

## Fuzzy system identification using an adaptive learning rule with terminal attractors

Chuan-Kai Lin, Sheng-De Wang\*

*Department of Electrical Engineering, EE Building, Rm. 441, National Taiwan University, Taipei 106, Taiwan*

Received December 1995; received in revised form March 1997

---

### Abstract

A fuzzy model with an adaptive learning rule is proposed to approximate a class of nonlinear systems. The proposed adaptive learning rule uses the concept of terminal attractors and guarantees that the identification is stable and fast convergence in finite time. Another subject of this paper is to determine the appropriate rule number of the fuzzy model required for a desired modeling error. A learning algorithm is also introduced to identify the centers of membership functions of fuzzy outputs. © 1999 Elsevier Science B.V. All rights reserved.

*Keywords:* Fuzzy model; Terminal attractor; Adaptive learning rule; System identification

---

### 1. Introduction

The mathematical models of complex and nonlinear systems are often very difficult to obtain. However, control engineers always face the problems of controlling and modeling such systems. In recent years, there has been growing interest in using neural networks and fuzzy set theory for modeling nonlinear and complex systems because fuzzy models offer a framework for combining linguistic information and numerical data in a unified fashion. However, most tasks are dedicated to tuning parameters of fuzzy models [14,11]. As for the topic of determining the rule number, it is still worth for researching a systematic approach.

When designing a fuzzy model, we shall first determine the number of rules and the modeling accuracy.

Then, use a learning algorithm to find out the free parameters of the model. The determination of an appropriate number of rules is not an easy task since it is a nontrivial function of modeling accuracy. In previous research work [14], the number of rules is, in general, given a priori by experienced designers. Therefore, we present a systematic method to determine the number of rules.

This paper is written with two main objectives. The first is to introduce a method for determining the number of rules of fuzzy model. The second objective is to develop a fast on-line learning rule to adjust free parameters of the proposed model. The concept of terminal attractors [18], which has fast convergence property, is incorporated into the adaptive learning rule. Hence, the adaptive learning rule provides properties of stability and convergence in finite time. The convergence rate of our approach is faster than conventional system identification with the backpropagation algorithm.

---

\* Corresponding author.

A fuzzy model or a fuzzy logic system, can be composed of four components: a fuzzifier, a fuzzy rule base, a fuzzy inference engine and a defuzzifier. The detailed description of the four components can be found in [5, 6]. Therefore, we just review them briefly.

Consider a fuzzy system:  $U \subset R^n \rightarrow R^m$ , where  $U$  is a compact set of universe discourse and  $n$  is the dimension of the input space and  $m$  is the dimension of the output space. The purpose of the fuzzifier is to map the real inputs to the fuzzy sets, which are characterized by membership functions  $\mu_F : U \rightarrow [0, 1]$  and represent linguistic terms such as “small”, “medium” and “large”. The shape of membership functions may be triangular, trapezoidal and Gaussian-like functions.

After the fuzzification process, we enter the fuzzy inference engine that employs fuzzy rules to map from fuzzy sets in input space to the fuzzy sets in output space. Assume that there are  $M$  rules in the fuzzy rule base and all are in the following form:

IF  $x_1$  is  $\tilde{A}_{j1}$  and  $x_2$  is  $\tilde{A}_{j2}$  and ... and  $x_n$  is  $\tilde{A}_{jn}$

THEN  $y_1$  is  $\tilde{B}_{1j}$  and ... and  $y_m$  is  $\tilde{B}_{mj}$ .

In the above rules,  $x_i$  are the real inputs into the fuzzy system,  $y_k$  are the output variables of the fuzzy system and  $\tilde{A}_{ji}$  and  $\tilde{B}_{kj}$  are linguistic terms characterized by membership functions  $\mu_{\tilde{A}_{ji}}(x_i)$  and  $\mu_{\tilde{B}_{kj}}(y_k)$ , respectively. By using fuzzy logic operations such as T-norm, the fuzzy inference engine determines the fuzzy outputs. The last procedure is to find a corresponding value in the problem domain using a defuzzifier that usually employs the center-of-area method. In this paper, a form of center-average defuzzification and the singleton fuzzification will be used in the proposed fuzzy model.

The paper is organized as follows. In Section 2, we will introduce the architecture of a fuzzy model and discuss the methodology to construct the fuzzy model. In Section 3, we construct the nonlinear identifiers with an adaptive learning rule for continuous systems. The stability and convergence of estimated errors are also analyzed in this section. A design algorithm summarizes the approach proposed in Sections 2 and 3. Section 4 then shows the simulation results. At last, Section 5 concludes the paper.

## 2. The proposed fuzzy modeling approach

The fuzzy logic systems with a center-average defuzzifier and a singleton fuzzifier can be represented as linear combinations of functions of normalized firing strength. From the  $k$ th output of the center-average defuzzifier,  $\hat{y}_k$ , is as follows:

$$\hat{y}_k(\mathbf{x}) = \frac{\sum_{j=1}^M \hat{w}_{kj} \prod_{i=1}^n \mu_{\tilde{A}_{ji}}(x_i)}{\sum_{j=1}^M \prod_{i=1}^n \mu_{\tilde{A}_{ji}}(x_i)}, \tag{1}$$

$$k = 1, 2, \dots, m,$$

where  $\hat{w}_{kj}$ 's are the centers of fuzzy outputs. Here  $\hat{y}_k$  can be viewed as the linear combination of normalized firing strength  $\varphi_j(\mathbf{x})$  which is defined as follows.

**Definition 1** (Normalized firing strength). Define the normalized firing strength as

$$\varphi_j(\mathbf{x}) = \frac{\beta_j}{\sum_{j=1}^M \beta_j} \tag{2}$$

$$\beta_j = \prod_{i=1}^n \mu_{\tilde{A}_{ji}}(x_i), \quad j = 1, 2, \dots, M, \tag{3}$$

where  $\mu_{\tilde{A}_{ji}}(x_i)$  are membership functions,  $M$  is the rule number,  $\prod$  is the implication operator and  $\sum_{j=1}^M \varphi_j = 1$ .

From the above description, we can view the defuzzifier as a summation of products of the normalized firing strengths and centers. As a result, the fuzzy model  $\hat{f}$  can be represented in the following form:

$$\hat{f} = \hat{W}^T s \tag{4}$$

and

$$y(t) = f(\mathbf{x}) = \hat{f}(\mathbf{x}) + e(t), \tag{5}$$

where  $\mathbf{y} = [y_1 \ y_2 \ \dots \ y_m]^T$ ,  $e(t)$  is the model error vector,  $\mathbf{s} = [s_1 \ s_2 \ \dots \ s_M]^T$  is the firing strength and  $\hat{W}$  is an  $M \times m$  real adjustable parameters matrix with entry  $\hat{w}_{kj}$ . The fuzzy model is shown in Fig. 1. For the single-output case, we assume  $f(\mathbf{x})$  is continuously differentiable  $(q+1)$  times and the  $(q+1)$ th derivative of  $f(\mathbf{x})$  is also bounded, i.e.,  $\|\nabla^{q+1} f(\mathbf{x})\|_2 < M_f$  for all  $\mathbf{x} \in U$  ( $\|\cdot\|$  is the induced norm). For the case of the Gaussian membership function, we can use a

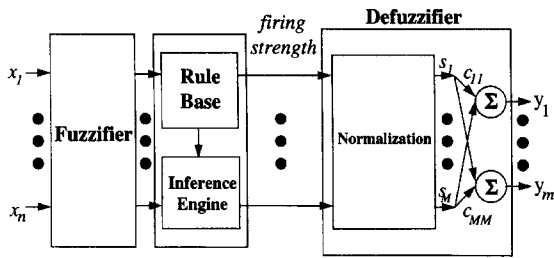


Fig. 1. The structure of fuzzy model.

center vector  $c_j$  to represent the  $n$  input centers of the  $j$ th rule. In [3], an inequality is obtained by Taylor's expansion to guarantee the existence of  $\hat{f}(x)$ :

$$\sum_{j=1}^M g_j(x) \beta_j(x) < 0, \tag{6}$$

where  $g_j(x) = \|x - c_j\|_2^{q+1} - \varepsilon(q+1)!/M_f$  and  $\varepsilon$  is an arbitrary real number. If inequality (6) holds for all  $x \in U$ , then  $\|f(x) - \hat{f}(x)\|_2 < \varepsilon$ . A geometric interpretation is illustrated in Fig. 2.

Johansen and Foss used inequality (6) to construct a global nonlinear ARMAX (NARMAX) model by interpolating local ARMAX models that are valid within certain operating regimes [3]. The local model validity functions are defined to indicate the relative validity of local models at a given operating point. Each local model validity function,  $\beta_j(x)$ , which is a Gaussian function, is defined to indicate the relative validity of the corresponding local model at a given operating point. The local model validity function bears analogy to the firing strength; the operating regime has analogy to the rule which has strongest firing strength. However, we can find out the differences between NARMAX models and fuzzy models. Johansen and Foss suggested that the width of the local model validity functions go to zero for satisfying inequality (6). From the point of view of fuzzy systems, it means that there is only one rule to be fired at each state. This phenomenon distinguishes the NARMAX model from the fuzzy model since fuzzy systems always fire several rules as a result of fuzzy reasoning. Hence, we can modify the method proposed by Johansen and Foss for fuzzy systems to determine the rule number of fuzzy rule base such that the modeling error

is within the specified upper error bound. The  $\alpha$ -cut fuzzy set is incorporated into the modified method. The set of elements that belong to the fuzzy set  $\tilde{A}$  at least to the degree  $\alpha$  is called the  $\alpha$ -level fuzzy set or  $\alpha$ -cut fuzzy set:

$$\tilde{A}_\alpha = \{x \in U \mid \mu_{\tilde{A}}(x) \geq \alpha\}.$$

Hence, we define a modified firing strength function as

$$\beta_j^\alpha = \begin{cases} 0, & \beta_j < \alpha, \\ \beta_j, & \beta_j \geq \alpha. \end{cases} \tag{7}$$

In Fig. 3, it is obvious that the width of  $\beta_j(x)$  is not necessarily small enough to satisfy inequality (6) after applying the  $\alpha$ -cut fuzzy set. We also propose a modified firing strength function which is capable of enlarging the width and can be incorporated into the adaptive learning rule as will be discussed in Section 3.

**Definition 2 (Modified normalized firing strength function).** Define the modified normalized firing strength function  $s_j$  as

$$s_j(x) = \frac{\beta_j^\alpha}{\|\beta^\alpha\|_2}, \quad j = 1, 2, \dots, M \tag{8}$$

where  $\beta^\alpha = [\beta_1^\alpha \ \beta_2^\alpha \ \dots \ \beta_M^\alpha]^T$  and  $\|\beta^\alpha\|_2 = \sqrt{\sum_{j=1}^M (\beta_j^\alpha)^2}$ . And the normalized firing strength function has the property  $\sum_{j=1}^M s_j^2 = 1$ .

As  $M_f$  and  $\varepsilon$  are given, the shapes of  $g_j$ -functions are determined. The location of a  $g_j$ -function is associated with the  $\beta_j$  and it is free to choose  $\alpha$  and  $\beta_j$  such that inequality (6) holds. However, the rule number  $M$  must be sufficiently large to cover the whole input space of interest. The basic idea of designing premises of fuzzy rules is to cover all input space by satisfying inequality (6). Take a simple example to illustrate the above discussion. Assume  $f(x) = x + \sin(x/2)$ ,  $q$  is chosen as 1 and the accuracy  $\varepsilon$  is set to be 0.01, and the  $g_j$ -functions are given by  $g_j(x) = \|x - c_j\|_2^2 - 0.08$  ( $M_f = 0.25$ ). For covering the interval  $[0.0, 2\pi]$ , there are 11 rules with  $\alpha = 0.3$  as shown in Fig. 3. The boundary of the operating region of each rule is determined by the  $\alpha$ -cut fuzzy set.

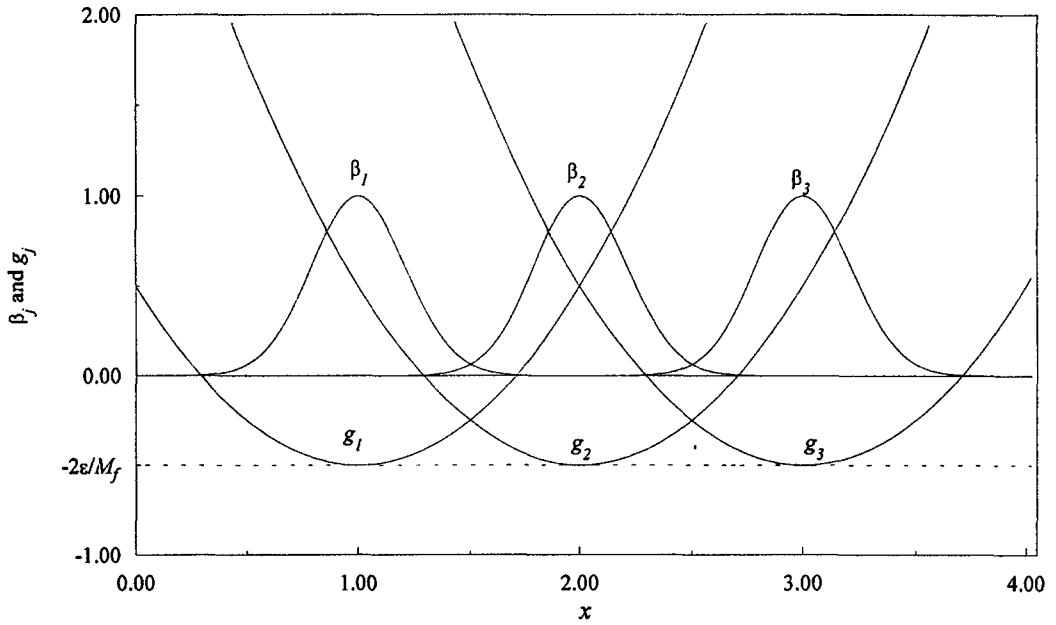


Fig. 2. Geometrical interpretation of Johansen and Foss method.

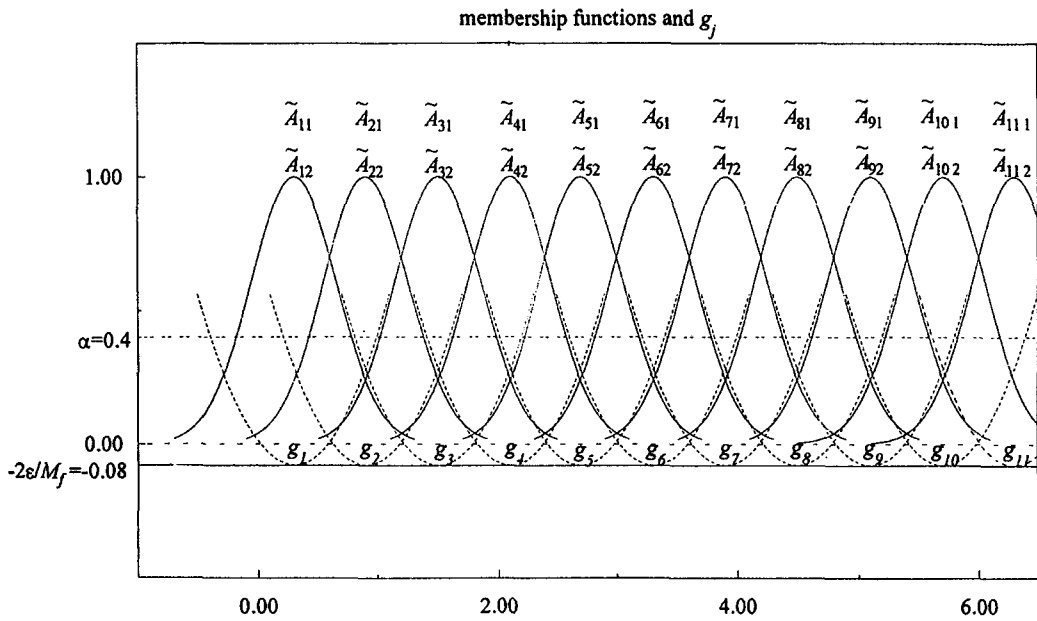


Fig. 3. Geometrical interpretation of Eq. (6):  $f(x) = x + \sin(x/2)$ .

### 3. Adaptive learning rule with terminal attractors

In the previous section, the method of constructing the premise structure (IF parts) of a fuzzy model has been developed and there remain free parameters in the consequent structure (THEN parts) to be identified by adaptive learning rule which will be discussed in this section. The proposed adaptive learning rule makes use of the concepts of terminal attractors and has the fast convergence rate and stability properties. On the other hand, there is a trade-off between the fuzzy rule number and the modeling error bound. The smaller the error bound is, the more the fuzzy rules will be obtained. Thus, we prefer to use a somewhat larger error bound to make the rule base small. Therefore, a robust adaptive technique [9–11] is added to reduce the approximating error. As a result, the specified upper error norm is not required to be very small to obtain the required approximation accuracy.

In this paper, we are concerned with the model identification of single-output–multi-input systems. The multi-input and multi-output fuzzy model can be divided into multi-input and single-output groups. It has been proven that if the mapping is well-defined then the fuzzy model is capable of approximating any real continuous function to arbitrary accuracy [14]. Therefore, we make the following assumption.

**Assumption 1.** There exists a coefficient vector matrix  $w_j^*$  such that the optimal fuzzy model  $f^*$  approximates the continuous function  $f$ , with accuracy  $\varepsilon$  over a compact set  $U$ , that is,  $\exists w_j^*$  such that

$$\max |f(\mathbf{x}) - f^*(\mathbf{x}, w_j^*)| \leq \varepsilon, \quad \forall \mathbf{x} \in U.$$

According to the above assumption, there exists  $w_j^*$  and  $\varepsilon_k$  such that the actual  $y_k(t)$  is given by

$$y_k(t) = \sum_{j=1}^M w_{kj}^*(t) s_j(\mathbf{x}) + \varepsilon_k, \quad (9)$$

and the estimated output  $\hat{y}_k(t)$  is given by

$$\hat{y}_k(t) = \sum_{j=1}^M \hat{w}_{kj}(t) s_j(\mathbf{x}) + \hat{\varepsilon}_k, \quad (10)$$

where  $\hat{w}_{kj}$ 's are an approximation to the optimum centers of membership functions in THEN parts from

the  $j$ th rule to the  $k$ th output. Hence,

$$y_k(t) - \hat{y}_k(t) = \sum_{j=1}^M \tilde{w}_{kj}(t) s_j(\mathbf{x}) + \tilde{\varepsilon}_k = e_k, \quad (11)$$

where  $\tilde{w}_{kj} = w_{kj}^* - \hat{w}_{kj}$  and  $\tilde{\varepsilon}_k = \varepsilon_k - \hat{\varepsilon}_k$ . As the number of rules is fixed, the smallest achievable  $\varepsilon_k$  can be determined. The modeling problem can thus be solved if we can use an adaptive learning law for tuning  $\hat{w}_{kj}$  and driving the errors  $e_k$  to zero. The weights of the  $k$ th output of the fuzzy model and the estimated error can be continuously, respectively, updated as

$$\dot{\hat{w}}_{kj} = \eta_{1k} e_k^p s_j(\mathbf{x}), \quad 0 < p < 1 \quad (12)$$

and

$$\dot{\hat{\varepsilon}}_k = \eta_{2k} e_k^p, \quad (13)$$

where  $\eta_{1k}$  and  $\eta_{2k}$  are positive constants determining the adaptation rates. Eqs. (12) and (13) constitute the adaptive learning rule. The introduction of estimated errors in (13) makes the equilibrium points of dynamic equations (12) and (13) become terminal attractors [8] that can guarantee the stability and convergence to the arbitrary small neighborhood of zero in finite time. Define the squared error  $E = \frac{1}{2} e_k^2(t) = \frac{1}{2} (y_k(t) - \hat{y}_k(t))^2$  as the error measure for the identification task. Now, a stability theorem is presented for the adaptive learning rule.

**Theorem 1.** Applying the adaptive learning rule (12) and (13) for fuzzy model (10). We have the following properties.

- (a) Output error  $e_k = 0$  is a terminal attractor of the dynamic equation describing  $\dot{E}$  and
- (b) the estimated parameters  $\hat{w}_{kj}$  of the fuzzy model will remain bounded.

**Proof.** (a) For proving that  $e_k = 0$  is a terminal attractor, compute the time derivative of  $E$  as follows:

$$\dot{E} = e_k \dot{e}_k = e_k \left( \sum_{j=1}^M \dot{\tilde{w}}_{kj} s_j(\mathbf{x}) + \dot{\tilde{\varepsilon}}_k \right)$$

$$\begin{aligned}
 &= e_k \left( \sum_{j=1}^M -\eta_{1k} e_k^p s_j^2(\mathbf{x}) - \eta_{2k} e_k^p \right) \\
 &= -\eta e_k^{1+p} = -\eta E^\gamma, \tag{15}
 \end{aligned}$$

where  $\eta = 2^{(1+p)/2}(\eta_{1k} + \eta_{2k})$  is a positive constant and  $0 < \gamma = (1 + p)/2 < 1$ . For  $E(0) \geq 0$ , the above equation implies that  $\dot{E} \leq 0$  with terminal attractor  $E = 0$ , i.e.  $e_i = 0$ .

(b) Consider the Lyapunov function candidate:

$$V = \frac{1}{2} e_k^2 + \frac{1}{2\eta_{1k}} \sum_{j=1}^M \tilde{w}_{kj}^2 + \frac{1}{2\eta_{2k}} \tilde{e}_k^2. \tag{16}$$

Using Eqs. (12) and (13), we have

$$\begin{aligned}
 \dot{V} &= -\eta E^\gamma + \frac{1}{\eta_{1k}} \sum_j \tilde{w}_{kj} (-\eta_{1k} e_k^p s_j(\mathbf{x})) \\
 &\quad + \frac{1}{\eta_{2k}} \tilde{e}_k (-\eta_{2k} e_k^p) \\
 &= -\eta E^\gamma. \tag{17}
 \end{aligned}$$

Thus, we have  $\dot{V} \leq -\eta E^\gamma$  for all  $t \geq 0$ , and for  $E \geq 0$ ,  $V$  will be asymptotically converged. Therefore, we can conclude that if  $\tilde{w}_{kj}$  are bounded at time  $t = 0$ , they will remain bounded for all  $t > 0$ .  $\square$

From Sections 2 and 3, we can conclude an algorithm to design the fuzzy model for identifying the given system.

### Design Algorithm

*Input:* 1. Given a desired modeling error  $e$  and  $M_f$ , that is the upper bound of first order derivative of the vector function  $f$  over  $\mathbf{x}$ .

2. Number of data patterns for identification:  $N$ .

3. Data patterns for identification and output dimension  $m$ .

*Output:* Rule number  $M$  and coefficient  $\hat{w}_{kj}$  ( $j = 1, \dots, M, k = 1, \dots, m$ ).

*Step 1:* Determine  $M$  by Eq. (6).

*Step 2:* For  $l = 1$  to  $N$  do Steps 3–8.

*Step 3:* For  $j = 1$  to  $M$

$$\beta_j = \prod_{i=1}^n \mu_{\tilde{A}_{ji}}(x_i).$$

*Step 4:* For  $j = 1$  to  $M$  do Steps 5–6.

*Step 5:* Calculate  $s_j$ .

*Step 6:* For  $k = 1$  to  $m$  do Steps 7–8.

*Step 7:* Calculate  $k$ th output error  $e_k$ .

*Step 8:*  $\hat{w}_{kj} = \eta_{1k} e_k^p s_j(\mathbf{x})$  and  $\hat{e}_k = \eta_{2k} e_k^p$ .

### 4. Simulation

There are two examples in this section. The first is the prediction of chaotic time series, and the second is to identify a multi-input and multi-output model. The plant dynamics were simulated in C language with Runge–Kutta algorithm on a Pentium 75 MHz PC.

**Example 1.** In this example, the fuzzy model is used to predict chaotic time series [10]. Chaotic time series, which can be generated by deterministic method, are complicated as random time series. The following discrete equation can generate chaotic time series:

$$y[k] = \gamma y[k - 1](1 - y[k - 1]),$$

where the chaotic behavior appears for certain choices of  $\gamma$ , for example,  $\gamma = 3.75$ . Fig. 4 shows a part of the chaotic series that are used in the simulation.

In the simulation, the rulebase with 225 rules has the following form:

IF  $y[k - 1]$  is  $\tilde{A}_{j1}$  and  $y[k - 2]$  is  $\tilde{A}_{j2}$

THEN  $y[k]$  is  $\tilde{B}_{j1}$ ,  $j = 1, 2, \dots, 225$ .

Each input variable is assigned to 15 membership functions for satisfying (6) and  $g_j(\mathbf{x}) = \|\mathbf{x} - \mathbf{c}_j\|_2^2 - e(2/M_f)$  ( $e = 0.005$  and  $M_f = 7.5$ ). The Gaussian-like membership functions are  $\mu_{\tilde{A}_{ji}}(x_i) = \exp(-600(x_i - c_{ji})^2)$  where  $c_{ji} = 0.2 + 0.6(j - 1)$ . Therefore, the firing strength  $\beta_j = \exp\{-600[(x_1 - c_{j1})^2 + (x_2 - c_{j2})^2]\}$  and  $\alpha$  and  $\eta_{1k}$  ( $= \eta_{2k}$ ) are set to 0.3 and 0.02, respectively. This simulation took about 40 s on a Pentium-75 MHz PC with 16 MB RAM. The recent proposed approach in [14] is a similar model with a

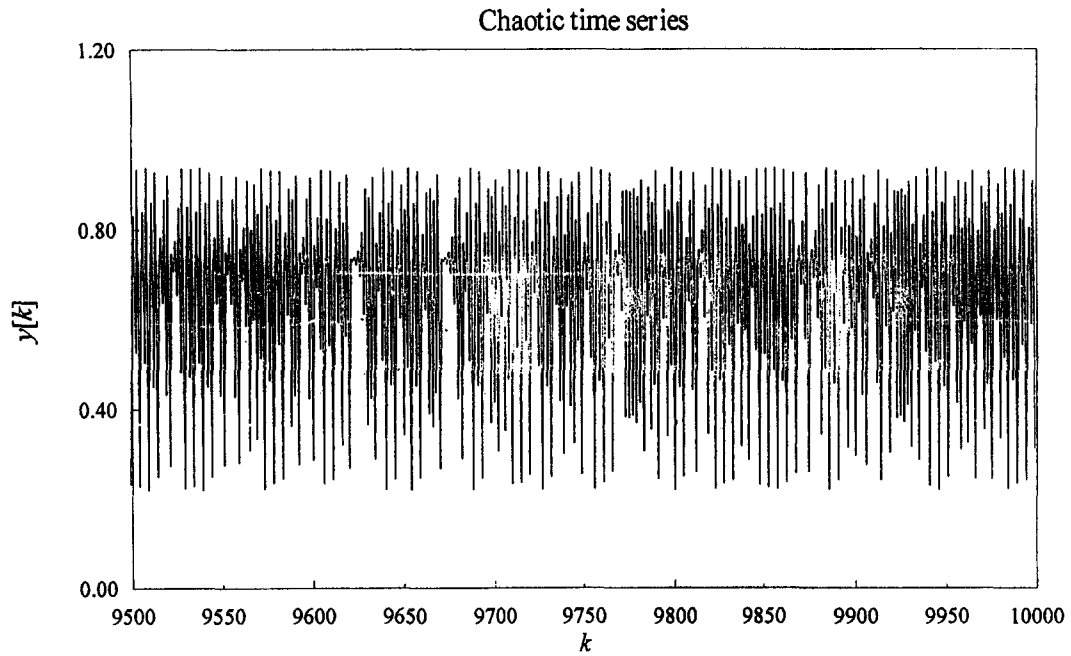


Fig. 4. The chaotic time series  $y[k] = \alpha y[k - 1](1 - y[k - 1])$  from  $k = 9500$  to  $10000$  ( $\alpha = 3.75$ ).

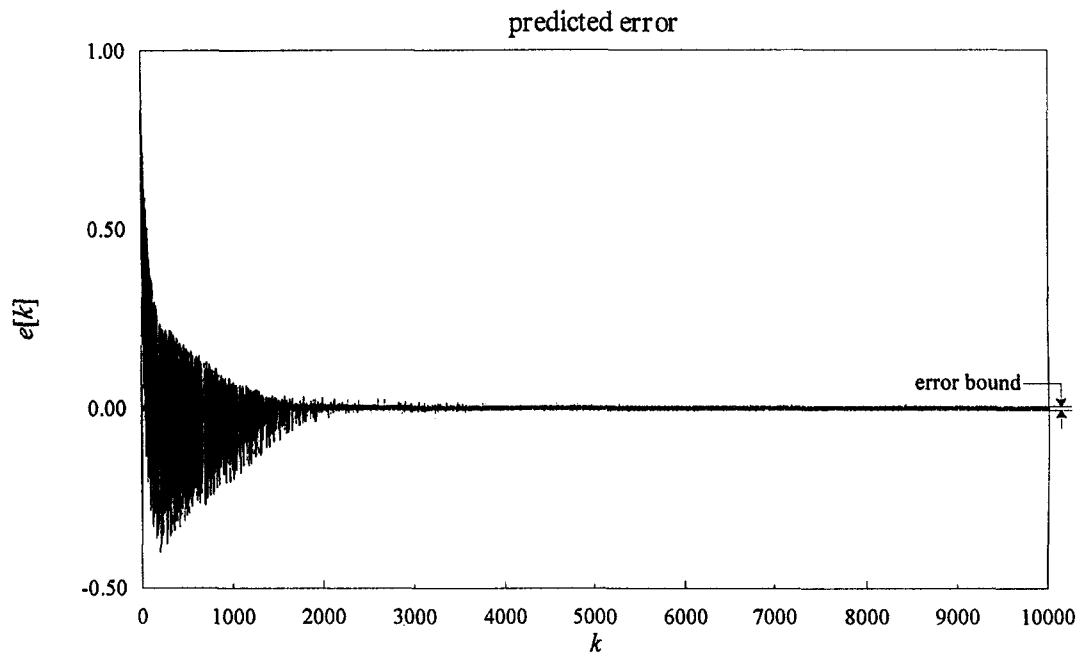


Fig. 5. The predicted error of the chaotic time series  $y[k] = \alpha y[k - 1](1 - y[k - 1])$  ( $\alpha = 3.75$ ).

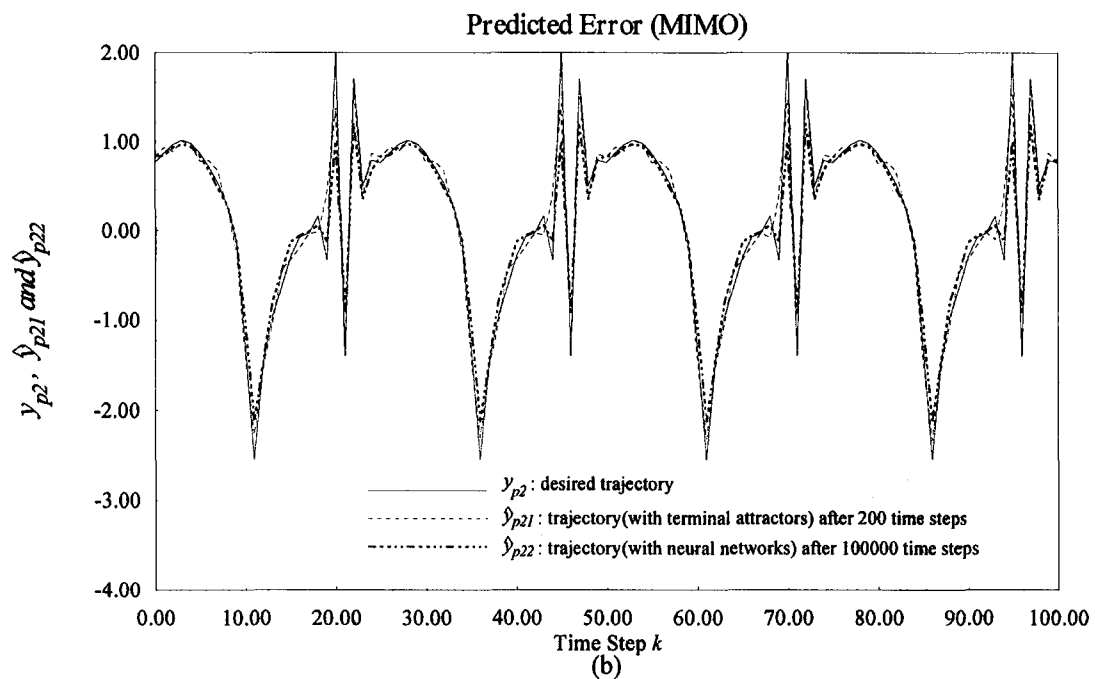
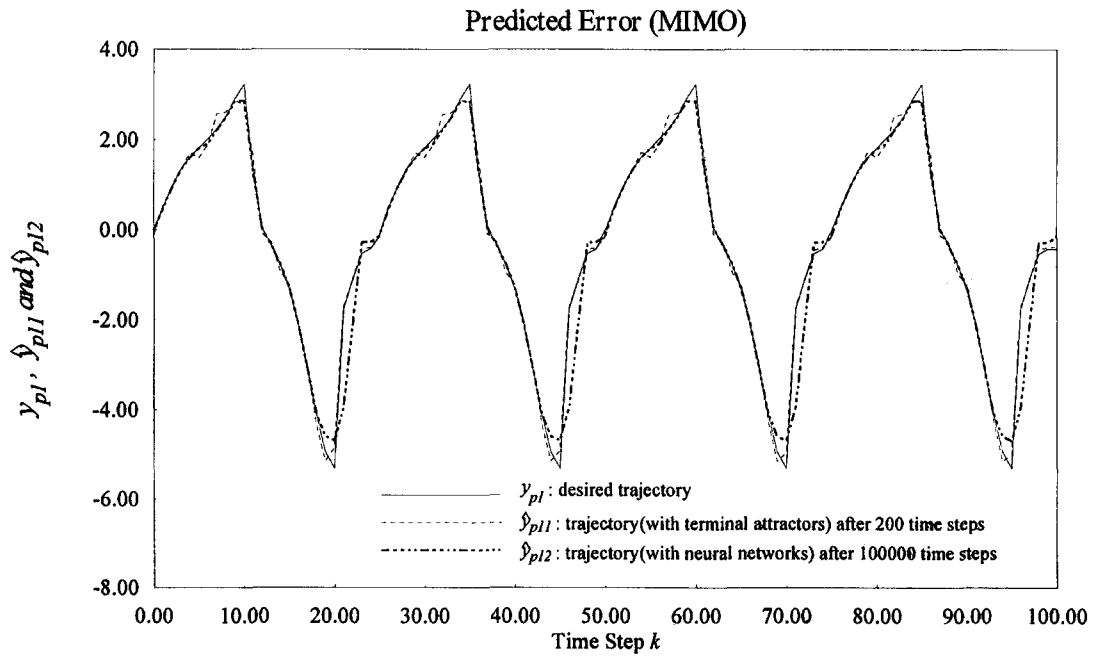


Fig. 6. MIMO model identification for (a)  $y_{p1}[k]$ , (b)  $y_{p2}[k]$ .



different learning algorithm which needs to compute an inverse matrix. Another approach is Gaussian neural networks. The architecture of Gaussian neural networks is too large (a total of 5776 nodes [10]) and it cannot represent the knowledge of linguistic rules. The approach proposed in this paper can complement the drawbacks.

The predicted error  $e(t)$  is defined as the difference between the chaotic time series  $y[k]$  and the estimated time series  $\hat{x}(t)$ , i.e.  $e(t) = \hat{x}(t) - x(t)$ . The upper error bound is set to 0.005. Fig. 5 shows most of the predicted error fall into the error bound as expected.

**Example 2.** In this example, it is shown that the MIMO plants can be identified by a MIMO fuzzy model which is composed of several multi-input and single-output groups. A nonlinear two-input and two-output plant is described as the following equations [7]:

$$\begin{bmatrix} y_{p1}[k+1] \\ y_{p2}[k+1] \end{bmatrix} = \begin{bmatrix} \frac{y_{p1}[k]}{1 + y_{p2}^2[k]} \\ \frac{y_{p1}[k]y_{p2}[k]}{1 + y_{p2}^2[k]} \end{bmatrix} + \begin{bmatrix} u_1[k] \\ u_2[k] \end{bmatrix},$$

where  $u_1[k]$  and  $u_2[k]$  are inputs and  $y_{p1}[k]$  and  $y_{p2}[k]$  are outputs. In the simulation,  $[u_1[k] \ u_2[k]]^T = [\sin(2\pi k/25) \ \cos(2\pi k/25)]^T$  and the resultant fuzzy rulebase has 81 rules. Since there are four fuzzy variables  $y_{p1}[k]$ ,  $y_{p2}[k]$ ,  $u_1[k]$  and  $u_2[k]$  are in the IF part and each fuzzy variable has three linguistic terms. The rulebase can be interpreted in the following forms:

IF  $y_{p1}[k]$  is P and  $y_{p2}[k]$  is Z and  $u_1[k]$  is N and  $u_2[k]$  is P then  $y_{p1}[k+1]$  is Z and  $y_{p2}[k+1]$  is Z.

The simulation results are shown in Fig. 6. The neural network proposed in [7] using 100 000 time steps to identify the system; however, our method only needs 200 time steps to achieve acceptable results. This implies the proposed model can identify a nonlinear MIMO dynamic system and has the fast convergence rate.

## 5. Conclusion

In this paper, we have considered the problem of system identification using neural fuzzy models and

achieves two objectives successfully: one is to determine the appropriate size of the fuzzy rulebase and another is to propose a new adaptive rule to learn the centers of output fuzzy sets. The first step of designing a fuzzy rulebase is to determine the rule number by the method described in this paper. As the number of rules and the linguistic IF part has been determined, the THEN part can be on-line tuned by the adaptive learning rule while the learning algorithm in [14] is off-line. Due to the introducing of terminal attractors, the convergence rate is very fast. Therefore, the methods proposed in this paper are effective.

## References

- [1] J.J. Buckley, Numerical relationships between neural networks, continuous functions, and fuzzy systems, *Fuzzy Sets and Systems* 60 (1993) 1–8.
- [2] J.J. Buckley, Y. Hayashi, Hybrid neural nets can be fuzzy controllers and fuzzy expert systems, *Fuzzy Sets and Systems* 60 (1993) 135–142.
- [3] T.A. Johansen, B.A. Foss, Constructing NARMAX models using ARMAX models, *Internat. J. Control* 58 (1993) 1125–1153.
- [4] J.M. Keller, R.R. Yager, H. Tahani, Neural network implementation of fuzzy logic, *Fuzzy Sets and Systems* 45 (1992) 1–12.
- [5] C.C. Lee, Fuzzy logic in control systems: Fuzzy logic controller, part I, *IEEE Trans. Systems Man Cybernet.* 20 (1990) 404–418.
- [6] C.C. Lee, Fuzzy logic in control systems: Fuzzy logic controller, part II, *IEEE Trans. Systems Man Cybernet.* 20 (1990) 419–435.
- [7] K.S. Narendra, K.P. Parthasarathy, Identification and control of dynamic systems using neural networks, *IEEE Trans. Neural Networks* 1 (1990) 4–27.
- [8] W. Pedrycz, Fuzzy neural networks and neurocomputations, *Fuzzy Sets and Systems* 56 (1993) 1–28.
- [9] R.M. Sanner, J.-J.E. Slotine, Gaussian networks for direct adaptive control, *IEEE Trans. Neural Networks* 3 (1992) 837–863.
- [10] R.M. Sanner, J.-J.E. Slotine, Stable recursive identification using radial basis function networks, in: *Proc. American Control Conf.*, 1992, 1829–1833.
- [11] C.-Y. Su, Y. Stepanenko, Adaptive control of a class of nonlinear systems with fuzzy logic, *IEEE Trans. Fuzzy Systems* 2 (1994) 285–294.
- [12] M. Sugeno, K. Tanaka, Successive identification of a fuzzy model and its application to prediction of a complex system, *IEEE Trans. Systems Man Cybernet.* 42 (1991) 315–334.
- [13] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, *IEEE Trans. Systems Man Cybernet.* 15 (1985) 116–132.

- [14] L.X. Wang, J.M. Mendel, Fuzzy basis functions, universal approximation, and orthogonal least-squares learning, *IEEE Trans. Neural Networks* 3 (1992) 807–814.
- [15] L.X. Wang, J.M. Mendel, Generating fuzzy rules by learning from examples, *IEEE Trans. Systems Man Cybernet.* 22 (1992) 1414–1427.
- [16] C. Xu, Y. Lu, Fuzzy model identification and self learning for dynamic systems, *IEEE Trans. Systems Man Cybernet.* 17 (1987) 683–689.
- [17] S. Yeong, M.J. Chung, Identification of fuzzy relational model and its application to control, *Fuzzy Sets and Systems* 59 (1993) 25–33.
- [18] M. Zak, Terminal attractors in neural networks, *Neural Networks* 2 (1989) 259–274.