Reduction of Fuzzy Rules and Membership Functions and Its Application to Fuzzy PI and PD Type Controllers

Seema Chopra, Ranajit Mitra, and Vijay Kumar

Abstract: Fuzzy controller's design depends mainly on the rule base and membership functions over the controller's input and output ranges. This paper presents two different approaches to deal with these design issues. A simple and efficient approach; namely, Fuzzy Subtractive Clustering is used to identify the rule base needed to realize Fuzzy PI and PD type controllers. This technique provides a mechanism to obtain the reduced rule set covering the whole input/output space as well as membership functions for each input variable. But it is found that some membership functions projected from different clusters have high degree of similarity. The number of membership functions of each input variable is then reduced using a similarity measure. In this paper, the fuzzy subtractive clustering approach is shown to reduce 49 rules to 8 rules and number of membership functions to 4 and 6 for input variables (error and change in error) maintaining almost the same level of performance. Simulation on a wide range of linear and nonlinear processes is carried out and results are compared with fuzzy PI and PD type controllers without clustering in terms of several performance measures such as peak overshoot, settling time, rise time, integral absolute error (IAE) and integral-of-time multiplied absolute error (ITAE) and in each case the proposed schemes shows an identical performance.

Keywords: Extraction of rules, fuzzy control, fuzzy subtractive clustering, membership functions.

---

1. INTRODUCTION

The applicability of classical control methods have been demonstrated in many control problems in industry, however the ever-increasing demand of flexibility will demand a response which does not change due to parameter variations at all levels of automation. Its simplicity has been the main reason for its wide application in the industry. Since classical controllers are fixed gain feedback controllers they cannot compensate the parameter variation in the plant easily and cannot adapt to changes in the environment. The difficulties that arise in this methodology are broadly classified into three categories. The first is the computational complexity due to mathematical modelling, second is the presence of the non-linear processes with many degrees of freedom and third is uncertainty (presence of noise and load disturbances etc.). The greater the ability to deal with these difficulties, the more intelligent is the control system. Because many living systems do implement some sort of intelligent control, it has been natural to look into computational paradigms used by nature. Fuzzy logic and Artificial Neural Networks represent such a biologically inspired paradigm.

Fuzzy Logic Controllers (FLC) have been introduced and successfully applied. One of the hallmarks of fuzzy logic is that it allows nonlinear input/output relationships to be expressed by a set of qualitative "if – then rules." Nonlinear control and process models may all be expressed in the form of fuzzy rules. Most fuzzy systems are hand crafted by human expert to capture some desired input/output relationships that the expert has in mind. However often an expert cannot express his or her knowledge explicitly and for many applications, an expert may not even exist. Hence there is considerable interest in being able to extract fuzzy rules from experimental input/output data. The motivation for capturing data behavior in the form of fuzzy rules is easy to understand [1]. An expert can check the rules for completeness and fine-tune or extend the system by editing the rule base. Obviously, it is difficult for human experts to examine all the input/output data from complex system to find the number of proper rules for fuzzy system. To cope with this difficulty, much research effort has been devoted to develop

Manuscript received March 4, 2005; revised April 17, 2006; accepted June 1, 2006. Recommended by Editorial Board member Jietae Lee under the direction of Editor Jin Young Choi.

Seema Chopra, Ranajit Mitra, and Vijay Kumar are with the Department of Electronics and Computer Engineering, Indian Institute of Technology, Roorkee, India (e-mails: seemadec, rmtrafec, vijecfec)@iitr.ernet.in).

alternative design methods. Generally, these methods consist of two learning phases, structure learning phase and parameter learning phase. The structure learning phase is employed to decide the structure of fuzzy rules and parameter learning phase is used to tune the coefficients of each rule (like the shape and positions of membership functions). An important task in the structure identification is the partition of the input space, which influences the number of fuzzy rules generated. Recently, methods for extracting fuzzy rules have incorporated clustering techniques. These methods require the user to prespecify the structure of the rule base, i.e., number of rules per class or number of membership functions per input feature, along with initial values for the adjustable parameters.

Clustering is the unsupervised classification of patterns (observations, data item, or feature vectors) into groups (clusters). But fuzzy clustering is also very useful for constructing fuzzy if-then rules from data. The structure of the rules depends on the considered application. For fault diagnosis and other classification tasks the rules aim at deciding to which class in a finite set of classes (like ok/tolerable/faulty) a given datum should be assigned. In system identification or function approximation the rules describe a usually continuous connection between different variables (like in fuzzy control). Clustering algorithms typically require the user to prespecify the number of cluster centers and their initial locations; the locations of the cluster centers are then adapted in a way such that the cluster centers can better represent a set of archetypical data points covering the range of data behavior. The Fuzzy c-Means algorithm (FCM) [2] and Kohonen's Self-Organizing Map [3] method are well-known examples of such clustering algorithms. For these algorithms, the quality of the solution, like that of most nonlinear optimization problems, depends strongly on the choice of initial values (i.e., the number of cluster centers and their initial locations). Pal *et al.* [4] in 1997 survey the use of clustering for identification of various parameters of fuzzy systems. Issues discussed include the proper domain for clustering, the clustering algorithm used, validation of clusters, and system validation.

Kusiak and Chow [5] in 1987 give an efficient clustering algorithm which has relatively low computational time complexity. Cheng *et al.* [6] presents a multistage random sampling fuzzy c-means based clustering algorithm, which is used to create fuzzy rules in the domain of magnetic resonance images where over 60,000 patterns and 3 features of attributes are common. Yang [7] in 1993 presents the survey of fuzzy set theory applied in cluster analysis and gives a survey of fuzzy clustering in three categories. The first category is the fuzzy clustering based on fuzzy relation. The second one is the fuzzy clustering based on objective function. Finally, the author gives an overview of a nonparametric classifier. Runkler and Palm [8] in 1996 develops a regular fuzzy c-elliptotype clustering algorithm for the direct extraction of regular fuzzy systems from measured data. In contrast to the conventional fuzzy c-elliptotype clustering, the modified algorithm identifies clusters located on a regular grid. Regular fuzzy clustering has a low computational complexity and good convergence properties. A new approach [9] to the design of fuzzy systems is presented by Sin and Rui, assuming that the system specification is given in terms of a large number of sample input/output pairs. In this approach, there is no need to guess the number and shapes of fuzzy sets in the input and output universe of discourse, and the number of clusters can be determined by using an appropriate measure of cluster validity.

In 1985, Takagi and Sugeno [10] present a mathematical tool to build a fuzzy model of a system where fuzzy implications and reasoning are used. The premise of an implication is the description of fuzzy subspace of inputs and its consequence is a linear input-output relation. The method of identification of a system using its input-output data is then shown. Hanss [11] in 1999 presents a special fuzzy modeling method for developing multivariable fuzzy model on the basis of measured input and output data. The fuzzy model identification procedure is carried out by applying fuzzy c-elliptotype method, to provide the parameters of the fuzzy model. Gedeon *et al.* [12] in 2002 present a method which extracts rules directly from numerical data for a Sedimentary Rock Data Set. This paper shows how pre-processing input data using clustering may help the classification accuracy in some cases.

Yager and Filev [13] proposed a simple and effective algorithm, called the mountain method, for estimating the number and initial location of cluster centers. Their method is based on gridding the data space and computing a potential value for each grid point. Although this method is simple and effective, the computation grows exponentially with the dimension of the problem. Chiu [14] proposed an extension of Yager and Filev's mountain method, called subtractive clustering, in which each data point, rather than the grid point, is considered as a potential cluster center. Using this method, the number of effective "grid points" to be evaluated is simply equal to the number of data points, independent of the dimension of the problem. Another advantage of this method is that it eliminates the need to specify a grid resolution, in which tradeoffs between accuracy and computational complexity must be considered.

Chiu [15] in 1997 presents methods for extracting fuzzy rules for both function approximation and pattern classification. The rule extraction methods are based on estimating clusters in the data; each cluster obtained corresponds to a fuzzy rule that relates a region in the input space to an output region (or, in the case of pattern classification, to an output class). Chiu [16] again in 1997 presents an efficient method for extracting fuzzy classification rules from high dimensional data. A cluster estimation method called subtractive clustering is used to efficiently extract rules from a high dimensional feature space.

Pal and Mudi [17] used FCM [2] to identify the rule base needed to realize a self-tuning fuzzy PI-type controller and they are able to reduce 49 rules to 17 rules by strategy 1 (data generated by uniform sampling e and Δe). The performance of the identified system is not quite satisfactory. Then they suggested other methods to get the initial estimate of membership functions (MFs) and data generated by running the process in closed loop, called strategy 2, for the improvement in performance. Grabusts [18] aims at modeling the input-output relationship with fuzzy IF-THEN rules by using fuzzy clustering technique. The main difference between fuzzy clustering and other clustering techniques is that it generates fuzzy partitions of the data instead of hard partitions. The author examines two fuzzy-clustering algorithms: FCM and subtractive clustering algorithm.

The Fuzzy c-means (FCM) clustering algorithm, which has been widely studied and applied, needs a priori knowledge of the number of clusters. Whenever FCM requires a desired number of clusters and initial guess positions for each cluster center, the output rules depend strongly on the choice of initial values as the FCM algorithm forms iteratively a suitable cluster pattern in order to minimize an objective function dependent of cluster locations. The auto-generation capability for determining the number and initial location of cluster centers through search techniques was introduced in the mountain clustering method. This method considers each discrete grid point as a potential cluster center by computing a search measure called the mountain function at each grid point. It is a subtractive clustering method with improved computational effort, in which the data points themselves are considered as candidates for cluster centers instead of grid points. By using this method, the computation is simply proportional to the number of data points and independent of the dimension of the problem. In this method, a data point with highest potential which is a function of the distance measure is considered as a cluster center and data points close to new cluster center are penalized in order to control the emergence of new cluster centers. Fuzzy c-means is a supervised algorithm, because it is necessary to tell it how many clusters 'c' to look for. If 'c' is not known before, it is necessary to apply an unsupervised algorithm. Subtractive clustering is based on a measure of the density of data points in the feature space. The idea is to find regions in the feature space with high densities of data points. The point with the highest number of neighbours is selected as centre for a cluster. The data points within a prespecified, fuzzy radius are then removed (subtracted), and the algorithm looks for a new point with the highest number of neighbours. This continues until all data points are examined.

The preceding discussion shows that different researchers have used different clustering algorithms and different cluster validity indices to decide on the number of rules. Our search though the literature revealed that Subtractive Clustering is fast and robust method for estimating the number and location of cluster centers present in a collection of data points. Initial fuzzy rules with rough estimate membership functions are obtained from the cluster centers; the membership functions and other rule parameters are then optimized with respect to some output error criterion.

The problem of the clustering based partition is that corresponding membership functions in each input variable are always opaque to the user, especially in the case of high-input dimensions. This violates the spirit of fuzzy systems that what a fuzzy rule means and how it works should be easy to understand. This problem can be solved by projecting the generated cluster onto each dimension of the input space to form a projected one-dimensional (1-D) membership function for each input variable and represent a cluster by the product of the projected membership functions, as illustrated in Fig. 1 [20].

Compared with the grid-type partition, the clustering-based partition does reduce the number of generated rules, but not the number of membership functions of each input variable as in [23]. To verify this, suppose there are 'n' input variables and each input variable is partitioned into 'm' parts (fuzzy terms). Then the total number of membership functions used is 'nm' for the grid-type partition. As to the clustering-based partition, if there are 'k' clusters formed, then the number of membership functions

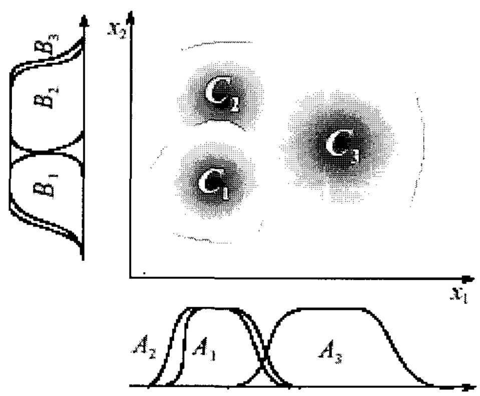


Fig. 1. Clustering-based partitioning.

generated is 'nk'. In general, k is larger than m, meaning that the clustering-based partition creates more membership functions than the grid-type one dose. In fact, by observing the projected membership functions in Fig. 1, it is found in [23] that some membership functions projected from different clusters have high similarity degrees. In this paper, these highly similar membership functions are eliminated. This phenomenon occurs not only in the clustering-based partitioning methods, but also in other approaches like those based on the orthogonal least square [OLS] method [24].

Firstly, Fuzzy Subtractive Clustering (FSC) approach is used to decide the number of rules. After that highly similar membership functions obtained from subtractive clustering are eliminated using similarity measure. Then FSC is used for identification of PD type Fuzzy Logic Controllers (FPDC) and PI type FLC (FPIC). A comparison between the clustering based Fuzzy Logic Controllers and conventional Fuzzy Logic Controllers using simulation of a wide range of linear and nonlinear processes is presented.

2. CLUSTER ESTIMATION

If a cluster tendency assessment technique signals existence of good substructure in the data, then it may be easier to find an "optimal" number of rules. However, irrespective of whether the input–output data has cluster substructure or not, it is always possible to partition it into a number of subsets and each such subset can be converted into a rule. If the data indeed has hyperspherical clusters [1], then the number of rules (subsets) would be smaller compared to the case when the data does not have any cluster substructure. For example, if the input–output relation is linear, the data will not exhibit any cluster structure, yet it can be partitioned into a number of small hyperspherical clusters to generate a set of rules to identify such linear systems.

Consider a collection of n data points $\{x_1, x_2, \dots x_n\}$ in an M dimensional space. Without loss of generality, we assume that the data points have been normalized in each dimension so that they are bounded by a unit hypercube. We consider each data point as a possible cluster center and define a measure of the potential of data point xi as

$$P_i = \sum_{j=1}^{n} e^{-\alpha \|x_i - x_j\|^2}, \tag{1}$$

$$\text{where } \alpha = 4/r_a^2. \tag{2}$$

||.|| denotes the Euclidean distance, and r_a is a positive constant. Thus, the measure of the potential for a data point is a function of its distances to all other data points. A data point with many neighboring data points will have a high potential value. The constant r_a is effectively the radius defining a neighborhood; data points outside this radius have little influence on the potential. After the potential of every data point has been computed, the data point with the highest potential is selected as the first cluster center. Let x_1* be the location of the first cluster center and P_1* be its potential value. The potential of each data point xi is revised by the formula

$$P_i \Leftarrow P_i - P_1 * e^{-\beta \|x_i - x_1^*\|^2}, \tag{3}$$

$$\text{where } \beta = 4/r_b^2, \tag{4}$$

and r_b is a positive constant. Next, from each data point, an amount of potential is subtracted as a function of its distance from the first cluster center. The data points near the first cluster center will have greatly reduced potential, and therefore will be unlikely to be selected as the next cluster center. The constant r_b is effectively the radius defining the neighborhood which will have measurable reductions in potential. To avoid obtaining closely spaced cluster centers, r_b is set to be somewhat greater than r_a; a good choice is r_b = 1.25 r_a. When the potentials of all data points have been revised according to (3), the data point with the highest remaining potential is selected as the second cluster center. The process is then continued further. In general, after the kth cluster center has been obtained, the potential of each data point is revised by the formula

$$P_i \Leftarrow P_i - P_k * e^{-\beta \|x_i - x_k^*\|_2}, \tag{5}$$

where x_k* is the location of the kth cluster center and P_k* is its potential value.

The process of acquiring new cluster center and revising potentials repeats until the remaining potential of all data points falls below some fraction of the potential of the first cluster center P_1*. In addition to this criterion for ending the clustering process are criteria for accepting and rejecting cluster centers that help avoid marginal cluster centers [15].

3. EXTRACTION OF RULES AND MEMBERSHIP FUNCTIONS

To extract the rules, firstly data is separated into groups according to their respective classes. Subtractive clustering is then applied to the input space of each group of data individually for identifying each class of data [16]. The clusters found in the data of a given group identify regions in the input space that map into the associated class. Hence, each cluster center may be translated into a fuzzy rule

for identifying the class. For example, if subtractive clustering was applied to the group of data for class and cluster center x_i* was found in the group of data for class c1, then cluster center provides the rule:
Rule i: If {x is near x_i*} then class is c1.

The degree of fulfillment of {x is near x_i*} is defined as

$$\mu_i = e^{-\alpha\left\|x-x_i^*\right\|_2}, \tag{6}$$

where α is the constant defined by (2).

One can also write this rule in the more familiar form:
Rule i: If X_1 is A_{i1} & X_2 is A_{i2} &... then class is c1, where X_j is the j'th input feature and A_{ij} is the membership function (Gaussian type) in the i'th rule associated with the j'th input feature.
The membership function A_{ij} is given by

$$A_{ij}(X_j) = \exp\left\{-\frac{1}{2}\left(\frac{X_j - x_{ij}^*}{\sigma^{ij}}\right)^2\right\}, \tag{7}$$

where x_{ij}* is the j'th element of x_i*, and

$$\sigma_{ij}^2 = 1/(2\alpha). \tag{8}$$

The degree of fulfillment of each rule is computed by using multiplication as the AND operator. By applying subtractive clustering to each class of data individually, a set of rules may be obtained for identifying each class. The individual sets of rules can then be combined to form the rule base of the classifier. When performing classification, the output class of the classifier is simply determined by the rule with the highest degree of fulfillment.

4. IMPLEMENTATION AND RESULTS

The FSC is used for identification of PD type FLC and PI type FLC. But for Fuzzy Controllers, parameter settings is necessary to determine universe ranges and perform hundreds of simulation experiments until acceptable values are not found. A retrieval of optimal parameter is very difficult, because the setting is dependent on lot of other parameters and desired value. One method with the unified universe range, stated in [19], considerably simplifies the setting of fuzzy PI/PD/PID controller. For the sake of completeness, a brief description of parameter setting for fuzzy PI type controller is given.

4.1. Fuzzy PI controller design

Fuzzy PID controllers are physically related to classical PID controller. A classical PI controller is described by (9) where K is the gain of PI controller, T_I is an integral constant, $e(t)$ is an error signal, $e(t) = r(t)-y(t)$, $r(t)$ is the desired value, $y(t)$ is the output from process and $u(t)$ is the output from controller.

$$u(t) = K(e(t) + \frac{1}{T_I}\int_0^t e(\tau)d\tau \tag{9}$$

When we derive (9) we get

$$u'(t) = K\left(e'(t) + \frac{1}{T_I}e(t)\right). \tag{10}$$

For a local extreme location we put

$$u'(t) = K\left(e'(t) + \frac{1}{T_I}e(t)\right) = 0, \tag{11}$$

$$e'(t) = -\frac{1}{T_I}e(t). \tag{12}$$

If we translate (11) to discrete form, we get the equation for action value change of discrete PI controller

$$\Delta u(k) = K\left(\Delta e(k) + \frac{1}{T_I}e(k)\right), \tag{13}$$

where $\Delta u(k) = (u(k) - u(k-1))/T$,
$\Delta e(k) = (e(k) - e(k-1))/T$.

T is the sampling period, k is the step. Equation (13) can also be written as

$$\Delta u(k) = K\frac{1}{T_I}\left(T_I\Delta e(k) + e(k)\right). \tag{14}$$

In next step it is necessary to map the rule base to the discrete state space $\Delta e(k)$, $e(k)$. We define the scale factor M for the universe range, M>0. This scale factor sets the universe ranges for the error and its first differences.
After extending (14) it becomes

$$\Delta u(k) = K\frac{M}{T_I}\left(\frac{T_I}{M}\Delta e(k) + \frac{1}{M}e(k)\right). \tag{15}$$

The placement of the base rules mapped into the state plane according to (15) is determined only by the chosen scale factor and the integral constant magnitude. Multiplication of the normalized universe is inversely proportional; therefore the values on axis are inverted up against (15).

Then apply fuzzification to input variables and after defuzzification,

$$\Delta u(k) = K\frac{M}{T_I}\boldsymbol{D}\left\{\boldsymbol{F}\left\{\frac{T_I}{M}\Delta e(k) + \frac{1}{M}e(k)\right\}\right\}, \tag{16}$$

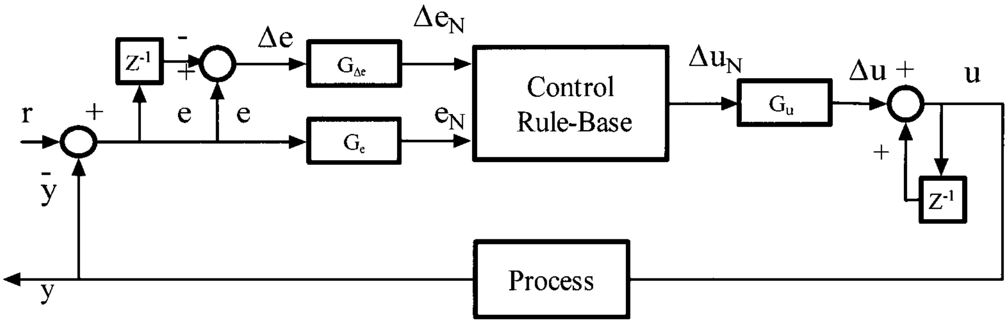


Fig. 2. Block diagram of FPIC.

where $\boldsymbol{F}$ is an operation for fuzzification and $\boldsymbol{D}$ for defuzzification.

For $\Delta u(k)$

$$\Delta u(k) = \frac{u(k)-u(k-1)}{T} = K\frac{M}{T_I}\boldsymbol{D}\left\{\boldsymbol{F}\left\{\frac{T_I}{M}\Delta e(k)+\frac{1}{M}e(k)\right\}\right\}. \tag{17}$$

The output of the fuzzy PI controller in the step k is then

$$u(k) = \frac{KMT}{T_I}\boldsymbol{D}\left\{\boldsymbol{F}\left\{\frac{T_I}{M}\Delta e(k)+\frac{1}{M}e(k)\right\}\right\}+u(k-1). \tag{18}$$

The block diagram of PI type FLC is shown in Fig. 2. The change in error is defined as

$$\Delta e(k) = e(k)-e(k-1), \tag{19}$$

where $e(k)$ is the error at the kth sample.

All membership functions (MFs) for controller inputs (i.e., e and Δe) and incremental change in controller output (i.e., Δu) are defined on the common normalized domain [-1,1]. The membership functions are shown in Fig. 3.

Here the input and output gains are G_e, $G_{\Delta e}$ and G_u.

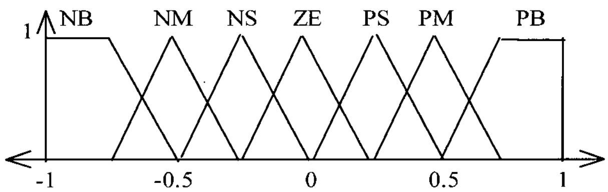
$$G_e = \frac{1}{M},\ G_{\Delta e} = \frac{T_I}{M},\ G_u = \frac{KMT}{T_I}. \tag{20}$$

The operation of PI type FLC can be described by

$$u(k) = u(k-1)+\Delta u(k). \tag{21}$$

In (21), Δu is the incremental change in controller output, which is determined by the rules of the form If e is E and Δe is ΔE, then Δu is ΔU. The rule base for computing Δu is shown in Fig. 4, which is a fairly standard one.

On the other hand, if the output of the FLC is u (not Δu) and there is no accumulation of controller output then fig is converted to a PD type FLC. In this case, the input and output gains G_e, $G_{\Delta e}$ and G_u are:



NB-Negative Big, NM-Negative Medium, NS-Negative Small, ZE-Zero Error, PS-Positive Small, PM-Positive Medium, PB-Positive Big

Fig. 3. MFs for e, Δe and Δu.

Δe/e	**NB**	**NM**	**NS**	**ZE**	**PS**	**PM**	**PB**
NM	NB	NB	NB	NM	NS	NS	ZE
NM	NB	NM	NM	NM	NS	ZE	PS
NS	NB	NM	NS	NS	ZE	PS	PM
ZE	NB	NM	NS	ZE	PS	PM	PB
PS	NM	NS	ZE	PS	PS	PM	PB
PM	NS	ZE	PS	PM	PM	PM	PB
PB	ZE	PS	PS	PM	PB	PB	PB

Fig. 4. Rule base.

$$G_e = \frac{1}{M},\ G_{\Delta e} = \frac{T_D}{M},\ G_u = KM, \tag{22}$$

where T_D is an derivative constant.

In this paper, PI and PD type FLC's (system with 49 rules) will be denoted by FPIC and FPDC, respectively, and their corresponding clustering based FLC's (system with reduced rule set) will be denoted by TFPIC and TFPDC.

4.2. Identification of fuzzy controllers

The FPIC in Fig. 2 use 49 rules and 7 membership functions in each variable to compute output, and exhibits good performance [21]. Next, we investigate the following – Given some data describing the output (Δu) as a function of Inputs (i.e., e and Δe), now main aim is to extract a smaller set of rules using FSC approach and then reduce membership functions to do the same. Then, the performance of the simple controller (identified system) compare with the original one. Now the following steps are followed.

4.2.1 Data generation

To identify the FPIC and FPDC, some data is needed, i.e., a set of two-dimensional input vectors $X=\{X_1, X_2, \ldots X_n\}$ and the associated set of one-dimensional output vectors as $Y=\{Y_1, \ldots Y_n\}$ where $X=\{e$ and $\Delta e\}$ and $Y=\{u\}$ is required. Here, the data has been generated by sampling input variables e and Δe uniformly and computing the value of $\{u\}$ for each sampled point. The number of data points generated is 442.

4.2.2 Rule extraction and membership functions

After generating the data, the next step is to estimate the rules. Although the number of rules (clusters) is automatically determined by this method, the user-specified parameter r_a (the radius of influence of cluster center) strongly affects the number of rules that will be generated. A large r_a generally results in fewer rules, while a small r_a can produce excessive number of rules. Thus r_a is an approximate specification. In this case data dimension is 3 (e.g., X has 2 columns and Y has 1 column). Here the radius of influence in the first data dimension is half the width of the data space and the range of influence in the second data dimension is one quarter the width of the data space and so on [21,25].

Then after applying Subtractive Clustering algorithm, eight clusters (rules) are extracted and eight MFs are formed. But using the similarity measure, the number of membership functions is reduced to 4 of input 1 'error (e)' and 6 of input 2 'Change in error (Δe)'. The membership functions of e and Δe after reduction are shown in Fig. 5.

4.2.3 Results

The FSC approach has been tested on a variety of linear and nonlinear processes, Type 0 and Type 1, of orders from 1 to 3, with different values of dead time (L). The objective here is to justify whether the system after clustering (with less no. of rules and membership functions can provide the same level of performance as that of the original one (system with 49 rules). This has been tested for the processes referred in [21,22,26] and observed satisfactory results in each case except in some systems using FPIC where after adjusting the values of gains (when all rules are fired), then it is observed that the performance of both the systems is close only on particular gains. However, four of them are reported here.

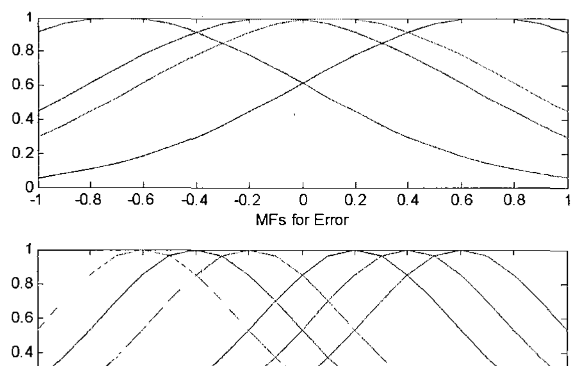


Fig. 5. MFs for e and Δe.

Table 1. Performance analysis for system 1 and 2.

System	FLC	%os	Ts	tr	ITAE	IAE
1	FPDC	16.0	5.7	2.7	17	10
	TFPDC	17.0	5.6	2.7	18	10
2	FPIC	-	71	56	704	32
	TFPIC	-	72	56	747	33

Table 2. Performance analysis for system 3.

System	FLC	%os	ts	tr	ITAE	IAE
3	FPDC	-	1.3	0.86	27	9
	TFPDC	-	1.35	0.95	28	9
3	FPIC	-	12	2.7	27	19
	TFPIC	-	13	2.7	28	19

Table 3. Performance analysis for system 4.

System	FLC	%os	ts	tr	ITAE	IAE
4	FPDC	15.2	4.2	2.1	907	235
	TFPDC	19.8	4.2	2.1	887	231
4	FPIC	1.5	8	7.5	27	19
	TFPIC	3	8	7.5	28	19

The process transfer functions $G(s)$ are reproduced as

$$G_1(s)=e^{-Ls}/s\,(s+1), \tag{23}$$

$$G_2(s)=(s+1)/(s^3+9s^2+26s+24), \tag{24}$$

$$G_3(s)=1/(s+1), \tag{25}$$

$$\ddot{y}+\dot{y}+0.25y^2=u(t-L). \tag{26}$$

The FPDC and FPIC as in Fig. 2 are used here with values of gains (G_e, $G_{\Delta e}$ and G_u) as 1 in almost all cases except those systems in which all rules are not fired (e and Δe are out of range) during simulation . In those systems, input and output gains are calculated from (20) and (22) as described in Section 4 (A). In subsequent discussion, the performance of TFPIC and TFPDC is considered good or satisfactory only when its performance is close to that of FPIC and FPDC. In this paper, it is emphasize that an identified system is called satisfactory only with respect to its closeness to the target system, here FPDC and FPIC. Response characteristics for all systems with and without clustering (with and without membership and rule reduction) are shown in Figs. 6 to 11. A number of performance indices such as peak overshoot (% os), settling time ($t_s(s)$) for ± 5% tolerance band, rise time

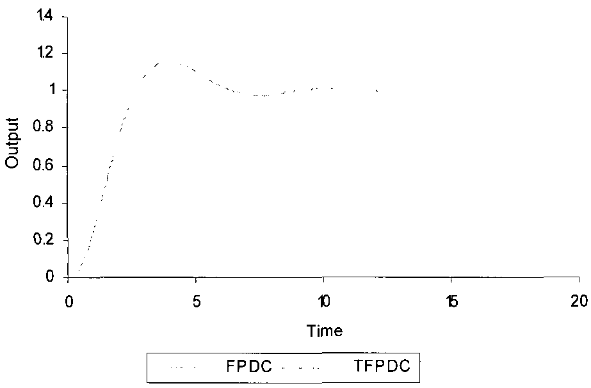


Fig. 6. Response of $G_1(s)= e^{-0.1}s/s(s+1)$ with FPDC.

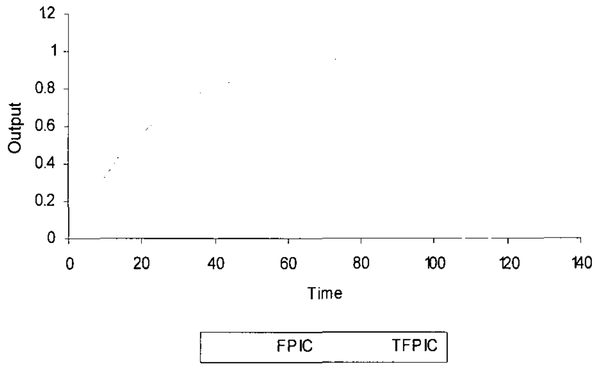


Fig. 7. Response of $G_2(s)=(s+1)/(s^3+9s^2+26s+24)$ with FPIC.

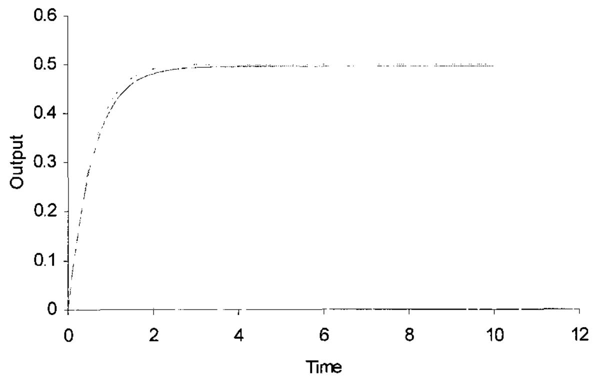


Fig. 8. Response of $G_3(s)= 1/(s+1)$ with FPDC.

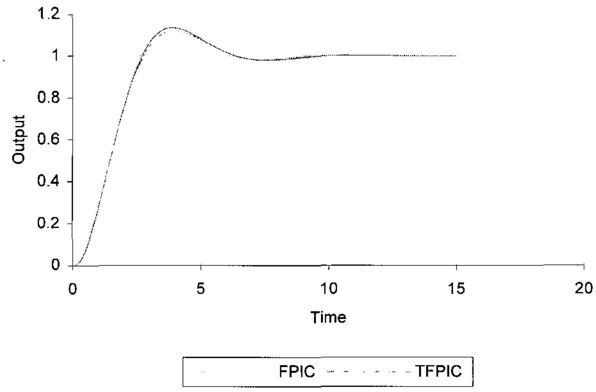


Fig. 9. Response of $G_3(s)= 1/(s+1)$ with FPIC.

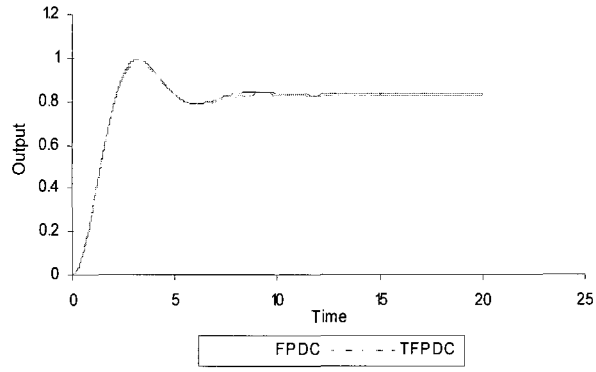


Fig. 10. Response of nonlinear system with FPDC.

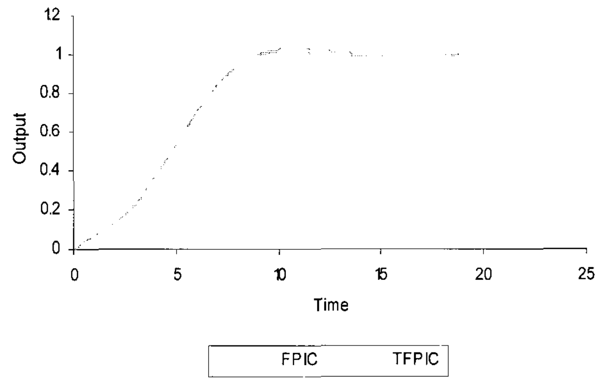


Fig. 11. Response of nonlinear system with FPIC.

($t_r(s)$), integral-absolute error (IAE), and integral of time-multiplied absolute error (ITAE) are computed [27,28] for a detailed performance comparison of the identified system and the original systems. These performance indices for both processes are provided in tabular forms (Table 1, 2, and 3). In each table, row corresponding to FPIC and FPDC presents the performance of the original system.

To summarize, when the data set is generated by the FLC with 49 rules, it exhibits cluster structure. Rules are then generated using the approach FSC which gives 8 rules and 8 membership functions. Then highly similar membership functions are eliminated. It reduces the MFs of error to 4 and MFs of change in error to 6. The overall performance of the clustering based Fuzzy Logic Controllers is compared with those of conventional Fuzzy Logic Controllers. Response characteristics of the identified system in both cases (FPDC and FPIC) are very close to the original one.

5. CONCLUSION

This paper presents two different approaches to deal with the most important design issues i.e., number of rules, number of membership function of Fuzzy Controllers. The FSC approach has been used to extract a rule base for the output 'u' of a FPIC and FPDC. This method is fast for estimating the number and location of cluster centers present in a collection of data points. After that highly similar membership functions obtained from subtractive clustering are eliminated using similarity measure. The proposed combination is able to reduce 49 rules to 8 rules and the number of MFs to 4 and 6 for error and change in error maintaining almost the same level of performance. The main advantage of the proposed approach is that by reducing the rules and membership functions, it can significantly reduce the time and effort needed to design a fuzzy controller directly from numerical data.

REFERENCES

[1] F. Klawonn and R. Kruse, "Constructing a fuzzy controller from data," *Fuzzy Sets and Systems,* vol. 85, pp. 177-193, 1997.

[2] J. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithm,* Plenum Press, New York, 1981.

[3] T. Kohonen, "The self-organizing map," *Proc. of IEEE*, vol. 78, no. 9, pp. 1464-1480, September 1990.

[4] R. Nikhil, K. Pal, J. C. Bezdek, and T. A. Runkler, "Some issues in system identification using clustering," *Proc. of International Conference on Neural Networks*, vol. 4, pp. 2524-2529, 1997.

[5] A. Kusiak and W. S. Chow, "An efficient cluster identification algorithm," *IEEE Trans. on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. SMC-17, no. 4, pp. 696-699, Jul.-Aug. 1987.

[6] T. B. Cheng, D. B. Goldgof, and L. O. Hall, *Fast clustering with application to fuzzy rule generation.* Available: http://www.morden.csee.usf.edu/~hall/mrfcm.pdf

[7] M. S. Yang, "A survey of fuzzy clustering," *Math. Comput. Modeling,* vol. 18, pp. 177-200, 1993.

[8] T. A. Runkler and R. H. Palm, "Identification of nonlinear systems using regular fuzzy c-elliptotype clustering," *Proc. of the Fifth IEEE International Conference on Fuzzy Systems*, vol. 2, pp. 1026-1030, 1996.

[9] S. K. Sin and Rui J. P. DeFigueiredo, "Fuzzy system design through fuzzy clustering and optimal pre defuzzification," *Proc. of IEEE International Conference on Fuzzy Systems, San Francisco*, pp. 190-195, 1993.

[10] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its application to modeling and control," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. SMC-15, no. 1, pp. 116-132, 1985.

[11] M. Hanss, "Identification of enhanced fuzzy models with special membership functions and rule base," *Engineering Application of Artificial Intelligence*, vol. 12, pp. 309-319, 1999.

[12] T. D. Gedeon, H. Kuo, and P. M. Wong, "Rule extraction using fuzzy clustering for a sedimentary rock data set," *International Journal of Fuzzy System*, vol. 4, no. 1, pp. 600-605, March 2002.

[13] R. R. Yager and D. P. Filev, "Generation of fuzzy rules by mountain clustering," *Journal of Intelligent and Fuzzy System*, vol. 2, pp. 209-219, 1994.

[14] S. L. Chiu, "Fuzzy model identification based on cluster estimation," *Journal of Intelligent and Fuzzy System*, vol. 2, pp. 267-278, 1994.

[15] S. L. Chiu, "Extracting fuzzy rules from data for function approximation and pattern classification," to appear as Chapter 9 in *Fuzzy Set Methods in Information Engineering: A Guided Tour of Applications*, D. Dubois, H. Prade, and R. Yager, ed., John Wiley, 1997.

[16] S. L. Chiu, "An efficient method for extracting fuzzy classification rules from high dimensional data," *J. Advanced Computational Intelligence*, vol. 1, no. 1, pp. 31-36, 1997.

[17] K. Pal, R. K. Mudi, and N. R. Pal, "A new scheme for fuzzy rule-based system identification and its application to self-tuning fuzzy controllers," *IEEE Trans. on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 32, no. 4, pp. 470-482, Aug. 2002.

[18] P. Grabusts, "Clustering methods in neuro-fuzzy modeling." Available: http://www.dssg.cs.rtu.lv/download/publications/2002/Garbusts-RA-2002.pdf

[19] P. Pivonka, "Comparative analysis of fuzzy PI/PD/PID controller based on classical PID controller approach." Available: http://www.feec vutbr.cz/~pivonka

[20] C.-F. Juang and C.-T. Lin, "An on-line self-constructing neural fuzzy inference network and its applications," *IEEE Trans. on Fuzzy System*, vol. 6, no. 2, pp. 13-31, 1999.

[21] R. K. Mudi and N. R. Pal, "A robust self-tuning scheme for PI and PD type fuzzy controllers," *IEEE Trans. on Fuzzy System.*, vol. 7, no. 1, pp. 2-16, 1999.

[22] J. Carvajal, G. Chen, and H. Ogmen, "Fuzzy PID controller: Design, performance evaluation, and stability analysis," *Information Sciences*, vol. 123, pp. 249- 270, 2000.

[23] S. Chopra, R. Mitra, and V. Kumar, "Identification of rules using subtractive clustering with application to fuzzy controllers," *Proc. of the Third International Conference on Machine Learning and Cybernetics*, pp. 4125-4131, 2004.

[24] M. Setnes, "Supervised fuzzy clustering for rule

extraction," *Proc. of the IEEE International Conference on Fuzzy System*, pp. 1270-1274, 1999.

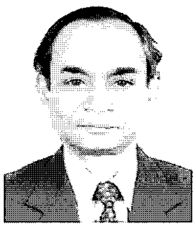
[25] D. Driankov, H. Hellendorn, and M. Reinfrank, *An Introduction to Fuzzy Control*, Springer-Verlag, New York, 1993.

[26] S. Chopra, R. Mitra, and V. Kumar, "Identification of self-tuning fuzzy PI type controllers with reduced rule set," *Proc. of the IEEE International Conference on Networking, Sensing and Control*, pp. 537-542, March 2005.

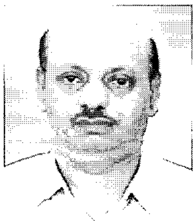
[27] K. Ogata, *Modern Control Engineering*, Prentice-Hall, Englewood Cliffs, NJ, 1970.

[28] M. Gopal, *Control Systems Principles and Design*, Tata McGraw-Hill, India, 1993.

Seema Chopra was born in Punjab, India in 1976. She received B. Tech. degree in Instrumentation and Control Engineering from Kurukshetra University in 2000, M. Tech. degree in Control Engineering from NIT Kurukshetra in 2002. Currently she is doing Ph.D. in Control and Guidance from the department of Electronics and Computer Engineering, Indian Institute of Technology, Roorkee, India under MHRD scheme. She is working as Lecturer in Instrument and Control Department in Haryana Engineering College Jagadhri, India and Centre of Advance Technology, Research Centre, Jagdhri, India since 2001. Her current area of research includes Intelligent Control, Fuzzy Logic, Neural Networks, Fuzzy Clustering and SCADA System.



Ranajit Mitra was born in Allahabad, India in 1945. He received the B.Sc. degree from Luckhnow University, M. Sc. (Tech.) from University of Allahabad, M.Tech. from Indian Institute of Technology, Kharagpur and Ph.D. degree in Control and Guidance from, University of Roorkee, India in 1974. Besides Post Doctoral work at DFVRL laboratory in West Germany, he had been in the Faculty of Department of System and Control University of Technology Baghdad and Thapar Engineering College, Patiala. For the Past 36 years Dr. Mitra is with Department of Electronics and Computer Engineering, Indian Institute of Technology, Roorkee, India (formerly University of Roorkee). His current area of research includes Fuzzy Control, development of new experimental systems in the area of Control and Instrumentation.



Vijay Kumar was born in Biharshariff, India in 1954. He received the B.E. degree from M.M.M. College of Engineering Gorakhpur, M.E. and Ph.D. degree in Control and Guidance from, Univer-sity of Roorkee, India in 1983 and 2001 respectively. He was a Lecturer in Department of Electronics and Computer Engineering, Indian Institute of Technology, Roorkee, India (formerly University of Roorkee) during 1983 to 2002, where currently he is working as Assistant Professor. His current area of research includes Fuzzy Control, Model Order Reductions.