Development of A Fuzzy System Model for Two-joint Robot Arm Control

M.H. Fazel Zarandi
Assistant Professor
Department of Industrial Engineering,
Amirkabir University of Technology

G. Karimian Khosrowshahi
Ph. D. Candidate
Department of Electrical Engineering,
Amirkabir University of Technology

Abstract

This paper presents a systematic fuzzy system model for control of complex systems. The proposed model conveys four distinctive features: 1) a unified parameterized reasoning formulation in antecedent; 2) input-output relation in a linear form in consequent; 3) application of evolutionary programming for selecting appropriate system parameters; and 4) an improved fuzzy clustering algorithm. Unlike the traditional approach of selecting inference mechanism a priori, the reasoning mechanism can adjust its parameters by means of evolutionary programming. Consequents of fuzzy rules are linear functions of the antecedent variables, describing the system with a set of local linear input-output relations. Thus, the model can take advantages of linear system properties in the consequent. The fuzzy rules are generated through fuzzy c-means (FCM) clustering algorithm. Proper selecting of cluster centers is also investigated. Finally, the system is tested and validated in a two-joint robot arm control.

Keywords
Approximate reasoning, fuzzy clustering, fuzzy modeling, evolutionary programming, robot

1-Introduction

Fuzzy models describe systems by establishing relations between the relevant variables in the form of IF-THEN rules. Traditionally, a fuzzy model is built by using expert knowledge in the form of linguistic rules. Recently, there is an increasing interest in obtaining fuzzy models from measured data. Automatic rule generation techniques have been introduced to generate models from training data. When sufficient training information is available, these techniques can produce highly precise models. A cost incurred in obtaining the precision is that the resulting models lack a linguistic interpretation and, in the case of rule-based models, frequently consist of a large number of rules [16]. Different approaches have been proposed for this purpose, such as fuzzy relational modeling [1], neural-network training techniques [2], and product-space clustering [3], [4].

Takagi-Sugeno-Kang (TSK) approaches of fuzzy modeling are gaining impetus recently in applications to complex systems, because of their computational efficiency, transparency and flexibility [17]. The main reason for this is their dual, quasi-linear nature. Generally, they are non-linear and therefore suitable for complex systems, but they could also be treated as linear in respect to the consequent parameters, which makes possible the application of efficient recursive techniques.

In this paper an improved fuzzy modeling system is developed. The focus of the work is on the process of system identification for fuzzy modeling. The problem of system identification can be divided into two parts: Structure identification and parameter identification [3-5]. In the structure identification stage, input-output relations in terms of IF-THEN rules are specified. In this paper, fuzzy clustering is considered as an approach of rule generation in fuzzy modeling [9,17]. In this direction, fuzzy c-mean clustering is used and the objectivity of this technique is improved by proper choosing of the number and the level of fuzziness of clusters [9,11,18]. A method for assignment of initial cluster center locations is also introduced. The parameter identification consists of derivation of the optimum inference parameters and adjustment of membership functions. We use an evolutionary algorithm for choosing optimum inference parameters. Another aspect of the proposed method is combination of linear consequent and nonlinear antecedent based on Takagi-Sugeno-Kang (TSK) approach of fuzzy modeling [5]. In this approach, the consequents of fuzzy rules are linear functions of the antecedent variables, describing the system with a set of local linear input-output relations. Thus, the model can take advantages of linear system properties in the consequent. The rest of the paper is organized as follows:

Section 2 presents rule generation with fuzzy clustering methods. Fuzzy inference parameter optimization is presented in section 3. Section 4 discusses fuzzy inference output parameter optimization. In section 5, a case study is presented for testing and verifying the proposed approach. Finally, the discussions and conclusions are appeared in section 6.

2-Rule generation using fuzzy clustering methods

A clustering algorithm partitions a given data set into subsets, called clusters, with the aim of minimizing the difference of the objects in the same cluster and maximizing the difference between the objects in different clusters. For fuzzy clustering from the available data sequence, a matrix X and an output vector y are constructed. Fuzzy clustering in Cartesian product-space $X \times Y$ is applied to partition the training data into characteristic regions. Combining X and y form the data set Z to be clustered:

$$Z = [X\ y] \tag{1}$$

Given the training data Z and the number of the clusters c, the c-means clustering algorithm is applied, which computes the fuzzy partition matrix U. The fuzzy sets in the antecedent of the rules are obtained from the partition matrix U, which ikth element $u_{ik} \in [0,1]$ is the membership degree of the data object z_k in cluster i. The ith row of U contains a pointwise definition of a multidimensional fuzzy set. One dimensional (1-D) fuzzy set A_{ij} are obtained from the multidimensional fuzzy sets by projections onto the space of the input variables x_j.

In this paper, we use fuzzy c-means (FCM) algorithm for data clustering. For proper choosing of number of clusters (c) and weighting exponent (m), several methods have been investigated.

Let $X=\{x_1,\dots,x_N\}$ be the set of input data, c number of clusters and m weighting exponent that demonstrates degree of fuzziness. An efficient method of choosing c is the parameter of cluster validity index (S_{cs}). The S_{cs} is defined as following [5]:

$$S_{cs} = \sum_{i=1}^{c}\sum_{k=1}^{N}(u_{ik})^m\left(\|x_k - v_i\|^2 - \|v_i - \bar{v}\|^2\right) \tag{2}$$

In the above expression, v_i is the cluster center of c_i and $\bar{v}$ is the total mean vector.

For obtaining proper c, one should minimize parameter of Scs with respect to c. However, this require that we know the proper value of m. For choosing m [6] and [7] introduce new parameters called total scatter matrix (S_T) and K.

$$S_T = \sum_{k=1}^{N}\left(\sum_{i=1}^{c}(u_{ik})^m\right)(x_k - \bar{v})(x_k - \bar{v})^T \tag{3}$$

$$K = \text{trace}\left(\sum_{K=1}^{N}\left[\left(x_K - \frac{1}{N}\sum_{k=1}^{N}x_k\right)\left(x_K - \frac{1}{N}\sum_{k=1}^{N}x_k\right)^T\right]\right) \tag{4}$$

Therefore, for clustering a data set, a suitable value of m may be fined in mid-domain of K, i.e., $K/2$ [7]. For this purpose, variations of S_T is plotted as a function of m with the different values of c. However, this methodology gives a reliable domain for selecting m instead of a single value.

For proper selection of number of clusters c, we use the subtractive clustering method [8]. The subtractive clustering method assumes each data point is a potential cluster center and calculates a measure of the likelihood that each data point would define the cluster center, based on the density of surrounding data points. Steps of the algorithm are as follows:

-Selects the data point with the highest potential to be the first cluster center,

-Removes all data points in the vicinity of the first cluster center (as determined by radii), in order to determine the next data cluster and its center location,

-Iterate on this process until all of the data is within radius of a cluster center,

The subtractive clustering method is an extension of the mountain clustering method proposed by Yager [9]. For the convenience of our description, let us rewrite subtractive clustering method as follows:

Taking each data in a set of data points as a candidate cluster center. Defining a measure as $P_i = \sum_{j=1}^{n} e^{-\xi\|x_i - x_j\|^2}$ for each data point, where, $\xi = \frac{4}{r_a^{2/(m-1)}}$, r_a is a positive constant; P_i denotes the function of the distance x_i to any other data pint in a data set. Each value of P_i is calculated and then x_i with the maximum distance value (P_i) is taken as a first cluster center x_1^*. The next cluster center is calculated and determined by $P_i = P_i - P_1^* e^{-\sigma\|x_i - x_1^*\|^2}$, where $\sigma = \frac{4}{r_b^2}$, $r_b = 1.5r_a$. The nearer a data point to x_1^* results in the smaller related P_i. Thus, the data point will have less possibility to be taken as the next cluster center. Then, the third data point is taken and calculated as above. This process is continued until $P_k^* < 0.15P_1^*$, where $P_i = P_i - P_k^* e^{-\sigma\|x_i - x_k^*\|^2}$.

After selecting the proper value of c by means of subtractive clustering, we can find the proper value of m. For this purpose, the trace of total scatter matrix (S_T) is plotted. A suitable value of m is that which gives a value of S_T equal to $K/2$ [6] and [7].

Another problem in FCM algorithm arises from the fact that this algorithm may produce only local minimum or partial optimal points. Therefore, different initial guesses for cluster centers may lead to different optimum results. In order to efficiently obtain a preference for initial location of cluster prototypes, the subtractive clustering method is used, too.

Next, the values of parameters for parameter identification should be computed. In a fuzzy model, the parameters are those concerned with membership functions. So, after the clustering process and obtaining the membership functions, we approximate trapezoidal membership functions. For this purpose we first obtain the trapezoidal fuzzy set parameters [*at bt ct dt*] by cutting the membership functions by $\alpha-\mathrm{cut}$ of level $\alpha=0.1$ and $1-\alpha$. The points of intersections of $\alpha-\mathrm{cut}$ are used as an initial guess for parameters [*at bt ct dt*]. Then, these parameters are optimized to minimize the difference between the initial fuzzy sets and trapezoidal membership functions. Here, SIMPLEX optimization scheme is implemented, which gives the global minimum point.

3-Fuzzy Inference Parameter Optimization

We use the following linguistic fuzzy model, which consist of a set of fuzzy rules, each describing a local input-output relation in a linear form [10].

$$\begin{aligned} &R_i : \mathrm{IF}\ x_1\ \mathrm{isr}\ A_{i1}\ \mathrm{AND} \ldots \mathrm{AND}\ x_n\ \mathrm{isr}\ A_{in} \\ &\quad \mathrm{THEN}\ \hat{y}_i = a_i x + b_i \quad i = 1,2,\ldots,c \end{aligned} \tag{5}$$

Here, R_i is the *ith* rule, $X=[x_1,\ldots,x_n]$ is the vector of input variables, $A_{i1},\ldots,A_{in}$ are fuzzy sets defined in the antecedent space, and $\hat{y}_i$ is the rule output, *c* denotes the number of rules in the rule base, and *isr* means "is related to". The aggregated output of the model $\hat{y}$ is calculated by Basic Defuzzification Distribution (BADD) method [11]:

$$\hat{y} = \frac{\sum_{i=1}^{c} \beta_i^{\alpha}(x)\hat{y}_i}{\sum_{i=1}^{c} \beta_i^{\alpha}(x)} \tag{6}$$

Obviously, BAAD is essentially a family of parameterized defuzzification methods. By continuously varying α in the real interval, it is possible to have more appropriate mapping from fuzzy set to a crisp value, depending on the system behavior.

In the above formula $\beta_i(x)$ is degree of firing of the i^{th} rule:

$$\beta_i(x) = T_{DP}(A_{i1},\ldots,A_{in}) \tag{7}$$

We use the parametric method of Dubios & Prade [12], for calculating degree of firing of rules in the formula (7):

$$T_{DP}(a,b) = \frac{ab}{\max(a,b,\gamma)} \tag{8}$$

In the above formula a and b are two fuzzy sets and γ is the parameter between zero and one.

For proper choosing of α and γ, an evolutionary programming (EP) [13] is implemented, since it does not demand a rich knowledge of system the behavior. Suppose that the consequent parameters a_i and b_i are determined by the process of determination of them explained in section 4. We apply the following EP procedure for obtaining proper parameters:

1-Generate an initial population of μ individuals randomly and set k=1. Each individual is a pair of real-valued vectors, $(w_i, \eta_i), \forall i \in \{1,...,\mu\}$, where w_i^s are system parameters, i.e, α, γ and η_i^s are variance vectors for Gaussian mutations (also known as strategy parameters in self-adaptive EA's). Each individual corresponds to a fuzzy model.

2-Each individual (w_i, η_i), $i = 1,..., \mu$, creates a single offspring (w_i', η_i'):

$$\eta_i'(j) = \eta_i(j)\exp(\tau' N(0,1) + \tau N_j(0,1)) \tag{9}$$

$$w_i'(j) = w_i(j) + \eta_i'(j) N_j(0,1) \quad \text{for } j = 1,..., n \tag{10}$$

Where, $w_i(j), w_i'(j), \eta_i(j)$, and $\eta_i'(j)$ denote the jth component of the vectors w_i, w_i', η_i, and η_i', $N(0,1)$ is a normally distributed one-dimensional random number with mean zero and variance equal to one, $N_j(0,1)$ indicates that a random number is generated anew for each value of j, and parameters τ and τ' are commonly set to $\left(\sqrt{2\sqrt{n}}\right)^{-1}$ and $\left(\sqrt{2n}\right)^{-1}$, respectively.

3-Determine the fitness of every individual, including all parents and offsprings, based on the training error. Here, different error functions may be used. We use the performance index (PI) as a training error:

$$PI = \frac{\sum_{i=1}^{N} (y^i - \hat{y}^i)^2}{N} \tag{11}$$

4-Conduct pairwise comparison over the union of parents (w_i, η_i) and offspring (w_i', η_i'), $\forall i \in \{1,...,\mu\}$. For each individual, q opponents are chosen uniformly random from all parents and offsprings. For each comparison, if individual's fitness is not smaller than opponent's, it receives a "win". Select μ individuals out of (w_i, η_i) and (w_i', η_i'), $\forall i \in \{1,..., \mu\}$, that have most wins to form the next generation.

5-Stop if the halting criterion is satisfied, otherwise, k=k+1 and go to step 2.

4- Fuzzy consequent parameter optimization

Least squares method identifies the consequent parameters from given data, on the assumption that the input partition is given. The normalized activation degree of the r[th] rule for the k[th] input pattern by:

$$\phi_r(k) = \frac{\beta_r^\alpha(k)}{\sum_{r=1}^{c} \beta_r^\alpha(k)} \tag{12}$$

Where β_r^α is the degree of firing of the rth rule and α is adjustment factor, which is determined by evolutionary programming. Combining these into matrix ϕ:

$$\phi = \begin{bmatrix} \phi_1(1) \ldots \phi_r(1) \ldots \phi_c(1) \\ \ldots \quad \ldots \quad \ldots \\ \phi_1(k) \ldots \phi_r(k) \ldots \phi_c(k) \\ \ldots \quad \ldots \quad \ldots \\ \phi_1(N) \ldots \phi_r(N) \ldots \phi_c(N) \end{bmatrix} \tag{13}$$

The output of fuzzy model computed over all input patterns (*k=1...N*) be equivalent to:

$$y = \phi\theta \tag{14}$$

Where θ are consequent parameters.

$$\theta = [p_{01}, \ldots, p_{0c}]^T, \; p_{0i} = [a_i, b_i]^T \tag{15}$$

and

$$y = [y(1), \ldots, y(N)]^T \tag{16}$$

For the given y (the vectors of output reference values), the output parameters can be estimated by:

$$\theta = [\phi^T\phi]^{-1}\phi^T y \tag{17}$$

It should be noted that for each step of evolutionary programming, the candidate values for parameters α and γ and their related values and the calculated value of $\hat{y}$ are chosen. Finally, the performance index (PI) of the model is evaluated. If PI satisfies the evolutionary programming criteria, the value of α and γ are selected and the algorithm is ended; otherwise the above procedure is repeated.

5-Application Example

In this section, proposed fuzzy modeling system is implemented to model the inverse kinematics of the two-joint planar robot arm shown in Figure 1 [14]. This problem involves learning to map from an end point Cartesian position (x, y) to joint angles (θ_1, θ_2). The forward kinematics equations from (θ_1, θ_2) to (x, y) are straightforward:

$$\begin{cases} x = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \\ y = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \end{cases} \tag{18}$$

Where, l_1 and l_2 are arm lengths, and θ_1 and θ_2 are their respective angles (see Figure 1). However, the inverse mappings from (x, y) to (θ_1, θ_2) are not clear:

$$\begin{cases} \theta_2 = \cos^{-1}\left[\left(x^2 + y^2 - l_1^2 - l_2^2\right)/(2 l_1 l_2)\right] \\ \theta_1 = \tan^{-1}(y/x) - \tan^{-1}\left[l_2 \sin\theta_2 / (l_1 + l_2 \cos\theta_2)\right] \end{cases} \tag{19}$$

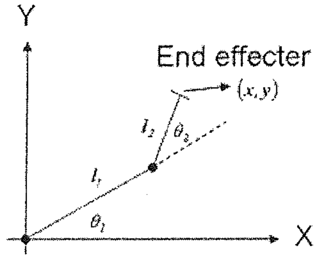


Figure (1) Two-joint planar robot

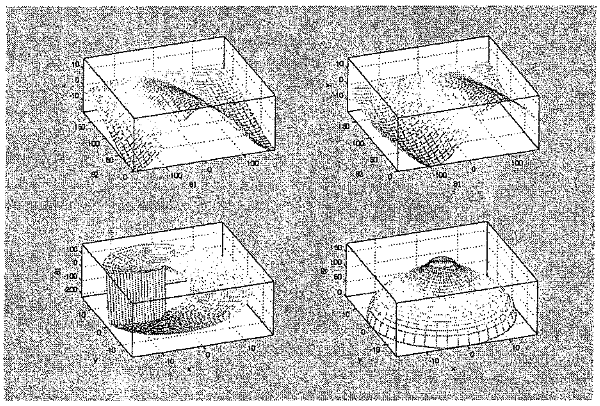


Figure (2) Direct and inverse kinematics of two-joint planar robot arm.

Figure (2) Demonstrates the forward mapping from (θ_1, θ_2) to (x, y) (the first row) and the inverse mapping from (x, y) to (θ_1, θ_2) (the second row). Here, it is assumed that $l_1 = 10, l_2 = 7$, and the value of θ_2 are restricted to $[0, \pi]$.

Even though it is possible to find the inverse mapping algebraically, the solutions are not generally available for multi-joint robot arm in 3-D space. Instead of using the above equations directly, we use two fuzzy modeling system to learn these inverse mappings. From the first quadrature, we collected 229 training data pairs of form (x, y, θ_1) and (x, y, θ_2), to train two fuzzy system, respectively (this data can be find in traininv m-file in MATLAB).

The parameters of two fuzzy models $(a_{ij}, b_i, c_{ij}, d_i, \alpha, \gamma)$ available in tables 1,2. In Figures 4, 5 membership functions of input variables (A_{ij}, B_{ij}) are shown. Numbers of rules of fuzzy systems are 7. The generated fuzzy rules are as bellow:

IF x is A_{11} AND y is A_{12} THEN $\theta_1 = [a_{11}\ a_{12}][x\ y]^T + b_1$

IF x is A_{21} AND y is A_{22} THEN $\theta_1 = [a_{21}\ a_{22}][x\ y]^T + b_2$

IF x is A_{31} AND y is A_{32} THEN $\theta_1 = [a_{31}\ a_{32}][x\ y]^T + b_3$

IF x is A_{41} AND y is A_{42} THEN $\theta_1 = [a_{41}\ a_{42}][x\ y]^T + b_4$

IF x is A_{51} AND y is A_{52} THEN $\theta_1 = [a_{51}\ a_{52}][x\ y]^T + b_5$

IF x is A_{61} AND y is A_{62} THEN $\theta_1 = [a_{61}\ a_{62}][x\ y]^T + b_6$

IF x is A_{71} AND y is A_{72} THEN $\theta_1 = [a_{71}\ a_{72}][x\ y]^T + b_7$

IF x is B_{11} AND y is B_{12} THEN $\theta_2 = [c_{11}\ d_{12}][x\ y]^T + e_1$

IF x is B_{21} AND y is B_{22} THEN $\theta_2 = [c_{21}\ d_{22}][x\ y]^T + e_2$

IF x is B_{31} AND y is B_{32} THEN $\theta_2 = [c_{31}\ d_{32}][x\ y]^T + e_3$

IF x is B_{41} AND y is B_{42} THEN $\theta_2 = [c_{41}\ d_{42}][x\ y]^T + e_4$

IF x is B_{51} AND y is B_{52} THEN $\theta_2 = [c_{51}\ d_{52}][x\ y]^T + e_5$

IF x is B_{61} AND y is B_{62} THEN $\theta_2 = [c_{61}\ d_{62}][x\ y]^T + e_6$

IF x is B_{71} AND y is B_{72} THEN $\theta_2 = [c_{71}\ d_{72}][x\ y]^T + e_7$

The antecedent spaces of the above models, which are two-dimensional space of the inputs *x* and *y*, are partitioned into seven fuzzy subspaces. The above model can be regarded as a quasi-linear system (i.e., a linear system with input dependent parameters). To see this, we combine equations (5) and (6) and calculate the output of the model, $\hat{y}$

$$\hat{y} = \frac{\sum_{i=1}^{c} \beta_i^{\alpha}(x) \times (a_i x + b_i)}{\sum_{i=1}^{c} \beta_i^{\alpha}(x)} = k_1 x + k_2$$

$$k_1 = \frac{\sum_{i=1}^{c} \beta_i^{\alpha}(x) \times a_i}{\sum_{i=1}^{c} \beta_i^{\alpha}(x)}, k_2 = \frac{\sum_{i=1}^{c} \beta_i^{\alpha}(x) \times b_i}{\sum_{i=1}^{c} \beta_i^{\alpha}(x)}$$

Figure 3 demonstrates θ_1 output from both fuzzy modeling system and equation (19). Obviously, fuzzy system properly approximates (with *PI*=.0018) the equation (19), so we can use the fuzzy modeling system in the case that it is not possible the inverse mapping algebraically.

Figure 3, also, demonstrates the result of comparing proposed fuzzy modeling system with ANFIS (adaptive neuro fuzzy inference system). The ANFIS has 9 rules consist of trapezoidal membership functions. The performance index (*PI*) for ANFIS is .0018. So, our proposed system approximates the inverse kinematics with smaller number of rules and with the same performance index.

At the end we study the example of the planar 3R manipulator. From basic trigonometry, the position and orientation of the end effector can be written in terms of the joint coordinates in the following way:

$$\begin{aligned} x &= l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ y &= l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3) \\ \varphi &= \theta_1 + \theta_2 + \theta_3 \end{aligned} \tag{20}$$

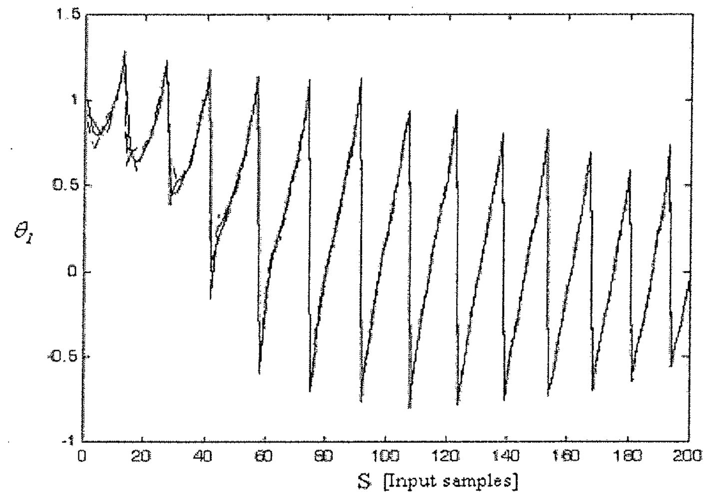


Figure (3) ___ Algebraic output ------ Proposed fuzzy system output -.-.-.-. ANFIS system output

We assume that we are given the Cartesian coordinates, x, y, and φ and we want to find the joint angles θ_1, θ_2 and θ_3 in terms of the Cartesian coordinates. Instead of solving the above equations directly, we use three fuzzy modeling systems to learn these inverse mappings. We can collect training data pairs of form $(x, y, \varphi, \theta_1)$, $(x, y, \varphi, \theta_2)$ and $(x, y, \varphi, \theta_3)$ to train three fuzzy systems.

6-Conclusions

In this paper a systematic fuzzy modeling is introduced in which the inference mechanism, number of rules and order of fuzziness of model are identified from data. For this purpose, a parameterized formulation was applied by which the suitable inference mechanism is adjusted for the system based on evolutionary programming. Moreover, to take advantages of linear system properties, linear functions of the antecedent variables were implemented in the consequence of the model, thus, we could describe the system with a set of local linear input-output relations and take advantages of linear systems properties.

The modeling methodology described in this article has been successfully applied to inverse kinematics control of two-joint planar robot arm. The results show that since the system is self-tuning, we don't need to tune the system parameters again. The potential future work for this research is that the proposed method can be generalized so that all system structure and parameters can be obtained by the optimization methods such as genetic algorithms.

Table (1) Parameters of Fuzzy modeling system 1.

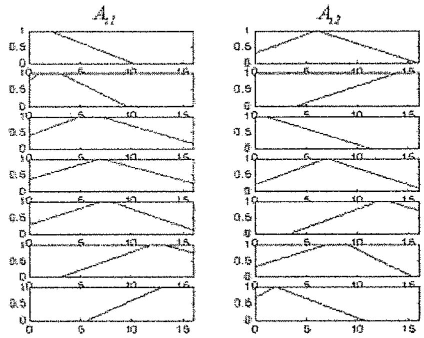
a_{i1}	a_{i2}	b_i	α	γ	PI
-0.3256	-0.2192	1.8794			
-0.0473	0.0271	0.9012			
0.0041	0.4338	-0.5187			
-0.2966	0.7833	-2.6069	0.8609	0.5118	0.0018
0.0622	0.4518	-6.8508			
0.0517	0.0259	-0.2489			
0.0438	0.0785	-1.0941			

Table (2) Parameters of Fuzzy modeling system 2.

c_{i1}	d_{i2}	e_i	α	γ	PI
1.2232	0.4456	-2.3191			
-1.4525	-1.6835	26.4234			
1.2160	0.3862	-16.5236			
-0.0865	-1.0860	10.8796	0.3139	0.1564	0.0080
-0.7162	-1.5388	16.3696			
-1.0411	0.4972	3.8106			
0.9470	1.9124	-12.0559			

IF Part

THEN Part

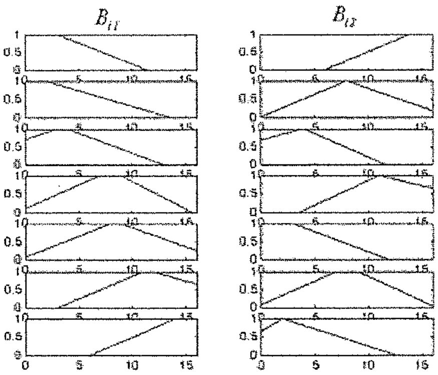


$$\theta_1 = -0.3256x - 0.2192y + 1.8794$$
$$\theta_1 = -0.0473x + 0.0271y + 0.9012$$
$$\theta_1 = 0.0041x + 0.4338y - 0.5187$$
$$\theta_1 = -0.2966x + 0.7833y - 2.6069$$
$$\theta_1 = 0.0622x + 0.4518y - 6.8508$$
$$\theta_1 = 0.0517x + 0.0259y - 0.2489$$
$$\theta_1 = 0.0438x + 0.0785y - 1.0941$$

Figure (4) Membership functions and parameters for fuzzy system 1.

IF Part

THEN Part



$$\theta_2 = 1.2232x + 0.4456y - 2.3191$$
$$\theta_2 = -1.4525x - 1.6835y + 26.4234$$
$$\theta_2 = 1.2160x + 0.3862y - 16.5236$$
$$\theta_2 = -0.0865x - 1.0860y + 10.8796$$
$$\theta_2 = -0.7162x - 1.5388y + 16.3696$$
$$\theta_2 = -1.0411x + 0.4972y + 3.8106$$
$$\theta_2 = 0.9470x + 1.9124y - 12.0559$$

Figure (5) Membership functions and parameters for fuzzy system 2.

References

[1] W. Pedrycz, fuzzy Sets Engineering. Boca Raton, FL: CRC, 1995.

[2] B. Kosko, Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence. Englewood Cliffs, NJ: Prentice-Hall, 1992.

[3] J. H. Nie and T. H. Lee, "Rule-based modeling: Fast construction and optimal manipulation," IEEE Transactions on Systems, Man and Cybernetics, A, vol. 26, pp. 728-738, Nov. 1996.

[4] R. Babuska and H. B. Verbruggen, "Fuzzy set methods for local modeling and identification," in Multiple Model Approaches to Nonlinear Modeling and Control, R. Murray-Smith and T. A. Johansen, Eds. London, U.K.:Taylor & Francis, 1997, pp. 75-100.

[5] M. Sugeno and T. Yasukawa, "A fuzzy-logic-based approach to qualitative modeling," IEEE Transactions on Fuzzy Systems, 1, No. 1:7-31, 1993.

[6] M. R. Emami, I. B. Turksen, A. A. Goldenberg, "An improved fuzzy modeling algorithm, part II: System identification," NAFIPS, 1996.

[7] I. B. Turksen and M.H. Fazel Zarandi, "Fuzzy system models for aggregated scheduling analysis," International Journal of Approximate Reasoning (IJAR), Vol.19, pp. 119-143, 1998.

[8] S. Chiu, "Method and software for extracting fuzzy classification rules by subtractive clustering," NAFIPS, 1996.

[9] R. R. Yager and D. P. Filev, "approximate clustering via the mountain method," IEEE Transaction on Systems, Man and Cybernetics, 24: 1279-1284, 1994

[10] M. Setnes, R. Babuska, and H.B. Verbruggen. "Rule-based modeling: Precision and transparency." IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews, 28(1): 165-169, 1998.

[11] R. R. Yager and D. P. Filev, Essentials of Fuzzy Modeling and Control, John Wiley & Sons, Inc, 1994.

[12] D. Dubois, H. Prade, Fuzzy Sets and Systems: Theory and Applications. Academic Press: New York, 1980.

[13] X. Yao, "Evolving artificial neural networks," Proceedings of the IEEE, 87(9): 1423-1447, September 1999.

[14] J.-S. R. Jang and N. Gulley, "The Fuzzy Logic Toolbox for Use with MATLAB," The MathWorks, Inc., Natick, Massachusetts, 1995.

[15]www.mathworks.com

[16] T. Sudkamp, A. Knapp, and J. Knapp, "Model Generation by Domain Refinement And Rule Reduction", IEEE Transaction on Systems, Man and Cybernetics, Vol. 33, NO. 1, 2003.

[17] P. Angelov, "A fuzzy controller with evolving structure", Journal of Information Sciences, Volume 161, Issues 1-2, 5, Pages 21-35, April 2004.

[18] B. Feil, J. Abonyi, F. Szeifert, "Model order selection of nonlinear input–output models—a clustering based approach", Journal of Process Control, Volume 14, Issue 6, Pages 593-602, September 2004.

Instruction to authors

a: Amirkabir scientific and Engineering Journal appears quarterly and contains papers, describing original investigations in all areas of physical sciences and engineering that have not been published or submitted for publication before. Review articles in specialized fields are also considered for publication.

b: Papers submitted for publication should be presented concisely in Word with double space format not exceed 20 pages. Articles in Persian and English are accepted for consideration.

c: Manuscripts should be prepared based on the template given in ***www.akujournal.com***.

d: The title of the paper should be concise and informative. It is highly desirable to divide the paper into different sections, such as Introduction, Problem statements, Experiments, Results, Discussion, and Conclusion. All papers should have an Abstract both in Persian and English.

Amirkabir
Journal of Science & Technology
Vol. 15. No. 59-A, (Electrical Engineering) Summer 2004
ISSN. 1015-0951

Address for correspondence:

Amirkabir University of Tech., Office of Vice Chancellor in Research Affairs, Tehran 15. Iran.
Tel: 0098-21-6491123
Fax: 0098-21-6491123
Web site: *www.akujournal.com*
Email Address: *Info@akujournal.com*