Designing a Fuzzy Logic Controller with a Non-Parametric Similarity-Based Clustering Algorithm

Essam Natsheh
Dep. of Informatics Engineering, College of Engineering, AMA International University, Bahrain.
dr_natsheh@hotmail.com

***Abstract*—Clustering algorithms are used to produce powerful extension to design a three-term (proportional plus integral plus derivative—PID) fuzzy logic controller(FLC). They can be used to eliminate the presumption of the existence of expert information and extract rules that can satisfactorily represent the systems. In this paper, a non-parametric clustering algorithm based on data similarity, which is free of user-defined parameters is proposed. This algorithm is simple and fast. For comparison purposes, two methods of extracting the rules for a three-term FLC from the generated clusters were presented. These two methods entail on the use of the linguistic-type model and the Takagi-Sugeno-Kang (TSK)-type model. Two applications representing second-order systems and third-order systems were used to analyze the performance of the proposed design methods and compared with other design methods. The analysis shows that the proposed design methods are efficient and superior to other design methods with respect to transient response, accuracy, and robustness to variation of defuzzification methods.**

Index Terms*—Clustering Algorithms; Fuzzy Logic Controller; PID Controller; Three-Term Controller.

I. Introduction

Three-term (proportional plus integral plus derivative—PID) fuzzy logic controller (FLC), or simply PID-like FLC, algorithms have been and continued to be a very active and fruitful research field since Mamdani and Assilian's pioneering work in 1974. In [1], the conventional and normal fuzzy logic PID controllers were compared. As there are many methods of designing PID-like FLCs, a performance comparison was carried out with some methods in [2].

The objective in this paper is to develop a new clustering algorithm that is free of user-defined parameters, called similarity measure-based algorithm (SMBA). The purpose of this algorithm is to eliminate the effect of user-defined parameters on the number of generated clusters. The generated clusters will be used to extract the rule-base for the three-term FLC. The proposed method is based on Gowda and Diday's similarity algorithm. Gowda and Diday [3], [4] proposed a simple and effective non-parametric algorithm for clustering symbolic data. Their method is built on a novel similarity measure, which is based on the "position," "span," and "content" of symbolic data. Although this measure is proposed for clustering symbolic data, in this study, it is modified to be used for clustering numerical data. Only the "position" component of this measure is used as a measure of similarity. Other modifications to the algorithm are proposed to develop the rules of the three-term FLC. The rule-base could be the linguistic-type model or the TSK-type model. In general, the proposed design method can be used in the design of any kind of FLC (PI-like FLC, PD-like FLC, etc.).

The remainder of this paper is organized as follows: Section II describes the novel of non-parametric similarity-based clustering algorithm. Section III proposes two methods to generate the rule-bases for the three-term FLC. These m ethods are built on the linguistic-type model and the TSK-type model. Section IV compares the performance of the proposed design method and of other design methods using a second-order armature-controlled DC motor and a third-order field-controlled DC motor as case studies. Finally, section V presents the conclusions of the proposed work.

II. Designing a PID-Like FLC With Non Parametric Similarity-Based Clustering

In the previous study [5], various clustering algorithms that could be used to design a PID-like FLC were discussed. These algorithms require the determination of some parameters that affects the number of clusters to be generated. In the next subsections, an algorithm based on Gowda and Diday's similarity measure is proposed. It is non-parametric, hierarchical, and agglomerative in nature, and its criterion is to merge the most similar mutual pair at each step.

A. Composite Objects

Successive merging lies at the heart of agglomerative clustering methods. Merging is the process of gathering together, on the basis of the similarity measure, two samples and assigning them to the same-cluster membership or a label for further clustering [3], [4]. For example, let A and B be two objects; Gowda and Diday proposed to compose these two objects by considering their minimum interval, which includes both A and B. Due to the non-interval data, the composite object O, resulting from the merging of A and B is proposed as:

$$O = \text{mean}\,(A, B) \tag{1}$$

B. Mutual Pair

Gowda and Krishna [6], [7] introduced the concept of mutual nearest neighborhood and successfully used it for agglomerative and disaggregate clustering, learning, condensed nearest-neighbor rule, editing, and error correction.

In a data set, on the basis of a similarity or dissimilarity measure, if an object Xi is the first nearest neighbor of an object Xj, and Xj is the first nearest neighbor of Xi, then Xi and Xj constitute a mutual pair. The mutual pair with the highest similarity value corresponds to the two objects of the data set having the highest similarity, while the mutual pair with the lowest dissimilarity value corresponds to the two objects of the data set having the lowest dissimilarity.

C. Dissimilarity and Similarity Measures

The dissimilarity and similarity measures indicate the relative positions of two point values on a real line. The dissimilarity D between two points A and B is defined as Equation (1), in which the ‖.‖ as any inner product norm metric is chosen as the Euclidean norm.

$$D(A,B) = ||A\text{-}B|| \tag{2}$$

Thus, it is represented by Equation (3) as follows:

$$D(A,B) = \sqrt{\sum_{i=1}^{n}(A_i - B_i)^2} \tag{3}$$

The values of the dissimilarity D are normalized within the range between 0 and 1 by the following linear transformation:

$$D' = \frac{D - D_{min}}{D_{max} - D_{min}} \tag{4}$$

where: D_{min} = Minimum dissimilarity value
D_{max} = Maximum dissimilarity value

Similarity is just another aspect of dissimilarity, which leads to the view that the more similar the two objects, the less dissimilar they are. On this basis, the similarity between the two points A and B is defined as follows:

$$S(A,B) = 1 - D'(A,B) \tag{5}$$

D. Similarity Measure-Based Algorithm (SMBA)

The similarity measure-based algorithm (SMBA) proceeds as follows [3], [4]:

1. Let {$X1$, $X2$, …, XN} be a set of N objects. Let the initial number of clusters be N, with each cluster having a cluster weight (number of objects) of 1.
2. Compute the weighted similarities S_w or weighted dissimilarities D_w between all points in the data set, as follows:

$$S_w(X_i, X_j) = S(X_i, X_j).\sqrt{\frac{n_i.n_j}{n_i + n_j}} \tag{6}$$

$$D_w(X_i, X_j) = D(X_i, X_j).\sqrt{\frac{n_i.n_j}{n_i + n_j}} \tag{7}$$

where: n_i = Cluster weights of X_i
n_j = Cluster weights of X_j
$S(X_i, X_j)$ = Similarity value is given by (5)
$D(X_i, X_j)$ = Dissimilarity value is given by (3)

Determine the mutual pair with the highest weighted similarity or the lowest weighted dissimilarity and form a composite object by merging the individuals of this pair. Reduce the number of clusters by 1.

3. Repeat step 2 until the number of clusters is equal to 1.

4. Repeat step 2 until the number of clusters is equal to 1.
5. Calculate the cluster indicator (CI) value using equation (8). The stage of maximum CI value indicates the number of clusters in the data. The composite objects of the stage describe the objects representing the clusters.

At each stage (p), maximum similarity (S_m) indicates the similarities between the individual objects of the mutual pair that are combined to form a composite object. The minimum similarity value at each stage indicates the similarity between the most distant clusters. The lower this value, the better the separation between the clusters [3].

The CI value at the pth stage is:

$$CI = \frac{|S_m \text{ at } (p+1) - S_m \text{ at } (p)|}{S_m \text{ at } (p)} \tag{8}$$

III. Developing The Rule-Base Of The Three-Term FLC Using Clustering Algorithms

Fuzzy system models basically fall into two categories, which differ fundamentally in their ability to represent different types of information [8]. The first includes linguistic models (LMs) that are based on collections of IF–THEN rules with vague predicates and use fuzzy reasoning. The second category of fuzzy models is based on the Takagi-Sugeno-Kang (TSK) method of reasoning [9]. These models are formed by logical rules that have a fuzzy antecedent part and a functional consequent; essentially, they are combination of fuzzy and non-fuzzy models. This section is a discussion of how both kinds of models can be applied in the design of a three-term FLC.

A. Linguistic Models as Tools for FLC Representation

One of the main directions in the theory of fuzzy systems is the linguistic approach, based on linguistically described models. Starting with the observed data pairs (e_k, de_k, se_k, U_k), the clustering methods provide a collection of clusters and their centers (e_{Ci}, de_{Ci}, se_{Ci}, U_{Ci}). Each center can be viewed as a prototypical fuzzy point in the relationship between input and output, as shown in Figure 1. Hence, each cluster center can be used as the basis of a rule that describes the system behavior.

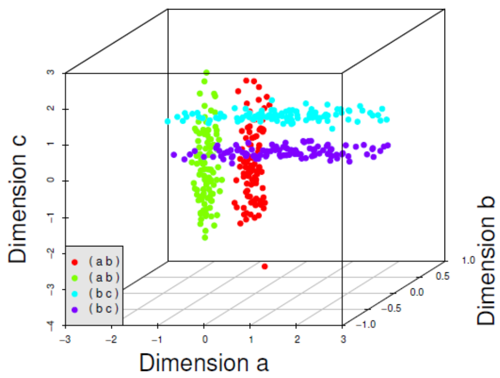


Figure 1: Three-dimensional input–output data clustering for rule determination

It is proposed that the rules in the PID-like FLC rule-based are formed linguistically as follows:

IF error is close to *cluster i*
AND error-change is close to *cluster i*
AND error-sum is close to *cluster i*,
THEN control action is close to *cluster i* (9)

(For $i = 1 \ldots C$)

where: C = Number of clusters

The cluster centers (*eCi*, *deCi*, *seCi*, *UCi*) of each fuzzy variable are considered to be the peaks of their membership functions.

To achieve a systematic method of defining the membership functions of the antecedent fuzzy sets, the use of the Gaussian curve membership function is proposed:

$$\mu_i(y) = exp\left(\frac{-(y - y_i^*)^2}{2\sigma^2}\right) \tag{10}$$

where: y = Input vector
y^*_i = Center of the cluster i
σ = Width of the cluster i

To compute the initial value of σ for the clusters in the fuzzy variable X (*e*, *de*, *se*, and *U*), the following equation is proposed:

$$\sigma = \frac{\text{Maximum point in } X - \text{Minimum point in } X}{\text{Number of clusters}} \tag{11}$$

To refine the σ value for each variable, the algorithm 1 is proposed:

Algorithm 1
Optimizing σ Values

```
Input: initial σ value for each variable
Output: optimized σ value for each variable
  Use eq. (11) to get the initial σ value to each variable;
  Initialize SSE (sum-squared error) to large value;
  For i=1 to maximum number of epochs to refine all σ
    If SSE < sse_goal, break, end if
     For j=1 to minimum no. of epochs to refinement one σ
        Run the experiment and get new_sse;
        If (new_sse ≤ SSE)
           SSE = new_sse;
           Save σ;
           σ = σ × increase_ratio;
        else
          σ = σ × decrease_ratio;
       end if
    end for
  end for
```

The use of a decrease ratio of 0.7 and an increase ratio of 1.05 is suggested.

B. TSK Models as Tools for PID FLC Representation

A known disadvantage of the LMs is that they do not contain in an explicit form of the objective knowledge of the system if such knowledge cannot be expressed and/or incorporated into the fuzzy set framework. A definition of the TSK rules in the PID-like FLC rule-based is proposed as follows:

IF error (y_1) is close to *cluster i*
AND error-change (y_2) is close to *cluster i*
AND error-sum (y_3) is close to *cluster i*,
THEN control action is $a_{i1}y_1 + a_{i2}y_2 + a_{i3}y_3 + a_{i4}$ (12)

(For $i = 1 \ldots C$)

This subsection contains a description of Chiu's method [10], [11] for fuzzy model identification. His method for fuzzy model identification from data is based on the use of a cluster estimation method to determine the number of rules and initial rule parameters and then on the application of optimization algorithms to tune the rule parameters. Chiu uses a *recursive least squares* estimation algorithm to optimize the fuzzy model. The issue with this algorithm is that it is iterative and slow. To optimize the fuzzy model, the use of a *singular value decomposition* method is proposed. This is a non-iterative algorithm capable of obtaining the parameter estimates very quickly and reliably.

i. Fuzzy Model Identification

Consider a set of C cluster centers $\{c_1, c_2, ..., c_C\}$ in an M dimensional space. Let the first N dimensions correspond to input variables and the last M-N dimensions correspond to output variables. Each vector c_i is decomposed into two components y^*_i and z^*_i, where y^*_i contains the first N elements of c_i (i.e., the coordinates of the cluster center in input space) and z^*_i contains the last M-N elements (i.e., the coordinates of the cluster center in output space). Given an input vector y, the degree to which rule i is fulfilled is defined as [10], [11]:

$$\tau_i = \exp(-\|y - y_i^*\|^2) \tag{13}$$

The output vector z can be computed via:

$$z = \frac{\sum_{i=1}^{C} \tau_i z_i^*}{\sum_{i=1}^{C} \tau_i} \tag{14}$$

To systematically define the membership functions of the antecedent fuzzy sets, the use of the Gaussian curve membership function as defined by (10) is proposed. Equation (11) should be used to compute the σ value for the Gaussian curves.

ii. Optimizing the Fuzzy Model

Equations (13) and (14) provide a simple and direct way to translate a set of cluster centers into a fuzzy model. Equation (14) can be used to optimize the rules by allowing z^*_i to be a linear function of the input variables, instead of a simple constant. That is:

$$z^*_i = G_i\, y + h_i \tag{15}$$

where: $G_i = (M\text{-}N) \times N$ constant matrix
h_i = Constant column vector with M-N elements

Expressing z^*_i as a linear function of the input allows a significant degree of rule optimization to be performed without adding much computational complexity. As pointed out by Takagi and Sugeno [9], given a set of rules with fixed premises, the optimization of the parameters in the consequent equations with respect to training data is reduced to a *least-squares* estimation problem. Such problems can be solved easily and the solution is always globally optimal. To convert the equation parameter-optimization problem into the least-squares estimation problem, let us define:

$$\rho_i = \frac{\tau_i}{\sum_{j=1}^{C} \tau_j} \tag{16}$$

Equation (14) can then be rewritten as equations (17) or (18):

$$z = \sum_{i=1}^{C} \rho_i \, z_i^* = \sum_{i=1}^{C} \rho_i \,(G_i y + h_i) \tag{17}$$

$$z^T = [\rho_1 y^T \;\; \rho_1 \;\; \dots \; \rho_C y^T \;\; \rho_C] \begin{bmatrix} G_1^T \\ h_1^T \\ \vdots \\ G_C^T \\ h_C^T \end{bmatrix} \tag{18}$$

where: z^T = Row vector
y^T = Row vectors

Given a collection of n input data points $\{y_1, y_2, ..., y_n\}$, the resultant collection of model output is given by:

$$\begin{bmatrix} z_1^T \\ \vdots \\ z_n^T \end{bmatrix} = \begin{bmatrix} \rho_{1,1} y_1^T & \rho_{1,1} & \dots & \rho_{C,1} y_1^T & \rho_{C,1} \\ & & \vdots & & \\ \rho_{1,n} y_1^T & \rho_{1,n} & \dots & \rho_{C,n} y_n^T & \rho_{C,n} \end{bmatrix} \begin{bmatrix} G_1^T \\ h_1^T \\ \vdots \\ G_C^T \\ h_C^T \end{bmatrix} \tag{19}$$

where: $\rho_{i,j} = \rho_i$ evaluated at y_j

Note that given $\{y_1, y_2,..., y_n\}$, the first matrix on the right-hand side of equation (19) is constant, while the second matrix contains all the parameters to be optimized. To minimize the squared error between the model output and that of the training data, the least-squares estimation problem in equation (19) is solved by replacing the matrix on the left-hand side with the actual output of the training data. Of course, implicit in the least-squares estimation problem is the assumption that the number of training data is greater than the number of parameters to be optimized.

Using the standard notation widely adopted in literature, the least-squares estimation problem in equation (19) is written as:

$$AX = B \tag{20}$$

where: B = Matrix of output values
A = Constant matrix
X = Matrix of parameters to be estimated

Let us choose X in such a way that the following objective function J is minimized [12], [13]:

$$J = \|B - AX\|_2^2 \equiv (B - AX)^{-T}(B - AX) \tag{21}$$

To carry out the minimization, J is differentiated with respect to X and the result is equated to zero. Thus:

$$\frac{\partial J}{\partial X} = -2A^T B + 2A^T X = 0 \tag{22}$$

From which X can be solved as:

$$X = (A^T A)^{-1} + A^T B \tag{23}$$

In practice, the most reliable method of computing the pseudo-inverse of a matrix is the *singular value decomposition* (SVD). The SVD method is based on the following theorem of linear algebra, whose proof is beyond the present scope [14]:

Any $M \times N$ matrix A where $M \geq N$, can be written as the product of an $M \times N$ orthogonal-columns matrix U, an $N \times N$ diagonal matrix S with nonnegative elements in decreasing order, and the transpose of an $N \times N$ orthogonal matrix V.

The SVD can also be carried out when $M < N$. In this case, the singular values s_j for $j = M + 1, \dots, N$ are all zero, and the corresponding columns of U are also zero. The SVD is one of the most powerful tools of numerical linear algebra and has been successfully applied in various areas such as statistical analysis, image and signal processing, system identification, and linear control. For an overview of the SVD, its theory, and numerical details, the reader is referred to [14].

IV. Performance Analysis Of The Proposed Method

Once a controller is designed, it is important to validate its performance and compare it with other types of controllers, possibly designed using other methodologies. In section *A*, it is suggested that two performance measures with two simulated systems be used. The objectives of the simulation are to demonstrate the feasibility of the proposed three-term design method when applied to second-order and third-order systems.

The following section presents the performance measures that will be used during the study and the applications that will be used in testing and analyzing the performance. The simulation results will be presented in section B.

A. Performance Study

To test the models, two performance measures have been chosen, which will be used to analyze the performance of the proposed method for designing a PID FLC. They are:

1. Accuracy: To design a PID-like FLC with clustering algorithms, the rise-time, overshoot, and settling-time performance measures will be omitted, because a teacher signal is used with these algorithms as a reference model. To validate the results, it is proposed that an accuracy criterion be employed. Accuracy means the correctness of the answer. In order to measure it, the use of the sum of squares for error is proposed. Thus, the smaller the error, the better the accuracy, and the larger the error, the worse the accuracy will be. The error here is the error between the output of the system under analysis and its reference model.
2. Robustness: A robust controller is capable of dealing with significant parameter variations. The examination of its performance for parameter values that are

different from the designed values is a method of assessing controller robustness. The analysis of the effects of parameter variations on PID-like FLC design methods provides a useful quantitative, albeit empirical, measure of robustness.

To measure robustness, it is proposed that the defuzzification method parameter be varied. During the design of the PID-like FLC, center of area (COA) was chosen as a defuzzification method. To measure the robustness of this controller, the use of bisector of area (BOA) as a defuzzification method [15] is suggested, which is defined by:

$$U = \left\{ x \middle| \int_{Min}^{x} \mu(x)dx = \int_{Min}^{x} \mu(x)dx \right\} \tag{24}$$

where: U = Control action
x = Running point in the universe
$\mu(x)$ = Membership
Min = Leftmost value of the universe
Max = Rightmost value of the universe

This method picks the abscissa of the vertical line that divides the area under the curve in two equal halves. The procedure used to implement this method, as shown in algorithm 2.

Algorithm 2:
Calculating the Control Action U

```
Input: define input vector x;
       define membership functions vector;
Output: control action U
    Total_area = sum of all membership functions;
    temp = 0;
    For i = 1 to length of input vector x
      temp = temp + membership_function [i];
      If temp > = Total_area / 2
        Break;
      end if
    end for
    U = x [i];
```

Two types of direct current (DC) motors are analyzed to examine the performance of proposed design methods: *armature-controlled* with fixed field and *field-controlled* with fixed armature current [16]. The same details and parameters as described in our previous studies [1], [2], [18] were used for these two systems. *MATLAB* with *Fuzzy Logic Toolbox* was used to simulate the PID-like FLC.

For the clustering technique, the reference model with inputs $[e, de, se]^T$ and output U is used to designate the desired performance. To design a PID-like FLC using the *fuzzy c-means* (FCM) algorithm, the weighting exponent parameter q for the membership functions matrix M was chosen as 2.0 [17], [10]. To design a PID-like FLC with a subtractive algorithm, the cluster radius r_a as 0.5 for all data dimensions, squash factor r_b as 1.5, accept ratio $\bar{\varepsilon}$ as 0.5, and reject ratio $\underline{\varepsilon}$ as 0.15 [10] were chosen.

B. Simulation Results

The performance of the PID-like FLC design methods is examined by analyzing the transient response and accuracy in subsection 1 and robustness in subsection 2.

i. Transient Response and Accuracy

The following subsection focuses on the performance of the armature-controlled DC motor, while subsection b focuses on the performance of the field-controlled motor.

a. Armature-Controlled DC Motor System

The *CI* for the armature-controlled DC motor system is shown in Figure 2. This figure shows that the SMBA generates only 5 clusters from 40 sampling points.

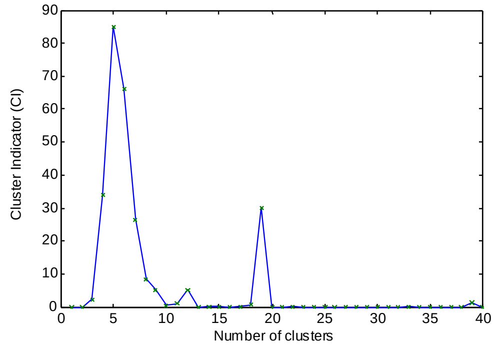
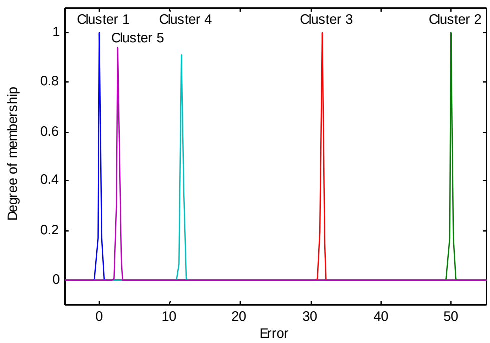
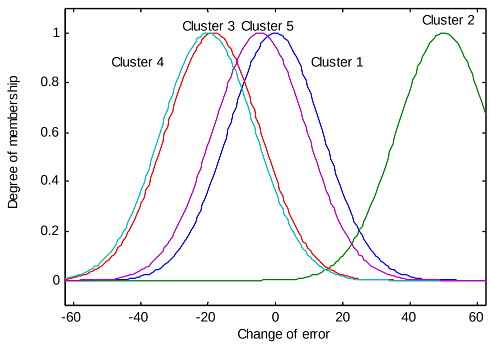


Figure 2: Cluster indicator for armature-controlled DC motor system

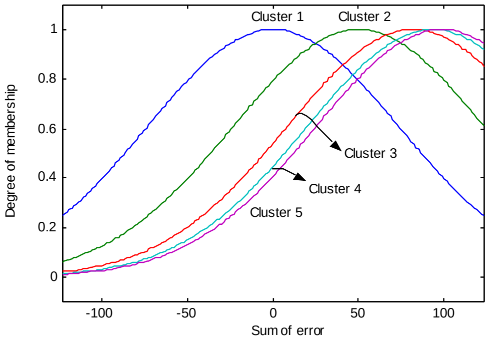
To design the linguistic-type SMBA, Figure 3 shows the membership functions generated by the proposed algorithm.



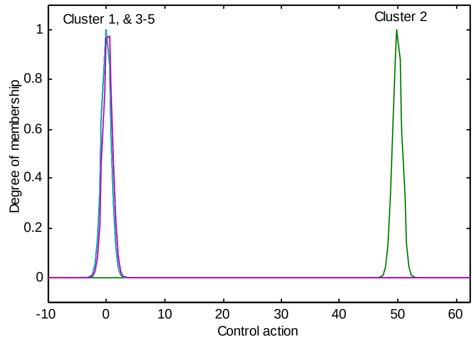
(a)



(b)



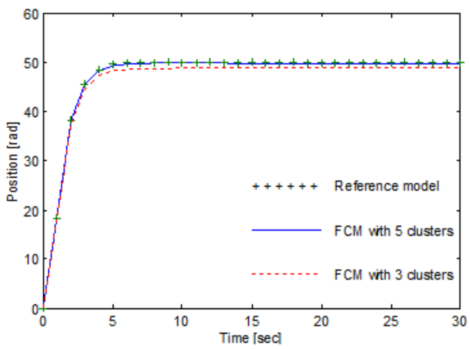
(c)



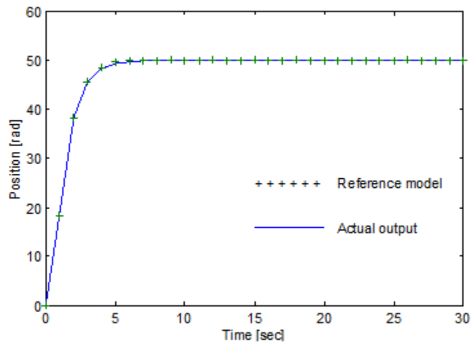
(d)

Figure 3: Membership functions generated for SMBA used to develop linguistic type model PID FLC for armature-controlled DC motor system

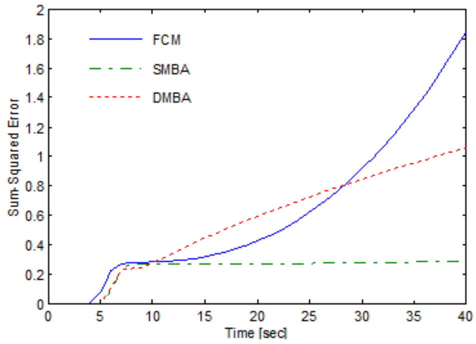
Figure 4 shows the step responses and accuracy of the armature-controlled DC motor system. The FCM algorithm used to generate three clusters and five clusters for comparison. It can be seen how the controller output gets close to the reference model as more clusters are considered. The subtractive algorithm generates only one cluster for this system, so it cannot be used with this model. A comparison between the SSE for the linguistic-type SMBA and other controller shows that they are comparable. Note that SSE for the SMBA is significantly smaller than the SSE for the other controller. Thus, we can conclude that no over-transient response occurs with SMBA method.



(a) Step response of FCM with 3 clusters and 5 clusters



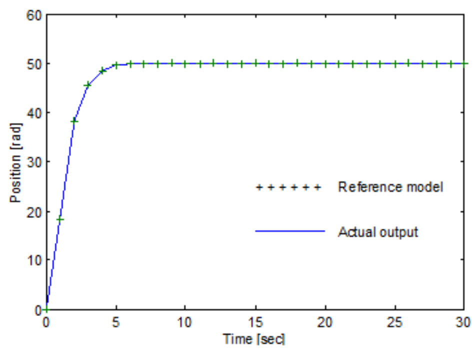
(b) Step response of proposed non-parametric SMBA



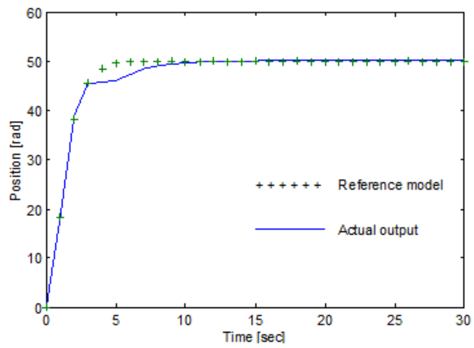
(c) Accuracy of clustering algorithms

Figure 4: Step responses and accuracy of armature-controlled DC motor system using clustering algorithms with PID-like FLC

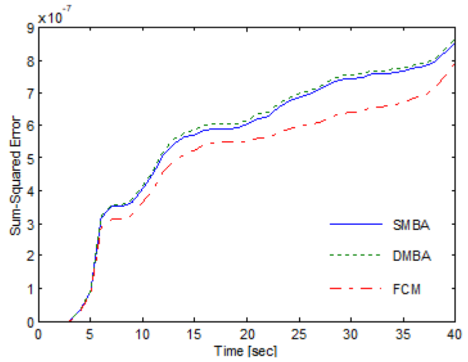
Figure 5 shows the step responses and accuracy of the armature-controlled DC motor system using the FCM algorithm (used to generate 5 clusters), the subtractive algorithm, and the SMBA for the TSK-type model of the PID-like FLC. It is possible to observe that SSE does not reach the 1m radian and again the transient response of the proposed controller is better than other controller performance.



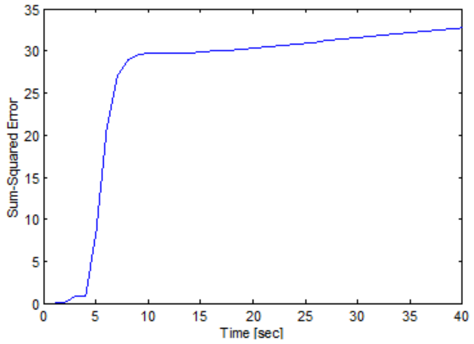
(a) Step response of FCM and SMBA



(b) Step response of subtractive clustering algorithm



(c) Accuracy of FCM, SMBA, and DMBA



(d) Accuracy of subtractive clustering algorithm

Figure 5: Step responses (A) and accuracy (B) of armature-controlled DC motor system using clustering algorithms used to develop TSK type model PID-like FLC

b. Field-Controlled DC Motor System

The cluster indicator (CI) for the armature-controlled DC motor system using the non-parametric clustering algorithm (SMBA) is shown in Figure 6. This figure shows that the SMBA generates 10 clusters from 40 sampling points.

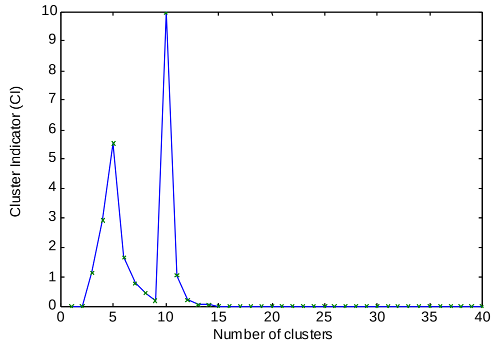
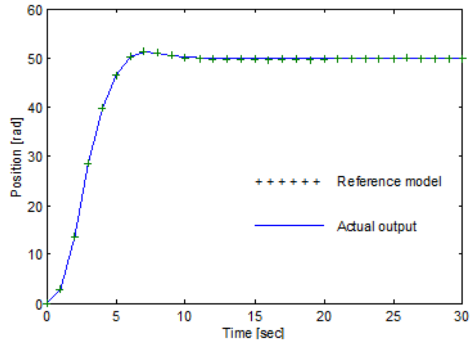
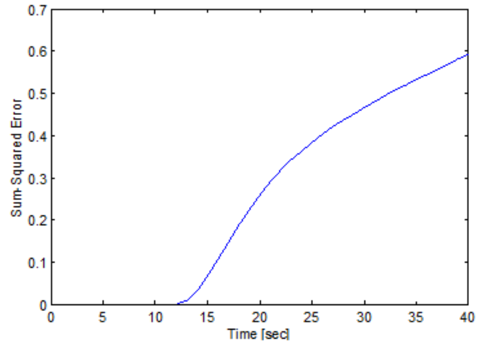


Figure 6: Cluster indicator for field-controlled DC motor system using non-parametric SMBA

Figure 7 shows the step responses and accuracy of the field-controlled DC motor system using the SMBA to develop a linguistic-type model of the PID-like FLC. The subtractive algorithm generates only one cluster for this system, so it cannot be used to develop a linguistic-type model of the PID-like FLC.



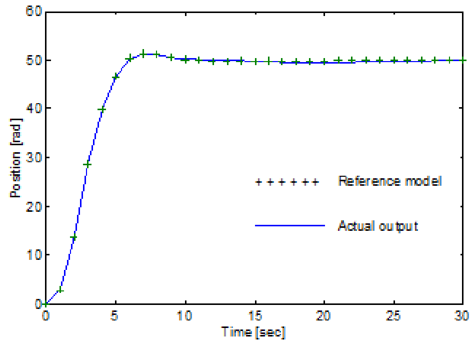
(a) Step responses of proposed non-parametric SMBA



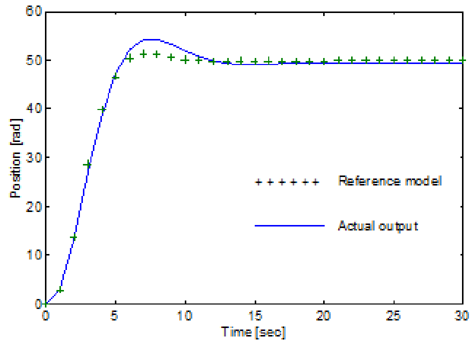
(b) Accuracy of proposed non-parametric SMBA

Figure 7: Performance of field-controlled DC motor system using FLC

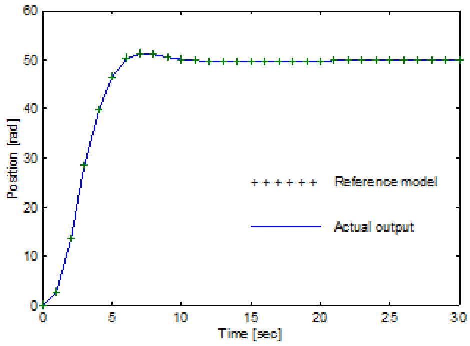
From Figure 8, it can be seen that the step response and accuracy results by the proposed TSK-type SMBA algorithm is better than the FCM algorithm (used to generate 5 clusters) and the subtractive clustering algorithm. Where, zero overshoot achieved in FCM and SMBA methods. The results show that the novel proposed tuning method works more precisely than other tested controllers.



(a) FCM algorithm with 5 clusters



(b) Subtractive clustering algorithm

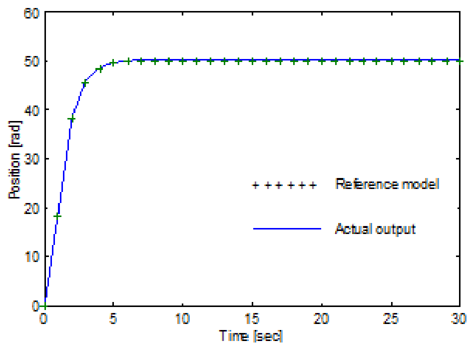


(c) Proposed non-parametric SMBA

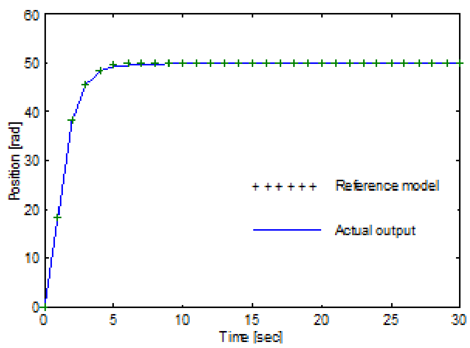
Figure 8: Step responses (of field-controlled DC motor system) using clustering algorithms used to develop TSK type model PID-like FLC

ii. Robustness Test

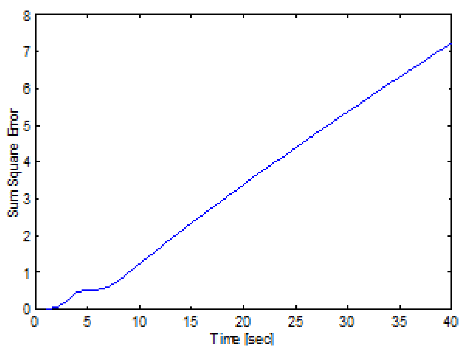
The following subsection analyzes the robustness of the FLC design methods when varying the defuzzification method from COA to BOA. This test cannot be used with the TSK-type model. The only defuzzification method that can be used with this model is the weighted average method, because the fuzzy output does not have a geometric shape.



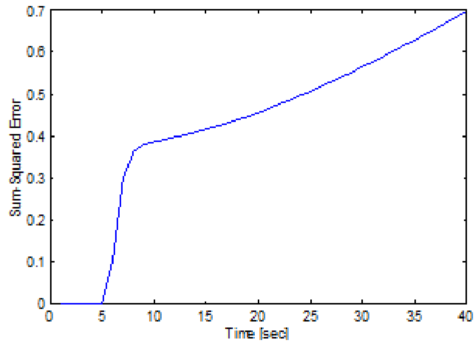
(a) Step response of FCM clustering algorithm



(b) Step response of non-parametric SMBA



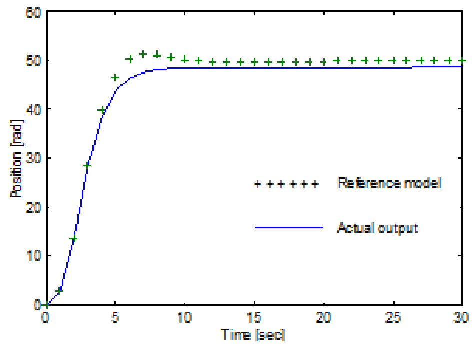
(c) Accuracy test of FCM clustering algorithm



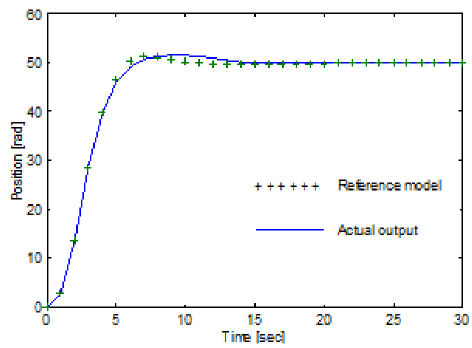
(d) Accuracy test of non-parametric SMBA

Figure 9: Robustness test for armature-controlled DC motor system

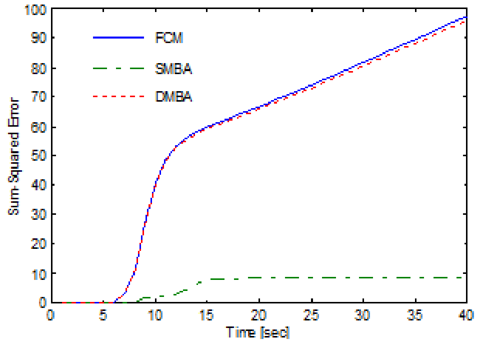
Figures 9 and 10 show the step responses and accuracy of the armature-controlled DC motor and the field-controlled DC motor respectively, using the FCM algorithm (used to generate 5 clusters) and the SMBA to develop a linguistic-type model of the PID-like FLC. It is shown that there is a substantial improvement in the time domain specification in terms of lesser rise time, settling time and overshoot using SMBA algorithm. Hence, this method is a robust design method for determining the PID controller parameters.



(a) Step response of FCM clustering algorithm



(b) Step response of non-parametric SMBA



(c) Accuracy test of clustering algorithms

Figure 10: Robustness test for field-controlled DC motor system

In general, the proposed non-parametric clustering algorithm based on the SMBA is more accurate than the FCM and subtractive algorithms for both systems used in the study. This enhanced accuracy is observed for both models used to design a PID-like FLC (linguistic and TSK models). Enhanced robustness is also observed for both the linguistic and the TSK models. However, designing a PID-like FLC using the SMBA produces more robust results than the FCM algorithm for both systems used in the study. From the simulation, it can be noticed that when designing a PID-like FLC, the TSK-type model is more accurate than the linguistic-type model for both systems used in the study.

V. Conclusion

The problem of clustering algorithms used in fuzzy systems is having non-deterministic parameters that must be defined by the user before the algorithms start. These parameters affect the number of clusters that the algorithms generate. To eliminate this, a non-parametric clustering method based on similarity algorithm was proposed. When designing a three-mode FLC using the clustering method, the performance analysis shows that the proposed *non-parametric clustering method* based on the SMBA is more accurate than the FCM and subtractive algorithms for both systems used in the study. This enhanced accuracy is observed for both models used to design the three-mode FLC (linguistic and TSK). Combining the cluster estimation method with a SVD estimation procedure provides a fast algorithm for identifying fuzzy models from numerical data. When designing a three-mode FLC, the TSK-type model is more accurate than the linguistic-type model.

Acknowledgment

The author would like to express his appreciation to College of Engineering at AMA International University Bahrain for supporting this research.

References

[1] E. Natsheh and Khalid Buragga, "Comparison between Conventional and Fuzzy Logic PID Controllers for Controlling DC Motors", *International Journal of Computer Science Issues (IJCSI)*, vol. 7, no. 5, pp. 128-134, 2010.
[2] E. Natsheh and Khalid Buragga, "A Performance Comparison for Various Methods to Design the Three-Term Fuzzy Logic Controller", *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 10, no. 5, pp. 142-151, 2010.
[3] K. C. Gowda and E. Diday, "Symbolic Clustering Using a New Similarity Measure", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-22, No. 2, pp. 368-378, 1992.
[4] K. C. Gowda and E. Diday, "Symbolic Clustering using a New Dissimilarity Measure", *Pattern Recognition*, Vol.24, No. 6, pp. 567-578, 1992.
[5] E. Natsheh, "Taxonomy of Clustering Methods Used in Fuzzy Logic Systems", *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol.4, no.1, pp. 65-72, 2012
[6] K. C. Gowda and G. Krishna, "Disaggregative Clustering Using the Concept of Mutual Nearest Neighborhood", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-8, No. 12, pp. 888-895, 1978.
[7] K. C. Gowda and G. Krishna, "The Condensed Nearest Neighbor Rule Using the Concept of Mutual Nearest Neighborhood", *IEEE Transaction on Information Theory*, Vol. 25, No. 4, pp. 488-490, 1979.
[8] R. R. Yager and D. P. Filev, "*Essentials of Fuzzy Modeling and Control*", Wiley India, 2002.
[9] T. Takagi and M. Sugeno, "Fuzzy Identification of Systems and Its Applications to Modeling and Control", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-15, NO. 1, pp.116-132, 1985.
[10] S. Chiu, "Fuzzy Model Identification Based on Cluster Estimation", *J. of Intelligent & Fuzzy Systems*, Vol. 2, No. 3, pp. 267-278, 1994.
[11] S. Chiu, "Selecting Input Variables for Fuzzy Models", *J. of Intelligent & Fuzzy Systems*, Vol. 4, No. 4, pp. 243-256, 1996.
[12] J. Yen and R. Langari, "*Fuzzy Logic: Intelligence, Control and Information*", Prentice-Hall, 1999.
[13] J. R. Jang, C. Sun, and E. Mizutani, "*Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligent*", Prentice-Hall, 1997.
[14] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, "*Numerical Recipes in C: The Art of Scientific Computing*", 3rd Edition, Cambridge University Press, 2007.
[15] J. Jantzen, "*Foundations of Fuzzy Control: A Practical Approach*", 2nd Edition, Wiley, 2013.
[16] M. Golob, "Decomposition of a Fuzzy Controller Based on the Inference Break-up Method", *Intelligent Data Analysis*, Vol. 3, Issue 2, pp. 127-137. 1999.
[17] I. Gath and A. B. Geva, "Unsupervised Optimal Fuzzy Clustering", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 7, pp. 773-781, 1989.
[18] E. Natsheh, "Designing and Tuning PID Fuzzy Controllers for Armature-Controlled DC Motor", *International Journal of Human and Technology Interaction (IJHaTI)*, vol.2, no.2, pp. 17-24, 2018.