An Auto-Generating Method in the Fuzzy System Design

Ching-Chang Wong and Shyang-Ming Her

Department of Electrical Engineering, Tamkang University,
Tamsui, Taiwan 25137, Republic of China

ABSTRACT

In this paper, a parameter structure of fuzzy system is presented along with a fuzzy set elimination method is proposed, which enables an auto-generating method not only to determine an appropriate fuzzy system with a high performance but also to favor the selected system with fewer fuzzy rules. An inverted pendulum control problem is utilized to illustrate the efficiency of the proposed method in the fuzzy system design.

1. Introduction

When designing fuzzy systems in practice, we can not perfectly represent the expert's knowledge by linguistic rules and choose appropriate membership functions for fuzzy sets. Suitable fuzzy rules and membership functions may be given by a very time-consuming trial-and-error procedure. Unsystematic approach in the design of fuzzy systems has reduced the applicability of fuzzy systems. Therefore, much work has been done on the analysis of the structures and parameters of fuzzy systems so that a fuzzy system can be constructed efficiently [3-11,14,15]. Furthermore, it is desirable for designers to tune the parameters to achieve a good performance. In general, we can not make sure that the selected fuzzy systems by conventional trial-and-error methods will provide the system with a better performance. In this paper, we propose a method based on Genetic Algorithms (GAs) for selecting parameters of fuzzy system to solve this problem. GAs are search procedures based on the mechanics of natural selection and natural genetics and are efficient for global searches [1,2,14]. In view of the simple structure of GAs and the feature that they do not rely on the characteristics of the considered system, we propose a method based on GAs to efficiently and automatically find an appropriate fuzzy system so that a better performance can be assured and a time-consuming trial-and-error parameter selection method can be prevented.

One of restriction of conventional approaches to fuzzy system designs is that the number of fuzzy sets of each input variable must be defined in advance. They may have redundant fuzzy sets such that a large number of fuzzy rules are generated for a complete rule base of fuzzy system. It will result in computation complexity and memory overloading. To reduce complexity of computation and to minimize requirement of large memory, we propose a fuzzy set elimination procedure such that some redundant fuzzy sets can be eliminated to reduce the rule number and a fewer rule number of the constructed fuzzy system can be found.

This paper is organized as follows. The considered fuzzy system structure is described in Section 2. An efficient method based on GAs to generate fuzzy systems with fewer fuzzy rules is proposed in Section 3. In Section 4, the feasibility of the proposed method is examined through controlling an inverted pendulum system. Finally, Section 5 concludes the paper.

2. Fuzzy System Structure

Without loss of generality, we consider two input variables (x_1,x_2) and a single output variable (y) in this paper. A rule base of a fuzzy system can be expressed by the following [12,13]:

(j_1,j_2)-th rule:

IF x_1 is $A_{(1,j_1)}$ and x_2 is $A_{(2,j_2)}$, THEN y is $y_{(0,f(j_1,j_2))}$

$$j_i \in \{-n_i,\ldots,-1,0,1,\ldots,n_i\}, i \in \{1,2\},$$
$$f(j_1,j_2) \in \{-n_0,\ldots,-1,0,1,\ldots,n_0\} \quad (1)$$

where

$$A_{(i,j_i)} \in T(x_i) = \{A_{(i,-n_i)},\ldots,A_{(i,-1)},A_{(i,0)},A_{(i,1)},\ldots,A_{(i,n_i)}\} \quad (2)$$

$$y_{(0,f(j_1,j_2))} \in T(y) = \{y_{(0,-n_0)},\ldots,y_{(0,-1)},y_{(0,0)},y_{(0,1)},\ldots,y_{(0,n_0)}\} \quad (3)$$

The term sets $T(x_i)$ and $T(y)$ have $2n_i+1$ and $2n_0+1$ fuzzy sets in the description of the input variable x_i and the output variable y, respectively. $A_{(i,j_i)}$ is the j_i-th term of

0-7803-3796-4/97/$10.00©1997IEEE

the term set $T(x_i)$. $f(j_1,j_2)=\langle j_1+j_2\rangle$ is a constant index function so that $f(j_1,j_2)\in\{-n_0,\ldots,-1,0,1,\ldots,n_0\}$, where

$$\langle b\rangle = \begin{cases} -n_0, & if\ b<-n_0 \\ b & if\ -n_0\le b\le n_0 \\ n_0, & if\ n_0<b \end{cases} \quad (4)$$

which decides $y_{(0,f(j_1,j_2))}$ is the $f(j_1,j_2)$-th term of the term set $T(y)$. In this study, the following triangular-shaped fuzzy numbers and singletons are used for defining the antecedent and consequent membership functions, respectively.

$$A_{(i,j)}(x_i) = \begin{cases} \dfrac{x_i-c_{(i,j)}+l_{(i,j)}}{l_{(i,j)}}, & if\ c_{(i,j)}-l_{(i,j)}\le x_i\le c_{(i,j)} \\ \dfrac{-x_i+c_{(i,j)}+r_{(i,j)}}{r_{(i,j)}}, & if\ c_{(i,j)}\le x_i\le c_{(i,j)}+r_{(i,j)} \\ 0, & otherwise \end{cases}$$

$$x_i\in[c_{(i,-n_i)},c_{(i,n_i)}] \quad (5)$$

$$y_{(0,j)}(y) = \begin{cases} 1, & if\ y=c_{(0,j)} \\ 0, & otherwise \end{cases} \quad y\in[c_{(0,-n_y)},c_{(0,n_y)}] \quad (6)$$

$c_{(i,j)}$ denotes the center of the membership function where the membership grade is equal to 1, and $l_{(i,j)}$ and $r_{(i,j)}$ denote the left and right spread of the triangular-shaped membership function, respectively (See Figure 1). In this paper, the values of $c_{(i,j)}$, $l_{(i,j)}$ and $r_{(i,j)}$ are described by

$$c_{(i,j)} = \begin{cases} -(\sum_{k=j}^{-1}d_{(i,k)}), & if\ -n_i\le j\le -1 \\ 0, & if\ j=0 \\ \sum_{k=1}^{j}d_{(i,k)}, & if\ 1\le j\le n_i \end{cases} \quad i\in\{0,1,2\} \quad (7)$$

$$l_{(i,j)} = \begin{cases} d_{(i,j-1)}, & if\ j\le 0 \\ d_{(i,j)}, & if\ j>0 \end{cases} \quad i\in\{1,2\} \quad (8)$$

and

$$r_{(i,j)} = \begin{cases} d_{(i,j)}, & if\ j<0 \\ d_{(i,j+1)}, & if\ j\ge 0 \end{cases} \quad i\in\{1,2\} \quad (9)$$

where $d_{(i,k)}$ are positive real values needed to be determined. That is, $d_{(i,j)}$, $j\in\{-n_i,\ldots,-1,1,\ldots,n_i\}$, are considered as parameters to describe membership functions of the term set $T(x_i)$ (See Figure 2).

When the input $x=(x_1,x_2)$ is given, the truth value of the premise of the (j_1,j_2)-th rule is calculated by

$$w_{(j_1,j_2)}=A_{(1,j_1)}(x_1)A_{(2,j_2)}(x_2) \quad (10)$$

There are $(2n_1+1)(2n_2+1)$ rules. By taking the weighted average method, the output of the fuzzy system can be calculated by

$$y=\frac{\sum_{j_1=-n_1}^{n_1}\sum_{j_2=-n_2}^{n_2}w_{(j_1,j_2)}\bullet y_{(0,f(j_1,j_2))}}{\sum_{j_1=-n_1}^{n_1}\sum_{j_2=-n_2}^{n_2}w_{(j_1,j_2)}} \quad (11)$$

The output of the fuzzy system consists of the parameters $d_{(i,j)}$ ($i\in\{0,1,2\}$; $j\in\{-n_i,\ldots,-1,1,\ldots,n_i\}$). These parameters can be grouped as the following parameter set:

$$R_i=(r_1,\ldots,r_q)=\{d_{(1,-n_1)},\ldots,d_{(1,-1)},d_{(1,1)},\ldots,d_{(1,n_1)};\ d_{(2,-n_2)},\ldots,d_{(2,-1)},d_{(2,1)},\ldots,d_{(2,n_2)};\ d_{(0,-n_0)},\ldots,d_{(0,-1)},d_{(0,1)},\ldots,d_{(0,n_0)};\} \quad (12)$$

where $q=2n_1+2n_2+2n_0$. One parameter set represents one fuzzy system. The input-output relation of the corresponding fuzzy system will be changed by varying these parameters, so the selected parameter set affects the performance of the constructed fuzzy system to a large extent. Therefore, a searching problem can be formulated and GAs can be applied to select an appropriate parameter set. The parameter selection method is discussed in the following section.

3. Rule Extraction

Let $l(x_i)$ and $u(x_i)$ be the lower and upper bounds of a feasible domain of i-th input variable x_i, respectively. The fuzzy set elimination procedure can be proposed as follows: When the values $d_{(i,j)}$, $j\in\{-n_i,\ldots,-1,1,\ldots,n_i\}$, are determined and the following conditions are satisfied

$$d_{(i,1)}+d_{(i,2)}+\ldots+d_{(i,q_i-1)}<u(x_i)\le d_{(i,1)}+d_{(i,2)}+\ldots+d_{(i,q_i)} \quad (13)$$

and

$$d_{(i,-1)}+d_{(i,-2)}+\ldots+d_{(i,-p_i+1)}<|l(x_i)|\le d_{(i,-1)}+d_{(i,-2)}+\ldots+d_{(i,-p_i)} \quad (14)$$

then there are q_i+p_i-1 points $(c_{(i,-p_i+1)},\ldots,c_{(i,-1)},0,c_{(i,1)},\ldots,c_{(i,q_i-1)})$ in the interval $[l(x_i),u(x_i)]$ satisfying the following inequalities

$$l(x_i)<c_{(i,-p_i+1)}<\ldots<c_{(i,-1)}<0<c_{(i,1)}<\ldots<c_{(i,q_i-1)}<u(x_i) \quad (15)$$

and the required number of fuzzy sets is q_i+p_i+1 $(q_i\le n_i;\ p_i\le n_i)$. Description of the structure can be shown as Figure 2, where $n_i=3$, $q_i=3$, $p_i=2$, $c_{(i,-2)}=l(x_i)$ and $c_{(i,3)}=u(x_i)$. We denote $c_{(i,-p_i)}=l(x_i)$ and $c_{(i,q_i)}=u(x_i)$ to meet the conventional representation of membership functions.

If each required number of fuzzy sets for each variable is determined by the elimination procedure, then the output equation (11) of the fuzzy system can be simply calculated by

$$y=\frac{\sum_{j_1=-p_1}^{q_1}\sum_{j_2=-p_2}^{q_2} w_{(j_1 j_2)} \bullet y_{(0,f(j_1 j_2))}}{\sum_{j_1=-p_1}^{q_1}\sum_{j_2=-p_2}^{q_2} w_{(j_1 j_2)}} \quad (16)$$

This will minimize requirement of large memory and reduce complexity of computation in the implementation of the fuzzy system.

In order to select an appropriate set of parameters R_i by using GAs, first, we select R_i as a parameter set and code the parameter set as a finite-length string, then choose a fitness function f(.) so that GAs can be used to search for a better solution in the parameter space according to the direction of the fitness function. The goal of the proposed method is not only to find a fuzzy system with a high performance but also to favor the constructed system with fewer fuzzy rules. Therefore, we define the following fitness function:

$$f(R_i)=g_1(J(R_i))*g_2(m(R_i)), \quad (17)$$

where $f(R_i)$ is the fitness value of the i-th string (fuzzy system) R_i; $m(R_i)=\prod_{i=1}^{2}(q_i+p_i+1)$ is the required rule number of R_i based on the proposed fuzzy set elimination method; $J(R_i)$ is the performance of R_i; and $g_1(.)$ and $g_2(.)$ are any appropriate membership functions. Here, the concept of membership function is used to evaluate the grade of goodness of each performance measure so that all the grade values of performance measurement are all in the range [0,1]. The advantage is that no matter what unit of the measurement ($J(R_i)$ or $m(R_i)$), we can easily see the result is good or bad from the grade value is toward one or zero. In this paper, we first choose

$$g_1(J(R_i))=\exp(-\tfrac{J(R_i)}{\sigma_J}) \quad (18)$$

$$g_2(m(R_i))=\begin{cases}1, & if\ m(R_i)\le m_R\\ \exp(-(\frac{m(R_i)-m_R}{\sigma_R})^2), & otherwise\end{cases} \quad (19)$$

where σ_J, m_R, and σ_R are determined by the designer. The other advantage is that the shape of these functions can be adjusted according to the designer, so the representation provides us a flexible method to meet any specifications. For example, if the first task is the selected fuzzy system must satisfy a preset performance, we can choose

$$g_1(J(R_i))=\begin{cases}1, & if\ J(R_i)\le AP\\ 0, & if\ J(R_i)> AP\end{cases} \quad (20)$$

where AP denotes an acceptable performance of the selected fuzzy system which is determined by the designer. Note that the fitness function is chosen so that all the fitness values are all in the range [0,1]. The fitness value depends on the performance $J(R_i)$ and also on the number $m(R_i)$ of fuzzy rules needed. The smaller the performance and the smaller the number of rules, the higher the fitness value. In this way, as the fitness value $f(R_i)$ increases as greatly as possible based on the guidance of the proposed fitness function, the fuzzy system corresponding to the selected string will satisfy the desired objective as well as possible. That is, the selected fuzzy system can get a high performance and has fewer rules simultaneously.

4. Illustrated Example

In this section, an inverted pendulum control problem is used as an illustrated example. Let $x_1(t)=\theta(t)$ (angle of the pole with respect to the vertical axis) and $x_2(t)=\dot\theta(t)$ (angular velocity of the pole), then the state equation can be expressed by [4,7,14]

$$\dot x_1 = x_2 \quad (21)$$

$$\dot x_2=\frac{g\sin(x_1)+\cos(x_1)(\frac{-F-mlx_2^2\sin(x_1)}{m+M})}{l(\frac{4}{3}-\frac{m\cos^2(x_1)}{m+M})} \quad (22)$$

where g (acceleration due to the gravity) is 9.8 meter/sec^2, M (mass of cart) is 1.0 kg, m (mass of pole) is 0.1 kg, l (half length of pole) is 0.5 meter, and F is the applied force in Newton. In this example, the fuzzy rules can be considered by

(j_1,j_2)-th rule:

IF x_1 is $A_{(1,j_1)}$ and x_2 is $A_{(2,j_2)}$ THEN y is $y_{(0,f(j_1,j_2))}$

$j_1, j_2, f(j_1,j_2)=\langle j_1+j_2\rangle \in \{-4,-3,-2,-1,0,1,2,3,4\}$. (23)

where $n_1=n_2=n_0=4$. The adopted fuzzy control system is given 9x9=81 rules at the start. Since the symmetric scheme of the inverted pendulum system, we let $d_{(i,j)}=d_{(i,-j)}$, i=0,1,2 and j=1,2,3,4 for the simplicity. Therefore, 3x4=12 parameters $\{d_{(1,1)},d_{(1,2)},d_{(1,3)},d_{(1,4)}; d_{(2,1)},d_{(2,2)},d_{(2,3)},d_{(2,4)}; d_{(0,1)},d_{(0,2)},d_{(0,3)},d_{(0,4)}\}$ are required to be coded as a chromosome. It is assumed that no prior knowledge (from a human operator's point of view) about the inverted pendulum system is available [4]. We use 121 numerical data uniformly distributed in the square region [-10,10] * [-10,10] to train the fuzzy control system. The time step is 0.01 second. The control task is to balance the inverted pendulum system, therefore the difference between the desired state and the actual state can be considered as a control performance. In this problem, the control performance with respect to the i-th string R_i can be described by the following error index:

$$J(R_i)=E(R_i)=\frac{\sum_{k=1}^{121}|\theta_k(0.03)|}{121} \quad (24)$$

where $|\theta_k(0.03)|$ is represented by the absolute value of error between the desired final pole angle (0 degree) and the actual pole angle ($\theta_k(0.03)$ degree) of the third time stage corresponding to the k-th data.

In this simulation, the boundary conditions of $\theta, \dot\theta$ and y are [-20,20], [-60,60] and [-25,25], respectively. The boundary conditions can be chosen by the designer according to feasible domains of input variables. Following the proposed method in Section 3, the data of the final best string are shown in Table 1 after 50 generations, where $\sigma_J = 5$, $m_R = 9$, and $\sigma_R = 40$. Table 1 also shows that the input and output fuzzy partitions are automatically constructed by the proposed method. After eliminating the unnecessary fuzzy sets from the data of Table 1 according to the proposed fuzzy set elimination method, we find that the constructed fuzzy system only needs 5x5=25 fuzzy rules. The performance of the constructed fuzzy control system has been evaluated with several different initial points. Some of simulation results are illustrated in Figure 3, where $\theta = 10, 20, 40$ and $\dot{\theta} = 0$. These results show that the inverted pendulum system can be balanced within a short time. Furthermore, the constructed system also can balance the pole from some initial points that don't fall into the region of the training data set. This demonstrates the robustness of the acquired fuzzy control system with respect to unseen initial conditions.

If we use Equations (20) and (19) to define the functions g_1 and g_2, respectively, where m_R =9 and σ_R =40. Following the proposed method, some simulation results according to different preset performance AP are shown in Figure 4~6, respectively. The summary results of each performance measure are shown in Table 2. The results of Table 2 show that the selected rule number increases as the preset performance AP decreases. It is reasonable and can prove that the efficiency and flexibility of the proposed method. We only give an acceptable performance (AP), then a fuzzy system with fewer rule number can be easily selected and $E(R_i) \leq AP$ is maintained.

5. Conclusions

To improve the disadvantage of conventional fuzzy system designs that the number of fuzzy sets of each input variable must be defined in advance, we have proposed a method to eliminate unnecessary fuzzy sets so that a fewer rule number can be selected. Furthermore, the proposed method based on GAs has the ability to select an appropriate fuzzy system with a high performance and some time-consuming trial-and-error procedure can be prevented. From the simulated results, we have proved that the efficiency and flexibility of the proposed method.

Acknowledgement:

This research is supported in part by the National Science Council of the Republic of China under contract NSC-86-2213-E-032-006.

References:

[1] L. Davis (Ed.), Handbook of Genetic Algorithms, (Van Nostrand Reinhold, 1991).

[2] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning. Reading, (MA: Addison-Wesley, 1989).

[3] S.Z. He, S. Tan and C.C Hang and P.Z. Wang, Control of dynamical processes using an on-line rule-adaptive fuzzy control system, Fuzzy Sets and Systems 54 (1993) 11-22 .

[4] J.-S. R. Jang, Fuzzy Controller Design without Domain Experts, IEEE Int. Conf. on Fuzzy Systems, (1992) 289-296.

[5] J.-S. R. Jang, ANFIS: Adaptive-Network-Based Fuzzy inference system, IEEE Trans. on Syst., Man and Cybern., 23 (3) (1993) 665-685.

[6] C.L. Karr, Fuzzy control of pH using genetic algorithms, IEEE Trans. on Fuzzy Systems, 1 (1993) 46-53.

[7] M.A. Lee and H. Takagi, Integrating design stages of fuzzy systems using genetic algorithms, Proceedings of the 2nd IEEE International Conference on Fuzzy Systems (1993) 612-617.

[8] M. Maeda and S. Murakami, A self-tuning fuzzy controller, Fuzzy Sets and Systems 51 (1992) 29-40.

[9] K.S. Narendra, and K.Parthasarathy, Identification and control of dynamical systems using neural networks, IEEE Trans. on Neural Networks, 1 (1) (1990) 4-27.

[10] T. Takagi and M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, IEEE Trans. Syst., Man and Cybern. 15 (1) (1985) 116-132.

[11] L.X. Wang, *Adaptive Fuzzy Systems and Control: Design and Stability Analysis* (Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1994).

[12] P.Z. Wang, H.M. Zhang, and W. Xu, Pad-analysis of fuzzy control stability, Fuzzy Sets and Systems, 38 (1990) 27-42 .

[13] C.C. Wong, C.H. Chou and D.L. Mon, Studies on the output of fuzzy controller with multiple inputs, Fuzzy Sets and Systems, 57 (1993) 149-158.

[14] C.C. Wong and S.M. Feng, Switching-type fuzzy controller design by genetic algorithms, Fuzzy Sets and Systems, 74 (1995) 175-185.

[15] Z.Q. Wu, P.Z. Wang, T.H. Heng and S.S. Song, A rule self-regulating fuzzy controller, Fuzzy Sets and Systems 47 (1992) 13-21.

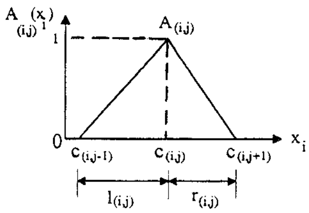

Fig. 1. A triangular-shaped membership function.

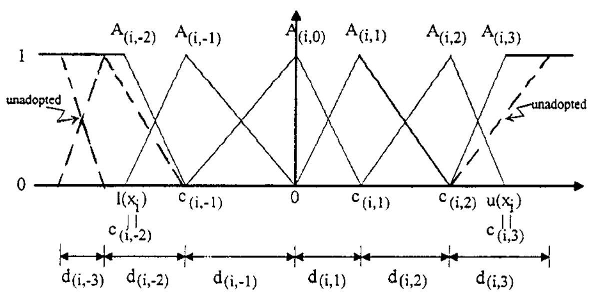

Fig. 2. Description of the membership functions of $T(x_i)$. (The membership functions represented by dash line are the undesired forms from the proposed fuzzy set elimination method.)

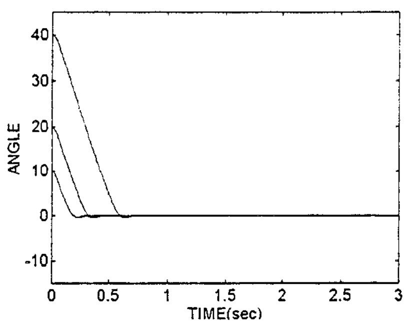

Fig. 3. Simulation results ($m(R_i)$=5x5 and $J(R_i)$ = 4.9084)

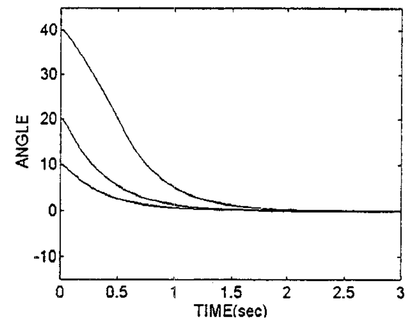

Fig. 4. Simulation results ($m(R_i)$=3x3 and $J(R_i)$ = 5.3124)

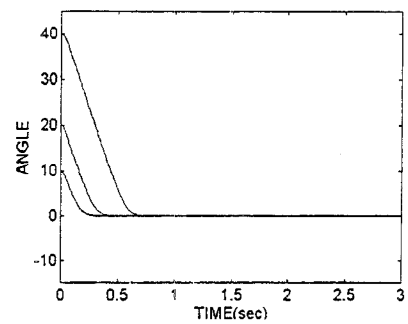

Fig. 5. Simulation results ($m(R_i)$=3x5 and $J(R_i)$ = 4.9213)

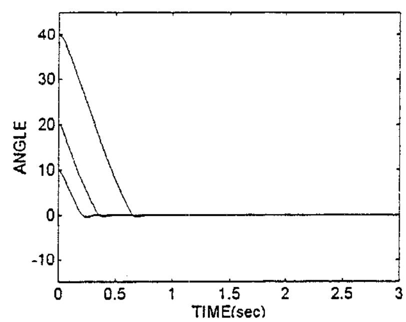

Fig. 6. Simulation results ($m(R_i)$=7x5 and $J(R_i)$ = 4.7046)

Table 1: Data of the delected fuzzy system ($m(R_i)$=5x5 and $J(R_i)$=4.9084) in Example.

Parameter Value Selected by proposed Method		Center Value Determined by Equation (6)	Final Selection (Adopted, Modified or Eliminated)
$d_{(1,-4)}$ [5,20]	14.081134	$c_{(1,-4)}$ = -36.383187	Eliminated
$d_{(1,-3)}$ [5,20]	2.14956	$c_{(1,-3)}$ = -22.302053	Eliminated
$d_{(1,-2)}$ [5,20]	18.152493	$c_{(1,-2)}$ = -20.152493	Modified by $c_{(1,-2)}$ = -20
$d_{(1,-1)}$ [5,20]	2	$c_{(1,-1)}$ = -2	Adopted
$d_{(1,0)}$	undefined	$c_{(1,0)}$ = 0	Adopted
$d_{(1,1)}$ [5,20]	2	$c_{(1,1)}$ = 2	Adopted
$d_{(1,2)}$ [5,20]	18.152493	$c_{(1,2)}$ = 20.152493	Modified by $c_{(1,2)}$ = 20
$d_{(1,3)}$ [5,20]	2.14956	$c_{(1,3)}$ = 22.302053	Eliminated
$d_{(1,4)}$ [5,20]	14.081134	$c_{(1,4)}$ = 36.383187	Eliminated
$d_{(2,-4)}$	37.331379	$c_{(2,-4)}$ = -156.95015	Eliminated
$d_{(2,-3)}$	37.214077	$c_{(2,-3)}$ = -119.618771	Eliminated
$d_{(2,-2)}$	32.404694	$c_{(2,-2)}$ = -82.404694	Modified by $c_{(2,-2)}$ = -60
$d_{(2,-1)}$	50	$c_{(2,-1)}$ = -50	Adopted
$d_{(2,0)}$	undefined	$c_{(2,0)}$ = 0	Adopted
$d_{(2,1)}$	50	$c_{(2,1)}$ = 50	Adopted
$d_{(2,2)}$	32.404694	$c_{(2,2)}$ = 82.404694	Modified by $c_{(2,2)}$ = 60
$d_{(2,3)}$	37.214077	$c_{(2,3)}$ = 119.618771	Eliminated
$d_{(2,4)}$	37.331379	$c_{(2,4)}$ = 156.95015	Eliminated
$d_{(0,-4)}$ [5,25]	16.085045	$c_{(0,-4)}$ = -68.035192	Eliminated
$d_{(0,-3)}$ [5,25]	11.950147	$c_{(0,-3)}$ = -51.950147	Eliminated
$d_{(0,-2)}$ [5,25]	20	$c_{(0,-2)}$ = -40	Modified by $c_{(0,-2)}$ = -25
$d_{(0,-1)}$ [5,25]	20	$c_{(0,-1)}$ = -20	Adopted
$d_{(0,0)}$	undefined	$c_{(0,0)}$ = 0	Adopted
$d_{(0,1)}$ [5,25]	20	$c_{(0,1)}$ = 20	Adopted
$d_{(0,2)}$ [5,25]	20	$c_{(0,2)}$ = 40	Modified by $c_{(0,2)}$ = 25
$d_{(0,3)}$ [5,25]	11.950147	$c_{(0,3)}$ = 51.950147	Eliminated
$d_{(0,4)}$ [5,25]	16.085045	$c_{(0,4)}$ = 68.035192	Eliminated

Table 2: Actual error and rule number of the selected fuzzy system corresponding to each prescribed

Prescribed Threshold (AP)	Actual Error of the Selected Fuzzy System ($E(R_i)$)	Rule Number of the Selected Fuzzy System ($m(R_i)$)
8	5.3124	9
5.3	4.9213	15
4.9	4.7046	35