International Journal of Fuzzy Logic and Intelligent Systems, vol. 7, no. 3, September 2007 pp. 182-187

Fuzzy Logic Modeling and Its Application to A Walking-Beam Reheating Furnace

Bin Zhang[1] and Jingcheng Wang[2]

1. School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30341
2. Department of Automation, Shanghai Jiaotong University, Shanghai China, 200240

Abstract

A fuzzy modeling method is proposed to build the dynamic model of a walking-beam reheating furnace from the recorded data. In the proposed method, the number of membership function on each variable is increased individually and the modeling accuracy is evaluated iteratively. When the modeling accuracy is satisfied, the membership functions on each variable are fixed and the structure of fuzzy model is determined. Because the training data is limited, in this process, as the number of membership function increase, it is highly possible that some rules are missing, i.e., no data in the training set corresponds to the consequent part of a missing rule. To complete the rulebase, the output of the model constructed at the previous step is used to generate the consequent part of the missing rules. Finally, in the real time application, a rolling update scheme to rulebase is introduced to compensate the change of system dynamics and fine tune the rulebase. The proposed method is verified by the application to the modeling of a reheating furnace.

Key Words : Fuzzy System, Structure Identification, Complete Rulebase, Rulebase Update, Walking-Beam Reheating Furnace Model

1. Introduction

Fuzzy logic modeling has been proposed as a viable alternative of traditional modeling approach and successfully employed in various fields. Due to this, considerable efforts have been devoted to the fuzzy modeling approaches in the past several decades.

Takagi, Sugeno and Kang proposed a method to combine the linguistic description with available mathematical description of the process to construct a fuzzy system model [1, 2]. Wang and Mendel proposed a general method to generate fuzzy rules from the data and then to design the fuzzy systems [3]. The key of this approach is that fuzzy rules can be extracted from recorded input-output data pairs and combined with linguistic rules to create the rulebase. The drawback is that the designer must divide the input space into fixed sections, and must decide the number of membership function in advance. To solve this difficulty, Nie proposed an approach to construct a multivariable fuzzy model from numerical data through a self-organizing counter-propagation network [4]. Abe and Lan developed a method to extract fuzzy rules directly from numerical input-output data for pattern classification and function approximation [5, 6]. Some optimization methods are also employed in fuzzy modeling [7-12]. In these approaches, the fuzzy rules are defined by activation hyperboxes which show the existence region of data for a class and are extracted by resolving overlaps between two classes recursively.

The existed methods depend on the well-distributed numerical data to extract rules. But, in practice, the numerical data are often unevenly distributed, or sparsely sampled. It is highly likely that the obtained rulebase is incomplete and even human experts cannot complete it. In this paper, we proposed a method to determine the structure of the fuzzy model by increasing the number of membership functions on each variable individually and step by step, and evaluating the performance of model at each step. To complete the missing rules, at the beginning, a small number of membership functions on input variable are defined such that the obtained rulebase is complete or can be completed by prior knowledge easily. Then, with the increase of fuzzy sets on input, a rulebase with more rules can be obtained. If the generated rulebase is incomplete, the model constructed at previous step is used to complete the rulebase.

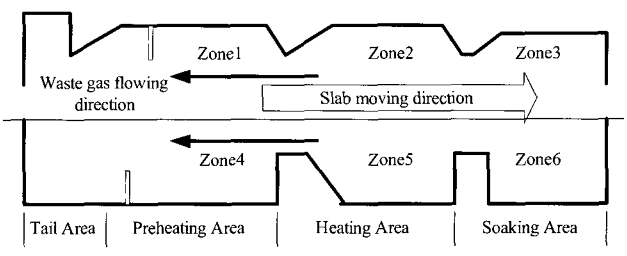


Figure 1. The structure of the walking-beam reheating furnace

Although this method can generate complete rulebase, the uncertainties and the changes of system dynamics in practice will definitely degrade the modeling performance. In addition, as new data come in, the rulebase should be adjusted accordingly to improve the modeling performance. Therefore, in

Manuscript received Aug. 3, 2007; revised Sep. 10, 2007.
This work was supported by National 863 Plan of China (No. 2006AA040308, No. 2007AA041403) and Shanghai Rising-Star Program (No. 07QA14030).

real time application, a certain length of past-recorded data is use to fine tune the rulebase and these rules are used to update the rulebase. Since these newly generated rules contain the present system dynamic information, the model output will become more accurate.

2. Problem Description

Walking-beam reheating furnace is an important device in the steel industry. In the furnace, all slabs are heated to reach a predefined discharging temperature and a balance of temperature distribution in slabs. The distribution and move of the slab in the furnace is subject to the heating capacity of the reheating furnace and state of rolling line such as rolling pacing, etc. The structure of the walking beam reheating furnace discussed here is shown in Figure 1. It contains tail area, preheating area, heating area, and soaking area. Tail area has no fuel input and the slabs in this zone are heated by waste gas. The function of preheating and heating areas is to heat the slabs. The aim of soaking area is to adjust the temperature gradient so that the inner temperature and surface temperature of the slabs can reach a balance. During the operating, slabs in the furnace move from tail area to soaking area. For modeling purpose, each area is further divided into upper and lower zones. The zones in the furnace are denoted as zone1 to zone6, respectively.

Due to the high nonlinearity and various disturbances, it is difficult to build the model by using conventional mathematical methods. The universal approximation theorem [13] provides a theoretical basis for using fuzzy system to build the dynamic model of the reheating furnace. In this paper, the model of each zone is constructed individually. The output and inputs are decided in advance as follows:

$y(k+1)$ = the predictive temperature of a zone at time instant $k+1$, the output of the system.

$x1(k)$ = heat absorbing ability of slabs in a zone at instant k.

$x2(k)$ = the fuel flux of a zone at time instant k.

$x3(k)$ = the temperature of a zone at instant k

The system model has the form of:

$y(k+1) = f(x1(k), x2(k), x3(k))$

3. The Determination of the Fuzzy Model Structure

The structure of the fuzzy model is determined as follows.

Step 1: Define a small number (for example, two) of membership functions on each input variable.

Without loss of generality, the evenly spread membership functions with triangular shape are defined to cover each variable. At this stage, the numbers of membership functions on these variables are not fixed. For simplicity, these variables are denoted as unfixed variable.

Step 2: Construct the fuzzy system based on Table Lookup method [3] on training data, and evaluate its performance on both training data and testing data.

For the mentioned walking-beam reheating furnace, at time instant k, the extracted fuzzy rules can be expressed as follows:

R^k : IF $x_1(k)$ is $A_1^{\,l}$ and $x_2(k)$ is $A_2^{\,m}$ and $x_3(k)$ is $A_3^{\,n}$, THEN $y(k)$ is B^j.

where $x_1(k)$, $x_2(k)$, $x_3(k)$ are the input signal, $y(k)$ is the output signal. $A_1^{\,l}$, $A_2^{\,m}$, $A_3^{\,n}$ are the fuzzy sets on the input variables x_1, x_2, x_3, respectively. B^j is the fuzzy sets on the output variable. Then, the model can be constructed and the modeling accuracy is evaluated.

Step 3: Increase the number of membership functions on an unfixed variable, then repeat step 2 to decide whether to keep this increase operation or not.

If the performance improves, then this increase operation is kept and the number of membership functions on this variable remains unfixed and will be increased in the following iteration. If the performance does not improve or the improvement is below a predefined threshold, then this increase operation is cancelled. In this case, the number of membership functions on this variable is determined and this variable is marked as a fixed variable. The number of membership functions on this variable will not be increased in the following iteration.

Step 4: Repeat step 3 until all the variables are marked as fixed variables. Then the structure of the model is determined.

The flow chart of this process is illustrated in Figure 2.

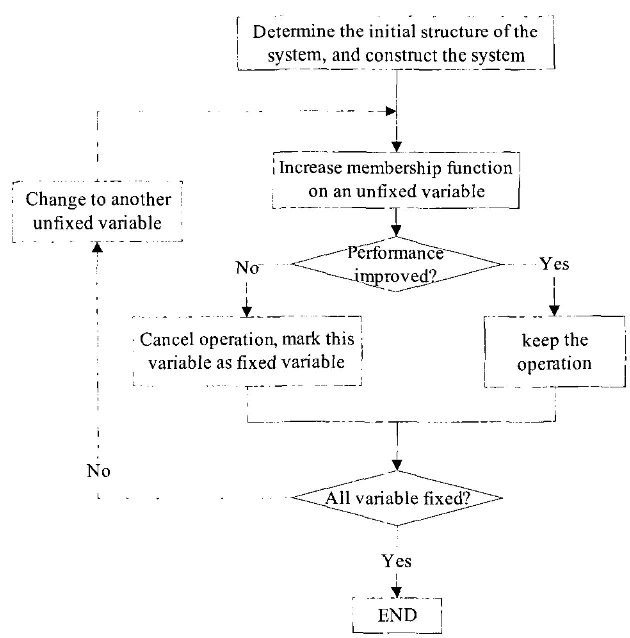


Figure 2. The flow chart of the determination of fuzzy system structure

Note that, at the end of **Step 4**, optimization algorithms can be introduced to adjust the membership function parameters, such as position and support, to further improve the modeling accuracy. This optimization is not applied in this paper.

4. Complete Missing Rules

Suppose the recorded N data pairs are $(x_1^p, x_2^p, \ldots, x_n^p; y^p)$, p=1,2,...,N, where x is the input of the model in the space $[x_{1,\min}, x_{1,\max}]\times[x_{2,\min}, x_{2,\max}]\times\ldots\times[x_{n,\min}, x_{n,\max}]$ and y is the output in the space $[y_{\min}, y_{\max}]$. Our objective is, in every iteration of the model structure identification process mentioned in Section 3, to get complete rulebase from this set of data and construct a fuzzy model $f(x)$ to describe the behavior of the system.

As described in the Step 1 in Section 3, at the beginning, for each input variable $[x_{i,min}, x_{i,max}]$, i=1,.2, ..., n, a small number of fuzzy sets is defined to cover it. In the below description, triangular membership functions in a two-input-one-output problem with two evenly distributed fuzzy sets A^1, A^2 on input variables and four evenly distributed fuzzy sets B^1, B^2, B^3, B^4 on output variable is used as an example. This initial condition is shown with solid line in Figure 3. Obviously, four fuzzy rules should be extracted from recorded data in this step. Because of the small number of rules at this step, even if there is a missing rule, it can be completed by human knowledge very easily.

Then, rules are extracted from training data pairs and a fuzzy model is constructed to describe the system behavior. By using singleton fuzzifier, center average defuzzifier, and product inference engine, the fuzzy model can be written as follows:

$$f_j(x)=\frac{\sum_{l=1}^{M} y_l(\prod_{i=1}^{n}\mu_{A_i^l}(x_i))}{\sum_{l=1}^{M}(\prod_{i=1}^{n}\mu_{A_i^l}(x_i))}$$

Where x is the input to the fuzzy system, $f_j(x)$ is the output of the fuzzy system, $\mu_{A_i^l}(x_i)$ is the membership value of x_i on fuzzy set A_i^l and y is the center of fuzzy sets on output variable.

By using the constructed model, the modeling accuracies can be evaluated on both training and testing data sets. If the modeling performance is acceptable, or a certain predefined criterion is satisfied, then this constructed fuzzy model can be used to describe the behavior of the real system. Otherwise, the structure of the model should be changed to approximate the real system. When the number of fuzzy sets on each input variable is increased as the dashed line in Figure 3, more fuzzy rules should be generated. However, due to the limited number of data in training set, it is highly possible that there are missing rules as the number of rules increases. To complete the rulebase, the constructed model at the previous step is utilized to generate the consequent part of the missing rule.

After the increase of the number of fuzzy sets on an input variable, if there is a missing rule, the centers of the fuzzy sets on the input variables of this missing rules $[x_{c,1}, x_{c,2}, \ldots, x_{c,n}]$ are input to the model constructed at the previous iteration and the model output is assigned to a fuzzy set on the output variable and employed as the consequent part of the missing rule. For example, if the missing rule is:

IF x_1 is A_1^1 and x_2 is A_2^3 Then y is ?

Suppose the center of A_1^1 is $x_1^1$and the center of A_2^3 is x_2^3, then, with the input being $x=[x_1^1\ x_2^3]$ to the previous model $f_j(x)$, we have an output of y_a. Additionally, suppose that y_a has the highest membership value on fuzzy set B^m, then the missing rule can be completed as:

IF x_1 is A_1^1 and x_2 is A_2^3 Then y is B^m.

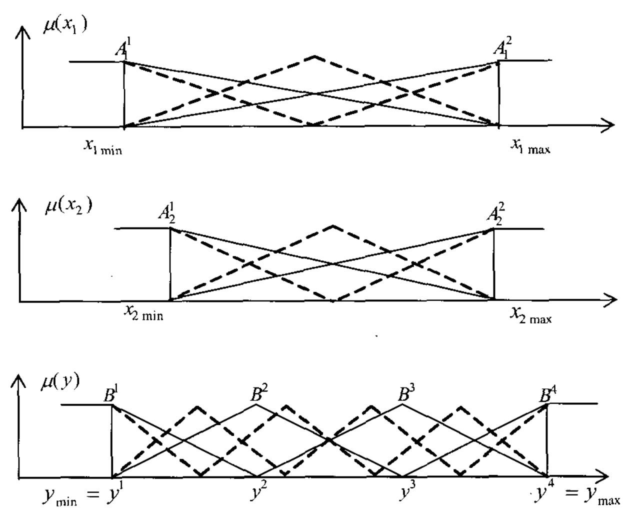


Figure 3 Example of generating rules from data in a two-input case

Through this approach, the rulebase generated at each iteration will be complete. This rulebase is used to build the model in the next iteration $f_{j+1}(x)$ and it is expected that the modeling accuracy is improved.

If the modeling of $f_{j+1}(x)$ is not satisfactory, increase the number of fuzzy sets on another input variable and repeat the process as described in Section 3. From theorem of universal approximation [3], it is clear this model can approximate any continuous function on a compact set to any accuracy. Therefore, a satisfactory fuzzy model can be obtained to approximate the real system by increasing the number of fuzzy sets.

Since the increase of fuzzy set on input variables is a repeated process and results in sparsely distribution of the training data in the whole space, a criterion should be defined to terminate this process. The definition of criterion is problem dependent. In our paper, the criterion is that the modeling accuracies on training data and testing data are similar and reach a predefined level. Meanwhile, optimization method can be employed to tune model parameters.

The advantage of the proposed method is that the generated rulebase is always complete. So there is a unique rule corresponding to a case of all the possible combinations of the fuzzy sets defined in the input space even if the number of recorded data is small.

5. Real Time Update of Rulebase

Although the scheme in Section 4 can generate complete rulebase and improve the modeling accuracy, the missing rules are generated through a rough approximation. In addition, the rulebase is built offline once for all through the training data and usually the rulebase does not change. However, during the operating of the system, the dynamics of system may change and unmodelled uncertainty may degrade the performance of the model. To make the model accommodate the change of system dynamics and uncertainties, an online real time rulebase update scheme is developed here.

As illustrated in a two-dimension example in Figure 4, suppose the square is the entire working domain of the discussed system. The training data, in most cases, cannot cover the entire working domain and suppose it only covers the shaded zone. Hence, from the recorded data, it is almost always impossible to obtain the behavior of system in the entire working domain. Consequently, when fuzzy system is used to model system dynamics based on this training data, the model is not only inaccurate, but also unreliable, especially when the working condition and environment has changed. Moreover, even with a complete rulebase, the number of those rules that describes system behavior in the working domain is limited. Thus, in practical application, the rulebase should be, if possible, updated during the operating of the system.

In the modeling process, to check the validity of the constructed model, the recorded data are usually divided into training set and testing set. As mentioned above, if the model errors on the two sets reach a certain level and get a balance, the model can be utilized to describe the system behavior. Let's investigate Figure 4. Suppose the initial working domain of system is $\boldsymbol{A}$ and the model is put into use at this time instance. After a period of time, the working domain of system may change to $\boldsymbol{B}$. Then the system may fire some rules that cannot be generated by the recorded data. Hence, a method should be introduced to guarantee that the constructed model could describe system behavior at working domain $\boldsymbol{B}$.

If the rules in the rulebase cannot reflect the mapping relationship from the input space to the output space in the current working domain, no matter how hard we tune the parameter set, satisfactory performance may not be guaranteed. Suppose working domain changes from working domain $\boldsymbol{C}$ to working domain $\boldsymbol{A}$, although $\boldsymbol{A}$ is covered by formerly generated rules, these rules may not describe system behavior exactly under current operating condition due to the uncertainty and change of system dynamics. In other words, a new set of rules should be extracted, if necessary, to reflect the system behavior under the current operating condition. Then an approach should be introduced to update rules from time to time so that these rules can reflect the current system dynamics.

To reach this goal, a certain length of recorded data, say L data, is determined, such that the current dynamics of system is well-described by these L most-recent recorded data. In the operation of the system, always keep the last L pairs so that these L data are used to update the rulebase as illustrated in Figure 5. By using the methods mentioned in Section 4, a new set of complete rulebase is obtained. If there are conflicting rules, the rule with the greatest degree of reliability is selected.

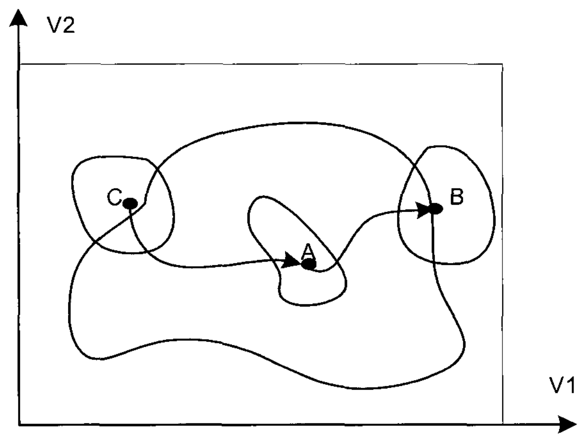


Figure 4. The change of working domain in a two-dimension sample

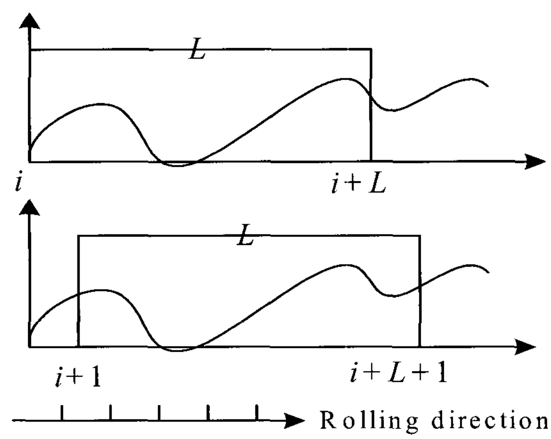


Figure 5. The process of generating fuzzy rules from most-recent recorded data

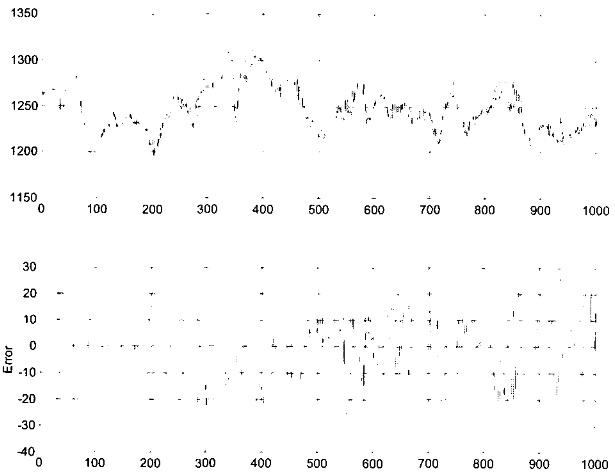


Figure 6. The model output (blue) and measurement (red) of zone 3 and the modeling error. The absolute average error $c1$ is 8.5379, 8.2740, and 8.7138 on entire data set, training set, and testing set, respectively. The square error $c2$ is 116.7924, 111.5349, and 120.2974, respectively.

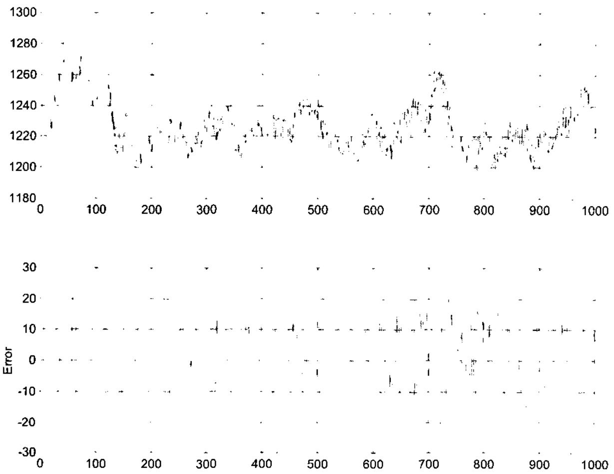


Figure 7. The model output (blue) and measurement (red) of zone 4 and the modeling error. The absolute average error $c1$ is 8.3210, 8.0209, and 8.1410 on entire data set, training set, and testing set, respectively. The square error $c2$ is 104.0946, 97.2042, and 108.6107, respectively.

From above description, it is clear that the update of rulebase is a rolling process. Suppose the current time instance is $i+L$, the newly recorded data are from time instance i to $i+L$, and the rulebase is updated by rules generated from these L data pairs. While at the next time instance $i+L+1$, the rules are generated from data recorded from time instance $i+1$ to $i+L+1$. In the same way, a strategy of parameter optimization can be introduced to optimize model parameter at every period if necessary and there is enough computation time. Since the structure of the fuzzy model has been determined, this update operation of rulebase will not cost too much computation and, therefore, can be realized on-line suppose the sampling rate is not very high. If, in some cases, the sampling rate is high and there is no enough computation time for update, this update operation can be performed every several sampling period rather than every sampling period. The idea is the same, using some recent recorded data to capture the current system dynamics.

6. Application to Reheating Furnace Modeling

In order to evaluate the performance of the model, some criteria need to be defined. The criteria should be able to support efficient algorithms to solve the optimization problem when the optimization is needed. In this section, two criteria are introduced to assess the performance of the model. The first one is absolute average error, which is formulated as follows:

$$c_1 = \frac{1}{N}\sum_{i=1}^{N}\left|y_m(i) - y(i)\right|$$

The second one is square error

$$c_2 = \frac{1}{N}\sum_{i=1}^{N}\left(y_m(i) - y(i)\right)^2$$

where y_m, y are the model output and the measurement from the reheating furnace, respectively.

In the following simulations, 1000 recorded data pairs are utilized to verify the proposed method. The first 700 pairs of data are used as training set and the other 300 pairs as testing set.

The number of fuzzy sets on each input variable is started from two and that on output variable is started from four. That is, two membership functions are defined to cover each input variable and four membership functions are defined to cover output space. The membership functions are triangular which are evenly spread. Then, the number of fuzzy sets on each input variable is increased as described in Section 3.

In the first example, the model of zone 6 is verified with L=120 and the result is shown in Figure 6. The sampling period of the recorded data is 1 minute. Hence, the L data sets contain the dynamics of the furnace in the past 2 hours. The model output, measurement are plotted in the upper subfigure and the error between them is plotted in lower subfigure of Figure 6. The value of the mentioned two criteria c_1 and c_2 on the entire data set, training data set, and testing data set are listed in the caption, respectively.

In the second example, shown in Figure 7, is the model of zone 4 with L=100. The result and performance evaluation, as in Figure 6, are given in the caption of Figure 7. It is clear from the comparisons that the proposed method can improve the modeling accuracy efficiently.

7. Conclusion

In this paper, a modeling method is proposed to identify the structure of fuzzy model, complete the rulebase, and update the rulebase in real time. The method starts from a small number of membership functions on each variable to build a rough model. Then, the number of membership functions on each variable is increased to improve the accuracy of model. In this process, an approach is applied to complete the rulebase as the increase of fuzzy rules. The update of the rulebase is carried out in every computation period in real application in a rolling way to accommodate the change of system dynamics and uncertainties. The proposed method is verified on the modeling of a walking beam reheating furnace.

References

[1] Sugeno M. and G. T. Kang, "Structure Identification of Fuzzy Model," *Fuzzy Sets and Sys*tems, vol.28, pp. 15-33, 1988.

[2] Takagi T and Sugeno M., "Fuzzy Identification of Systems and Its Applications to Modeling and Control," *IEEE Trans. on Systems, Man, and Cybernetics*, vol.15, no. 1, pp. 116-132, 1985.

[3] L.X. Wang and J.M. Mendel, "Generating Fuzzy Rules by Learning from Examples," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 22, no. 6, pp. 1414-1427, 1992.

[4] J. Nie, "Constructing Fuzzy Model by Self-Organizing Counter-Propagation Network," *IEEE Trans. on System, Man, and Cybernetics*, vol. 25, no. 6, pp. 963-970, 1995.

[5] S. Abe and M.S. Lan, "A Classifier Using Fuzzy Rules Extracted Directly from Numerical Data," *Proceedings of Second IEEE International Conference on Fuzzy Systems*, San Francisco, CA, March 1993, pp. 1191-1198, 1993.

[6] S. Abe and M.S. Lan, "Fuzzy Rules Extraction Directly from Numerical Data for Function Approximation," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 25, no. 1, pp. 119 – 129, 1995.

[7] L. De Mecatrnica, "Autonomous Mountain-Clustering Method Applied to Fuzzy Systems Modeling," *Smart Engineering Systems: Fuzzy Logic and Evolutionary Programming, ASME Press*, New York, USA, pp. 311-316, 1995.

[8] Huarng, K.-H.; Yu, T. H.-K.; Hsu, Y. W., "A Multivariate Heuristic Model for Fuzzy Time-Series Forecasting," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 37, no. 4, pp. 836-846, 2007.

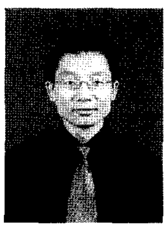
[9] J. Jang, "Structure Determination in Fuzzy Modeling: A Fuzzy CART Approach," *Proc. of IEEE International Conference on Fuzzy System*, Orlando, FL, June 1994, pp. 480 – 485, 1994.

[10] Kuang C.C. and Su J. Y., "Affine Takagi-Sugeno Fuzzy Modeling Algorithm by Fuzzy C-Regression Models Clustering with a Novel Cluster Validity Criterion," *Control Theory and Application, IET*, vol. 1, no. 5, pp. 1255-1265, 2007

[11] Moreno-Garcia, J.; Castro-Schez, J.; Jimenez, L., "A Fuzzy Inductive Algorithm for Modeling Dynamic Systems in a Comprehensible Way," *IEEE Transactions on Fuzzy Systems*, vol. 15, no.4, pp. 652-672, 2007.

[12] Y. Wu, J. Lu, J. Xu, Y. Sun, "Bioprocess Modeling Using Fuzzy Regression Clustering and Genetic Programming," *The 6th World Congress on Intelligent Control and Automation*, vol.2, Dalian China, pp. 9337-9341, June 2006.

[13] L.X. Wang, *Adaptive Fuzzy Systems and Control: Design and Stability Analysis,* Prentice Hall, Englewood Cliffs, NJ. USA, 1994.



Bin Zhang

He received his BE and MSE degrees from Nanjing University of Science and Technology, China, in 1993 and 1999, respectively. He received his Ph.D. degree from Nanyang Technological University, Singapore, in 2007. He is currently a research engineer at the School of Electrical and Computer Engineering at Georgia Institute of Technology, Atlanta, GA. His research interests are intelligent control, digital signal processing, fault diagnosis, learning control and their applications to robotics, power electronics and various mechanical systems.

Jingcheng Wang

He received his Bachelor and master degrees in Northwestern Polytechnic University of China and Ph.D. in Zhejiang University of China. He was a recipient of Alexander von Humboldt fellowship in Rostock University of Germany. Currently, he is a Professor in Shanghai Jiaotong University, Shanghai. His research interests are robust control, intelligent control, real-time control and simulation etc.