Jurnal Teknologi, 34(A) Jun 2001: 45–60

EFFECTS OF USER SELECTED CONDITIONS ON MODELING OF DYNAMIC SYSTEMS USING ADAPTIVE FUZZY MODEL

MOHD SHAFIEK YAACOB[1] & HISHAMUDDIN JAMALUDDIN[2]

Abstract. In this paper, major properties of an adaptive fuzzy model as a system identifier when trained by the back-propagation algorithm are discussed. The standard rule-based fuzzy models were used to identify discrete-time nonlinear dynamic systems. The method of selection of the input variables, the number of rules, and the learning rate are briefly discussed. Three methods for choosing the initial parameter of the fuzzy model are considered, namely the on-line, the off-line, and the random initial parameters. The implementation and the computational aspects of the training algorithm are also highlighted. Three examples of discrete-time nonlinear systems are used in the simulation study to show the effects of user selected conditions on the identification process. The results of the identification procedure show that they approximate the dynamic plants quite well. The correlation based model validity tests are used to validate the identified fuzzy model.

Keywords: System identification, modeling, fuzzy system, back-propagation algorithm, dynamic systems.

Abstrak. Kertas kerja ini membincangkan sifat-sifat utama satu pengenalpasti sistem iaitu model kabur suai yang dilatih dengan algoritma perambatan balik. Model kabur piawai berasaskan aturan kabur telah digunakan untuk mengenal pasti sistem dinamik diskret tak lelurus. Kaedah pemilihan pemboleh ubah masukan, bilangan aturan dan kadar latihan ada dibincangkan dengan ringkas. Tiga kaedah pemilihan parameter awal telah dipertimbangkan, iaitu kaedah pemilihan dalam talian, luar talian dan rawak. Aspek pelaksanaan dan pengiraan algoritma ini turut diketengahkan. Tiga contoh sistem dinamik tak lelurus telah digunakan untuk menunjukkan kesan-kesan keadaan latihan yang dipilih oleh pengguna dalam proses pengenalpastian ini. Keputusan daripada proses pengenalpastian model ini menunjukkan ia boleh menganggarkan sistem dinamik dengan baik. Ujian pengesahan model secara sekaitan telah digunakan untuk mengesahkan kecukupan model berkenaan.

Kata kunci: Pengenalpasti sistem, model kabur, algoritma perambatan balik, sistem dinamik.

1.0 INTRODUCTION

In many scientific problems an essential step toward their solutions is to establish modeling and identification of some objects or systems under investigation in order to understand and predict the behavior of the systems. *System identification* is defined as the process of deriving a mathematical model from observed data, sometimes called input-output data, in accordance with some predetermined criterion [1]. The resultant of the identification process is called a *model*. There are many areas of appli-

[1 & 2] Faculty of Mechanical Engineering, University Teknologi Malaysia, 81310 Skudai, Johor Bahru, Malaysia. e-mail: shafiek@fkm.utm.my and hishamj@fkm.utm.my

cation, such as control engineering, electrical engineering, economics, biomedicine, etc. where adequate mathematical models of the real systems are desirable [1]. Three main purposes of the identification process include control, signal processing, and prediction [2].

Here, the application and implementation of fuzzy models as system identifier are discussed. Basically, fuzzy models approximate mathematical functions and they are usually called *model-free estimators*. They can be considered as universal approximators that can approximate any real nonlinear function to any arbitrary degree of accuracy if they use enough fuzzy rules [3]. Historically, Lotfi Zadeh almost single handedly brought about the second wave of multivalued research under the banner and language of fuzzy logic [4]. In 1965, Zadeh published the landmark paper 'Fuzzy Sets', which started the birth of fuzzy technology and later became the backbone of fuzzy set theory. In 1974, Ebrahim H. Mamdani developed the first fuzzy logic controller to control a steam engine [5]. Mamdani's work marks the start of fuzzy engineering after which a plethora of related papers were published [6].

The basic problem to be addressed here is the use of fuzzy models for identification, namely how to construct a fuzzy model from numerical data. That is, given some function g: $U \subset R^n \to R$, where U is compact, a fuzzy model f: $U \subset R^n \to R$ that approximates the function g is to be constructed. Here, only multi-input-single-output fuzzy models were considered. A multi-output system can always be separated into a group of single-output systems [3]. Generally, three main types of fuzzy structures have been presented in the literature [7], namely the Rule-based systems, the *Fuzzy relational systems*, and the *Fuzzy functional systems*, sometimes referred to as Takagi-Sugeno fuzzy system. Here, only the rule-based adaptive fuzzy model (AFM) as proposed by Wang and Mendel [3], was proposed. Back-propagation (BP) algorithm was used to train the fuzzy model.

2.0 FUZZY SYSTEMS AND BACK-PROPAGATION ALGORITHM

In this section, the description of the standard rule-based fuzzy system and the BP training algorithm are discussed.

2.1 Fuzzy Systems

In general, the input and output of a dynamic plant to be identified are real-valued variables. The most straightforward way of utilizing a fuzzy system is to add a fuzzifier to the input and a defuzzifier to the output of the pure fuzzy logic system shown in Figure 1, as suggested by Mamdani [8]. Here, only a multi-input-single-output fuzzy system, f: $U \subset R^n \to V \subset R$, where U is compact, is considered. The *fuzzy rule base* consists of a collection of fuzzy *IF-THEN* rules to determine a mapping from fuzzy sets in the input universe of discourse $U \subset R^n$ to fuzzy sets in the output universe of discourse $V \subset R$ based on fuzzy logic principles.

The *fuzzifier* maps crisp points in U to fuzzy sets in U. A fuzzy set A in U is characterized by a membership function μ_A: $U \rightarrow [0, 1]$, with $\mu_A(u)$ representing the grade of membership of $u \in U$ in the fuzzy set A. The *fuzzy rule base* is a set of linguistic rules in the form of "*IF* a set of conditions are satisfied, *THEN* a set of consequences are inferred". For a given fuzzy system with n input variables $x_1, x_2, \ldots, x_n$ and one output variable $\hat{y}$, these rules can be formally written a

$$R^l\text{: } IF\ x_1 \text{ is } A_1^{\,l} \text{ and } \ldots \text{ and } x_n \text{ is } A_n^{\,l},\ THEN\ \hat{y} \text{ is } G^l \tag{1}$$

where l = 1, 2, …, M is the rule number, A_i^l and G^l are fuzzy sets in $U_i \subset R$ and $V \subset R$ respectively, $x = (x_1, \ldots, x_n)^T \in U_1 \times \ldots \times U_n$ and $y \in V$ are input and output linguistic variables [9]. The *fuzzy inference engine* is a decision-making logic that employs the fuzzy rule base to map its fuzzified inputs to fuzzy output set using a procedure known as the *compositional rule of inference*. The most commonly used fuzzy logic principle is the so-called sup-star composition, which is some form of fuzzy relation [9]. Normally, a crisp output is required from the fuzzy rule base and is computed by a process known as *defuzzification*. The *defuzzifier* maps fuzzy sets in V to a crisp point $\hat{y} \in V$.

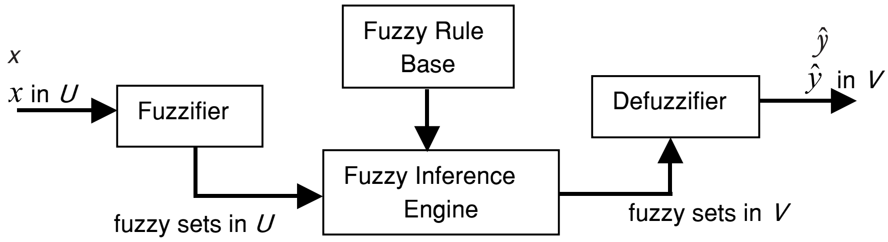

Figure 1 Basic configuration of fuzzy system

When *sup-product* compositional rule of inference, singleton fuzzifier, and center average defuzzifier are used together with Gaussian membership function in the form

$$\mu_{A_i^l}(x_i) = a_i^l \exp\left[-\left(\frac{x_i - \bar{x}_i^l}{\sigma_i^l}\right)\right]^2 \tag{2}$$

where a_i^l, $\bar{x}_i^l$, and σ_i^l are real-valued adjustable parameters with $0 < a_i^l < 1$, the fuzzy logic system can then be derived to be

$$\hat{y} = f(x) = \frac{\sum_{l=1}^{M} \bar{y}^l \left(\prod_{i=1}^{n} a_i^l \exp\left[-\left(\frac{x_i - \bar{x}_i^l}{\sigma_i^l}\right)^2\right]\right)}{\sum_{l=1}^{M} \left(\prod_{i=1}^{n} a_i^l \exp\left[-\left(\frac{x_i - \bar{x}_i^l}{\sigma_i^l}\right)^2\right]\right)} \tag{3}$$

Equation (3) is in fact represents the adaptive fuzzy model (AFM), as proposed by [9]. The parameter $\bar{x}_i^l$ and σ_i^l represent the center and the spread of the input membership functions respectively and $\bar{y}^l$ represents the center of the output membership functions. Meanwhile, a_i^l represents the height of the membership functions.

2.2 Back-Propagation Algorithm

The BP training algorithm is an iterative gradient descent algorithm designed to minimize the mean square error between the fuzzy model output $f(x)$ and the desired output y. This algorithm was initially used to train the multi-layer feed-forward neural networks. By observing the functional form of Equation (3), Wang showed that the fuzzy logic system can be represented by a three-layer feed-forward network as shown in Figure 2 [9]. Therefore the BP algorithm can be used to train them. For a given input-output pair (x^p, y^p), $x^p \in U \subset R^n$, $y^p \subset V \subset R$, the fuzzy logic system $f(x)$ in the form of Equation (3) is designed such that the error

$$e^p = \frac{1}{2}\left[f\left(x^p\right) - y^p\right]^2 \tag{4}$$

is minimized. By assuming that M is given and $a_i^l = 1$, the problem becomes training the parameters $\bar{y}^l$, $\bar{x}_i^l$, and σ_i^l such that e^p of Equation (4) is minimized.

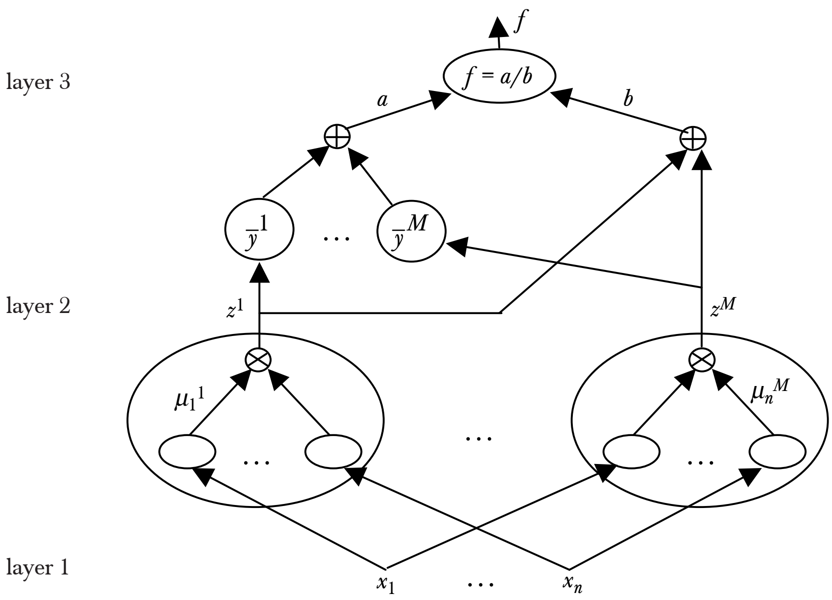

Figure 2 Network representation of the fuzzy systems

To update $\bar{y}^l$, the gradient descent method gives

$$\bar{y}^l(k+1) = \bar{y}^l(k) - \alpha \frac{\partial e^p}{\partial \bar{y}^l}\Big|_k , \tag{5}$$

where l = 1, 2, ..., M is the rule number, k = 1, 2, ... is the iteration step, and α is a constant step size known as the learning rate. The chain rule from calculus gives

$$\frac{\partial e^p}{\partial \bar{y}^l} = \left(f(\underline{x}^p) - y^p\right)\frac{\partial f(\underline{x}^p)}{\partial \bar{y}^l} = \left(f(\underline{x}^p) - y^p\right)\frac{z^l}{b} \tag{6}$$

where

$$z^l = \prod_{i=1}^{n} \exp\left[-\left(\frac{x_i - \bar{x}_i^l}{\sigma_i^l}\right)^2\right] \tag{7}$$

$$z^l = \sum_{l=1}^{M} z^l \tag{8}$$

Substituting Equation (6) into Equation (5), the training algorithm for $\bar{y}^l$ becomes

$$\bar{y}^l(k+1) = \bar{y}^l(k) - \alpha\left(f(\underline{x}^p) - y^p\right)\frac{z^l}{b}. \tag{9}$$

To update $\bar{x}_i^l$, the gradient descent method gives

$$\bar{x}_i^l(k+1) = \bar{x}_i^l(k) - \alpha \frac{\partial e^p}{\partial \bar{x}_i^l}\Big|_k . \tag{10}$$

Similarly, the chain rule gives

$$\frac{\partial e^p}{\partial \bar{x}_i^l} = \left(f(\underline{x}^p) - y^p\right)\frac{\partial f(\underline{x}^p)}{\partial z^l}\frac{\partial z^l}{\partial \bar{x}_i^l} = \left(f(\underline{x}^p) - y^p\right)\frac{\bar{y}^l - f(\underline{x}^p)}{b} z^l \frac{2\left(x_i^p - \bar{x}_i^l\right)}{\sigma_i^{l^2}} \tag{11}$$

Substituting Equation (11) into Equation (10), the training algorithm for $\bar{x}_i^l$ becomes

$$\bar{x}_i^l(k+1) = \bar{x}_i^l(k) - \alpha\left(f(\underline{x}^p) - y^p\right)\frac{\bar{y}^l - f(\underline{x}^p)}{b} z^l \frac{2\left(x_i^p - \bar{x}_i^l(k)\right)}{\sigma_i^{l^2}(k)} \tag{12}$$

To update σ_i^l, the spreads of the membership function, the above procedures give

$$\sigma_i^l(k+1) = \sigma_i^l(k) - \alpha \frac{\partial e^p}{\partial \sigma_i^l}\Big|_k = \sigma_i^l(k) - \alpha\left(f(\underline{x}^p) - y^p\right)\frac{\partial f(\underline{x}^p)}{\partial z^l}\frac{\partial e^p}{\partial \sigma_i^l}\Big|_k \tag{13}$$

After evaluating the partial derivatives, the training algorithm for σ_i^l becomes

$$\sigma_i^l(k+1) = \sigma_i^l(k) - \alpha\left(f(\underline{x}^p) - y^p\right)\frac{\overline{y}^l - f(\underline{x}^p)}{b} z^l \frac{2\left(x_i^p - \overline{x}_i^l(k)\right)^2}{\sigma_i^{l^3}(k)} \tag{14}$$

There are some evidences that the convergence properties of the gradient method can sometimes be improved via the addition of a "momentum term" to each of the update laws in Equation (5), (10), and (13) [10]. For instance, Equation (5) could be modified into

$$\overline{y}^l(k+1) = \overline{y}^l(k) - \alpha \frac{\partial e^p}{\partial \overline{y}^l}\Big|_k + \beta\left(\overline{y}^l(k) - \overline{y}^l(k+1)\right) \tag{15}$$

where β is the gain of the momentum term. Similar changes can be made to Equation (10) and (13). In general, the momentum term will help to keep the updated parameters moving in the right direction. Some of the fundamental properties of AFM with BP algorithm can be found in [11].

3.0 AFM APPLIED TO SYSTEM IDENTIFICATION

A discrete nonlinear dynamic plant to be identified in general is governed by the difference equation

$$y(t+1) = g(y(t), ..., y(t-q+1); u(t), ..., u(t-r+1)) \tag{16}$$

where g is the unknown function to be identified, u and y are the input and output of the dynamic plant respectively, q and r are positive integers representing the number of output and input lags, describing the dynamic of the model, and t represents the time instant [9]. In this case, the number of input variables of the fuzzy system is $n = q + r$. In this paper, the *series-paral lel model* [9] or one-step-ahead prediction [12] was adopted. The identification model is then in the form of

$$\hat{y}(t+1) = f(y(t), ..., y(t-q+1); u(t), ..., u(t-r+1)) \tag{17}$$

where y is the output of the plant. If $x = \{y(t), ..., y(t-q+1); u(t), ..., u(t-r+1)\}$ is the input of the fuzzy system and f is the AFM in the form of Equation (3), then determining the parameter of AFM is basically a system identification problem. The representation of the one-step-ahead prediction model, where the output of the plant is fed to the fuzzy system, is shown in Figure 3.

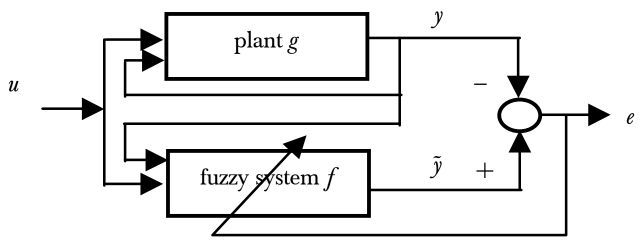

Figure 3 One-step-ahead prediction identification model using fuzzy system

The fuzzy system can also be used to predict a time series $y(t)$ where $t = 1, 2, \ldots$. Here, the mapping from $[y(t-q+1), y(t-q+2), \ldots, y(t)] \in R^q$ to $[y(t+s)] \in R$ is to be determined, where q and s are positive integers. For one-step-ahead prediction, s is set to be equal to 1 and the identification model becomes

$$\hat{y}(t+1) = f(y(t), y(t-1), \ldots, y(t-q+1)) \tag{18}$$

Again, if $x = \{y(t), \ldots, y(t-q+1)\}$ is the input of the fuzzy system and f is the AFM in the form of Equation (3), then determining the parameter of AFM is basically a system identification problem for time series prediction.

Three methods of choosing the initial value of the AFM parameters are considered in this paper, namely the on-line, the off-line, and the random initial parameter choosing method. Since the parameters of the AFM have clear physical meaning, namely the centers of the input and output fuzzy sets, $\bar{x}_i^l$ and $\bar{y}^l$ respectively, the initial parameters for f can be approximated based on initially available information. For on-line initial parameter choosing method as suggested by Wang [9], we use the first M data pairs as the basis of the initial parameters as follows:

$$\bar{y}^l(0) = y(l) \qquad \text{where } l = 1, 2, \ldots, M \tag{19a}$$

$$\bar{x}_i^l(0) = x_i(l) \qquad \text{where } l = 1, 2, \ldots, M \text{ and } i = 1, 2, \ldots, n \tag{19b}$$

Note that since the adaptation is done off-line, the information about $x_i(M)$ and $y(M)$ are available at the beginning of the training. Alternatively, we can start training from time step $M + 1$.

We also consider the off-line initial parameter choosing method where the initial parameters are sampled uniformly throughout the estimation data set as follows:

$$\bar{y}^l(0) = y(l\Delta) \qquad \text{where } l = 1, 2, \ldots, M \tag{20a}$$

$$\bar{x}_i^l(0) = x_i(l\Delta) \qquad \text{where } l = 1, 2, \ldots, M \text{ and } i = 1, 2, \ldots, n \tag{20b}$$

Here, the integer $\Delta = int(N/M)$, where *int* is the integer operator and N is the number of data pairs in the estimation set. Generally, we can expect that the off-line initial parameters will give better estimate since it gives better representation of the entire estimation data set.

Finally, for random initial parameter choosing method, the initial parameters are chosen randomly in the interval of the universe of discourse of the input-output data as follows:

$$\bar{y}^l(0) = rand[\max(y(t) : t = 1, 2, ..., N), \min(y(t): t = 1, 2, ..., N)] \tag{21a}$$

$$\bar{x}_i^l(0) = rand[\max(m_i(t) : t = 1, 2, ..., N), \min(m_i(t): t = 1, 2, ..., N)] \tag{21b}$$

Here, *rand* is the random number operator, randomly choosing a real number between the intervals in the square brackets. Meanwhile, small initial values of spread σ_i^l are chosen for all three methods of choosing the initial parameters. Here, we choose a value of 20% of the range of the universe of discourse to be the initial values of the spreads.

The predictive accuracy of the identification model was computed by defining the normalized root mean square of the residuals as an error index, Q, and is given by

$$Q = \left[\frac{\sum(\hat{y}(k) - y(k))^2}{\sum y^2(k)}\right]^{\frac{1}{2}} \tag{22}$$

Model validation should form the final stage of any identification procedure. The model structure and the estimated parameters are considered adequate if they produce unbiased predictions over different data sets. If the model is valid then the prediction error sequence $\varepsilon(t)$ should be unpredictable from all linear and non-linear combination of past inputs and outputs [11]. This condition will hold if

$$\begin{aligned}
\phi_{\varepsilon\varepsilon}(\tau) &= E[\varepsilon(t-\tau)\varepsilon(t)] = \delta(\tau) \\
\phi_{u\varepsilon}(\tau) &= E[u(t-\tau)\varepsilon(t)] = 0, \forall\tau \\
\phi_{u^{2'}\varepsilon}(\tau) &= E\left[(u^2(t-\tau) - \overline{u^2}(t))\varepsilon(t)\right] = 0, \forall\tau \\
\phi_{u^{2'}\varepsilon^2}(\tau) &= E\left[(u^2(t-\tau) - \overline{u^2}(t))\varepsilon^2(t)\right] = 0, \forall\tau \\
\phi_{\varepsilon(\varepsilon u)}(\tau) &= E[\varepsilon(t)\varepsilon(t-1-\tau)u(t-1-\tau)] = 0, \tau \geq 0
\end{aligned} \tag{23}$$

where ϕ is the standard correlation functions and τ is the lag number [12]. The model is regarded as adequate if these functions fall within 95% confidence bands define as $\pm 1.96/\sqrt{N}$, where N is the data length.

5.0 SIMULATION

In order to illustrate the effects of user selected conditions on system identification using AFM trained by BP algorithm, two dynamic plants and one time series data sets were used. The first plant, P1, to be identified is governed by the difference equation

$$g(t+1) = (0.3\,g(t) + 0.6\,g(t-1) + 0.6\sin(\pi u(t)) + 0.4\sin(3\pi u(t)))/5.5, \tag{24}$$

and the plant output data are corrupted by random measurement noise with values between the interval of –0.1 and +0.1 as follows:

$$y(t+1) = g(t+1) + rand[-0.1, +0.1] \tag{25}$$

The input $u(t)$ is chosen to be

$$u(t) = \sin(2\pi t/250) \tag{26}$$

One thousand input-output data pairs were generated. The original plant output g, the measured output data y, and the input u of plant P1 are as shown in figure 4.

The second plant, P2, is the benchmark data originating from the work of Box and Jenkins concerning the identification of a gas oven [14]. It consists of 296 pairs of input-output measurements. The input $u_o(k)$ of the original plant is the gas flow rate into the furnace, and the output $y_o(k)$ is the percentage concentration of CO_2 gas in the outlet. The sampling interval is 9 s. The original data pairs were normalized so that they lie within the interval of –1.0 and +1.0 as follows:

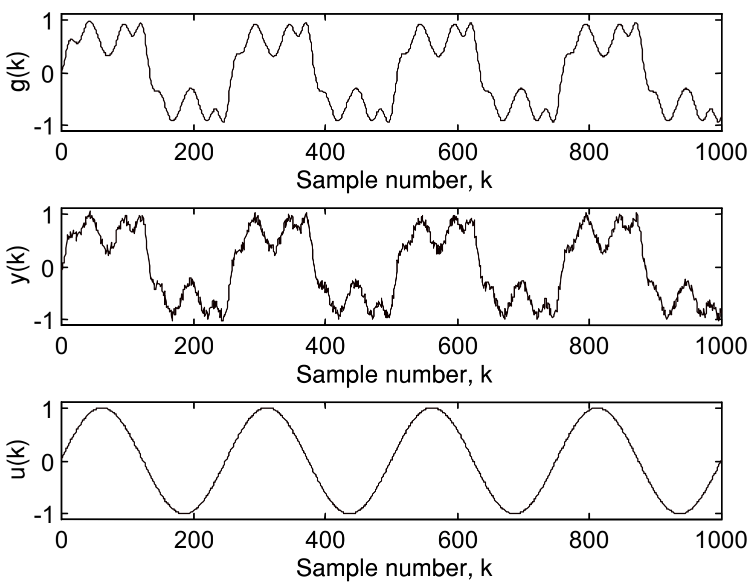

Figure 4 The original plant g, measured output data y, and the input u of P1

$$u(k) = u_o(k)/2.9 \tag{27}$$

$$y(k) = y_o(k) - 53.0)/7.5 \tag{28}$$

The normalized input and output data of P2 are as shown in figure 5. The third plant, P3, is the time series data of the Wolfer annual sunspot [14]. The data consists of 288 values of the sunspot numbers tabulated for a period from the year 1700 to 1987. The data set was divided by 200 so that values of the series lie within the interval of [0, +1]. The normalized time series data set of P3 is as shown in Figure 6.

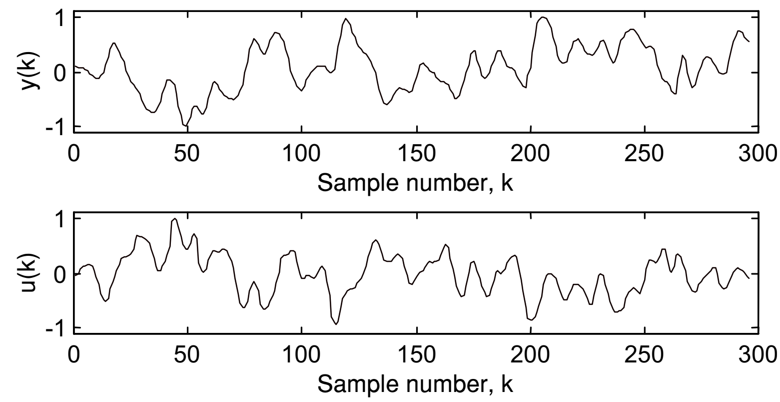

Figure 5 The normalized input and output data of P2

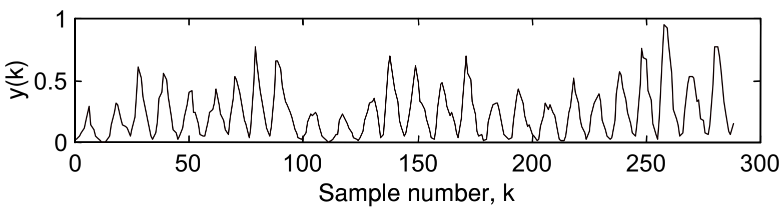

Figure 6 The normalized time series data for P3

For plant P1, the first 500 data pairs were used as the estimation set and the next 500 data pairs were reserved as test set. Since the data pairs for plant P2 and P3 are not enough to be separated into estimation set and test set, all data pairs were used as estimation set.

5.1 Effects of the Number of Rules and Input Variables

Before further training can be done, the number of fuzzy rules, M, and the number of input variables, q and r, must be selected. Figure 7 and 8 show the one-dimensional search for the number of rules M and the output lags q. Here, the on-line initial parameter choosing method and the learning rate $\alpha = 0.2$ were used for the training. The input lags $r = 1$ were chosen for the training of P1 and P2. The error index was evaluated after 10 passes were made through the estimation data set. In general, the predictive accuracy of the AFM increases with the number of rules. However, the number of output lags q must be carefully chosen since unnecessary large value may result in a higher value of error index as shown in Figure 8.

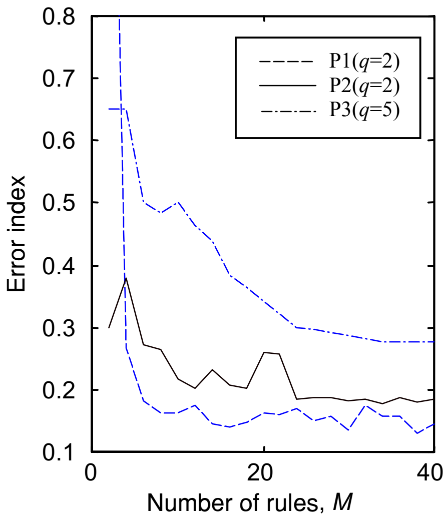

Figure 7 Effects of the number of rules on identification of P1, P2, and P3

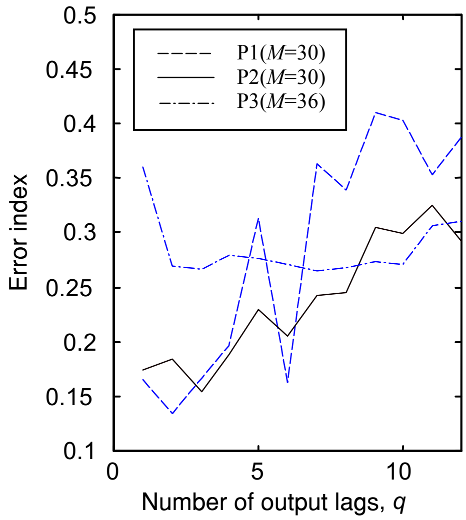

Figure 8 Effects of the number of output lags on identification of P1, P2, and P3

5.2 Effects of Learning Rate and Momentum Gain

To show effects of the learning rate α, the plant P2 was trained using the on-line initial parameter choosing method with $M = 30$, $q = 3$, and $r = 1$ for different values of learning rate. Figure 9 shows the effect of learning rate α on the convergence properties of AFM. Small value of α usually results in slower convergence rate, while bigger value of α tends to make the AFM settled at a sub-optimal level. figure 10 shows the effect of momentum gain β when plant P3 was trained using the on-line initial parameter choosing method with $M = 36$, $q = 6$, $r = 0$, and $\alpha = 0.1$ for different values of momentum rate. There are some evidences that the convergence properties of gradient method can sometimes be improved via the addition of the momentum term in updating the parameters. For a good value of learning rate, the addition of momentum term is usually not necessary.

5.3 Effects of Initial Parameters

Conditions for the subsequent training of the AFM are as shown in Table 1. Figures 11, 12, and 13 show the effect of different initial values of the AFM parameters for plant P1, P2, and P3 respectively. The results clearly show that the performance of BP algo-

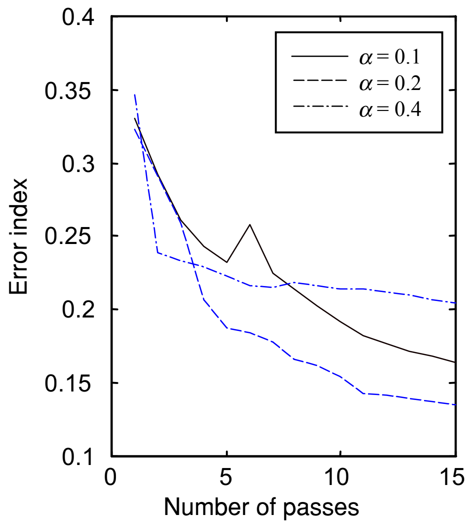

Figure 9 Effects of the learning rates on identification of P2

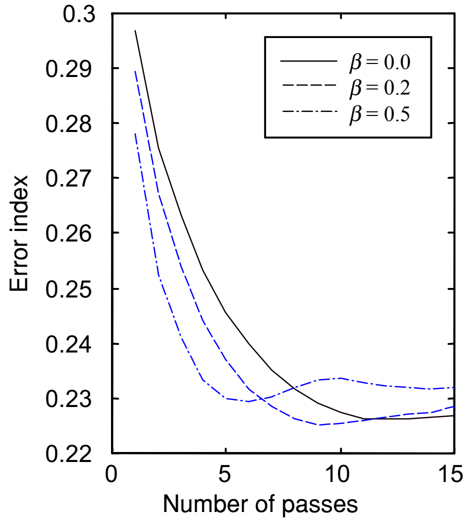

Figure 10 Effects of the momentum gain on identification of P3

Table 1 The subsequent training conditions for P1, P2, and P3.

Plant	M	q	r	α	β
P1	30	2	1	0.2	0
P2	30	3	1	0.2	0
P3	36	6	0	0.1	0

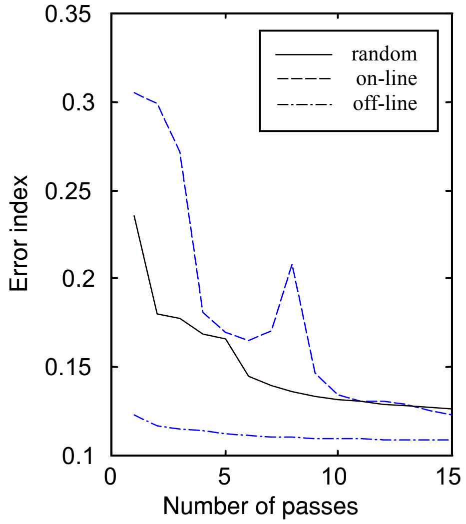

Figure 11 Effects of the initial parameters on identification of P1

rithm depends very much on the initial values of these parameters. Most of the time the AFM will be trapped at the local minimum for a randomly chosen initial parameters. The off-line method of choosing the initial parameters consistently gave good identification. However, if the number of rule M is large enough for the input-output data pairs to cover large range of the universe of discourse in the first M time steps, then the on-line initial parameter choosing method gave similar good results as shown in Figures 12 and 13.

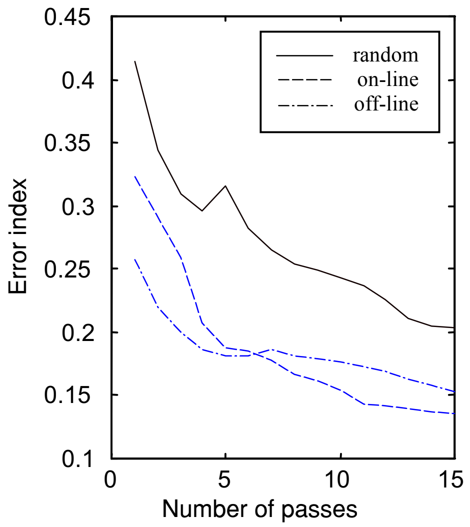

Figure 12 Effects of the initial parameters on identification of P2

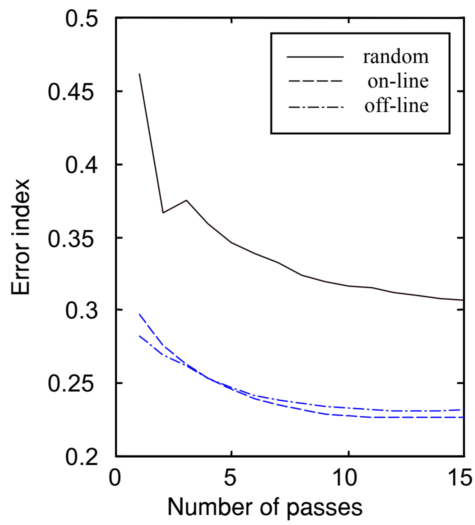

Figure 13 Effects of the initial parameters on identification of P3

Figures 14, 15, and 16 show the one-step-ahead prediction of the identification model (dashed line) superimposed on the plant output (solid line) when the training was

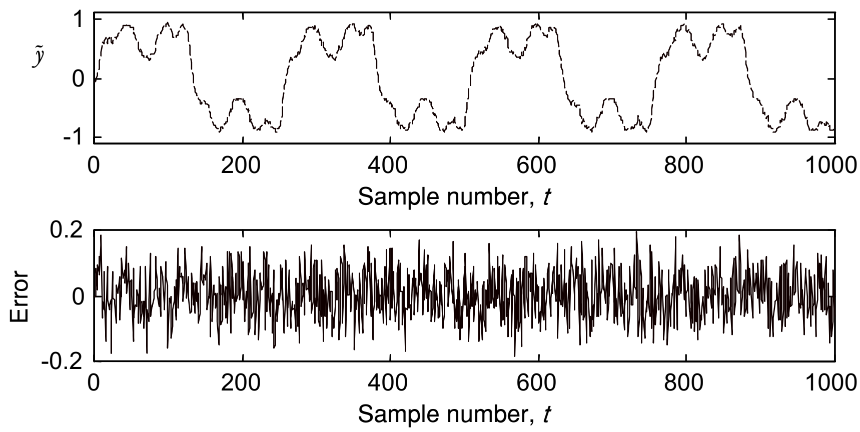

Figure 14 Output of AFM (dashed line) with the off-line initial parameters andthe prediction error for P1

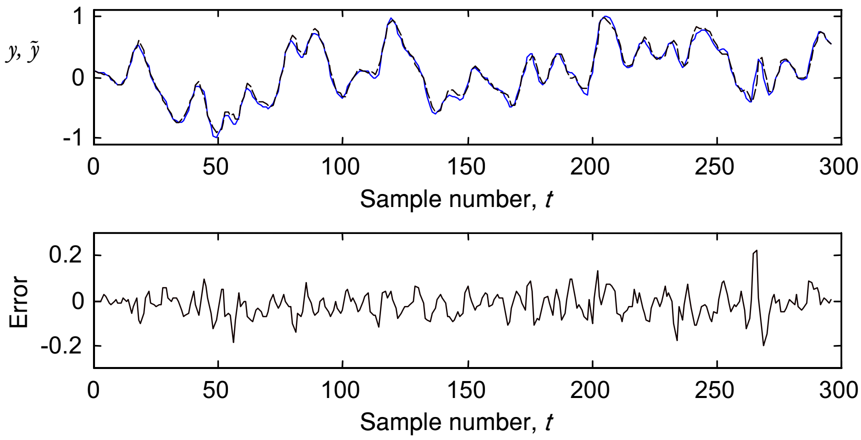

Figure 15 Normalized plant output (solid line), output of AFM (dashed line) with the on-line initial parameters, and the prediction error for P2

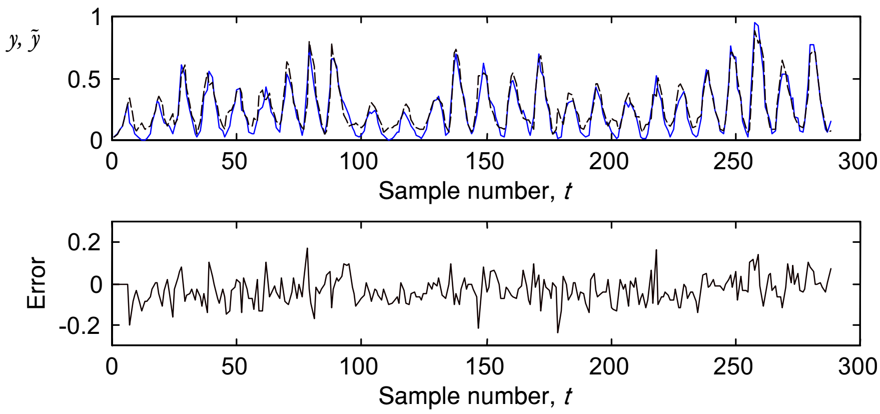

Figure 16 Normalized plant output (solid line), output of AFM (dashed line) with on-line initial parameters, and the prediction error for P3

stopped after 15 passes through the estimation set for plants P1, P2, and P3 respectively. The original plant output is not shown in figure 14 since it would be difficult to distinguish the lines. However, it is clear that the AFM has filtered out the noise and the model resembles the original function g it supposes to identify in the first place. The corresponding estimation errors at each time instant are also provided in those figures in order to give the insight of the accuracy of the identification model.

5.4 Model Validity Test

The final step in system identification is to conduct the model validity tests. The correlation based model validity tests were conducted on the one-step-ahead identification model of P1 in Figure 14. It was found that the model is not adequate since portions of the correlation function $\phi_{\varepsilon\varepsilon}$ lie outside the 95% confidence band as shown in Figure 17. However, when the number of rules were increased from 30 to 45 and the plant P1 was trained under the same condition as before, the correlation tests revealed that the model is adequate as shown in Figure 18.

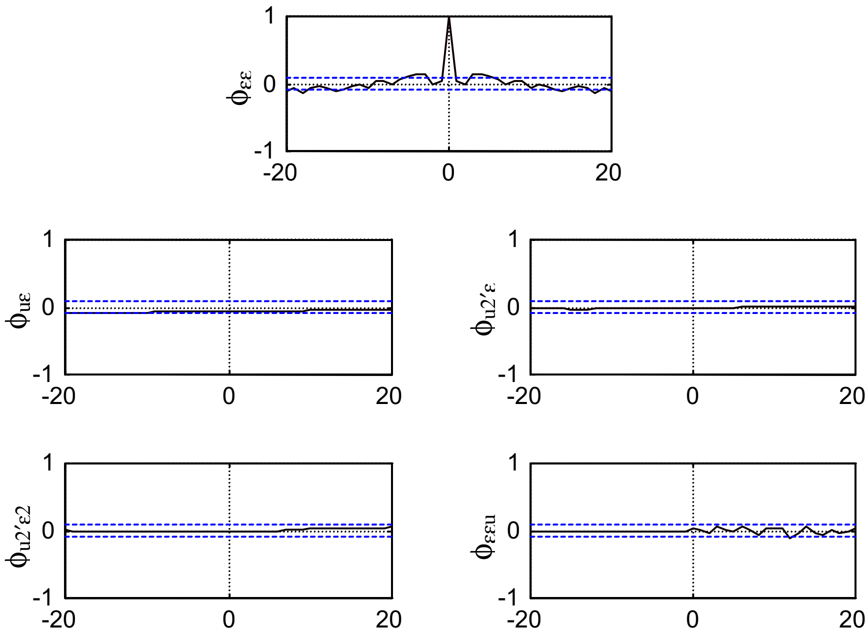

Figure 17 Correlation tests for identification of P1 using AFM with 30 fuzzy rules

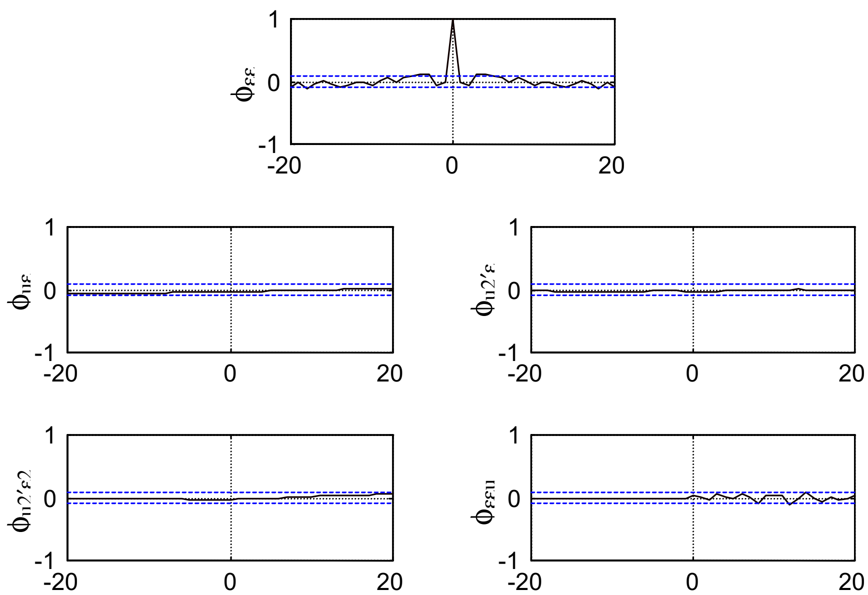

Figure 18 Correlation tests for identification of P1 using AFM with 45 fuzzy rules

6.0 CONCLUSION

In this paper, the standard rule-base fuzzy model was used for the purpose of identifying the nonlinear dynamic systems. The BP algorithm, a gradient descent method, was used to train them. Three methods for choosing the initial parameter of the AFM were considered. The use of additional momentum terms to update the centers of the input and output membership functions was also explored. Some of the fundamental properties of the AFM with BP algorithm were highlighted, illustrating the advantages and shortages of these approaches.

Through simulations, the effects of the number of rules, the input variables, the learning rate, the momentum gain, and the initial values of the parameter on the convergence of AFM have been demonstrated. The results clearly show that the performance of BP algorithm depends very much on the initial values of the parameters. Most of the time, the AFM will be trapped at the local minimum for a randomly chosen initial parameters. The off-line method of choosing the initial parameters consistently gave good identification results.

REFERENCES

[1] Johansson, R. 1993. *System Modeling and Identification*. Englewood Cliffs: Prentice Hall.

[2] Ljung, L. and T. Soderstrom. 1983. *Theory and Practice of Recursive Identification*. Massachusetts: The MIT Press.

[3] Wang, L. X. and J. M. Mendel. 1992. Back-propagation Fuzzy System as Nonlinear Dynamic System Identifiers. *Int. Proc. of the IEEE Conf. on Fuzzy Systems*. 1409 – 1418.

[4] Zadeh, L. A. 1965. Fuzzy Sets. *Information and Control*. 8: 338 – 353.

[5] Yen, J. 1999. Fuzzy Logic – A Modern Perspective. *IEEE Trans. Knowledge and Data Engineering*. 11(1): 153 – 165.

[6] Kosko, B. 1997. *Fuzzy Engineering*. New Jersey: Prentice Hall.

[7] Branco, P. J. C. and J. A. Dente. 2000. A Fuzzy Relational Identification Algorithm and its Application to Predict the Behavior of a Motor Drive System. *Fuzzy sets and Systems*. 109: 343 – 354.

[8] Mamdani, E. H. 1974. Application of Fuzzy Algorithms for Simple Dynamic Plant. *Proc. IEE*. 121(12): 1585 – 1588.

[9] Wang, L. X. 1994. *Adaptive Fuzzy Systems and Control*. Englewood Cliffs: Prentice Hall.

[10] Passino, K. M. and S. Yurkovich. 1998. *Fuzzy Control*. Menlo Park: Addison Wesley Longman.

[11] Yaacob, M. S. and H. Jamaluddin. 2001. Properties of Adaptive Fuzzy Model for System Identification of Dynamic Systems using Back-propagation Algorithm. *First Int. Conf. on Mechatronics (ICOM'01). Kuala Lumpur*, 12 – 13 Feb. 2001.

[12] Billings, S. A., H. B. Jamaluddin and S. Chen. 1991. A Comparison of the Back-propagation and Recursive Prediction Error Algorithms for Training Neural Network. *Mechanical Systems and Signal Processing*. 5(3): 233 – 255.

[13] Billings, S. A. and W. S. F. Voon. 1986. Correlation Based Model Validity Tests for Non-linear Models. *Int. Journal of Control*, 44: 235 – 244.

[14] Box, G. E. P. and G. M. Jenkins. 1970. *Time Series Analysis, Forecasting, and Control*. San Francisco: Holden Day.