ELSEVIER

Fuzzy Sets and Systems 104 (1999) 183–197

FUZZY sets and systems

Simplifying fuzzy modeling by both gray relational analysis and data transformation methods

Yo-Ping Huang*, Hung-Chi Chu

Department of Computer Science and Engineering, Tatung Institute of Technology, 40 Chungshan N. Rd, 3rd Sec., Taipei, Taiwan 10451, ROC

Received December 1996; received in revised form May 1997

Abstract

Instead of following the traditional approaches which utilize original data patterns to construct the fuzzy model, this paper proposes to exploit both gray relational analysis and data transformation techniques to simplify the modeling procedures. The transformation method allows us to map the original data to other domains such that there is no need to adjust the membership functions and the fuzzification process is simply taking place on the fixed ones. Since too many system variables involved may complicate the fuzzy modeling, the gray relational method is exploited to select the crucial variables from a finite set of candidates. Based on the calculated relational degrees between the output and the prospective input variables, we can decide which are the important premise variables. The proposed methods have definite effects on the model's performance; therefore, the way to systematically adjust the transformation functions is also investigated. Ease in selecting the premise variables and minimal effort needed to adjust system parameters are the merits of the proposed work. Simulation results from two different examples are presented to demonstrate the superiority of the proposed model to the conventional methodologies.

Keywords: Fuzzy modeling; Membership functions; Structure and parameter identifications; Gradient descent method; Gray relational analysis

1. Introduction

Generally speaking, the identification of a fuzzy model includes the structure and parameter identifications [6, 10, 11]. As we know, the structure identification plays a more important role than the parameter identification in fuzzy modeling. If we know the input candidates to the system, our problem to the fuzzy modeling is almost solved [10]. In case a lot of input variables are to be adopted, the designed fuzzy system may become too complicated. To simplify the modeling procedures, we normally try to neglect the less important input variables before constructing the fuzzy model. However, there is no systematic approach to selecting the appropriate inputs from a finite set of candidates.

Takagi and Sugeno [11] used a tree-like searching method to determine the premise variables. Each time a single prospective variable with its

* Corresponding author. Fax: +886(02)5925252 ext. 2288; e-mail: yphuang@cse.ttit.edu.tw.

0165-0114/99/$ – see front matter
PII: S0165-0114(97)00212-1

domain divided into two fuzzy subspaces was taken into the model to determine the performance index. Among the candidates, the one resulting in the best performance was chosen as the primary variable. Note that the optimization work must be done for each individual model such that the performances can be fairly compared. Starting from the first-round picked variable, each variable (including the primary variable itself) was joined one by one into the primary model to find whether the performance can be further improved. If yes, the one with the strongest contribution to the primary model was selected as the secondary variable. The search of important variables continued until no more improvement of performance index was found or some other preset criteria were met. Although effective, the Takagi–Sugeno algorithm choses the input candidates one by one and has to properly partition the input domains as well as do the optimization work in each step. It was not a time-efficient methodology.

In [10], the authors suggested to use the so-called regularity criterion (RC) to search the premise variables. The regularity criterion defined on two groups of data is formulated as follows:

$$RC = \left[\sum_{i=1}^{k_A} (y_i^A - y_i^{AB})^2 / k_A + \sum_{i=1}^{k_B} (y_i^B - y_i^{BA})^2 / k_B \right] \Big/ 2, \tag{1}$$

where k_A and k_B are the numbers of data in groups A and B, respectively. y_i^A and y_i^B are the output data of groups A and B, respectively. The model output for group A input estimated by the model identified using the group B data is denoted by y_i^{AB}. Similarly, the model output for group B input estimated by the model identified using the group A data is denoted by y_i^{BA}. Similar to the Takagi–Sugeno method, the searching procedures continued until the RC value cannot be further improved by adding other variables. As usual, we had to optimize both fuzzy models to calculate the RC in each step. The final candidates came out from a repeated modeling of fuzzy systems and comparisons of RC values. Again, this was also a time-consuming approach.

To optimize the system parameters several different schemes, such as genetic algorithms [4, 9], gradient descent method [3, 5, 7] and others [1, 12], can be considered. Among the different methods, the gradient descent method allows us to systematically adjust the membership functions. Assuming each fuzzy rule has a crisp consequent part, w_i, and the triangular membership functions are used. A triangular membership function is characterized by the center, a_i, and the base, b_i. Having defined an error function, E, we have to take the partial derivative of the error function with respect to each to-be-adjusted parameter, such as $\partial E/\partial a_i$, $\partial E/\partial b_i$, and $\partial E/\partial w_i$, to decide the quantity of adjustment in the next step. Besides, the learning rate for each parameter has to be decided in advance. A good initial setting for the parameters normally results in a better outcome. However, how to properly set up the initial conditions to expedite the training process remains to be solved.

The paper is organized as follows. Section 2 investigates how the gray relational method can be applied to choose the premise inputs for the fuzzy rules. Having determined the premise variables, we present a new scheme which maps the data patterns from their own defined domains into other domains to simplify the fuzzy modeling in Section 3. How to systematically adjust the transformation functions is also discussed in this section. Simulation results from different approaches for the same problem are compared in Section 4. Conclusions are made in the final section.

2. Choosing premise variables by gray relational method

The gray relational space (GRS) was originally proposed to relate the main factor to the other reference factors in a given system [2, 5]. We applied this technique here to select some input variables which show stronger impact to the system output. Let $\gamma(y_0(k), x_i(k))$ be called the gray relational coefficient at point k between output sequence y_0 and input sequence x_i. The following describe the four basic axioms for the gray relational space [2]:

(1) $\gamma(y_0(k), x_i(k)) \in (0, 1], \forall k$.
(2) $\gamma(y_0(k), x_i(k)) = \gamma(x_i(k), y_0(k))$ if and only if it is a single-input and single-output system.

(3) $\gamma(y_0(k), x_i(k)) \neq \gamma(x_i(k), y_0(k))$ almost holds if and only if it is a multi-input and single-output system.
(4) $\gamma(y_0(k), x_i(k))$ decreases with the increase of $\Delta(k)$, where $\Delta(k) = |y_0(k) - x_i(k)|$.

From the above axioms we can understand that if an input sequence shows a higher similarity to the output one than the others, then this input variable can be said to be more important to the output. To calculate the gray relational degrees between the output and input variables and then to compare the relative importance, the following procedures are usually used:

Step 1. Let the output sequence be $y_0 = (y_0(1), y_0(2), \ldots, y_0(n))$, where n stands for the number of data.

Step 2. Denote the m sequences to be compared by $x_i = (x_i(1), x_i(2), \ldots, x_i(n))$, $i = 1,2, \ldots, m$.

Step 3. Calculate

$$\gamma(y_0(k), x_i(k)) = \frac{\min_j \min_k |y_0(k) - x_j(k)| + \zeta \max_j \max_k |y_0(k) - x_j(k)|}{|y_0(k) - x_i(k)| + \zeta \max_j \max_k |y_0(k) - x_j(k)|},$$

where $\zeta \in (0,1]$ is the distinguishing coefficient and $\zeta = 0.5$ is normally used. $j = 1,2, \ldots, m$. $k = 1,2, \ldots, n$. $\gamma(y_0(k), x_i(k))$ is called the gray relational coefficient at point k. Note that a normalization operation on the data sequences is normally required since the range or unit in one data sequence may differ from the others. Aggregating the gray relational coefficient calculated at each point, we can obtain the gray relational grade for an entire sequence.

Step 4. The gray relational grade between the output and a specific input sequence is derived as follows:

$$\gamma(y_0, x_i) = \frac{1}{n} \sum_{k=1}^{n} \gamma(y_0(k), x_i(k)).$$

Here, $\gamma(y_0, x_i)$ represents to what degree of influence the sequence x_i can exert on the output sequence y_0. In other words, the system output can grasp some useful information about the variation of data points from input sequences. Thus, the analysis of gray relational grade provides us an alternative to decide which input variables show the crucial effect to the output. This in turn can be applied to identifying the structure of a fuzzy model.

To verify the applicability of the gray relational method to the selection of input variables, an example simulating the control action of a human operator on water cleaning process [11] is investigated. The purpose of the water cleaning process is to determine the amount of PAC to be added to the water. The six variables influencing the sedimentation process are: TUB1 (turbidity of the original water), TUB2 (turbidity of the treated water), PAC (amount of PAC), TE (temperature of water), PH, and AL (alkalinity). In [11], the authors used about 600 data patterns to construct the fuzzy model but only part of the data were given in their paper. Table 1 lists those 19 data patterns. Having initialized those 19 data patterns, we can analyze the gray relational degrees and then easily decide which are the crucial premise variables. Table 2 compares the gray relational degrees between the variables influencing the water cleaning process when the distinguishing coefficient $\zeta = 0.5$ is used.

Based on the results listed in Table 2, it is very clear that the gray relational degrees between PAC and PH, TE, or AL are larger than to TUB1 and TUB2. This implies that PH, TE, and AL have stronger impact to both TUB1 and TUB2 than to PAC. As a result, PH, TE, and AL are chosen as the premise variables in the fuzzy rules. This conforms to the results analyzed by the tree-like searching method given in [11]. From the viewpoint of task required in choosing the influential variables, the gray relational method is much simpler than the previous approach.

Having selected the effective variables, we then apply the linear regression method [8] to verifying whether the chosen variables can faithfully play their roles in approximating the real data. Since the variable PH shows the strongest relationship to PAC, we use PAC and PH to build the first regression model and obtain the following result:

$$\text{PAC} = 4332.5 - 454.5\text{PH}. \quad (2)$$

Based on the above equation, the average of absolute errors for PAC is 123.39. Similarly, by adding the variable TE to the first regression model we

Table 1
The 19 data patterns used for the water cleaning process [11]

TUB1	PH	TE	AL	PAC	TUB2	PAC from our model	Abs(error)
10.0	7.1	18.8	53.0	1300	1.0	1272.75	27.25
17.0	7.0	18.6	50.0	1300	1.0	1293.32	6.68
22.0	7.3	19.4	46.0	1400	2.0	1364.75	35.25
50.0	7.1	19.5	40.0	1400	1.0	1370.49	29.51
9.0	7.3	23.3	48.0	900	4.0	896.75	3.25
11.0	7.1	20.7	50.0	900	1.0	943.89	43.89
12.0	7.2	21.3	50.0	900	3.0	890.95	9.05
14.0	7.2	23.6	53.0	900	4.0	908.62	8.62
35.0	7.0	17.8	35.0	1200	1.0	1182.03	17.97
20.0	7.0	16.6	40.0	1100	1.0	1053.20	46.8
20.0	6.9	17.8	42.0	1100	1.0	1058.62	41.38
18.0	7.1	17.3	40.0	1100	1.0	1092.30	7.7
12.0	7.2	18.8	55.0	900	3.0	876.25	23.75
8.0	7.2	18.0	50.0	1000	1.5	1073.12	73.12
11.0	7.1	19.2	49.0	1000	2.0	1032.42	32.42
50.0	7.0	18.0	37.0	1200	1.5	1239.83	39.83
35.0	7.0	17.7	42.0	1200	1.5	1155.67	44.33
30.0	7.0	17.3	41.0	1100	1.5	1164.71	64.71
16.0	7.1	19.3	42.0	1100	3.0	1105.07	5.07

Table 2
The gray relational degrees obtained for the water cleaning process

	TUB1	PH	TE	AL	PAC	TUB2
TUB1	1.0000	0.7292	0.7332	0.7118	0.6964	0.6335
PH	0.7185	1.0000	0.9699	0.9343	0.9267	0.7795
TE	0.7240	0.9701	1.0000	0.9270	0.9259	0.7709
AL	0.7118	0.9383	0.9310	1.0000	0.9277	0.7430
PAC	0.6870	0.9277	0.9264	0.9241	1.0000	0.7440
TUB2	0.6190	0.7791	0.7695	0.7316	0.7412	1.0000

have the following result:

$$\mathrm{PAC} = 1747.9 + 19.5\mathrm{PH} - 40.9\mathrm{TE} \tag{3}$$

The average of absolute errors for PAC is 116.71. When the variable AL is added into the first regression model we derive another result:

$$\mathrm{PAC} = 2420.9 - 113.7\mathrm{PH} - 11.2\mathrm{AL}, \tag{4}$$

and the average of absolute errors for PAC is 117.62. Due to the fact that variable TE has a higher relational degree than the variable AL to PAC, the average of absolute errors from Eq. (3) is a little bit smaller than from Eq. (4). When both TE and AL are added to the first model we have

$$\mathrm{PAC} = 764.6 + 192.4\mathrm{PH} - 33.2\mathrm{TE} - 8.6\mathrm{AL}, \tag{5}$$

and the average of absolute errors for PAC is 111.73. If all the influential variables are taken into account, we obtain the final regression model as follows:

$$\mathrm{PAC} = -2786.5 + 553.9\mathrm{PH} - 10\mathrm{TE} + 3.5\mathrm{AL} - 91.5\mathrm{TUB2} + 7.7\mathrm{TUB1}, \tag{6}$$

and the average of absolute errors for PAC reduces to 82.28. Although the averages of absolute errors keep on decreasing when more and more variables are adopted, the results from the linear regression models are still far from our expectation.

3. Data transformation technique

Traditionally, a fuzzy model can be directly constructed from the given data patterns [1, 6, 10–12]. To obtain a satisfactory performance, both the structure and parameter identifications of the fuzzy model are indispensable. Instead of attempting to directly optimize the membership functions to satisfy the given data, we approach from the opposite direction which transforms the data to other domains to satisfy the fixed membership functions. As a result, the tuning of membership functions can be ignored. This simple but effective transformation method allows us to take the equally partitioned fuzzy sets to fuzzify the transformed data such that the firing degree of each fuzzy rule can be easily calculated.

The given data patterns are transformed by a function, $k(x)$, to another domain. The transformed data are then fuzzified by the equally partitioned fuzzy sets positioned on the universe of discourse. Note that the universe of discourse is now defined for the transformed data, not for the original ones. Fig. 1 compares the proposed method with the conventional approaches. Since the inferred outputs from the proposed model heavily depend on the transformed scheme, a more systematic approach to adjusting the transformation functions is necessary.

Different types of fuzzy rules have been proposed before. In this paper a fuzzy rule has the following type:

Rule i. If x_1 is A_{1i} and x_2 is A_{2i} and ... and x_n is A_{ni}, then y is w_i.

Note that x_i's are input variables, y is the output variable, A_{ji}'s are linguistic descriptors, and w_i's are singletons. Assume that the product operation is used, the firing degree of a data pattern from the antecedent part of the ith rule is obtained as follows:

$$\mu_i = \prod_{j=1}^{n} A_{ji}(k(x_j)), \tag{7}$$

where $A_{ji}(k(x_j))$ represents the membership degree for the transferred x_j on fuzzy set A_{ji}. Based on the firing degrees, we can infer the desired output from an m-rule base as follows:

$$y_c = \left(\sum_{i=1}^{m} \mu_i w_i\right) \Bigg/ \sum_{i=1}^{m} \mu_i. \tag{8}$$

From the inferred output we can define the following error function for a given data pattern:

$$E = \tfrac{1}{2}(y_c - y_d)^2, \tag{9}$$

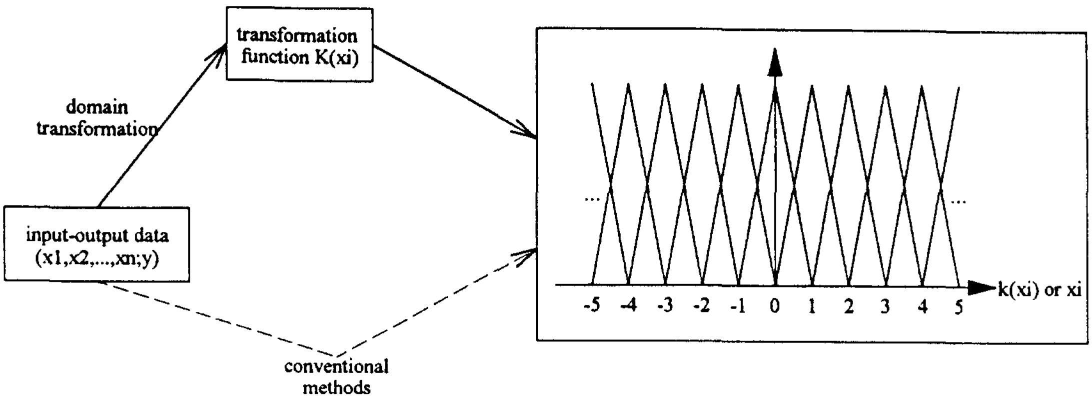

Fig. 1. The comparison between the proposed and conventional approaches.

where y_d denotes the actual output. Assume that the parameters to be adjusted in the transformation functions are p_j's. The quantity of p_j to be adjusted in each iteration can be obtained from $\partial E/\partial p_j$. By using the chain rule we have the following result:

$$\frac{\partial E}{\partial p_j} = \frac{\partial E}{\partial y_c}\frac{\partial y_c}{\partial \mu_i}\frac{\partial \mu_i}{\partial A_{ji}(k(x_j))}\frac{\partial A_{ji}(k(x_j))}{\partial k(x_j)}\frac{\partial k(x_j)}{\partial p_j}. \tag{10}$$

As long as the quantity of p_j to be adjusted has been determined, the new p_j can be decided:

$$p_j(t+1) = p_j(t) - \gamma_p \frac{\partial E}{\partial p_j}, \tag{11}$$

where γ_p is the learning rate.

Similarly, the consequent part, w_i, in each fuzzy rule is randomly assigned at the beginning, it needs to be adjusted, too. By using the chain rule again, the following can be derived:

$$\begin{aligned} w_i(t+1) &= w_i(t) - \gamma_w \frac{\partial E}{\partial w_i} \\ &= w_i(t) - \gamma_w \frac{\partial E}{\partial y_c}\frac{\partial y_c}{\partial w_i}, \end{aligned} \tag{12}$$

where γ_w is the prearranged learning rate.

4. Simulation results and analyses

Let us use the following double-input and single-output system given in [6, 10] for simulations:

$$y = (1 + x_1^{-2} + x_2^{-1.5})^2, \quad 1 \leqslant x_1, \quad x_2 \leqslant 5 \tag{13}$$

The 50 data patterns listed in [10] are redrawn in Fig. 2. In [10], two other dummy variables, x_3 and x_4, are also included to verify the applicability of the regularity criterion. By using the gray relational method we can also derive the same result that both x_1 and x_2 are more important than the other two. Since the domains of both inputs, x_1 and x_2, range 1–5, we use the following transformation function to map the original data to another domain:

$$k(x_j) = 6/[1 + \mathrm{e}^{-p_j(x_j-1)}], \quad j = 1, 2 \tag{14}$$

Eq. (14) indicates that $k(x_j)$ has a value of 3 when x_j equals 1. When x_j is large enough and the parameter p_j is positive, $k(x_j)$ approaches 6. In case the parameter p_j is adjusted to a negative value, $k(x_j)$ drops from 3 to nearly zero. This suggests that we set up the domain of $k(x_j)$ between zero and 6 to cover every possibility. Since each transformed data pattern will activate at least one rule, we can calculate the quantity of adjustment for the

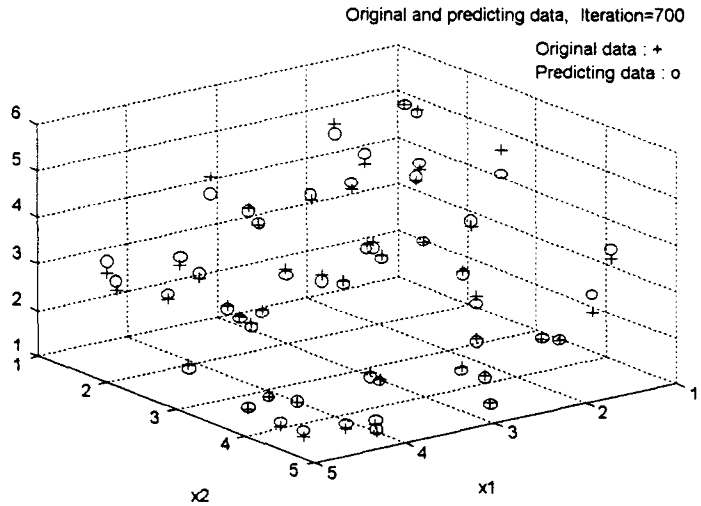

Fig. 2. The 50 data given in [10] and the corresponding training outputs when each variable has five labels.

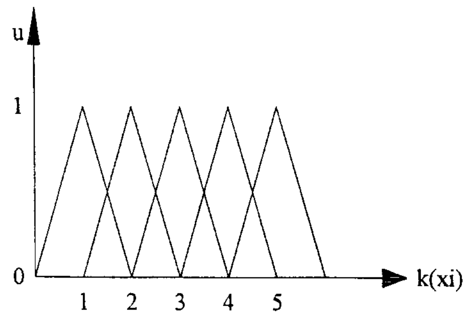

Fig. 3. The fixed membership functions defined for the transferred data.

parameter p_j in each cycle from Eq. (10) as follows:

$$\frac{\partial E}{\partial p_j} = (y_c - y_d)\left(\frac{w_i - y_c}{\sum \mu_i}\right)\left(\frac{\mu_i}{A_{ji}(k(x_j))}\right) \times \left(\frac{-sgn(k(x_j) - a_{ji})}{b_{ji}}\right)\left(x_j - 1\right)k(x_j)\left(1 - \frac{k(x_j)}{6}\right) \tag{15}$$

where a_{ji} and $2b_{ji}$ represent the center and base of the fixed triangular membership function, respectively, and $sgn(d)$ stands for the sign of d. Besides, from Eq. (12) we can obtain

$$\frac{\partial E}{\partial w_i} = (y_c - y_d)\frac{\mu_i}{\sum \mu_i}. \tag{16}$$

The initial values for both p_j's are 0.05, the learning rate γ_p in Eq. (11) is kept at 0.02, and the learning rate γ_w in Eq. (12) is set to 0.05. Since there are two inputs and both have the same domain, only one transformation function is needed and the quantity of adjustment for p simply takes the average of $\partial E/\partial p_1$ and $\partial E/\partial p_2$, i.e.,

$$p(t+1) = p(t) - \gamma_p\left(\frac{\partial E}{\partial p_1} + \frac{\partial E}{\partial p_2}\right)\Big/2. \tag{17}$$

In case the input variables have different domains, different transformation functions should be used.

In performing the simulations, we assume that the transferred domain of x_1 or x_2 is equally partitioned into five labels and the base for each isosceles membership function has 50% overlap with its neighbors. Fig. 3 plots the five fixed membership functions defined for the transferred inputs $k(x_1)$ and $k(x_2)$. Assigning five labels for each variable may result in at most 25 rules generated in the fuzzy rule base. The inferred outputs from the five-label case for the 50 data patterns are also drawn in Fig. 2 for comparison. The differences between the actual and the inferred outputs are depicted in Fig. 4. In [10], the mean square errors, i.e., the performance index (PI), from before and after parameter identifications were 0.318 and 0.079, respectively. The same data set run by the linguistic fuzzy

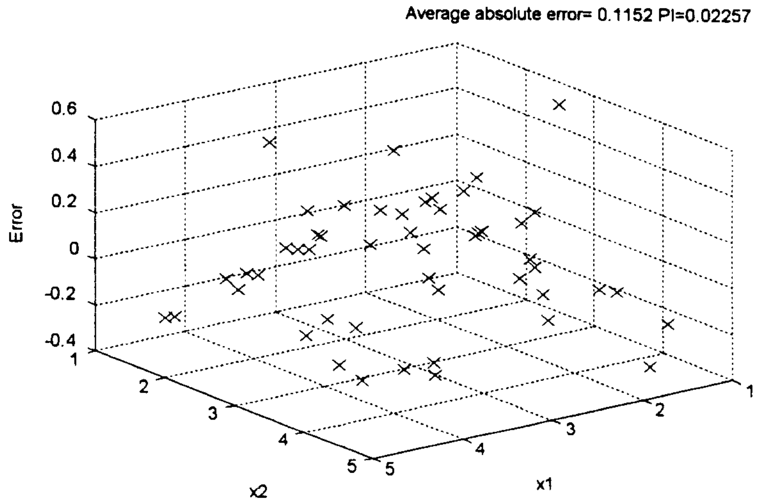

Fig. 4. The training errors for the 50 training data in the five-label case.

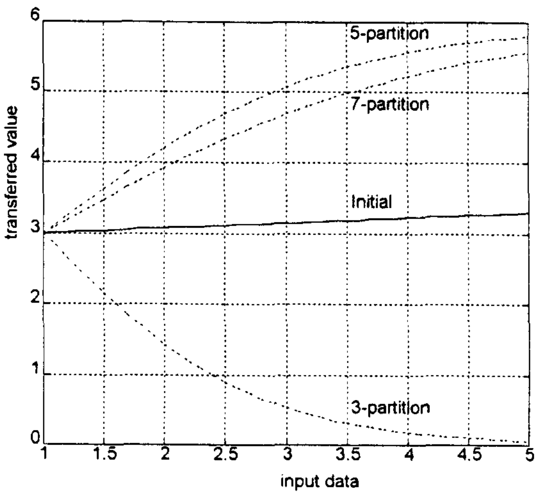

Fig. 5. The curves for the initial and finally-adjusted transformation functions in three-, five-, and seven-label cases. The initial p_j's are 0.05 for all cases and the final values are −1.163617, 0.6373, and 0.85068 for the three-, five-, and seven-label cases, respectively.

model [6] had an error of 0.073, while our model only produces an error of 0.02257 which is the smallest among the methods just mentioned. Fig. 5 plots the curves for the initially given and after adjusted transformation functions. Obviously, the transferred values from the final transformations functions cover either the range of 3–6 or 3–0 depending on the sign of parameter p_j. This in turn demonstrates that at most half of the labels are fired by the input data and the rest remains unfired. Those unfired labels can be treated as dummy ones; therefore, only 9 out of 25 fuzzy rules are actually used for the five-label case. In the sequel, a combination of transformed technique with the adjustment of consequent parts allows us to obtain a fascinating result.

To observe the effects of coarser and finer partitions of input domains on the final results, two other simulations are performed. When each input domain is partitioned into three and seven subsections, the training errors are plotted in Figs. 6 and 7, respectively. The curves for the initially given and post-adjusted transformation functions are also depicted in Fig. 5 for comparisons. Based on the results shown in both Figs. 6 and 7, it is clear that a coarser partition of the input domain worsens the outcome. On the contrary, a finer partition gives a better result. For the seven-label case, the final training error is 0.008385 which is even smaller than using the position-gradient type in [10]. Note that in the above simulations the iteration number was set to 700. Extending the iterations may further improve the results. However, the relationship

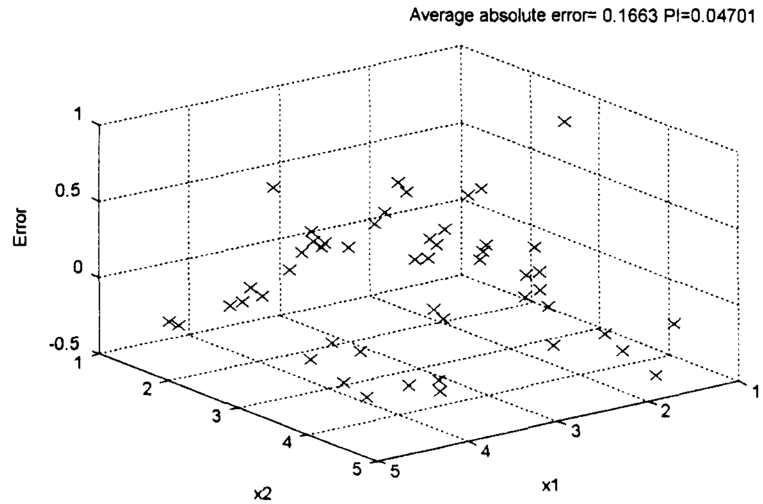

Fig. 6. The training errors for the 50 training data in the three-label case.

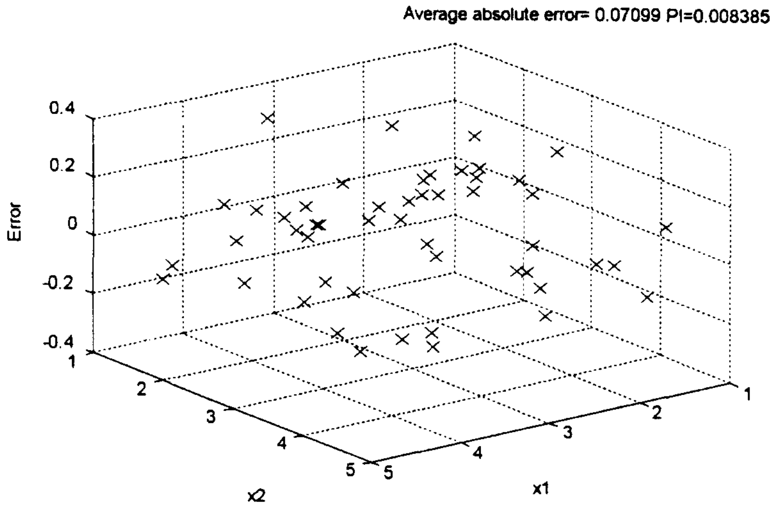

Fig. 7. The training errors for the 50 training data in the seven-label case.

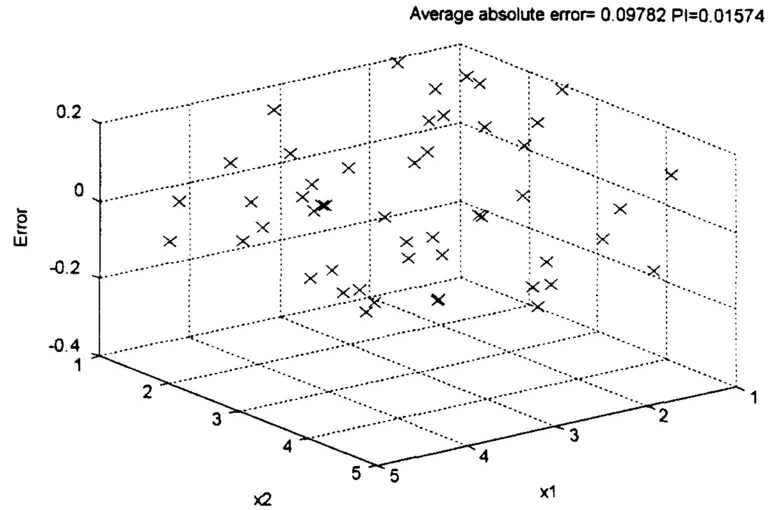

Fig. 8. The training errors caused by the gradient descent method.

between the number of partitions and the final results is implicit.

Based on the case of five labels for each input variable, Fig. 8 displays the training errors caused by the gradient descent method running for 2500 iterations. In using the gradient descent method, the learning rates for the centers and the bases of the triangular membership functions are, respectively, fixed at 0.01 and 0.001 and the learning rate for the consequent parts in the fuzzy rules is 0.001. The mean squared errors resulted from different methods are compared in Table 3. From the viewpoints of mean squared errors and the needed task in modeling, it is obvious that the proposed method performs very satisfactorily.

Table 3
The comparison of mean squared errors (MSE) from different methods

Method	MSE
Our model (3-label for each input)	0.04701
Our model (5-label for each input)	0.02257
Our model (7-label for each input)	0.008385
Gradient descent method	0.01574
From [10] – without parameter identification	0.318
From [10] – with parameter identification	0.079
From [10] – position-gradient type	0.010
From [6]	0.073

We next use 441 training data plotted in Fig. 9 to construct another fuzzy model. The same transformation function as formulated in Eq. (14) is used and the five-label case is investigated. After running for 500 times, we plug the 1681 test data shown in Fig. 10 to the established fuzzy model and obtain the results given in Fig. 11. The inferred errors are plotted in Fig. 12. The mean squared errors from our model are only 0.03653. Carefully observing the results shown in Fig. 12, we found a peak error occurring at both x_1 and x_2 equal to 1 which corresponds to a maximum output 9 for y in Eq. (13). This implies that more efforts should be taken to minimize the error.

When the transformation function is slightly modified to the following:

$$k(x_j) = 6/[1 + e^{-p_j(x_j - 3)}], \quad j = 1,2 \tag{18}$$

and the initial consequent values in the fuzzy rules are subjectively made, we have the results plotted in Figs. 13–16. Obviously, different transformation functions may not work as well for the same problem. This is due to the fact that Eq. (18) has little effect to change the input when x_j is close to 3, especially at 3. Thus, the transformation function has a definite effect on the system performance. How to systematically choose the transformation functions holds a challenge. To see how much improvement the proposed model has done to the system, the original data shown in Fig. 10 were plugged directly into the fixed fuzzy sets shown in Fig. 3 (except the domain for x_j) to infer the outputs. Figs. 17 and 18 depict the inferred outputs and errors, respectively. As can be seen, the results are very poor. This demonstrates that our method can turn a poor model into a well-behaved one.

We now use the data given in Table 1 to further validate our model. Only three variables, PH, TE, and AL, are exploited to determine the PAC. Since the minimum and maximum values for PH are 6.9 and 7.5, respectively, the transformation function is

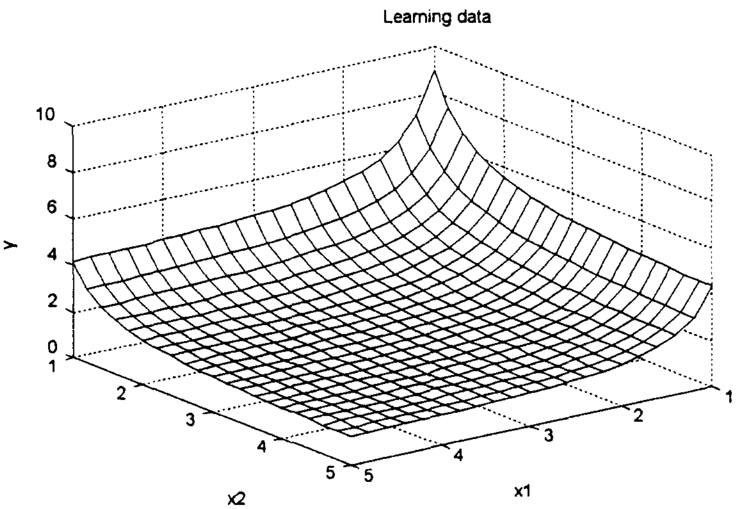

Fig. 9. The 441 training data used for another simulation.

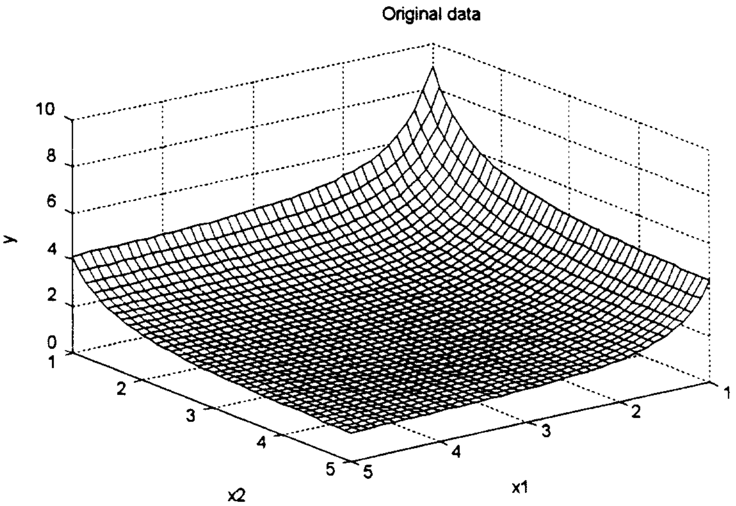

Fig. 10. The 1681 data used for the test simulation.

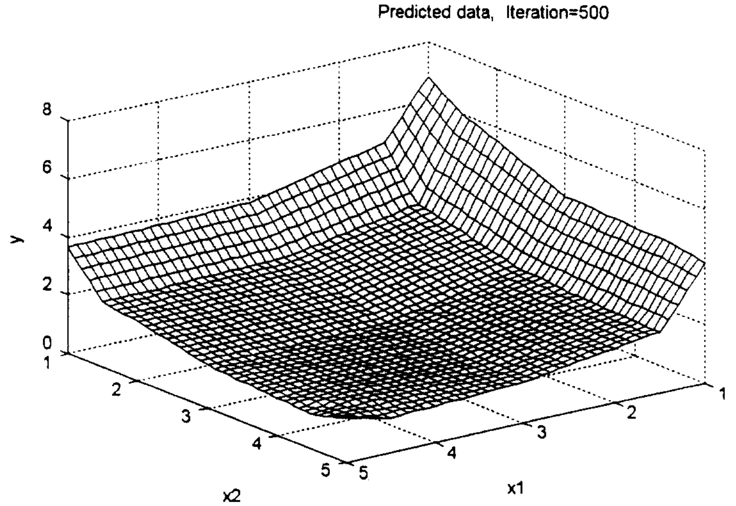

Fig. 11. The 1681 test outputs from our model.

formulated as follows:

$$k(x_1) = 7.5/[1 + e^{-p_1(x_1 - 6.9)}]. \tag{19}$$

The centers of the five labels are located at 1, 3, 5, 7, and 9. The initial value and learning rate for the parameter p are 0.05 and 0.0002, respectively. Similarly, the transformation functions for TE is

$$k(x_2) = 23.6/[1 + e^{-p_2(x_2 - 16.6)}], \tag{20}$$

and the five labels are centered at 9, 11, 13, 15, and 17. Similarly, the transformation functions for AL is

$$k(x_3) = 60/[1 + e^{-p_3(x_3 - 35)}], \tag{21}$$

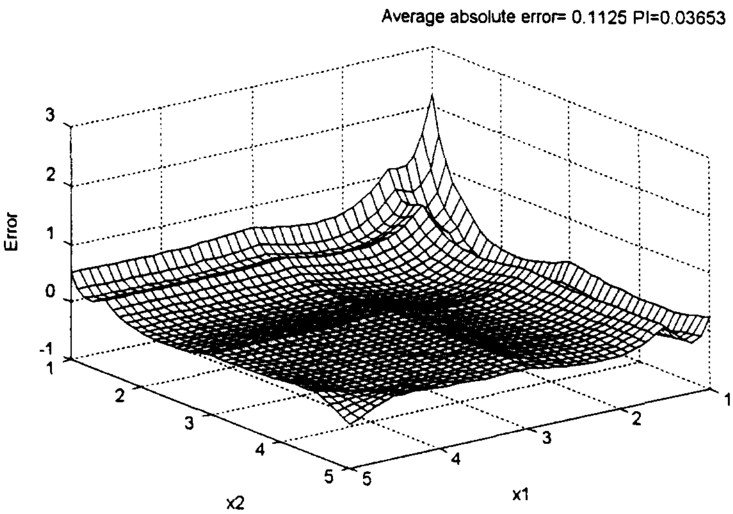

Fig. 12. The 1681 test errors from our model.

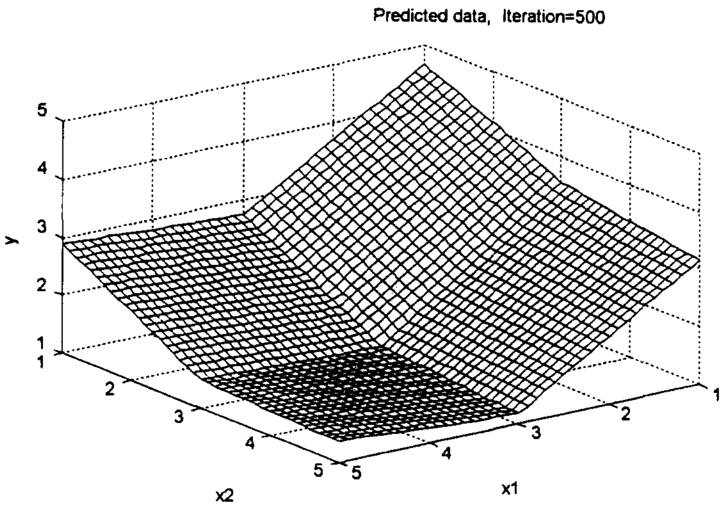

Fig. 13. The test outputs resulted from using the transformation function given in Eq. (18) and randomly setting the initial consequent values in the fuzzy rules.

and the centers of the five labels are positioned at 22, 27, 32, 37, and 42. We stopped the iterations at 18 500 and obtained an average absolute error of 29.50 for the PAC. The last two columns of Table 1 list the inferred outputs and absolute errors from our model. It is clear that our result outperforms the linear regression model. Again, our transformation method always allows us to establish a fascinating fuzzy model.

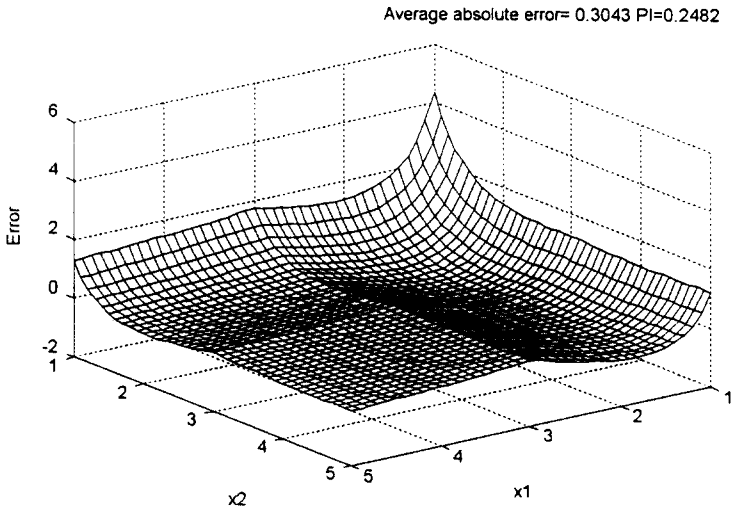

Fig. 14. The test errors resulted from using the transformation function given in Eq. (18) and randomly setting the initial consequent values in the fuzzy rules.

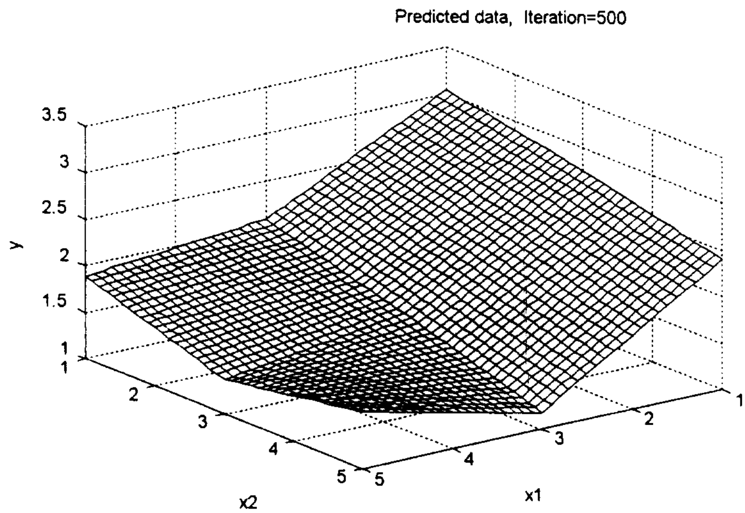

Fig. 15. The test outputs resulted from using the transformation function given in Eq. (18) and setting the same initial consequent values in the fuzzy rules at 6.

5. Conclusions

We have presented a transformation technique which maps the inputs into other domains to simplify the fuzzy modeling. The transferred data were then fuzzified by the fixed set of membership functions. Instead of adjusting the membership functions to satisfy the given data, we optimized the transformation functions to fuzzify our data such that a satisfactory performance can also be maintained. Some simulation results were given to verify our new approach. Besides, in the structure

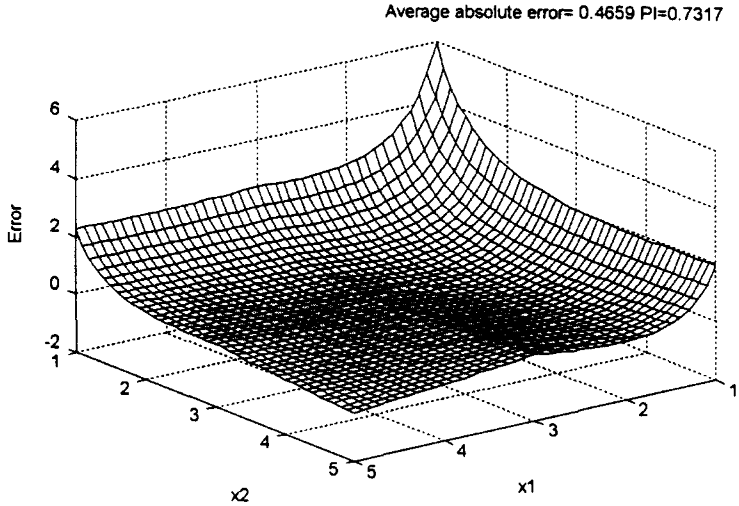

Fig. 16. The test errors resulted from using the transformation function given in Eq. (18) and setting the same initial consequent values in the fuzzy rules at 6.

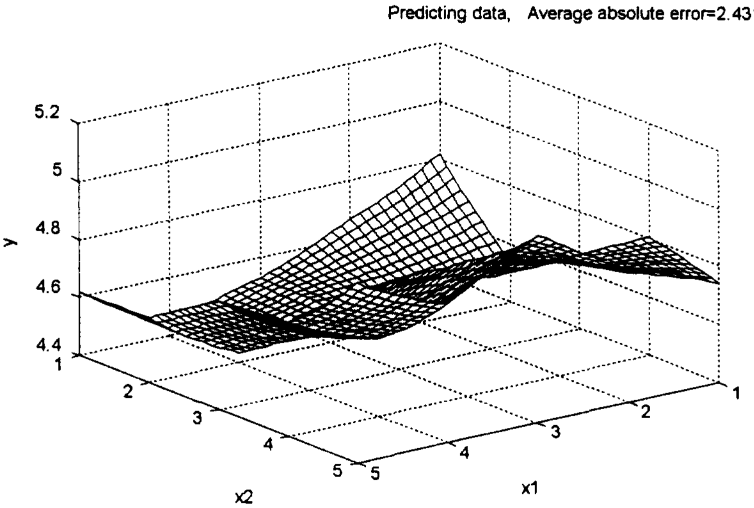

Fig. 17. The training outputs obtained from directly plugging the original data into the fixed five-label fuzzy model.

identification of a fuzzy model, the gray relational method played an important role in choosing some crucial premise variables from a finite set of candidates. Examples were given to demonstrate that the gray relational method can work satisfactorily in selecting the premise variables. Different transformation functions may show different effects to the results. By observation, the distribution of available data can normally give us some ideas as how to create the transformation functions. What kinds of transformation functions fit the given data patterns better or how to mathematically formulate

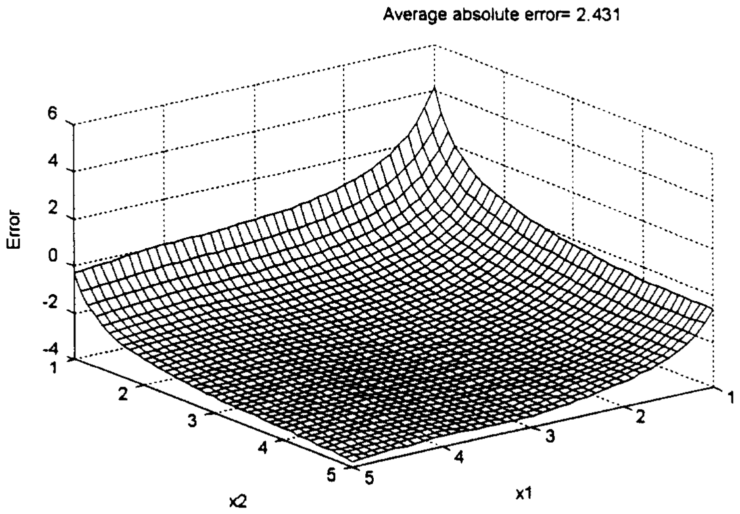

Fig. 18. The test errors obtained from direction plugging the original data into the fixed five-label fuzzy model.

the transformation functions remains to be solved. This is under our investigation. The work presented here is intended to provide the readers a new direction to model a fuzzy system.

Acknowledgements

This work was supported by the National Science Council, Taiwan, ROC, under Grant NSC 86-2212-E-036-006.

References

[1] K.C. Chan, G.C.I. Lin, S.S. Leong, A more accurate adaptive fuzzy inference system, Comput. Industry 26 (1995) 61–73.
[2] J.L. Deng, Introduction to gray system theory, J. Grey System 1(1) (1989) 1–24.
[3] F. Guely, P. Siarry, Gradient descent method for optimizing various fuzzy rule bases, Proc. FUZZ-IEEE (1993) 1241–1246.
[4] A. Homaifar, E. McCormick, Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms, IEEE Trans. Fuzzy Systems 3 (2) (1995) 129–139.
[5] Y.-P. Huang, C.-C. Huang, The integration and application of fuzzy and gray modeling methods, Fuzzy Sets and Systems 78(1) (1996) 107–119.
[6] H.-S. Hwang, K.-B. Woo, Linguistic fuzzy model identification, IEE Proc.-Control Theory Appl. 142(6) (1995) 537–544.
[7] H. Nomura, I. Hayashi, N. Wakami, A learning method of fuzzy inference rules by descent method, Proc. FUZZ-IEEE (1992) 203–210.
[8] L. Ott, An Introduction to Statistical Methods and Data Analysis, 3rd ed., Pws-Kent Publishing Company, Boston, MA, USA, 1988.
[9] K. Shimojima, T. Fukuda, Y. Hasegawa, Self-tuning fuzzy modeling with adaptive membership function, rules, and hierarchical structure based on genetic algorithm, Fuzzy Sets and Systems 71 (1995) 295–309.
[10] M. Sugeno, T. Yasukawa, A fuzzy-logic-based approach to qualitative modeling, IEEE Trans. Fuzzy Systems 1(1) (1993) 7–31.
[11] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, IEEE Trans. Systems Man Cybernet. 15(1) (1985) 116–132.
[12] L.-X. Wang, J.M. Mendel, Generating fuzzy rules by learning from examples, IEEE Trans. Systems Man Cybernet. 22(6) (1992) 1414–1427.