The Third AFSS(1998), 453-458

A Multiple Model Approach to Fuzzy Modeling and Control of Nonlinear Systems

Chul Heui LEE , Seon Hak SEO, Young Ki HA

Dept. of Electrical Engineering, Kangwon National University
Chuncheon, Kangwon-do, 200-701, Korea
Tel:+82-361-250-6296 Fax:+82-361-241-3775 E-mail:chlee@cc.kangwon.ac.kr

Abstract

In this paper, a new approach to modeling of nonlinear systems using fuzzy theory is presented. So as to handle a variety of nonlinearity and reflect the degree of confidence in the informations about system, we combine multiple model method with hierarchical prioritized structure. The mountain clustering technique is used in partition of system, and TSK rule structure is adopted to form the fuzzy rules. Back propagation algorithm is used for learning parameters in the rules. Computer simulations are performed to verify the effectiveness of the proposed method. It is useful for the treatment of the nonlinear system of which the quantitative math-approach is difficult.

Keywords: Multiple Model, Hierarchical Prioritized Structure(HPS), Mountain Clustering, TSK Rule

1. Introduction

Even though most physical systems intrinsically contain the nonlinearity, they are usually approximated as linear model in the vicinity of nominal point because the mathematical treatment of nonlinearity is somewhat difficult and there's no general tools to cover all sorts of the nonliearities. However, the linearization method may not succeed in coping with the uncertainty of the system and the change of the operational environment[1].

To attack these problems, some complicated control mechanisms such as adaptive control and variable structure system control have been proposed. However, there still exists a problem to be solved: the construction of a available mathematical model of system for them. Therefore, as indicated in Zadeh's principle of incompatibility, it is desirable to apply the fuzzy theory to system modeling and control in case that the mathematical formulation of the system is troublesome to handle[2].

Fuzzy modeling and control use the qualitative characteristics of the system, and it provides a nonlinear relationship induced by membership functions, rules, and inference. From a viewpoint of the flexibility, fuzzy modeling and control implicitly possesses the properties similar to variable structure system or multiple model method, since many "IF - THEN - " rules are arranged in parallel in fuzzy rule base. It can also have the adaptive capability by learning procedure of the membership functions and rules[2,3,4,5].

If a single rule base is constructed for the system of which the nonlinearity is severe and complicated, the size of rule base becomes very large, and the consistency between the rules is hardly to be retained, and the computing time for inference increases. Therefore, it's better to express the system characteristics with several different workframes (i.e. multi-models) partitioned appropriately according to its operational characteristics. Moreover, a special rule structure which reflects the degree of specification of the rules is preferred because they don't have the same degree of belief even if they belong to the same rule base.

Thus, in this paper, a new approach to modeling and control of nonliear systems using fuzzy theory is presented, where we combine the multiple model method[6,7] with hierarchical prioritized structure(HPS)[8] to express effectively the complex behavior of various nonlinear systems.

The first thing to be solved in the proposed method is partitioning of the system into several models. We solve this problem by clustering the observed data in input-output space, and the mountain clustering(MC) technique[9] is used for fuzzy clustering.

Next problem to be handled is the construction of rule base for each partitioned model. The hierarchical prioritized structure make it possible to express the difference in the degree of belief on the rules, since the more specific the rules are the higher levels of priority they have.

For individual rules, TSK(Takagi-Sugeno-Kang) type fuzzy rules are adopted, where the consequents employ the functions of input variables, usually linear functions, instead of fuzzy subsets. So rule generation requires the determination of the parameters of the functions in the consequents. We use mountain clustering technique again in fuzzy space partitioning, and the back propagation algorithm is used for the determination of the parameters. Of course, another rule generation methods may be used, if necessary.

2. Partitioning of system into multiple models

2.1 Multiple Model Method.

Models encountered in fuzzy modeling and control are concerned with approximating the relationship between input and output variables. However, a single fixed model is liable to fail in many cases since this relationship varies nonlinearly depending on system characteristics and operational environment. Therefore it is natural to take a family of models than single model for the representation of system.

The multiple model is a collection of models equipped with some mechanisms aimed at a relevant triggering between the models or aggregating the results furnished by the individual models[6,7]. From the operational standpoint, it is essential to elaborate on how the partition of the input-output space can be carried out efficiently. The main idea is to reveal the structural relationship between the input and output variables via a method of specialized fuzzy clustering: the mountain clustering method.

First, we partition the system into several sub-models on input-output space in accordance with their characteristic similarity by using mountain clustering. Next, for each subsystem, we construct the TSK type fuzzy rules as follows:

IF x_1 is A_{i1} .... and x_n is A_{in} THEN $y=f_i(x_1, ..., x_n)$

In general, the function f_i in the consequent part is linear. If that is the case, the above rule implies a linear approximation of system characteristics for a specified fuzzy input subspace. Therefore, each sub-model becomes nothing but a collection of linear approximations expressed by a number of fuzzy rules.

Fig 1. shows the example of the partition of system. In this case, the system is divided into 5 sub-models with cluster centers denoted by *, and lines in sub-model C represent the linear approximations by TSK fuzzy rules.

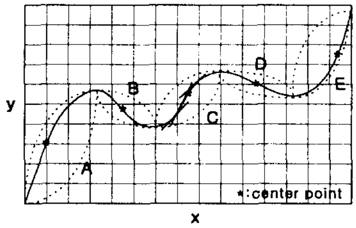

Fig. 1. Partitioning of Model

If the partition is disjointed, there's no need to introduce a special mechanism for triggering between the sub-models or aggregating their results, because only one sub-model is matched to a specified input data, and the detailed behavior around the boundary can be taken into account in its rule base.

2.2 Mountain Clustering

As stated above, partitioning of system is carried out through fuzzy clustering procedure. Although different clustering methods such as ISODATA, Fuzzy C-Means, and Fuzzy Bounded Classification methods can be applied for this purpose[2,3]. We use here a recently developed clustering approach called the mountain clustering. This clustering technique is a very simple grid-based three-step process for identifying the approximate location of cluster centers in data sets with clustering tendencies.

In the first step we discretize the object space by griding with lines and generate the potential cluster centers as the intersection of these grid lines, called nodes. The second step uses the observed data to construct the mountain function, which is constructed by adding an amount to each node exponentially inversely proportional to that node's distance from the data point. The third step generates the cluster centers by an iterative destruction of the mountain function, where the effect of the just-identified cluster center is eliminated from the total score of each node.

We note that it is not necessary for the griding lines to be equidistant in this method; they can be finer or coarser in different regions of the space according to data characteristics. A finer griding increases the number of potential cluster centers but also increases the calculation required.

Assume the data consists of a set of K points, $O_k(x_k,y_k)$, in the input-output space $X \times Y$, and the number of grid nodes $N_{ij}(X_i,Y_j)$ is $N(=r_1 \times r_2)$. The mountain function M is constructed from the observed data by adding an amount to each node in N proportional to that node's distance from the data point; it takes nonnegative values only. More formally for each point $N_{ij},(X_i,Y_j)$

$$M(N_{ij}) = \sum_{k=1}^{n} e^{-(\alpha d(N_{ij},O_k))} \quad (1)$$

where α is a positive constant, and $d(N_{ij}, O_k)$ is a measure of distance between N_{ij} and O_k typically, but not necessarily, measured as

$$d(N_{ij},O_k) = (X_i - x_k)^2 + (Y_j - y_k)^2 \quad (2)$$

The values of mountain functions are approximations of the density of the data points in the vicinity of each node. The higher the mountain-function value at a node the larger its potential for a cluster center.

Therefore, we designate the node with maximal total score M_1^* as the first cluster center C_2^*. In order to get the next cluster center, we must remove the effect of the peak associated with the just-identified cluster center and revise the mountain function because this peak is usually surrounded by a number of grid points that also have high score. This destruction of the mountain function is realized by subtracting from the total score of each node a α product of M_1^* and an exponential term inversely proportional to the distance of the node from the just-identified cluster center.

That is, we form a revised mountain function M_2 such that:

$$M_2(N_{ij}) = M_1(N_{ij}) - M_1^* e^{-\beta d(C_1^*,N_{ij})} \quad (3)$$

where β is a positive constant.

It guarantee that those nodes close to the identified cluster center have mountain values that are reduced more strongly than those further away. Now we repeat the above procedure iteratively until the whole cluster centers are obtained. At k-th iteration, we proceed as follows.

i. Find $M_k^* = \text{Max}\{M_k(N_{ij})\}$

ii. Designate k-th cluster center C_k^* at the location of maximal node found in i

iii. Formulate revised mountain function, M_{k+1}, as

$$M_{k+1}(N_{ij}) = M_k(N_{ij}) - M_k^* e^{-\beta d(C_k^*,N_{ij})}$$

The process of destroying the mountain function ends at step m with the estimation of m cluster centers after next M^*_{m+1} becomes less than a given stopping constant δ:

$$M^*_{m+1} < \delta$$

This means that there are only a very few points around the $(m+1)^{th}$ cluster center and that the cluster center can be omitted.

The cluster centers obtained by applying mountain clustering can be regarded as nominal points of corresponding sub-models. The determination of the boundaries between disjointed rule-models is very simple and easy in mountain clustering. It is proposed that the grid point which yields almost some values of mountain functions with respect to 2 adjacent cluster centers is chosen as the boundary between these sub-models.

3. Construction of HPS Rule Base

3.1 Hierarchical Prioritized Structure(HPS)

After partitioning the system, the rule base for each partitioned model must be constructed. Usually rules are induced from the observed data via some identification procedures. In practice, the behavior of nonlinear system can be exactly described by observed data at some specific operating points, while it is approximated more roughly at the points further away from them. Moreover, the rules made from more ambiguous or less specific information give lower degree of confidence since, the consequent of TSK fuzzy rules present only averaged property with respect to the fuzzy subsets in the antecedent. In consequence, the flat representation of fuzzy rules in a rule base leads to unsatisfactory results, and it is desirable to use a structure that reflects the difference in the degree of belief on the rules.

Yager's HPS[8] is such a rule structure, where the more specific the rules are the higher levels of priority they have, and thereby it prevent the problem that the more specific information is swamped by the less specific information. Fig. 2 shows in a systematic view the form of HPS representation.

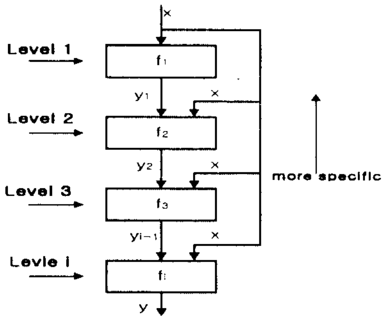

Fig. 2. Hierarchical Prioritized Structure(HPS)

In the HPS each subbox, denoted f_i, is a collection of rules relating the system input, x, and the current iteration of the output, y_{i-1}, to a new iteration of the output. At the highest level we have specific point information. The next level encompasses these points and in addition provides a more general and perhaps fuzzy knowledge. We note that the lowest level can be used to tell us what to do if we have no knowledge up to this point. In some sense the lowest level is a default value. Therefore, the higher priority levels of rules would have less general information, and consist of rules with more specific antecedents than those of lower priority, which generally reflects into more specific consequent information.

In our HPS rule base for a partitioned model, rules corresponding to a cluster center and/or some specific points are placed at the highest level. At the next level the specificity of the antecedent linguistic variables would decrease. Thus the support sets of the antecedents would be wide and will as the range of the consequent. As we go to lower levels, the cardinality of the support set decreases.

If an input is not consistent with the rules in the higher priority level, then the rules in the lower priority levels compensate this inconsistency to produce a suitable result. On the other hand, an input doesn't bother to fire any of the less specific rules in the lower priority levels if it perfectly matches one or more of the rules in the higher priority level.

The final result of fuzzy inference with HPS rule base is not obtained until the sequential determination of the output of each level is finished. The aggregation of rules at each level is of the standard Mamdani type[2], and the contribution of the lower level is limited by the compatibility of the input to the rules in the higher level.

Let τ_{ij} be the firing level of j-th rule in i-th priority level given by

$$\tau_{ij} = \wedge \mu_{A_{ij}^k}(x_k) \quad (4)$$

The aggregation of the rules T_i in the i-th level is obtained by

$$T_i = \bigcup_{j=1}^{n_i} \tau_{ij} \wedge f_{ij} \quad (5)$$

Assume that the output of (i-1)-th level y_{i-1} is G_{i-1}. Also we define a measure of how much matching we have up to (i-1)-th level, denoted by g_{i-1}, such that

$$g_{i-1} = \max \mu_{G_{i-1}}(y) = \text{Possibility}(G_{i-1}) \quad (6)$$

Then the output of i-th level is determined from G_{i-1}, g_{i-1}, and T_i as follows.

$$G_i(y) = (T_i(y) \wedge \text{low}(g_i)) \vee G_{i-1}(y) \quad (7)$$

where low(·) bounds the allowable contribution of the i-th level to the overall output. Here the following function is used for low(·).

$$\text{low}(g_i) = 1 - g_i \quad (8)$$

As in eq.(7), if $g_i=1$, lower levels cannot contribute any more. As long as we have not found one y with membership grade 1, $\text{Poss}[G_{i-1}] \neq 1$, we add some of the output of the current subbox to what we already have. Each element y gets $1\text{-Poss}[G_{i-1}]$ portion of the contribution at that level, T_i. The amount $\text{Poss}[G_{i-1}]$ is determination by the highest membership grade of any element in G_{i-1}.

3.2 Parameter learning of rules

To obtain TSK fuzzy rules, we need to solve the problems such as fuzzy space partition, assignment of memberships for the antecedents, and determination of the parameters of function f_i in the consequent. For fuzzy space partition, fuzzy clustering methods or neural networks can be used[2,3]. Learning of memberships and parameters can also be performed by neural networks[3].

Here we apply again the mountain clustering to fuzzy space partition, and the parameters of memberships and f_i are learned by back propagation, which is a powerful technique for learning neural networks.

Consider TSK fuzzy rules expressed by

IF x_1 is A_{i1} AND ···· AND x_r is A_{ir}

THEN $y_i = b_{i0} + b_{i1}x_1 + \cdots + b_{ir}x_r$, $i=1, \cdots, m$

Assume that the antecedent fuzzy sets are defined by Gaussian membership functions with parameters x^*_{ij} and σ_{ij} :

$$\mu_{A_{ij}}(x_j) = \exp\left(-\frac{1}{2}\left(\frac{x_j - x^*_{ij}}{\sigma_{ij}}\right)^2\right) \quad (9)$$

Then the firing level of the rule is given by

$$\begin{aligned}\tau_i &= \mu_{A_{i1}}(x_1) \wedge \mu_{A_{i2}}(x_2) \wedge \ldots \wedge \mu_{A_{ir}}(x_r) \\ &= \exp\left(-\frac{1}{2}\sum_{j=1}^{r}\left(\frac{x_j - x^*_{ij}}{\sigma_{ij}}\right)^2\right)\end{aligned} \quad (10)$$

Therefore, the crisp output inferred by this fuzzy rule base is as follows :

$$\begin{aligned} y &= \frac{\sum_{i=1}^{m} \tau_i (b_{i0} + b_{i1}x_1 + \ldots + b_{ir}x_r)}{\sum_{i=1}^{m} \tau_i} \\ &= \frac{\sum_{i=1}^{m}\left[\exp -\left(\frac{1}{2}\sum_{j=1}^{r}\left(\frac{x_j - x^*_{ij}}{\sigma_{ij}}\right)^2\right)\right](b_{i0} + b_{i1}x_1 + \ldots + b_{ir}x_r)}{\sum_{i=1}^{m}\left[\exp -\left(\frac{1}{2}\sum_{j=1}^{r}\left(\frac{x_j - x^*_{ij}}{\sigma_{ij}}\right)^2\right)\right]} \\ &= \sum_{i=1}^{m} v_i (b_{i0} + b_{i1}x_1 + \ldots + b_{ir}x_r) \end{aligned} \quad (11)$$

$$v_i = \frac{\tau_i}{\sum_{i=1}^{m}\tau_i} = \frac{\left[\exp -\left(\frac{1}{2}\sum_{j=1}^{r}\left(\frac{x_j - x^*_{ij}}{\sigma_{ij}}\right)^2\right)\right]}{\sum_{i=1}^{m}\left[\exp -\left(\frac{1}{2}\sum_{j=1}^{r}\left(\frac{x_j - x^*_{ij}}{\sigma_{ij}}\right)^2\right)\right]} \quad (12)$$

Thus, for a fixed number of rules m, the problem of setting up the rules is essentially a parameter estimation problem. For a given collection of crisp input-output data we can formulate this problem as a minimization of the square of instantaneous errors between the output of the fuzzy model eq.(11) y and the current output reading y_k with respect to unknown parameters x^*_{ij}, σ_{ij}, and $b_{i0}, \cdots, b_{ir}$. The criterion for minimization is given by

$$E_k = \frac{1}{2}(y-y_k)^2 = \frac{1}{2}e^2 = \frac{1}{2}\left(\sum_{i=1}^{m} v_i(b_{io}+b_{i1}u_1+\ldots+b_{ir}u_r)-y_k\right)^2 \quad (13)$$

We obtain the following formulas for back-propagation learning of TSK rules by application of the chain rule.

$$b_{io}(k+1) = b_{io}(k) - \alpha\frac{\partial E_k}{\partial b_{io}} = b_{io}(k) - \alpha v_i e \quad (14)$$

$$b_{ij}(k+1) = b_{ij}(k) - \alpha\frac{\partial E_k}{\partial b_{ij}} = b_{ij}(k) - \alpha v_i x_j e \quad (15)$$

$$x^*_{ij}(k+1) = x^*_{ij}(k) - \alpha\frac{\partial E_k}{\partial x^*_{ij}} = x^*_{ij}(k) - \alpha v_i(b_{io}+b_{i1}x_1+\ldots+b_{ir}x_r-y)e\frac{x_j-x^*_{ij}(k)}{\sigma^2_{ij}(k)} \quad (16)$$

$$\sigma_{ij}(k+1) = \sigma_{ij}(k) - \alpha\frac{\partial E_k}{\partial \sigma_{ij}} = \sigma_{ij}(k) - \alpha v_i(b_{io}+\ldots+b_{ir}x_r-y)e\frac{(x_j-x^*_{ij}(k))^2}{\sigma^3_{ij}(k)} \quad (17)$$

where α is the learning rate.
The initial estimation of parameters are chosen by
$x^*_{ij}(0)$= i-th cluster center

$$\sigma_{ij}(0) = \sqrt{\frac{1}{2\beta}} \quad (18)$$

$b_{i0}(0) = b_{i1}(0) = \cdots = b_{ir}(0) = 0$

Fig. 3 presents the block diagram of the learning algorithm combine with the three-layer network.

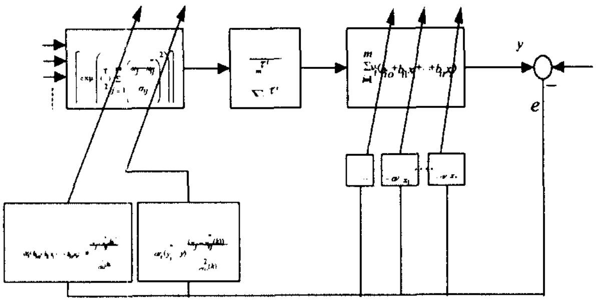

Fig.3 Block Diagram of Back propagation

4. Numerical Experiments

Computer simulations in two cases are performed to show the effectiveness of the proposed method. One is a curve fitting problem of highly nonlinear data corrupted by noise, and the other is a control problem of the inverted pendulum.

Example 1. (Modeling)

This example is concerned with the construction of proper model from the noisy data. The characteristics of system is governed by

$$y = \frac{x^3+x^2}{x^6+x^4+3} \quad (19)$$

This system is highly nonlinear. The noise added to the original system is a white noise with σ=0.05. The original system and noisy data are presented in Fig.4, and the model obtained by proposed method is also shown in Fig.4. Fig. 5 shows the value of mountain functions with respect to 5 luster centers. As shown in Fig. 4, the proposed method find a good model for system from the noisy data.

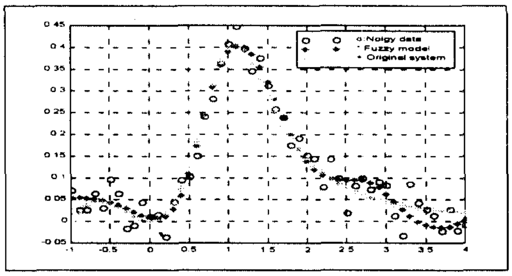

Fig. 4. Cluster Center Points

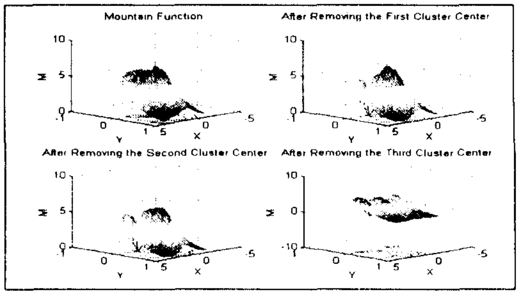

Fig. 5. Mountain Function

Example 2. (Control)

This example is a well known control problem, the control of inverted pendulum[10].

The structure of inverted pendulum system is shown in Fig.6, and its parameters are as follows:

g : 9.8 [m/s²] (acceleration of gravity)
m_c : 0.9 [kg] (mass of cart)
m_p : 0.1 [kg] (mass of pole)
l : 0.5 [m] (half length of pole)
F : [N] (force)

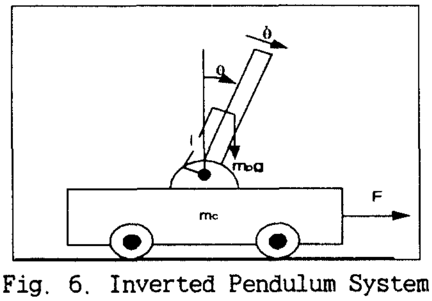

Fig. 6. Inverted Pendulum System

The state equation for this system are given by

$$\dot{x}_1 = x_2 \tag{20}$$

$$\dot{x}_2 = \frac{g\sin x_1 - \cos x_1\left[\frac{F + m_p l(\sin x_1)x_2^2}{m_c + m_p}\right]}{\frac{4}{3}l - \frac{m_p l(\cos x_1)^2}{m_c + m_p}} \tag{21}$$

$$\dot{x}_3 = \frac{F + m_p l\ [(\sin x_1)x_2^2 - (\cos x_1)\dot{x}_2]}{m_c + m_p} \tag{22}$$

where $x_1 = \theta$, $x_2 = \dot{\theta}$, and x_3 is the velocity of the cart.

Because of the nonlinear and unstable nature of inverted pendulum system, we limit initial angle θ and initial angular velocity to be within [0,0.873] rad and [0,0.5] rad/sec, respectively[10].

An exact control surface obtained by numerical integration of system equations using the shooting method is presented in Fig.6. The proposed method yields a control surface almost equal to exact one, as shown in Fig.7.

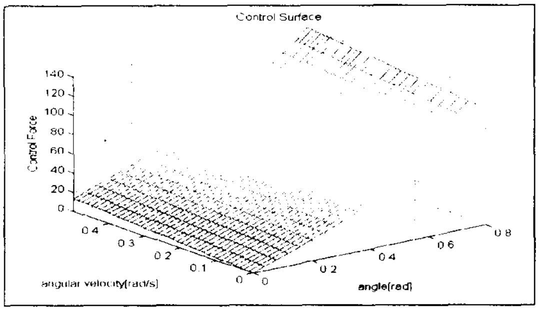

Fig. 7. Exact control Surface

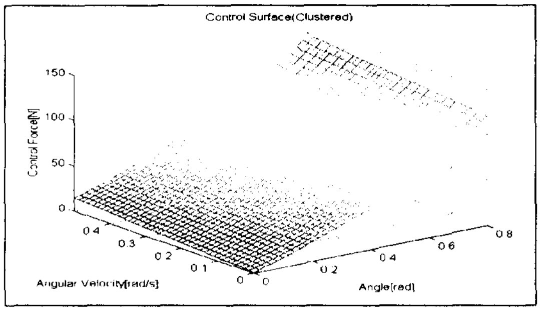

Fig. 8. Control Surface(Clustered)

5. Conclusions

Modeling and control of nonlinear system is difficult to solve due to the complexity of characteristics. Therefore, we suggest a new approach in which the partition of system into multiple model by fuzzy clustering is combined with hierarchical prioritized structure rule base. As the proposed method is capable of handling a variety of nonlinearity and reflecting the degree of confidence in the informations about system, it can be applied effectively to the nonlinear system of which the mathematically quantitative treatment is difficult and troublesome. The simulation results reveal that the proposed method is very useful for modeling and control of nonlinear systems.

ACKNOWLEDGEMENT

This work is supported in part by Multimedia Research Center of Kangwon National University in cooperation with IITA.

References

1 J.E. Slotine and W. Li, *Applied Nonlinear Control*, Prentice Hall, Englewood Cliffs, NJ, 1991

2 R.R. Yager and D.P. Filev, *Essentials of Fuzzy Modeling and Control*, John Wiley and Sons, New York, 1994

3 C.T. Lin and C.S.G. Lee, Neural Fuzzy Systems, Prentice Hall PTR, Upper Saddle River, NJ, 1996

4 M. Sugeno and T. Yasukawa, "A Fuzzy Logic Based Approach to Qualitative Modeling", IEEE Trans. Fuzzy Systems, Vol.1 pp.7-31, 1993

5 W. Pedrycz, "Fuzzy Modeling: Fundamentals, Construction and Evaluation", Fuzzy Sets and Systems, Vol.41, pp.1-15, 1991

6 D.G.Lainiotis, "Partitioning: A Unifying Framework for Adaptive Systems, I:Estimation", IEEE Proc. Vol.64, pp.1126-1143, 1976

7 W. Pedrycz, "Fuzzy Multimodels", IEEE Trans. Fuzzy Systems, Vol.4, pp.139-148, 1996

8 Ronald R. Yager, "Hierarchical Representation of Fuzzy IF-TEN Rules", Machine Intelligence Inst. Iona College Tech. Report #MII-1305,1992

9 Ronald R. Yager, Dimetar P. Filev, "Approximate Clustering Via the Mountain Method", Machine Intelligence Inst. Iona College Tech. Report #MII-1305, 1993

10 A. Mikulcic, J. Chen, "Experiments On Using Fuzzy Clustering For Fuzzy Control System Design", IEEE Proc, pp.2168-2174, 1996