The Sugeno Fuzzy Model Identification Method In Parameter Variation of Systems

Shravani Chandaka

Department of ECE,Chaitanya Institute of Technology, Visakhapatnam,India

Abstract **- This paper presents the sugeno model identification method by which a great number of systems whose parameters vary dramatically with working states can be identified via Fuzzy Neural Networks (FNN). The suggested method could overcome the drawbacks of traditional linear system identification methods which are only effective under certain narrow working states and provide a global dynamic description based on which further control of such systems may be carried out. The simulation results of a second-order parameter varying system demonstrate the effectiveness of the method.**

Keywords **- Parameter Varying Systems, TS Fuzzy Model, Fuzzy Neural Networks (FNN), Identification.**

1 INTRODUCTION

Controlled systems whose parameters vary dramatically with working states, namely parameter varying systems, are widely encountered in practical industrial situations. Although traditional linear system identification methods have been well established in the last twenty years, it can only be used under a certain narrow range of working conditions. Moreover, traditional controllers based on such models cannot cope with the changes in process dynamic effectively. Therefore, developing a global dynamic model and establishing the corresponding control schemes for the parameter varying systems are deeply expected.

Takagi and Sugeno [1,2,3] proposed a new type of fuzzy model (TS model) which has been widely used in many disciplines. Describing complex systems is one of the most important applications since the mathematical expression of the model is convenient to design controllers. Recently, the authors [4] suggested an identification method of the TS fuzzy model for nonlinear systems via Fuzzy Neural Networks (FNN). It has been proved effective in describing the systems. In this paper, the TS fuzzy model is generalized to the parameter varying systems, and an identification method based on FNN is presented. The simulation results of a second-order system verify the effectiveness of the method.

This paper is organized as follows: Section 2 gives mathematical expression of the TS fuzzy model and point out the reason why the model is fit for parameter varying systems. Identification method for the model via FNN is suggested in section 3. The structure and the training algorithm for the networks are also given in details. Section 4 gives the simulation results of a second-order parameter varying system. Finally, the conclusions of the paper are given in section 5.

2. TS FUZZY MODEL

Parameter varying systems which possess m working state characteristic variables, q inputs and single output can be described by the TS fuzzy model consisting of R rules where the i-th rule can be represented as:

Rule i: if z_1 is A_1^{i,k_1}, z_2 is A_2^{i,k_2}, $\cdots$, and z_m is A_m^{i,k_m}

$$\text{then } y^i = a_1^i x_1 + a_2^i x_2 + \cdots + a_q^i x_q \tag{1}$$

$$i = 1,2,\cdots,R.\ k_j = 1,2,\cdots,r_j.$$

Where R is the number of rules in the TS fuzzy model. $z_j\ (j=1,2,\cdots,m)$ is the j-th characteristic variable, which reflects the working state of the systems and can be selected as input, output or other variables affecting the parameters of system dynamics. $x_l\ (l=1,2,\cdots,q)$ is the l-th model input. y^i is the output of the i-th rule. For the i-th rule, A_j^{i,k_j} is the k_j-th fuzzy subset of z_j. a_l^i is the coefficient of the consequent. r_j is the fuzzy partition number of z_j. For simplicity of induction, we let $r_j = r$ and r is determined by both the complexity and the accuracy of the model.

Once a set of working state variables $(z_{10}, z_{20}, \cdots, z_{m0})$ and model input variables $(x_{10}, x_{20}, \cdots, x_{q0})$ are available, then the output of the TS model under such working states can be calculated by the weighted-average of each y^i:

$$y = \sum_{i=1}^{R} \mu^i y^i \Big/ \sum_{i=1}^{R} \mu^i \tag{2}$$

where y^i is determined by the consequent equation of the i-th rule. The truth-value μ^i of the i-th rule can be calculated as:

$$\mu^i = \bigwedge_{j=1}^{m} A_j^{i,k_j}(z_{j0}) \tag{3}$$

Furthermore, the equation (2) can be rewritten as:

$$y = \left(\sum_{i=1}^{R} \mu^i a_1^i x_1 + \cdots + \sum_{i=1}^{R} \mu^i a_q^i x_q \right) \Big/ \sum_{i=1}^{R} \mu^i \tag{4}$$

From (4), one can see that the TS fuzzy model can be expressed as an ordinary linear equation under certain working states since the truth-value μ^i is only determined by the working state variables. As μ^i varies with working state, TS fuzzy model is a coefficient-varying linear equation. For all possible varying ranges of working states, the TS fuzzy model reflects the relationships between model parameters and working states. Therefore, the global dynamic characteristics of the parameter varying systems can be represented.

3. FUZZY NEURAL NETWORKS TS FUZZY MODEL IDENTIFICATION METHOD

A. *Structure of the FNN*

According to (1~3), the structure of FNN presented here consists of a premise, consequent and fuzzy inference. For systems which posses m working state characteristic variables, q inputs and a single output, the FNN used for the TS model identification is shown in Fig. 1. The circles and the squares in the figure represent the units of the networks. The notations between the units denote the connection weights. The units without any notation just deliver the signals from input to output.

1) Normalization of the working state variables

Layers (A)~(B) of the FNN are used to normalize the working state variables in case of saturation of the premise nodes. Assuming P samples $\left(z_1^p, z_2^p, \cdots, z_m^p\right)$ $(p = 1,2,\cdots,P)$ are available for training the networks, the j-th working state variable of the p-th sample can be normalized as:

$$\bar{z}_j^p = \left(w_s\right)_j \left(z_j^p - \left(w_t\right)_j\right) \tag{5}$$

where $\bar{z}_j^p$ is the normalized working state variable of $\bar{z}_j^p$; $\left(w_s\right)_j$ and $\left(w_t\right)_j$ are the coefficients and biases of normalization respectively:

$$\left(w_s\right)_j = \frac{2}{\max\left(z_j^p\right) - \min\left(z_j^p\right)}$$

$$\left(w_t\right)_j = \frac{\max\left(z_j^p\right) + \min\left(z_j^p\right)}{2}$$

$$j = 1,2,\cdots,m. \; p = 1,2,\cdots,P$$

2) Premise

The premise parts of the FNN include Layers (C)~(F) which are used for fuzzy partition and truth-value calculations. Signature 'Σ' in layer (D) ,which is the sum node, realizes the following operations for the k-th fuzzy subset of $\bar{z}_j$:

$$\Sigma : \begin{cases} I_{j,k}^{(D)} = \left(w_g\right)_{j,k} \bar{z}_j - \left(w_c\right)_{j,k} \\ O_{j,k}^{(D)} = I_{j,k}^{(D)} \end{cases} \tag{6}$$

Signature 'Λ' in layer (F) is the fuzzy minimum node and the input-output relationships for the i-th rule can be written as:

$$\Lambda : \begin{cases} I_i^{(F)} = \min\limits_{j=1 \text{ and } k=\phi(i,j)}^{m} O_{j,k}^{(E)} \\ O_i^{(F)} = I_i^{(F)} \end{cases} \tag{7}$$

$$\phi(i,j) = \operatorname{int}\left(\frac{i - \sum_{l=0}^{j-1}\left[\left(\phi(i,l) - 1\right) r^{(m-l)}\right] - 1}{r^{(m-j)}} \right) + 1 \tag{8}$$

$$\phi(i,0) = 1. \; i = 1,2,\cdots,R. \; j = 1,2,\cdots,m. \; k = 1,2,\cdots,r$$

where $I_{j,k}^{(\cdot)}$ and $O_{j,k}^{(\cdot)}$ are input and output of the nodes which correspond to the k-th fuzzy subset of $\bar{z}_j$ in layer $(\cdot)$ respectively; $I_i^{(\cdot)}$ and $O_i^{(\cdot)}$ are input and output of the nodes which correspond to the i-th rule in layer $(\cdot)$ respectively; the central point and gradient of the k-th fuzzy subset for $\bar{z}_j$ are determined by both $\left(w_g\right)_{j,k}$ and $\left(w_c\right)_{j,k}$; $\phi(i,j)$ represents the connective relationship between the i-th rule and the k-th fuzzy subset of $\bar{z}_j$.

The membership functions of the working state variables are determined by activation functions of the nodes in layer (E). In this paper, the following activation functions are taken:

$$f_k(x) = \begin{cases} e^{-x} / \left(1 + e^{-x}\right) & , k = 1 \\ e^{-x^2} & , k = 2,3,\cdots,r-1 \\ 1 / \left(1 + e^{-x}\right) & , k = r \end{cases} \tag{9}$$

which realize fuzzy partition as shown in Fig. 2.

3) Consequent and fuzzy inference

Layers (G)~(J), which are used to implement the linear equations of the TS fuzzy model, are consequent parts of the FNN. As for the i-th rule of the consequent, input-output relation realized can be written as:

$$O_i^{(J)} = \sum_{j=1}^{q} \left(w_a\right)_{j,i} x_j \tag{10}$$

where $\left(w_a\right)_{j,i}$ is the coefficient of x_i in rule $\boldsymbol{i}$.

Layers (K)~(M) realize the fuzzy inference as shown in (2).

B. *Learning algorithm*

Two kinds of parameters need to be learnt by the FNN. One is $\left(w_g\right)_{j,k}$ and $\left(w_a\right)_{j,k}$, which determine central points and gradients of the membership functions in the premise. The other is $(w_a)_{j,i}$, which determines local linear relationships of the consequent.

Assuming P samples $\left(z_1^p, z_2^p, \cdots, z_m^p, x_1^p, x_2^p, \cdots, x_q^p\right)$ $(p = 1, 2, \cdots, P)$ are available for training the FNN and the corresponding teacher signal is t^p. Once the p-th sample is put on the networks, the actual output y^p of the networks can be obtained. Thus, the learning error function of the sample can be defined as:

$$E^p = \frac{1}{2}\left(t^p - y^p\right)^2 \tag{11}$$

Under this definition, the total error function of all the samples can be written as:

$$E = \sum_{p=1}^{P} E^p = \frac{1}{2}\sum_{p=1}^{P}\left(t^p - y^p\right)^2 \tag{12}$$

According to the Gradient-Descent learning algorithm, one can obtain:

$$\begin{aligned}\frac{\partial E}{\partial (w_a)_{l,i}} &= \sum_{P=1}^{p}\left\{\frac{\partial E^p}{\partial y^p}\cdot\frac{\partial y^p}{\partial O_l^{(J)p}}\cdot\frac{\partial O_l^{(J)p}}{\partial (w_a)_{l,i}}\right\} \\ &= \sum_{p=1}^{P}\left\{\left(t^p - y^p\right)\cdot\left(O_l^{(J)p}\Big/\sum_{k=1}^{R}O_k^{(F)p}\right)\cdot x_l^p\right\}\end{aligned} \tag{13}$$

$$\begin{aligned}\frac{\partial E}{\partial (w_g)_{j,k}} &= \sum_{p=1}^{P}\left\{\frac{\partial E}{\partial y^p}\cdot\frac{\partial y^p}{\partial (w_g)_{j,k}}\right\} \\ &= \sum_{p=1}^{P}\left\{\frac{\partial E}{\partial y^p}\cdot\sum_{i=1 \text{ and } k=\phi(i,j)}^{R}\left[\frac{\partial y^p}{\partial O_i^{(F)p}}\cdot\frac{\partial O_i^{(F)p}}{\partial O_{j,k}^{(E)p}}\right]\right. \\ &\qquad \left.\cdot\frac{\partial O_{j,k}^{(E)p}}{\partial I_{j,k}^{(E)p}}\cdot\frac{\partial I_{j,k}^{(E)p}}{\partial (w_g)_{j,k}}\right\} \\ &= -\sum_{p=1}^{P}\left\{\left(t^p - y^p\right)\right. \\ &\qquad \cdot\sum_{i=1 \text{ and } k=\phi(i,j)}^{R}\left[\left(O_i^{(F)p} - y^p\right)\Big/\sum_{l=1}^{R}O_l^{(F)p}\cdot\frac{\partial O_i^{(F)p}}{\partial O_{j,k}^{(E)p}}\right] \\ &\qquad \left.\cdot\frac{\partial O_{j,k}^{(E)p}}{\partial I_{j,k}^{(E)p}}\cdot z_j^p\right\}\end{aligned} \tag{14}$$

In order to solve $\partial O_i^{(F)p}/\partial O_{j,k}^{(E)p}$ in (14) , the following equivalent transition for (7) is needed:

$$\begin{aligned}O_i^{(F)} &= \min_{j=1 \text{ and } k=\phi(i,j)}^{m} O_{j,k}^{(E)} \\ &= \sum_{j=1 \text{ and } k=\phi(i,j)}^{m}\left[O_{j,k}^{(E)}\cdot\prod_{l=1 \text{ and } l\neq j}^{m}\mathrm{I}\left(O_{j,k}^{(E)} - O_{l,k}^{(E)}\right)\right]\end{aligned} \tag{15}$$

where:

$$\mathrm{I}\left(O_{j,k}^{(E)} - O_{l,k}^{(E)}\right) = \begin{cases}1\ , & O_{j,k}^{(E)} \le O_{l,k}^{(E)} \\ 0\ , & O_{j,k}^{(E)} > O_{l,k}^{(E)}\end{cases} \tag{16}$$

Therefore, $\partial O_i^{(F)p}/\partial O_{j,k}^{(E)p}$ can be calculated by

$$\begin{aligned}\frac{\partial O_i^{(F)p}}{\partial O_{j,k}^{(E)p}} &= \prod_{l=1 \text{ and } l\neq j}^{m}\mathrm{I}\left(\partial O_{j,k}^{(E)p} - \partial O_{l,k}^{(E)p}\right) \\ &= \begin{cases}1\ , & \partial O_{j,k}^{(E)p} \le \partial O_{l,k}^{(E)p} \\ 0\ , & \partial O_{j,k}^{(E)p} > \partial O_{l,k}^{(E)p}\end{cases}\end{aligned} \tag{17}$$

Moreover, $\partial O_{j,k}^{(E)p}/\partial I_{j,k}^{(E)p}$ can be obtained from (10) as follows:

$$\frac{\partial O_{j,k}^{(E)p}}{\partial I_{j,k}^{(E)p}} = \begin{cases}O_{j,1}^{(E)p}\cdot\left(O_{j,1}^{(E)p} - 1\right)\ , & k = 1 \\ -2I_{j,k}^{(E)p}\cdot O_{j,k}^{(E)p}\ , & k = 2,3,\cdots,r\text{-}1 \\ O_{j,r}^{(E)p}\cdot\left(1 - O_{j,r}^{(E)p}\right)\ , & k = r\end{cases} \tag{18}$$

From (14), (17) and (18), $\partial E/\partial (w_g)_{j,k}$ can be obtained.

Using the same method mentioned above, $\partial E/\partial (w_c)_{j,k}$ can also be represented by

$$\begin{aligned}\frac{\partial E}{\partial (w_c)_{j,k}} &= \sum_{p=1}^{P}\left\{\frac{\partial E^p}{\partial y^p}\cdot\frac{\partial y^p}{\partial (w_c)_{j,k}}\right\} \\ &= \sum_{p=1}^{P}\left\{\frac{\partial E^p}{\partial y^p}\cdot\sum_{i=1 \text{ and } k=\phi(i,j)}^{R}\left[\frac{\partial y^p}{\partial O_i^{(F)p}}\cdot\frac{\partial O_i^{(F)p}}{\partial O_{j,k}^{(E)p}}\right]\right. \\ &\qquad \left.\cdot\frac{\partial O_{j,k}^{(E)p}}{\partial I_{j,k}^{(E)p}}\cdot\frac{\partial I_{j,k}^{(E)p}}{\partial (w_c)_{j,k}}\right\} \\ &= -\sum_{p=1}^{P}\left\{\left(t^p - y^p\right)\right. \\ &\qquad \cdot\sum_{i=1 \text{ and } k=\phi(i,j)}^{R}\left[\left(O_i^{(F)p} - y^p\right)\Big/\sum_{l=1}^{R}O_l^{(F)p}\cdot\frac{\partial O_i^{(F)p}}{\partial O_{j,k}^{(E)p}}\right] \\ &\qquad \left.\cdot\frac{\partial O_{j,k}^{(E)p}}{\partial I_{j,k}^{(E)p}}\cdot(-1)\right\}\end{aligned} \tag{19}$$

Therefore, the final tuning equations of the premise and consequent parameters of the FNN can be written as:

$$\left(w_a\right)_{j,i}(n+1) = \left(w_a\right)_{j,i}(n) - \zeta \cdot \partial E \Big/ \partial\left(w_a\right)_{j,i} \quad (20)$$

$$\left(w_g\right)_{j,k}(n+1) = \left(w_g\right)_{j,k}(n) - \xi \cdot \partial E \Big/ \partial\left(w_g\right)_{j,k} \quad (21)$$

$$\left(w_c\right)_{j,k}(n+1) = \left(w_c\right)_{j,k}(n) - \xi \cdot \partial E \Big/ \partial\left(w_c\right)_{j,k} \quad (22)$$

where n is the training times; ζ and ξ are learning rates. In this paper, we use the adaptive back-propagation algorithm suggested by the authors [5].

4. SIMULATION EXAMPLE

Considering the following second-order parameter varying system:

$$\frac{y(s)}{u(s)} = \frac{1}{(1+Ts)^2} \quad (23)$$

where the time constant T is affected by a working state variable z $\left(z \in [0.3, 0.9]\right)$. Suppose the relationship between them is:

$$T = 20 + 20 \cdot (z - 0.3) \quad (24)$$

Once the sample time T_0 is given, the discrete time description of the system could be obtained:

$$y(k) = \Big[2 \cdot \left(T^2 + T \cdot T_0\right) \cdot y(k-1) - T^2 \cdot y(k-2) + T_0^2 \cdot u(k-1)\Big] \Big/ \left(T + T_0\right)^2 \quad (25)$$

In this paper, the sample time T_0 is taken as 5 seconds. Curves 1, 2 and 3 in Fig. 3 show the unit step response of the system at $z = 0.3$, $z = 0.6$ and $z = 0.9$ respectively and one can see that variations between the different z's are very large.

Using the suggested FNN TS model identification method, we select z as a working state variable for the input of premise in the FNN and take $u(k-1)$, $y(k-1)$ and $y(k-2)$ as input variables for the TS model. The aim of the identification is to obtain the global model which is suitable for all the possible working states of the system. First, ten states are selected randomly, and 310 groups of training data are obtained by exerting 5-order M sequels which have the range of 1 on the system. All of the weights in the consequent of the FNN are selected between -0.1 and 0.1 randomly, and the fuzzy partition number r is selected to 7 as shown in Fig. 2.

In order to fasten the convergence rate of the networks, the following parameters are used as the initial value of the adaptive BP algorithm shown in [5]:

$$\zeta(1) = 0.9,\ \xi(1) = 0.4,\ \alpha_0 = 1.4,\ \alpha_1 = 0.6,\ E_s = 0.5$$

The final convergence conditions are taken as:

1) The number of the samples which have satisfied $\left(t^p - y^p\right) \big/ t^p \le 0.05$ has exceeded 95 percent of the total samples.

2) Training times has exceeded the maximum times specified as 10000.

After training the FNN 868 times, the networks converged by satisfying condition 1) and the final simulation results are shown in Fig 4 and Fig. 5. As shown in Fig. 4, where the solid line and the dotted line denote the expected output of the system and the actual output of the networks respectively, most of the samples have good performance to describe the actual outputs of the system.

Finally, we use another ten groups of z to verify the performance of the resulted FNN TS fuzzy model and the results are shown in Fig. 6. The same conclusion can be drawn from it. Therefore, the suggested TS model identification method is strongly effective to obtain the global dynamic model of parameter varying systems.

5. CONCLUSIONS

This paper generalizes the TS model to the parameter varying systems and presents the corresponding identification method via FNN. The proposed method can effectively realize the identification of parameter varying systems whereas the traditional linear system identification methods can not. Furthermore, control of such systems based on the well-established TS fuzzy model can be carried out and this further research field creates for us. The simulation results of a second-order parameter varying system have fully verified the effectiveness of this method.

It should be noted that a more effective way is provided to establish fuzzy control rules for the multi-working-states situations. Based on the model, performance of fuzzy controller will be greatly improved under such situations.

7 REFERENCE

[1] T. Takagi and M. Sugeno, “Fuzzy identification of system and its applications to modeling and control”, *IEEE Trans. on System Man and Cybernetics*, vol. SMC-15, no. 1, 1985, pp. 116-132.

[2] M. Sugeno and G. T. Kang, “Structure identification of fuzzy model”, *Fuzzy Sets and Systems*, vol. 28, 1988, pp. 15-33.

[3] M. Sugeno and K. Tanaka, “Successive identification of a fuzzy model and its applications to prediction of a complex system”, *Fuzzy Sets and Systems*, vol. 42, 1992, pp. 315-334.

[4] Xie Keming and Zhang Jianwei. “A linear fuzzy model identification method based on fuzzy neural networks”, *in Proceedings of the 2nd Worldwide Chinese Intelligence Control and Intelligence Automation Conference*, 1997.

[5] Xie Keming and Zhang Jianwei, “An Adaptive Backpropagation Algorithm Based on Error Rate of Change”, *submitted to Journal of Taiyuan University of Technology*.

ISSN: 2278-0181
Vol. 1 Issue 9, November- 2012

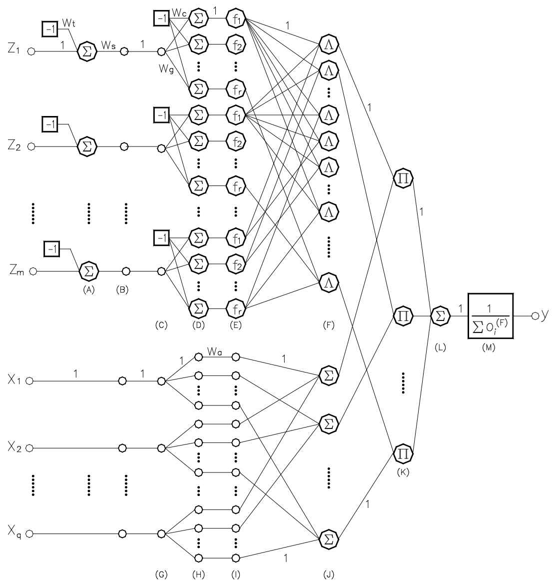

Fig. 1. Structure of FNN

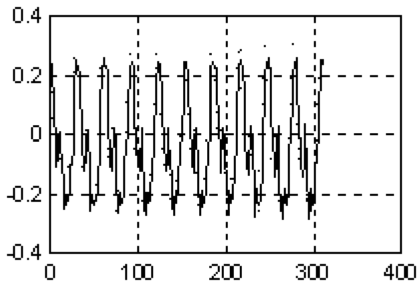

Fig. 4. Results of training

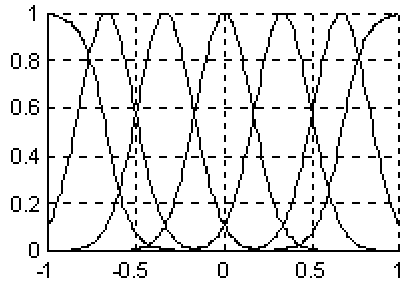

Fig. 2. Fuzzy subsets of working state variables

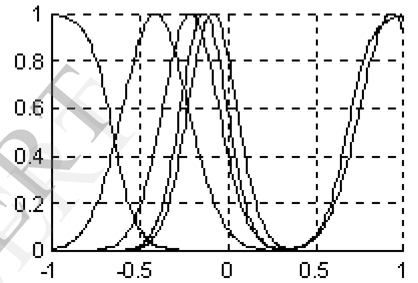

Fig. 5. Membership functions after training

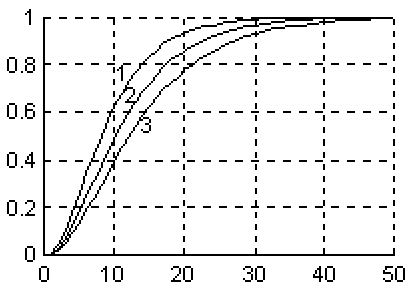

Fig. 3. Unit step response under different working state

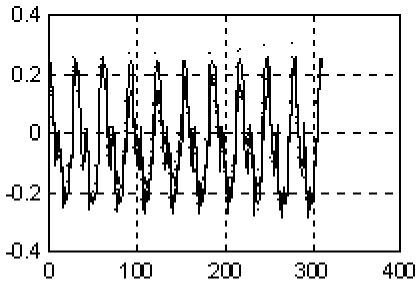

Fig. 6. Results of verification