in Fig. 8.

As shown in Fig. 8, our proposed controller returns more rapidly to the equilibrium point.

The other comparison is performed with Chen et al. [30]. They proposed a GA-based adaptive fuzzy sliding mode controller (GA-based AFSMC) and use it for the problem of balancing an inverted pendulum on a cart. The comparison between our controller and GA-based AFSMC with the same initial condition of $\theta = 60^\circ$ is shown in Fig. 9.

Response of our controller is slightly slower than GA-based AFSMC. However, our control force is significantly smaller than GA-based AFSMC. It may be concluded that the overall behavior of our proposed controller is better than GA-based AFSMC.

Finally, a comparison is performed with the fuzzy controller proposed by Sun and Er [31]. Results indicate that our proposed controller significantly out performs controller proposed by [31] (Fig. 10).

The controller proposed by [31] becomes unstable after 41.83°. Our proposed controller not only can stabilize the system faster

Table 3
Input–output training data.

	Inputs		Output
	θ (deg)	$\dot{\theta}$ (deg/s)	Force
1	−20	−70	−12
2	−20	0	−9
3	−20	70	−6
4	0	−70	−9
5	0	0	0
6	0	70	9
7	20	−70	6
8	20	0	9
9	20	70	12

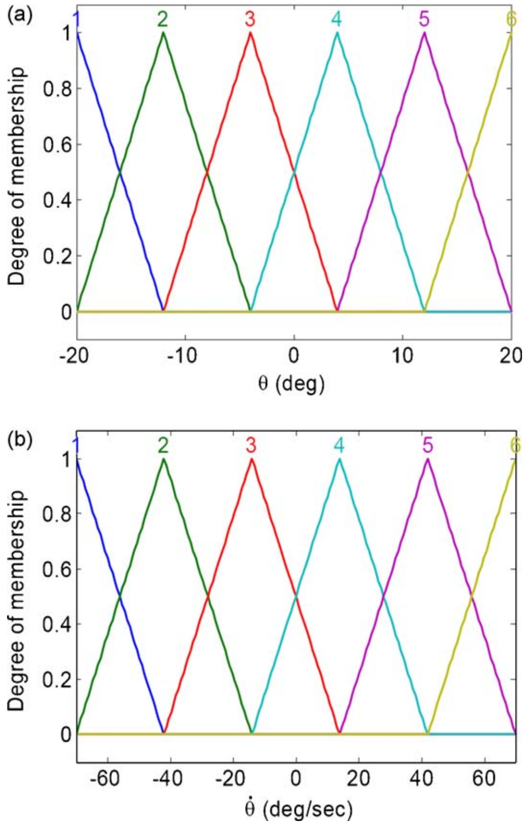

Fig. 5. Membership functions for inverted pendulum problem.

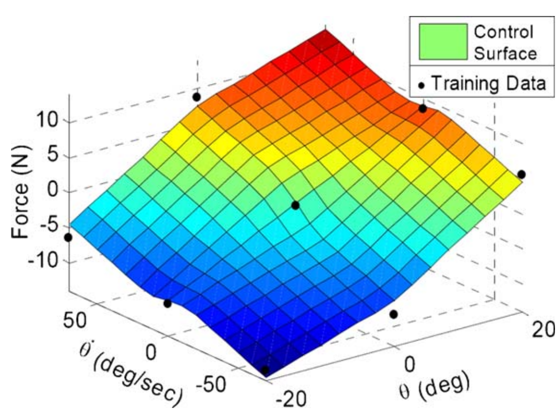

Fig. 6. Control surface.

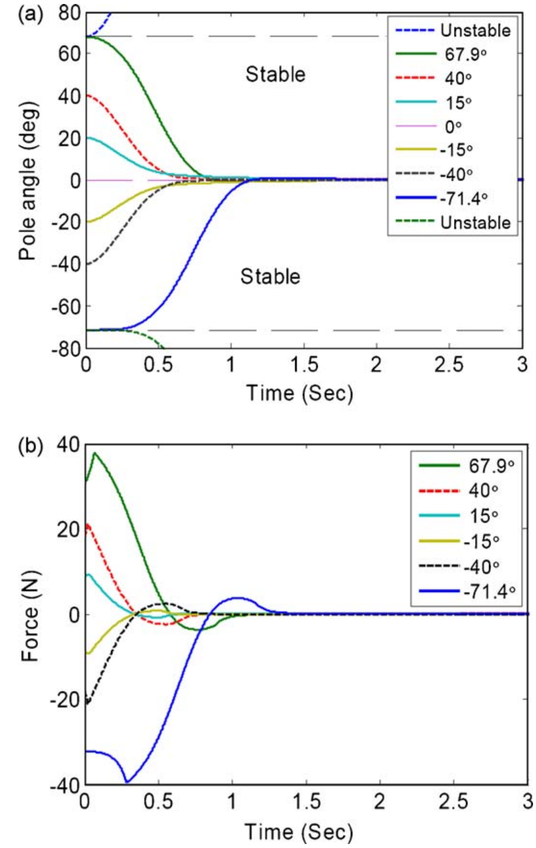

Fig. 7. The response of the system.

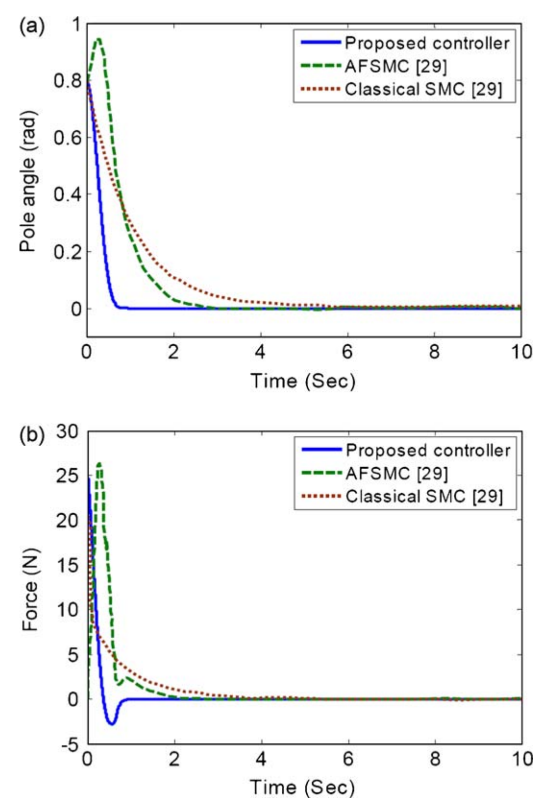

Fig. 8. Comparison with AFSMC and classical SMC.

than the [31] controller, but also as shown in Fig. 7, it remains stable for a significantly larger deviations from vertical, up to 67.9. Based on the three comparisons made, we conclude that our proposed controller is suitable for nonlinear control of a system such as inverted pendulum. It should be noted that all the controllers used for comparison each used a different initial condition. This is why the comparison is not presented in a one chart.

4.2. Nonlinear system modeling

Due to their nonlinear properties, T–S fuzzy systems are well known to be capable of modeling nonlinear functions. The modeling methods can be categorized into nonincremental or incremental, depending on how data is presented to the model. For

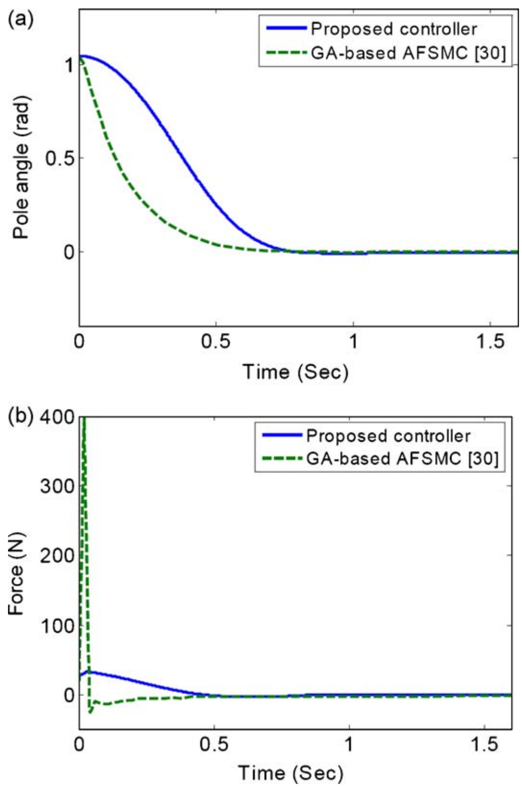

Fig. 9. Comparison with GA-based AFSMC.

nonincremental methods, similar to our proposed methods, the entire data set is considered all at once. On the other hand, the incremental modeling methods, consider training data one at a time [20].

To show the ability of the proposed method to model a nonlinear function, a nonlinear function given by the following formula [32] is used.

$$y = x_2 \sin x_1 + x_1 \cos x_2, \quad 0 \le x_1 \le \pi, \;\; 0 \le x_2 \le \pi \tag{18}$$

This function has no special meaning. It is meant to only provide a test-bed function for testing the proposed method. It should be noted that our objective in this example is not to improve the results obtained by other methods, but to investigate if the modeling performance of our proposed method with nine training data is satisfactory.

To obtain training data, the seven steps outlined in Section 2.2.1 is once again performed. The critical inputs and outputs may be

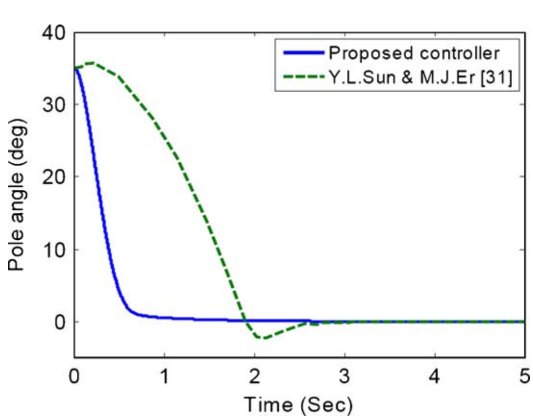

Fig. 10. Comparison between two controller.

Table 4
Input–output training data.

	Inputs		Output
	x_1	x_2	y
1	0	0	0
2	0	$\pi/2$	0
3	0	π	0
4	$\pi/2$	0	$\pi/2$
5	$\pi/2$	$\pi/2$	$\pi/2$
6	$\pi/2$	π	$\pi/2$
7	π	0	π
8	π	$\pi/2$	0
9	π	π	$-\pi$

obtained by physical examination of the nonlinear function that needs to be estimated. The 3^2 factorial is used. Each input is divided into three levels. The output values (Table 4) are then obtained by placing the input values into the original function (Eq. (18)).

Six membership functions are assumed for each input variable and are divided equally within their limits. See Fig. 11.

The output surfaces are produced by least square method.

$$\begin{aligned} \mathrm{RSM}_1 &= -2x_1 - x_2 + 7.0686 \\ \mathrm{RSM}_2 &= x_1 \\ \mathrm{RSM}_3 &= x_1 \\ \mathrm{RSM}_4 &= -x_2 + 2.3562 \end{aligned} \tag{19}$$

Fuzzy rules are generated according to Table 2. The overall behaviors of the modeled and actual system are shown in Fig. 12.

To show the validity of the predicted function, testing data has been taken by sampling x_1 and x_2 with a sampling period of $(\pi/20)$ in the range of $[\pi/40, 39\pi/40]$ for each input. The testing dataset will therefore contain 400 data points. Next, normalized root-mean-square-error (*NRMSE*) is used to measure the adequacy of the model. The model with the smallest value for *NRMSE* will be the

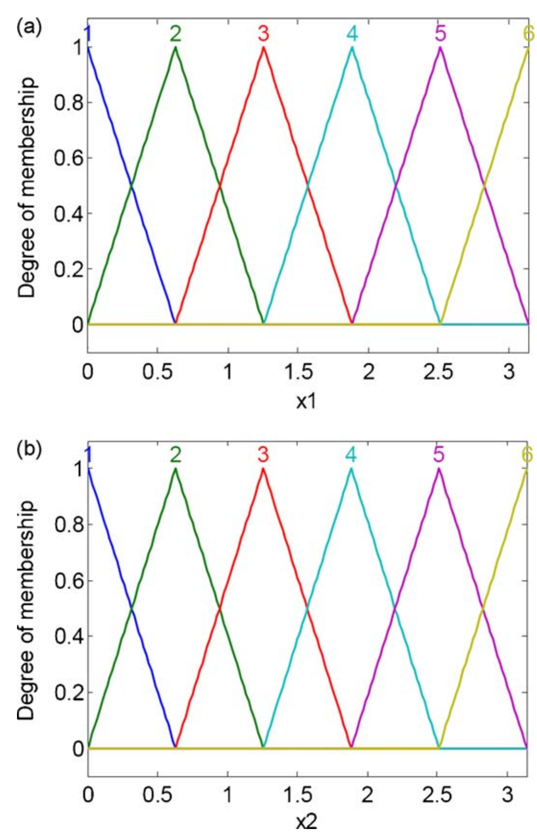

Fig. 11. Membership functions for nonlinear function.

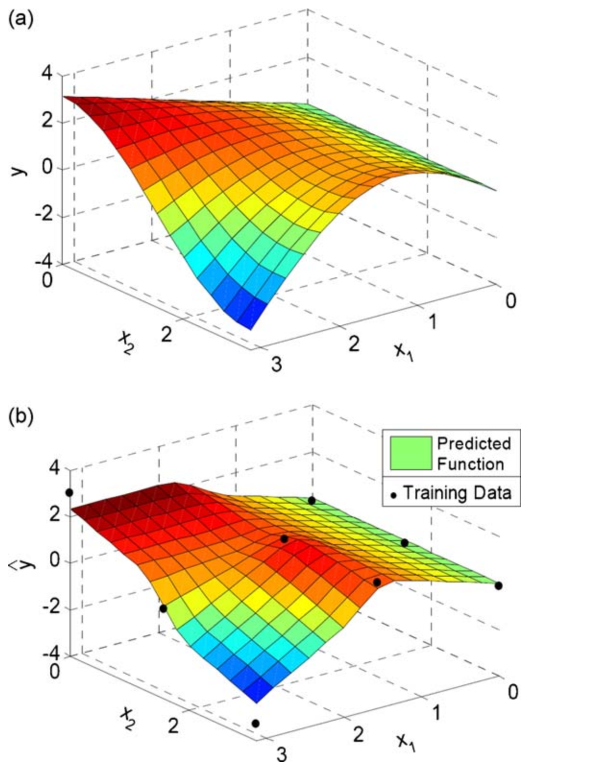

Fig. 12. (a) Output of the original function and (b) output of the T–S model (predicted).

superior model. The *NRMSE* is defined as follows [33]:

$$NRMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y}_i)^2}} \quad (20)$$

Using nine data points for training, the *NRMSE* values obtained for training and testing data is 0.25 and 0.3549, respectively. For comparison, the *NRMSE* values of other neurofuzzy modeling methods, MFC [20], ACA [34], SCRG [35], SM [36] and Rezaee's method [37] are presented. These methods use different technique to extract fuzzy rules from training data. Additionally, they all use 441 training data points. Ouyang et al. [20] developed an incrementally merge-based fuzzy clustering (MFC) method, Juang [34] used an aligned clustering-based algorithm (ACA), Lee and Ouyang [35] used a self-constructing rule generation algorithm (SCRG), Setnes et al. [36] proposed a merge based method for fuzzy rule simplification (SM), and Rezaee and Fazel Zarandi [37] used an initial rule base and update it by a module to reduce the system error. The results of the first four methods for modeling the nonlinear function (Eq. (18)) are reported in [20]. They arranged the 441 training data points in increasing, decreasing, interleaving and random order patterns. In each arrangement, they obtained a different value for *NRMSE*. Rezaee and Fazel Zarandi [37] varied

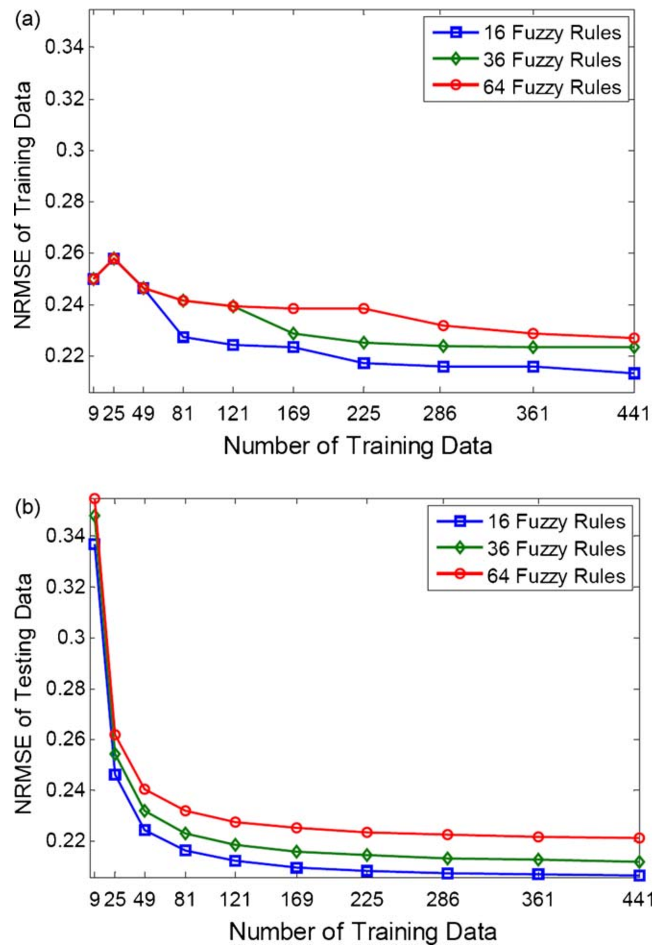

Fig. 13. *NRMSE* for training and testing data.

number of fuzzy rules and obtained different values of *NRMSE*. These values are reported as range in Table 5.

NRMSE value of our method for training is larger than the minimum but smaller than the maximum range value among the five methods. As expected, *NRMSE* value of our method for testing is not as impressive and is higher than all other methods. However, it should be noted that our method used significantly fewer number of training data, 9 versus 441, in comparison with the other methods. Therefore, the increase in *NRMSE* is rather logical. Moreover, the value of the *NRMSE* for MFC, ACA, SCRG and SM are significantly varied for different sequences of training data. The value of the *NRMSE* for Rezaee is also significantly varied for different number of fuzzy rules. However, our method is not sensitive to the order of the training data as well as the number of fuzzy rules. In summary, although in some cases, other methods may be more precise, but they seem to sacrifice the simplicity of the implementations.

Table 5
NRMSE values for different modeling methods.

	Training		Testing	
	Number of data	*NRMSE*	Number of data	*NRMSE*
MFC [20]	441	0.184–0.225	400	0.176–0.222
ACA [34]	441	0.161–0.291	400	0.147–0.287
SCRG [35]	441	0.144–0.273	400	0.121–0.259
SM [36]	441	0.155–0.291	–	0.150–0.289
Rezaee and Fazel Zarandi [37]	441	0.026–0.389	400	Not reported
Proposed method	9	0.25	400	0.355

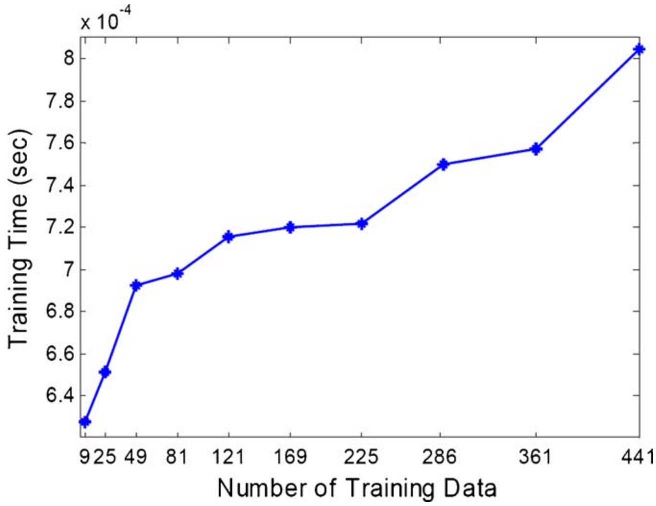

Fig. 14. Required training time.

Our proposed method is also not sensitive to the number of membership functions and consequently fuzzy rules. As shown in Fig. 13, with nine data points, the numbers of fuzzy rules were varied from sixteen to sixty-four while *NRMSE* values remained practically unchanged. Furthermore, if a large number of training data is available, we can still apply the proposed method. In this case, more accurate surfaces for consequences part of fuzzy rules could be obtained. To show the validity of these two advantages, *NRMSE* values for different number of rules as well as different number of training data is calculated. Results are shown in Fig. 13.

As shown in Fig. 13, as the number of the training data increases *NRMSE* values for both testing and training data decreases. Clearly obtaining a lower value of *NRMSE* for testing data is more important than that obtained for training data. Our best case *NRMSE* is 0.2066 which is in the mid-range (0.121–0.289) of all method compared in Table 5.

In addition to reduction in the number of training data, our proposed method has a low learning time. If training data is changed, it will only need to compute the four surfaces for the consequence part of the fuzzy rules. Input membership functions and fuzzy rules will not change. Fig. 14 illustrates the required time for training a new fuzzy model as the number of training data is increased.

Fast learning is especially valuable in real time processes. As an example a biped robot is a non-smooth nonlinear mechanical system [38] which requires a fast method for its modeling and control in different environmental conditions.

5. Conclusions

We proposed a simple and systematic procedure for design of T–S fuzzy systems, based on design of experiments methodology. The significant contribution of the study is the reduction of training data required for the T–S fuzzy systems while obtaining good system performance. The systematic approach facilitates conducting the T–S design in comparison with other methods. Membership functions and fuzzy rules were defined in a straightforward manner. Additionally, the method offers an extremely low learning time.

To demonstrate the effectiveness, the proposed method was applied to control of an inverted pendulum as well as modeling of a nonlinear function. In the case of inverted pendulum, we demonstrated a significant performance improvement. In the case of the nonlinear function modeling, we showed low approximation errors while significantly reducing the training data.

References

[1] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, IEEE Trans. Syst. Man Cybern. 15 (1985) 116–132.
[2] H. Du, N. Zhang, Application of evolving Takagi–Sugeno fuzzy model to nonlinear system identification, Appl. Soft Comput. 8 (1) (2008) 676–686.
[3] C.H. Lien, K.W. Yu, Robust control for Takagi–Sugeno fuzzy systems with time-varying state and input delays, Chaos Soliton Fract. 35 (5) (2008) 1003–1008.
[4] Y.B. Luo, Y.Y. Cao, Y.X. Sun, Robust stability of uncertain Takagi–Sugeno fuzzy systems with time-varying input-delay, Acta Automatica Sinica 34 (1) (2008) 87–92.
[5] W. Hong-Rui, L. Lei, Z. Zi-Hui, Real-time online fuzzy modeling for robotic manipulators, in: International Conference on Machine Learning and Cybernetics 1, 2007, 477–481.
[6] H. Shakouri, G.R. Nadimi, F. Ghaderi, A hybrid TSK-FR model to study short-term variations of the electricity demand versus the temperature changes, Expert Syst. Appl. 36 (2009) 1765–1772.
[7] R. Babu, H. Verbruggen, Neuro-fuzzy methods for nonlinear system identification, Annu. Rev. Control 27 (1) (2003) 73–85.
[8] J. Yen, R. Langari, L.A. Zadeh, Industrial Applications of Fuzzy Control and Intelligent Systems, IEEE Press, New York, 1995.
[9] H. Wongsuwarn, D. Laowattana, Neuro-fuzzy algorithm for a biped robotic system, Int. J. Appl. Math. Comput. Sci. 3 (4) (2006).
[10] D. Katić, M. Vukobratović, Survey of intelligent control techniques for humanoid robots, J. Intell. Robot. Syst. 37 (2003) 117–141.
[11] E.O. Ari, I. Erkmen, A.M. Erkmen, A FACL controller architecture for a grasping snake robot, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005, 1748–1753.
[12] Y. Seung, D. Shin, J.Y. Kim, S. Baek, S.H. Min, Obstacle recognition and collision avoidance of a fish robot based on fuzzy neural networks, in: ICFIE2007 Conference, 2007.
[13] G. Feng, A survey on analysis and design of model-based fuzzy control systems, IEEE Trans. Fuzzy Syst. 14 (5) (2006) 676–697.
[14] T.L. Seng, M.B. Khalid, R. Yusof, Tuning of a neuro-fuzzy controller by genetic algorithm, IEEE Trans. Syst. Man Cybern. B 29 (1999) 226–236.
[15] L. Wang, R. Langari, Complex systems modeling via fuzzy logic, IEEE Trans. Syst. Man Cybern. 26 (1996) 100–106.
[16] J. Jang, ANFIS: adaptive-network-based fuzzy inference system, IEEE Trans. Syst. Man Cybern. 23 (1993) 665–685.
[17] J. Yen, L. Wang, C.W. Gillepie, Improving the interpretability of TSK fuzzy models by combining global learning and local learning, IEEE Trans. Fuzzy Syst. 6 (4) (1998) 530–537.
[18] Y. Jin, Fuzzy modeling of high-dimensional systems: complexity reduction and interpretability improvement, IEEE Trans. Fuzzy Syst. 8 (2) (2000) 212–221.
[19] M.Y. Chen, D.A. Linkens, Rule-base self-generation and simplification for data-driven fuzzy models, Fuzzy Sets Syst. 142 (2) (2004) 243–265.
[20] C.S. Ouyang, W.J. Lee, S.J. Lee, A TSK-type neuro fuzzy network approach to system modeling problems, IEEE Trans. Syst. Man Cybern. B: Cybern. 35 (4) (2005) 751–767.
[21] D.C. Montgomery, Design and Analysis of Experiments, 6th ed., Wiley, 2005.
[22] C.Y. Takeiti, T.G. Kieckbusch, F.P. Collares-Queiroz, Optimization of the jet steam instantizing process of commercial maltodextrins powders, J. Food Eng. 86 (3) (2008) 444–452.
[23] J.B. Tang, H.M. Yang, S.L. Song, P. Zhu, A.G. Ji, Effect of glycine and Triton X-100 on secretion and expression of ZZ–EGFP fusion protein, Food Chem. 108 (2) (2008) 657–662.
[24] N. Jalbani, T.G. Kazi, B.M. Arain, M.K. Jamali, H.I. Afridi, R.A. Sarfraz, Application of factorial design in optimization of ultrasonic-assisted extraction of aluminum in juices and soft drinks, Talanta 70 (2) (2006) 307–314.
[25] R.M. Myers, D.C. Montgomery, Response Surface Methodology, 2nd ed., Wiley, New York, 2002.
[26] A.S. Hedayat, N.J.A. Solane, J. Stufken, Orthogonal Arrays: Theory and Applications, Springer-Verlag, New York, 1999.
[27] M. Landry, S.A. Campbell, K. Morris, C.O. Aguilar, Dynamics of an inverted pendulum with delayed feedback control, SIAM J. Appl. Dyn. Syst. 4 (2) (2005) 333–351.
[28] C.C. Wong, S.M. Her, A self-generating method for fuzzy system design, Fuzzy Sets Syst. 103 (1999) 13–25.
[29] M. Roopaei, M. Zolghadri, S. Meshksar, Enhanced adaptive fuzzy sliding mode control for uncertain nonlinear systems, Commun. Nonlinear Sci. Numer. Simulat. 14 (2009) 3670–3681.
[30] P.C. Chen, C.W. Chen, W.L. Chiang, GA-based fuzzy sliding mode controller for nonlinear systems, in: Mathematical Problems in Engineering, 2008 art. no. 325859.
[31] Y.L. Sun, M.J. Er, Hybrid fuzzy control of robotics systems, IEEE Trans. Fuzzy Syst. 12 (6) (2004) 755–765.
[32] Y. Lin, G.A. Cunningham, S.V. Coggeshall, Using fuzzy partitions to create fuzzy systems from input–output data and set the initial weights in a fuzzy neural network, IEEE Trans. Fuzzy Syst. 5 (4) (1997) 614–621.
[33] J. Gonzalez, I. Rojas, H. Pomares, J. Ortega, A. Prieto, A new clustering technique for function approximation, IEEE Trans. Neural Networks 13 (1) (2002) 132–142.
[34] C.F. Juang, A TSK-type recurrent fuzzy network for dynamic systems processing by neural network and genetic algorithms, IEEE Trans. Fuzzy Syst. 10 (2) (2002) 155–170.

[35] S.J. Lee, C.S. Ouyang, A neurofuzzy system modeling with self-constructing rule generation and hybrid SVD-based learning, IEEE Trans. Fuzzy Syst. 11 (2003) 341–353.
[36] M. Setnes, R. Babuška, U. Kaymak, H.R.v.N. Lemke, Similarity measures in fuzzy rules base simplification, IEEE Trans. Syst. Man Cybern. B: Cybern. B 28 (3) (1998) 376–386.
[37] B. Rezaee, M.H. Fazel Zarandi, A new approach For TSK-type fuzzy model design, in: Fifth International Conference on Fuzzy Systems and Knowledge Discovery, IEEE Computer Society, (2008), pp. 433–437.
[38] Y. Hurmuzlu, F. Genot, B. Brogliato, Modeling, stability and control of biped robots—a general framework, Automatica 40 (2004) 1647–1664.