Turk J Elec Engin, VOL.8, NO.2 2000, © TÜBİTAK

A Fuzzy Identification Method for Nonlinear Systems

İbrahim EKSİN, Osman Kaan EROL
Faculty of Electrical and Electronic Engineering,
İstanbul Technical University,
Maslak, 80626, İstanbul-TURKEY
e-mail: eksin@elk.itu.edu.tr, osmane@beko.com.tr

Abstract

In this paper, two mathematical ways of building a fuzzy model of both linear and nonlinear systems are presented and compared. In order to determine a model for a nonlinear system, the phase plane is divided into sub-regions and a linear model is assigned for each of these regions. This linear model is represented either in state-space or ARX model form. To determine the pre-selected parameters of the linear system model under study, least-square identification method is used. Then these linear models are arranged in a fuzzy manner to characterize the overall system behavior. The results show that this method can identify linear systems exactly and nonlinear ones quite satisfactorily with both system representations, assuming that the input-output data is not corrupted by noise.

1. Introduction

Fuzzy models become useful when a system cannot be defined in precise mathematical terms. The non-fuzzy or traditional representations require a well structured model and well defined model parameters. Even if the structure is known, numerical model representations usually become irrelevant and computationally inefficient as the complexity increases. Moreover, there may be a lot of uncertainties, unpredictable dynamics and other unknown phenomena that cannot be mathematically modeled at all. Therefore, when a system cannot be modeled with traditional methods for the reasons stated above, then fuzzy logic offers an efficient mathematical tool in handling many practical problems. The main contribution of fuzzy control theory is its ability to handle many practical problems that cannot be adequately handled by conventional control methods.

Fuzzy modeling of the systems have been observed in many scientific researches. Takagi and Sugeno have proposed a search algorithm for a fuzzy controller and generalized their research to fuzzy identification [1]. Sun has observed a modeling scheme for an adaptive-network-based fuzzy inference system [2]. Chen, Pham and Weiss have shown that state space model of a linear system can be modeled fuzzily and extended their study on nonlinear systems [3]. Eksin and Ayday have proposed an approach for fuzzy identification of nonlinear systems [4], Mouzouris and Mendel have implemented a search algorithm for dynamical non-singleton fuzzy control systems [5].

In the studies listed above, except the one by Takagi and Sugeno, the models require some human knowledge and experience. The operator's experience in the system is involved in the mathematics of fuzzy control theory as a collection of "if...then" rules, known as "heuristic rules". The main goal of this paper is to

present a mathematical way of defining a fuzzy model of a nonlinear system without necessitating any human knowledge. Instead of heuristic rules that many fuzzy models or controllers use, the presented method uses the state values or the input-output data pairs. The system either starts from an initial condition and is expected to reach an equilibrium point, or tries to follow a trajectory under some control. The state values or input/output data are collected during these runs of the system for use at the identification stage. It is clear that the more data is collected during various free runs, the more precise the model becomes. In this paper, the initial conditions are chosen in such a way that the data covers all the dynamical input/output range of the system.

2. Basic concepts and formulations

Linear systems can be represented in various structural forms such as state-space or auto-regressive moving average models. In this study, basically these two structural models are used in identifying the systems. The state space model of an n^{th} order linear system with m inputs and l outputs can be written as

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{H}\mathbf{x} \end{aligned} \tag{1}$$

where $\mathbf{x}$ is an $(n \times 1)$ vector which represents the "states" of the system, $\mathbf{y}$ is an $(l \times 1)$ output vector, $\mathbf{u}$ is an $(m \times 1)$ input vector, and $\mathbf{A}$ is an $(n \times n)$, $\mathbf{B}$ is an $(n \times m)$, and $\mathbf{H}$ is an $(l \times n)$ matrix. When there is no control signal, then $\mathbf{u=0}$, and hence this system is called "free".

The phase plot is a diagram which shows the trajectories of the states as $t \to \infty$. A free system model without output equality can be written as

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} \tag{2}$$

The solution point of a stable system given by (2) with the initial state vector $\mathbf{x}(0)$ as t approaches infinity is $\mathbf{x} = 0$. This means that all states of a stable linear system must converge to 0. The discrete-time free model of (2) is

$$\mathbf{x}(k+1) = \mathbf{A_d}\mathbf{x}(k) \tag{3}$$

where k denotes the steps of the iteration.

Any nonlinear system can be considered as piecewise linear and be modeled by using the linear state space equations given in (3). However, the equilibrium or the solution point as t approaches infinity must also be added in (3). When the system is not linear, then the equilibrium point can be different from 0. A linear shift of the states yields

$$\mathbf{x}(k+1) = \mathbf{A}(\mathbf{x}(k) - \mathbf{S}) \tag{4}$$

where $\mathbf{S}$ is a $(n \times 1)$ vector which holds the equilibrium coordinates of the nonlinear system. Leaving the constant values apart, (4) becomes

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{p} \tag{5}$$

where

$$\mathbf{p} = -\mathbf{AS} \tag{6}$$

A LTI discrete-time system can also be modeled by using autoregressive average (AR) or autoregressive average exogenous (ARX) difference equations, [6]. This model representation is useful when all the system states are not accessible or only the input/output data is available. Let n denotes the order of the ARX model, the model structure can be written as

$$y(k) = \sum_{i=1}^{n} (a_i y(k-i) + b_i u(k-i)) \tag{7}$$

where $y(k)$ is the output of the model at k^{th} iteration, a_i and b_i are the weighting factors. With the assumption that $u = 0$, a free system model or AR model can be obtained as

$$y(k) = \sum_{i=1}^{n} a_i y(k-i) \tag{8}$$

Let us assume that the equilibrium point of (8) is achieved after l^{th} step. Then for all $k > 1$,

$$y(k) = y(k-1) = \ldots = y(k-n) = y \tag{9}$$

$$\sum_{i=1}^{n} a_i = 1 \tag{10}$$

In (7), the dependence on the past output (and/or past input) values is due to finite time response of the system. When the system output reaches a steady equilibrium point different from zero in the nonlinear system case, then according to (9), all the output values become equal, and so the sum of the weighting factors is equal to unity as in (10). Therefore, (10) poses an extra condition in the calculation of the a_i coefficients. In order to get rid of this linear dependency and to have a nonzero equilibrium point in a nonlinear system, we can add a constant term p to (7) and obtain

$$y(k) = \sum_{i=1}^{n} (a_i y(k-i) + b_i u(k-i) + p) \tag{11}$$

The role of p in (11) is different from the one in (5). The model given in (7) is still capable of modeling nonlinear systems, but the addition of p improves the model response.

To characterize a nonlinear system better, the dynamic range is divided into a quantity of sub-ranges and a linear model is assigned for each range. The passage from one model to another has been smoothened using fuzziness to prevent sudden changes in the model. The selection of the number or type of the segmentation is done a priori. Increasing the number of partitions will give better results, but will in turn increase the number of calculations. Before giving details on how to implement models (5) and (11) to form a single fuzzy model, it is useful to explain the terms that are used in fuzzy language. The method will then be explained in detail in Section 3.

Fuzzy sets and membership functions In non-fuzzy control applications, the states **x**, control input **u** and the model output **y** receive crisp values. In fuzzy modeling, these variables are assumed to have

fuzzy values. Each variable is assigned a value called a "membership value" in the set [0,1], according to a certain number of "membership functions". Throughout this paper, the membership values are denoted by the letter W and membership functions by S.

Fuzzy interval partitioning For an input variable **x**, the input space is divided into N intervals, and a unique membership function is assigned to each of these intervals. The intervals can be chosen to be distinct from one another but they can overlap as well. According to the crisp value that **x** takes, it receives N membership values, but only two of them belonging to two adjacent intervals are nonzero.

If... then rules The dynamics of a system can be summarized by "if... then" statements, called heuristic rules. Any rule is composed of an antecedent and a consequence. The antecedent may receive more than one preposition. In mathematical terms, a "relation" or "if... then" rule can be expressed as
R: IF $x_1 \in S_1$ AND $x_2 \in S_2$ AND $\ldots$ $x_n \in S_n$ THEN $y \in S_3$
where $x_1, x_2, \ldots x_n$ are the crisp input variables, and y is the consequence or output of the relation **R**. For each antecedent, a unique membership value W is calculated as the minimum of the membership values W_i of each preposition, say "x_i is S_i", $i = 0, \ldots, n$.

$$W = min(W_i), i = 0, \ldots, n \tag{12}$$

Sun (1994) has proposed a multiplication operation instead of taking the minimum as

$$W = \prod_i W_i \tag{13}$$

for his network based identification algorithm.

Defuzzification Since the output of the model must be a non-fuzzy value, the output of the above relation must be "defuzzified". If there exist r relations, and $W^i, i = 1 \ldots, r$ being the membership value and y^i the output of each relation, and y_m the output of the model, then the defuzzified output is the weighted sum of the r fuzzy outputs and is expressed as

$$y_m = \frac{\sum_{i=1}^{r} W^i y^i}{\sum_{i=1}^{r} W^i} \tag{14}$$

3. The algorithm of identification

Any state variation from an initial condition towards the equilibrium point or input-output data from a running system is recorded. Then by using these data, the model parameters are calculated using the least squares identification algorithm.

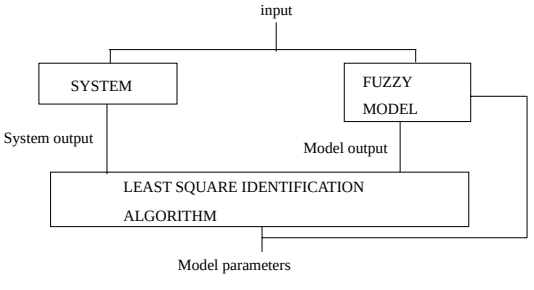

Figure 1. Outline of the identification algorithm

Having m number of different input and output data, the least square identification method searches the optimum fuzzy model parameters in order to minimize the sum of the squared errors between system and model outputs for $k = 1, \ldots, m$. The outline of the identification algorithm is given in Figure 1.

During the fuzzification stage for the fuzzy model, each crisp input data x is assigned a membership value W according to its negative, zero or positive value by using the membership function chosen as

$$W(x_i(k)) = \frac{1}{1 + ((x_i(k) - a)/2)^4} \quad i = 1, \ldots, n \tag{15}$$

In (15), the values of a are chosen with respect to a pre-selected dynamic range and should be scaled in accordance with the system dynamics. With these three membership functions, the input space is partitioned into three overlapping intervals.

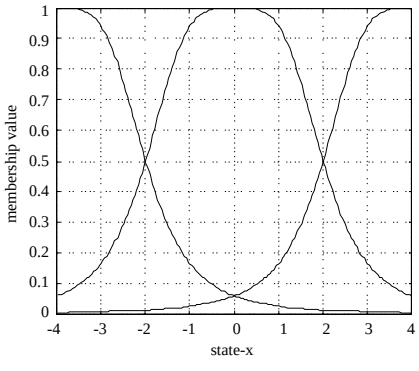

Figure 2. Fuzzy partitioning of the input space

As an example, the zero membership function will assign 1, the highest possible value to the input $x_i(k)$ if $x_i(k) = 0$ and decreasing values as $x_i(k), i = 1, \ldots, n$ is located on each side of 0. The symmetrical nature of the function used in (15) improves noise immunity since the average value of the noise is zero. The fuzzy input space partitioning is given in Figure 2.

If...then rules used for model approximation require no knowledge of the system behavior. The number of rules required to model the system in state-space representation is $r = o(N^n)$ and this number becomes $r = o(N)$ in the case of ARX model representation.

3.1. Fuzzy state-space model structure

Fuzzy state-space modeling of a second order system with $N = 3$ can be summarized as follows:

$\mathbf{R}^1$: IF $x_1(k)$ is negative AND $x_2(k)$ is negative THEN $\mathbf{y}^1(k) = \mathbf{A}^1\hat{\mathbf{x}}(k-1)+\mathbf{p}^1(k-1)$ with the membership value W^1

$\mathbf{R}^2$: IF $x_1(k)$ is negative AND $x_2(k)$ is zero THEN $\mathbf{y}^2(k) = \mathbf{A}^2\hat{\mathbf{x}}(k-1) + \mathbf{p}^2(k-1)$ with the membership value W^2

...

$\mathbf{R}^9$: IF $x_1(k)$ is positive AND $x_2(k)$ is positive THEN $\mathbf{y}^9(k) = \mathbf{A}^9\hat{\mathbf{x}}(k-1)+\mathbf{p}^9(k-1)$ with the membership value W^9

To carry out the identification algorithm in the MATLAB environment, it is required to rewrite the system equations so that they conform with the least squares identification (LSI) method. The LSI structure is given below:

$$\boldsymbol{\vartheta}\boldsymbol{\alpha}_1 = \boldsymbol{\varphi}_1 \tag{16}$$

and

$$\boldsymbol{\vartheta}\boldsymbol{\alpha}_2 = \boldsymbol{\varphi}_2 \tag{17}$$

where

$$\boldsymbol{\vartheta} = \begin{bmatrix} K_1(1)x_1(1) & K_1(1)x_2(1) & K_1(1) & \dots & K_9(1)x_1(1) & K_9(1)x_2(1) & K_9(1) \\ . & . & . & \dots & . & . & . \\ K_1(m-1)x_1(m-1) & . & . & \dots & . & . & K_9(m-1) \end{bmatrix} \tag{18}$$

$$\boldsymbol{\vartheta}_1 = [x_1(2)x_1(3)\dots x_1(m)]^T \tag{19}$$

$$\boldsymbol{\vartheta}_2 = [x_2(2)x_2(3)\dots x_2(m)]^T \tag{20}$$

$$K_i(k) = \frac{W^i(k)}{\sum_{l=1}^{9} W^l} \tag{21}$$

The solution vectors $\boldsymbol{\alpha}_1$ and α_2 of the algorithm will be

$$\boldsymbol{\alpha}_1 = [a_{11}^1 a_{12}^1 p_1^1 \dots a_{11}^9 a_{12}^9 p_1^9]^T \tag{22}$$

$$\boldsymbol{\alpha}_2 = [a_{21}^1 a_{22}^1 p_2^1 \dots a_{21}^9 a_{22}^9 p_2^9]^T \tag{23}$$

$a_{i,j}$ are the elements of the matrix $\mathbf{A}$ in (5).

When the system is running with a controller, the model will not be free; hence, the relation set becomes $i = 1, \dots, 9$

$\mathbf{R}^i$: IF $x_1(k)$ is positive AND $x_2(k)$ is positive THEN

$\mathbf{y}^i(k) = \mathbf{A}^i\mathbf{x}(k-1) + \mathbf{p}^i(k-1) + \mathbf{Bu}(k-1)$, with the membership value W^i.

3.2. Autoregressive model structure

On the other hand, if an AR model (11) is chosen with $N = 3, n = 4$, then the rule set and MATLAB implementation for one input and one output variable case can be written as follows:

$\mathbf{R}^1$: IF $y_m(k-1)$ is negative THEN $y^1(k) = \sum_{j=1}^{4} a_j^1 y_m(k-j) + p_1(k)$ with the membership value W^1

$\mathbf{R}^2$: IF $y_m(k-1)$ is zero THEN $y^2(k) = \sum_{j=1}^{4} a_j^2 y_m(k-j) + p_2(k)$ with the membership value W^2

$\mathbf{R}^3$: IF $y_m(k-1)$ is positive THEN $y^3(k) = \sum_{j=1}^{4} a_j^3 y_m(k-j) + p_3(k)$ with the membership value W^3

The $\boldsymbol{\vartheta}, \boldsymbol{\varphi}$, matrices are

$$\boldsymbol{\vartheta} = \begin{bmatrix} \dots & K_i(4)x_1(4) & K_i(4)x_1(3) & K_i(4)x_1(2) & K_i(4)x_1(1) & K_i(4) & \dots \\ . & . & & & & . & . \\ \dots & K_i(m-1)x_1(m-1) & K_i(m-1)x_1(m-2) & K_i(m-1)x_1(m-3) & K_i(m-1)x_1(m-4) & K_i(m-1) & \dots \end{bmatrix}$$

$$i = 1, 2, 3 \tag{24}$$

$$\boldsymbol{\varphi} = [x_1(5)x_1(6)..x_1(m)]^T \tag{25}$$

$$K_i(k) = \frac{W^i(k)}{\sum_{l=1}^{3} W^l} \tag{26}$$

The solution vector α_1 of the algorithm will be

$$\boldsymbol{\alpha}_1 = [a_1^1\, a_1^2\, a_1^3\, a_1^4\, p_1 \ldots a_1^3\, a_2^3\, a_3^3\, a_4^3\, p_3]^T \tag{27}$$

Similar calculations can be carried out to compute the solution vector $\boldsymbol{\alpha}_2$. The mean square fitness is calculated according to the following equation:

$$fit = \sqrt{\frac{\sum_{k=1}^{m}(x_1(k) - x_{m1}(k))^2 + (x_2(k) - x_{m2}(k))^2}{m}} \tag{28}$$

and the steady-state error for phase-plane plots is given as

$$e_{ss} = \| x(m) - x_m(m) \|_2 \tag{29}$$

It is to be noted that "modeling can be done by separating the premise identification from the consequence identification and estimating the parameters of these linear equations using an orthogonal estimator [7].

4. Simulations

Simulation 1

Let us consider a nonlinear system model chosen a priori as:

$$x_1(k) = 0.9x_1(k-1) - 0.2\sin(x_2(k-1))$$

$$x_2(k) = 0.2\cos(x_1(k-1)) + 0.9x_2(k-1)$$

For each unique initial condition, 200 $(x_i(k), x_i(k-1))$ $i = 1, 2$ data pairs are obtained from this nonlinear system. The entire phase-plane is tried to be covered by starting from 20 randomly selected initial conditions. The resulting phase-plane plot for one of these initial conditions is given in Figure 3.
The matrices found by using LSI algorithm are given below:

$$\mathbf{A}^1 = \begin{bmatrix} 0.8801 & 0.1458 \\ -0.1905 & 1.0023 \end{bmatrix}, \mathbf{p}^1 = \begin{bmatrix} 0.2448 \\ -0.5632 \end{bmatrix}$$

$$\mathbf{A}^2 = \begin{bmatrix} 0.8956 & 0.1474 \\ -0.1032 & 0.9717 \end{bmatrix}, \mathbf{p}^2 = \begin{bmatrix} 0.3035 \\ 0.6386 \end{bmatrix}$$

$$\mathbf{A}^3 = \begin{bmatrix} 0.9017 & 0.1004 \\ 0.0363 & 0.6885 \end{bmatrix}, \mathbf{p}^3 = \begin{bmatrix} 0.1482 \\ -1.2454 \end{bmatrix}$$

$$\mathbf{A}^4 = \begin{bmatrix} 0.8928 & -0.1851 \\ -0.2281 & 1.0189 \end{bmatrix}, \mathbf{p}^4 = \begin{bmatrix} -0.0172 \\ -1.0262 \end{bmatrix}$$

$$\mathbf{A}^5 = \begin{bmatrix} 0.9048 & -0.2213 \\ 0.0450 & 0.9085 \end{bmatrix}, \mathbf{p}^5 = \begin{bmatrix} -0.0076 \\ 0.4392 \end{bmatrix}$$

$$\mathbf{A}^6 = \begin{bmatrix} 0.8779 & -0.1718 \\ 0.3575 & 0.5893 \end{bmatrix}, \mathbf{p}^6 = \begin{bmatrix} 0.1060 \\ -1.6054 \end{bmatrix}$$

$$\mathbf{A}^7 = \begin{bmatrix} 0.9733 & 0.1789 \\ 0.1093 & 1.1513 \end{bmatrix}, \mathbf{p}^7 = \begin{bmatrix} -0.1681 \\ -0.7456 \end{bmatrix}$$

$$\mathbf{A}^8 = \begin{bmatrix} 0.8881 & 0.1292 \\ 0.1100 & 0.8631 \end{bmatrix}, \mathbf{p}^8 = \begin{bmatrix} -0.2743 \\ 0.5869 \end{bmatrix}$$

$$\mathbf{A}^9 = \begin{bmatrix} 0.8664 & 0.1836 \\ 0.1785 & 0.6764 \end{bmatrix}, \mathbf{p}^9 = \begin{bmatrix} -0.3406 \\ -0.0553 \end{bmatrix}$$

The mean square fitness and steady-state error values given in (28) and (29), respectively, are obtained as follows for $(x_1(1), x_2(1)) = (4, 4)$

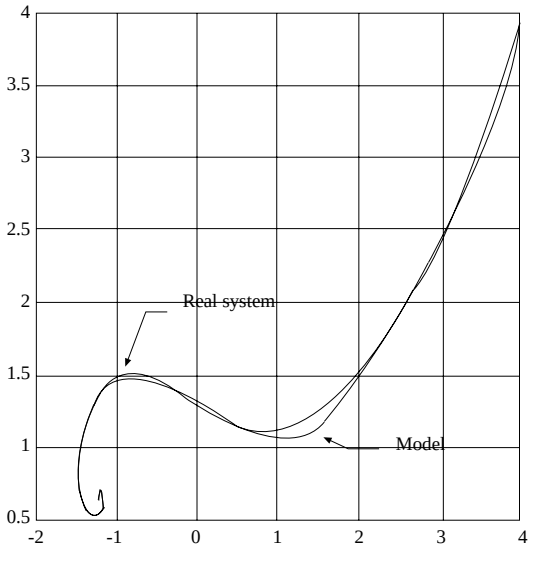

Figure 3. Phase-plane portrait for $x_1(0) = x_2(0) = (4,4)$

$$fit = 0.0204$$

$$e_{ss} = 0.0039$$

Simulation 2

The same system in Simulation 1 is taken into consideration with the same initial conditions. In this case, the AR model is used and the following results are obtained.

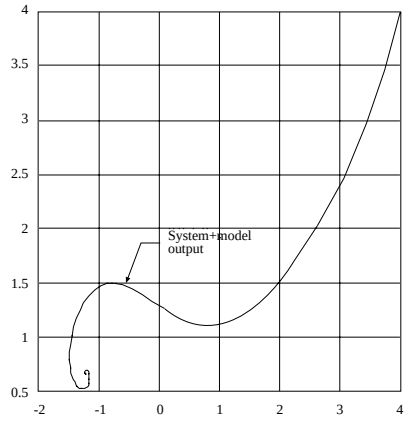

Figure 4. AR model performance for $x_1(1) = x_2(1) = (4,4)$

$$fit = 8.4509 \times 10^{-4}$$

$$e_{ss} = 2.7944 \times 10^{-4}$$

It is obvious that the results are much better when compared to state-space model representation. In fact, the system output and the model output coincide, which means that the model perfectly matches the system.

Simulation 3

In this case, a linear system with the state equations

$$x_1(k+1) = x_1(k) + 0.1x_2(k)$$

$$x_2(k+1) = -0.1x_1(k) + 0.8x_2(k)$$

is considered.

With the same starting points and equal number of iterations, the simulation has produced a perfect match between the model and the system. The phase-plane portrait, the fitness and the steady-state error values are given below:

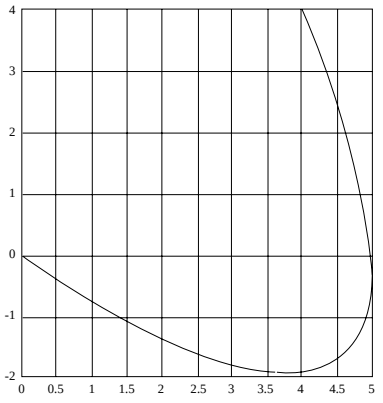

Figure 5. Phase-plane portrait of the a linear system for $x_1(0) = x_2(0) = (4,4)$

$$fit = 5.0839 \times 10^{-14}$$

$$e_{ss} = 4.7876 \times 10^{-15}$$

The model parameters are all found to be identical for 9 sub-regions, as expected. The term p is, quite naturally, calculated to be zero.

$$\mathbf{A} = \begin{bmatrix} 1 & 0.1 \\ -0.1 & 0.8 \end{bmatrix}, \mathbf{p} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Simulation 4

In this simulation, data have been collected from a laboratory scale process (Feedback's Process Trainer PT326). The process operates much like a common hand-held dryer. Air is blown through a tube after being heated. The input to the process is the voltage applied to a mesh of resistor wires that constitute the heating device. The output of the process is the air temperature measured in volts by a thermocouple sensor at the outlet.

One thousand input-output data points were collected from the process as the input was changed in a random fashion between two levels. The sampling interval is 80 ms. These data are available in the MATLAB identification toolbox.

The first 500 data are used for training. Then the outputs of the MATLAB's arx441 format model are compared to those of the presented " model in Table 1. The mean square error is expressed as

$$MSE = \sqrt{\frac{1}{500}\sum_{i=1}^{500}(y_m(i) - y_s(i))^2}$$

Table 1. Fitness values

" model with p	0.2710
" model without p	0.5007
Non-" ARX441 model	0.5168

As can be seen in Table1, the identification by using " model gives much better results.

5. Conclusion

It is well known that the identification of nonlinear systems by traditional means is very cumbersome [8]. For this reason, neural network or other intelligent function approximation approaches (such as radial basis functions) are devised or used [9].

The proposed system identification method offers a new approach and tool for building a mathematical model for nonlinear systems. When there is a lack of human knowledge or there are difficulties in obtaining crisp model parameters, the method described above can be used quite satisfactorily. The basic idea of the method is that phase-plane or input/output space is divided into sub-regions and a linear model is assigned to each region and then these models are composed together using " theory to describe the overall nonlinear system dynamics. The model obtained by using this method shows some discrepancies at the beginning, but it converges to the real system within a few iterations.

To apply this method with state-space representation, all the system states must be accessible. This also means that the order of the system must be known a priori. The identification by deduction of non-observable states through the use of derivative functions require a perfect model order known a priori. If this is not the case, the model will not be stable.

On the other hand, there is no need to know the order of the system nor all the states when using an AR or ARX structure based model. Only input/output data pairs are enough to identify the system. From the simulations, it is clear that the ARX model converges much faster and better than state-space model because it requires a lower number of " relations.

References

[1] T. Takagi, M. Sugeno, "identification of systems and its applications to modeling and control", *IEEE Trans. On Man and Cybernetics*, vol. smc-15, no. 1, pp. 116-132, 1985.

[2] C.T. Sun, "Rule-base structure identification in an adaptive-network based inference system," *IEEE Trans. On Systems*, vol. 2, no. 1, pp. 64-73, 1994.

[3] G. Chen, T.T. Pham, J.J. Weiss, "modeling of control systems," *IEEE Trans. On Aerospace And Electronic Systems*, vol. 31, no. 1, pp. 414-428, 1995.

[4] I. Eksin, C.T. Ayday, "identification of nonlinear systems," *International Conference On Industrial Electronics, Control And Instrumentation*, vol. 2/3, pp. 289-293, 1993.

[5] G.C. Mouzouris, J.M. Mendel, "Dynamic non-singleton logic systems for nonlinear modeling," *IEEE Trans. On Systems*, vol. 5, no. 2, pp. 199-208, 1997.

[6] T. Söderström, P. Stoica, *System Identification*, Prentice Hall International Series In Systems And Control Engineering, pp.147-158, 1989.

[7] L.Wang, R.Langari, "Building Sugeno-type models using discretization and orthogonal parameter estimation techniques," IEEE Trans. On Systems, vol.3, no. 4, pp. 454-458, 1995.

[8] N.K. Sinha, B. Kuszta, *Modeling And Identification Of Dynamic Systems*, Van Nostrand Reinhold Company, 1983.

[9] S. Chen, S.A. Billings, C.F.N. Cowan, P.M. Grant, "Practical identification of NARMAX models using radial basis functions," *Int. J. Control*, vol. 52, no. 6, pp. 1327-1350, 1990.