Designing and Tuning PID Fuzzy Controllers for Armature-Controlled DC Motors

Essam Natsheh

Dep. of Informatics Engineering, College of Engineering, AMA International University, Bahrain

dr_natsheh@hotmail.com

Abstract—**Designing and tuning proportional plus integral plus derivative (PID) fuzzy controllers is an active research field which looks at the optimal design for these controllers. In this paper, a novel optimization design method is proposed, which uses a performance rule-based model with any method of designing PID fuzzy controllers to achieve certain desired performance. Since constructing the membership functions is the most critical part of the fuzzy controller, a self-optimized membership functions algorithm is introduced. Armature-controlled DC motor, as an application representing second-order systems, was used to analyze the performance of the proposed design method and compare its performance with other various design methods. The accuracy analysis shows that the proposed design method is 2 seconds faster in rise-time, 2 seconds faster in settling-time and; at the same time, it decreases the overshoot by 1.7% compared to the original design methods. Meanwhile, the robustness analysis shows that the proposed design method is 2 seconds faster in rise-time, 2.6 seconds faster in settling-time and; at the same time, it decreases the overshoot by 3.4% compared to the original design methods.**

Keywords—**Three-term controller; PID controller; fuzzy systems; fuzzy logic controller**

I. INTRODUCTION

FUZZY logic is described as "computing with words rather than numbers." In the same manner, fuzzy control is described as "control with sentences rather than equations." A review of fuzzy logic controller usage can be found in [1], [2] and [3]. A three-term fuzzy logic controller consists of a proportional plus integral plus derivative controller that can be simply written as PID-like FLC. Examples of applications of fuzzy logic control (FLC) which is covering a wide range of practical areas can be found in [4] and [5].

Since methods of designing a three-term FLC has emulate the human control strategy, their principles are understandable for the non-control specialists. During the last two decades, the methods of designing a conventional three-term controllers have employed more and more advanced mathematical tools. This is needed in order to solve the difficult problems in a rigorous manner. However, the results showed that fewer and fewer practical engineers have a full understanding over these design methods. Therefore, the practical engineers who are on the front line of designing consumer products have tendency to use the approaches that are simple and easy to understand; hence, the proposed various methods of designing three-term FLCs are just such approaches.

The previous studies from [6] and [7] had identify, study and taxonomize the various design choices for a three-term FLC. In this paper, these design methods will be enhanced by a focus on the desired performance of the controlled process as measured by rise-time and by percentage overshoot. To justify this, the use of a performance rule-based model is proposed, which can be employed with any method of design of this controller. Therefore, this approach has narrows the gap between a practical three-term FLC and a desired one. A simple means for designating membership functions for a three-term FLC is presented, which allows a novice to construct a set of membership functions for a specific linguistic variable systematically.

Article history: Manuscript received 25 April 2018; received in revised form 24 August 2018; Accepted 30 August 2018.

The rest of this paper is organized as follows: Section II will proposes a novel method that relies on using a performance rule-based model to achieve the desired performance for the three-term FLC. Section III will compares the performance of the proposed design method and the other design methods that is using a second-order armature-controlled DC motor as a case study. Finally, the section IV will presents the conclusions of the proposed work.

II. PERFORMANCE-OPTIMIZATION DESIGN METHOD FOR A THREE-TERM FLC

To obtain optimal performance output, two modules are proposed: a *performance rule-based model* is meant to achieve a desired performance of any selected system and an *optimal membership functions* algorithm to obtain the optimal membership functions used to represent the fuzzy linguistic variables.

A. Performance rule-based model

Previous studies at [6] [7] presented various methods of designing a PID-like FLC, but these methods do not take into consideration of the required performance criteria for the controlled plant. In a practical situation, every plant must meet certain desired performance measures in order to function properly.

The nature of the rules contained in a PID-like FLC rule-base is discussed overleaf. The typical response of the second-order systems is shown in Fig. 1.

According to the sign and the magnitude of error $e\ (k)$ and its change $de\ (k)$, the response plane is roughly divided into five zones. The index used for identifying the response zones is defined as [8], [9]:

$$Z_1:\ e > 0 \text{ and } de < 0, \qquad Z_2:\ e < 0 \text{ and } de < 0,$$
$$Z_3:\ e < 0 \text{ and } de > 0, \qquad Z_4:\ e > 0 \text{ and } de > 0,$$
$$Z_5:\ e \approx 0 \text{ and } de \approx 0.$$

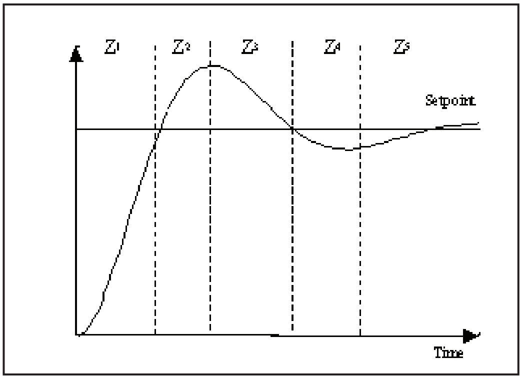


Fig. 1. The general response of the second-order systems

Five main homogenous zones in the PID-like FLC rule-base can be identified, as shown in Table I, where N, Z, and P are the linguistic labels negative, zero, and positive of the term sets error, change of the error, and sum of the error.

TABLE I. STATE PLANE OF THE PID-LIKE FLC RULE-BASE

Error / Change of the error

Error	N	N	N	Z	Z	Z	P	P	P
N	Z2						Z3		
Z				Z5					
P	Z1						Z4		
	N	Z	P	N	Z	P	N	Z	P

Sum of the error

Tang and Mulholand [10] propose the following three metarules:

MR1: If both $e\ (k)$ and $de\ (k)$ are zero, then maintain present control setting.

MR2: If conditions are such that $e\ (k)$ will go to zero at a satisfactory rate, then maintain present control setting.

MR3: If $e\ (k)$ is not self-correcting, then control action $U\ (k)$ is not zero and depends on the sign and magnitude of $e\ (k)$ and $de\ (k)$.

Yager and Filev [12] use these three metarules to analyze the general rule-bases of the FLC as follows:

- In zone 1 and 3 the error is self-correcting and $U\ (k)$ is almost zero; that is, the control variable remains at its present setting. The rules of these zones realize rule 2.
- In zone 2 and 4 the errors are not self-correcting and consequently negative and positive, respectively, control action $U\ (k)$

applies. The magnitude of *U* (*k*) changes with respect to the magnitudes of *e* (*k*) and *de* (*k*). The rules associated with these zones are related to rule 3.

- In zone 5 both *e* (*k*) and *de* (*k*) are close to zero. The system is in a steady state and *U* (*k*) is almost zero, that is, the present control setting is maintained. The rules that belong to this zone realize metarule 1.

To generate a fast rise-time, it is suggested that the control action *U* (*k*) in zone 1 must be positive independent of the three measured input values *e* (*k*), *de* (k), and *se* (*k*). To achieve an approximate reduction in overshoot, it is suggested that the control action *U* (*k*) in zone 2 to be negative. The designer should determine the proper rise-time and percentage overshoot that are optimal for the plant to function properly. Therefore, the following two rules are proposed:

R1: IF rise-time is FS1, THEN control action *U* (*k*) is positive

R2: IF overshoot is FS2, THEN control action *U* (*k*) is negative

where *FS1* is fuzzy set defined on fuzzy variable *rise-time*, and FS2 is fuzzy set defined on fuzzy variable *overshoot*. Fig. 2 shows how these fuzzy sets can be defined: Threshold 1 is the rise-time threshold that a plant must not exceed, while threshold 2 is the desired percentage overshoot that a plant must not exceed.

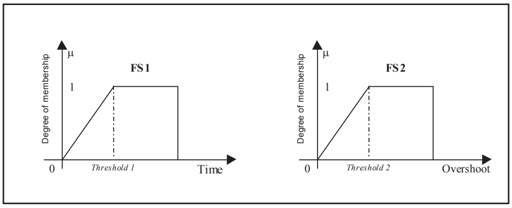


Fig. 2. Fuzzy sets for rise-time and overshoot

Before the system reaches threshold 1 (desired rise-time), the output of the PID-like FLC will rise to reach the desired setpoint. Since the system output will rise up faster, it will certainly create an overshoot, so metarule 2 is used to control it. If the overshoot exceeds threshold 2 (normal expected overshoot), the output of the PID-like FLC must deflate to stop the previous increase. These two metarules can be used with any method of designing a PID-like FLC described in the previous section.

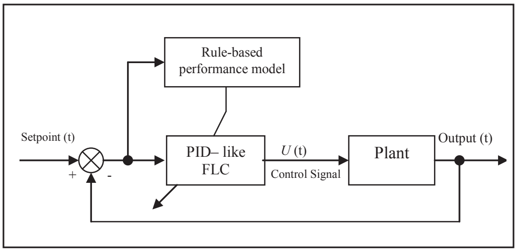


Fig. 3. PID-like FLC with performance rule-based model

To reduce the complexity of the rule-base design further and increase the efficiency, a division of the rule-base of the PID-like FLC into two modules is proposed, as shown in Fig. 3. One module uses a large number of input variables (*e* (*k*), *de* (*k*), and *se* (*k*)) to provide a nominal output. The other module uses only a few input variables (rise-time and percentage overshoot) to detect the performance and adjust the nominal output; it is called *performance rule-based model* (PRBM). The key idea behind this approach is that the performance conditions is corresponding only to a few regions in the input space and hence, need a few rules only to describe them (i.e., one does not need to consider all the possible combinations of input variables when writing the rules for performance handling) [11]. The output of the controller will be the mean between the output of the nominal control module and the output of the performance-handling module, as illustrated by the algorithm shown in Fig. 4.

```
Calculate by Fuzzy Inference System (FIS) control action $U [k]$ using the selected input.
% Input is selected depending on the kind of controller.

If time < desired rise-time
        Calculate by FIS $U 2 [k]$ using proposed PRBM;
        $U [k] = U [k] + U 2 [k] / 2$;
End if
If percent overshoot > desired percent overshoot
        Calculate by FIS $U 2 [k]$ using proposed PRBM;
        $U [k] = U [k] + U 2 [k] / 2$;
End if
```

Fig. 4. The procedure for tuning PID fuzzy controllers

B. *Optimizing the membership functions*

In real applications of FLC, membership functions are constructed by assembling the knowledge from experts; then they are modified by surveying the control response of the process laboriously. In most of the control cases, the FLC would not be effective without a careful arrangement of the membership functions.

Regarding to the theoretical analysis of the FLC, the majority of the researchers do not focus much on the selection of the membership functions. Most of them use isosceles triangular functions with equal spans as the membership functions for their FLCs throughout the whole universe of discourse [13], [14], [8], [9]. The main advantage of choosing this type of membership function is that it eases the difficulties in analyzing the structure of the FLC. However, almost all FLC applications adopt non-equal span membership functions to cope with the real control problems. Instinctively, the closer the control response to the setpoint (or normal condition), the narrow the membership function range should be. A FLC with an equal-span triangular membership function is not adequate to achieve a good control result for some highly nonlinear processes.

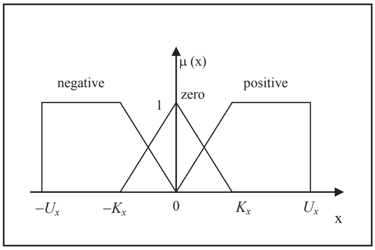


Fig. 5. Membership functions for fuzzy variable x

The membership functions shown in Fig. 5 are used to represent fuzzy variable x with universe of discourse $[-U_x, U_x]$ and three fuzzy sets *negative, zero,* and *positive*. The problem is simplified to be the determination of K_x point in each fuzzy variable.

To achieve a better performance and to devise a systematic method of obtaining optimal membership functions, the following algorithm is proposed:

```
Determine the universe of discourse $U_x$ for each fuzzy variable;
Initialize $K_x$ value to each fuzzy variable to be $K_x = U_x / 2$ ;
Initialize IAE and ITAE to large values;
For i=1 to maximum number of epochs to refinement all $K_x$
    For j=1 to minimum number of epochs to refinement one $K_x$
        Run the experiment and get new_IAE and new_ITAE ;
        If ((new_IAE < IAE) and (new_ITAE < ITAE))
            IAE = new_IAE ;
            ITAE = new_ITAE ;
            Save $K_x$ ;
        End if
        If ((new_IAE $\le$ IAE) and (new_ITAE $\le$ ITAE))
            $K_x = K_x \times$ increase_ratio;
        Else
            $K_x = K_x \times$ decrease_ratio;
        End if
    End for
End for
```

The use of decrease_ratio and increase_ratio as 0.9 and 1.05 respectively is suggested. IAE and ITAE are defined as follows:

1. Integral of the absolute of the error (IAE) defined by:

$$IAE = \int_0^\infty \left| e(t) \right| dt \tag{1}$$

2. The integral of time multiplied by the absolute of error (ITAE) defined by:

$$ITAE = \int_0^\infty t \left| e(t) \right| dt \tag{2}$$

where $e(t)$ is the measured error. The calculation in the studies was carried out by substituting an algebraic sum for the integrals [5]. *IAE* accounts mainly for errors at the beginning of the response and to a lesser degree for the steady state duration. *ITAE* records errors at the beginning but also emphasizes the steady state [15].

III. PERFORMANCE ANALYSIS OF THE PROPOSED DESIGN METHOD

When designed a controller, it is important to validate its performance and compare it with that of other types of controllers, most possibly designed is by using other methodologies. Such evaluations are usually achieved by analyzing time responses. However, a mere examination of the behavior of a controller is not enough to validate its performance or prove that it is better or worse than other controllers. In the next section, the use of three performance measures with two simulated systems is proposed. The objective of the simulation is to demonstrate the feasibility of the proposed three-term design method when applied to second-order systems. A comparison of its performance with the performance of various pure fuzzy design methods is carried out.

Section *A* introduces the performance measures used in the study. Section *B* presents the application employed in testing and analyzing the performance. Section *C* describes the implementation technique for all methods of designing a pure fuzzy three-term FLC and the proposed performance rule-based method. The simulation results of these methods are presented in section *D*.

A. Performance study

To test the models, three performance measures were chosen to analyze the performance of the proposed methods of designing a PID-like FLC:

1. *Transient response:* One of the most important characteristics of control systems is their transient response. The transient response is the response of a system as a function of time. It can be described in terms of two factors [16]:
   a. *The swiftness of response*, as represented by the rise-time Tr.
   b. *The closeness of the response to the desired response*, as represented by the overshoot *Os* and settling-time *Ts*.
2. *Error integral criteria:* The performance was evaluated by two frequently used error integral criteria—*IAE* and *ITAE*—described in the previous section.
3. *Robustness:* A robust controller is capable of dealing with significant parameter variations. Its robustness can be usually assessed by examining its performance for parameter values that are different from the designed values. The analysis of the effects of parameter variations on PID-like FLC design methods provides a useful quantitative, albeit empirical, measure of robustness.

   A varying defuzzification method parameter is suggested to measure robustness. 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, it is proposed that bisector of area (BOA) be used as a defuzzification method [1], which is defined by:

$$U=\left\{x \mid \int_{Min}^{x}\mu\left(x\right)dx=\int_{x}^{Max}\mu\left(x\right)dx\right\} \tag{3}$$

where *U* is the control action, x is the running point in the universe, μ (*x*) is its membership, *Min* is the leftmost value of the universe, and *Max* is the rightmost value. This method picks the abscissa of the vertical line that divides the area under the curve in two equal halves. The procedure shown in Fig. 6 will be used to implement this defuzzification method.

```
Define input vector x;
Define membership functions vector;
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
control_action = x [i];
```

Fig. 6. The procedure for implement defuzzification method

B. Armature-controlled DC motor

DC motors are classified into several broad categories, as described in [17]. DC motors have a separately excited field, in which the field winding is separate from the armature. They are either *armature-controlled* with fixed field, or *field-controlled* with fixed armature current [18]. Armature-controlled DC motor is used in this simulation. The block diagram of the systems is shown in Fig. 7. The control objective for this type of DC motors is to reach a specified motor position using an appropriate input drive voltage.

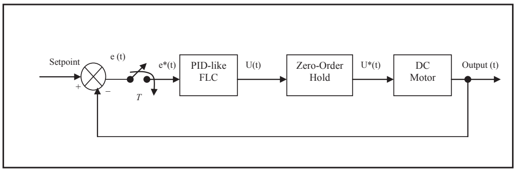


Fig. 7. Armature-controlled DC motor system with PID-like FLC

A zero-order holder device is used to keep a constant controller output during each interval. The PID controller inputs are defined as follows:

$$e(kT) = \{setpoint(t) - position(t)\}|_{t=kT} \tag{4}$$

$$de(kT) = \frac{e(kT) - e((k-1)T)}{T} \tag{5}$$

$$se(kT) = se((k-1)T) + T \times e((k-1)T) \tag{6}$$

where T is sampling interval time, while *setpoint* (*t*) and *position* (*t*) are reference and process output, which is the angular displacement of the motor shaft.

Ogata [18] provides the transfer function between the output angular displacement of the motor shaft θ (*t*) and the input control action U (*t*):

$$\frac{\theta(s)}{U(s)} = \frac{K_m}{s(T_m s + 1)} \tag{7}$$

where K_m is motor gain constant and T_m is motor time constant. For simplicity, it is assumed that $K_m = 1$ newton-m/amp and $T_m = 1$ second.

C. Implementation of various methods of designing PID-like FLCs

The *hybrid types* [7] of PID-like FLCs are not true fuzzy PID controllers because they include deterministic controls as well [19]. Therefore, only *pure* PID-like FLC types were used in the simulation.

To simulate all the methods of designing PID-like FLCs, *MATLAB* with *Fuzzy Logic Toolbox* is developed by the Mathworks and was used with the same configurations for the parameters of fuzzy system as described in [6] and [7].

To design an *ideal PID-like FLC*, a three-dimensional rule-base matrix described in Table II is proposed, where N, Z, and P are the linguistic labels *negative, zero,* and *positive* of the term sets *error, change of the error,* and *sum of the error.*

To design an *incremental PID-like FLC* as proposed by Yager and Filev [12], the same rule-base matrix in Table II is used, but the input of the controller is the error at *k, k*–1, and *k*–2 sampling intervals. The same rule-base matrix is also employed to design the *velocity algorithm* of the PID-like FLC, where the inputs of the controller are replaced with *e* (*k*), *de* (*k*), and *d2e* (*k*).

TABLE II. RULE-BASE MATRIX OF THE PID-LIKE FLC

Error / Change of the error	**N**	**N**	**N**	**Z**	**Z**	**Z**	**P**	**P**	**P**
N	N	N	N	N	N	N	Z	Z	Z
Z	N	N	N	Z	Z	Z	P	P	P
P	Z	Z	Z	P	P	P	P	P	P
Sum of the error	**N**	**Z**	**P**	**N**	**Z**	**P**	**N**	**Z**	**P**

To design the *break-up PID-like* FLC proposed by Golob [20], three rule-bases are used: the first rule-base for error, the second for *de* (*k*), and the third for *se* (*k*). An example of a rule-base of such an error term is the following set of vague rules:

Rule 1: IF error *e* (*k*) is negative THEN control action *U* (*k*) is negative.

Rule 2: IF error *e* (*k*) is zero THEN control action *U* (*k*) is zero.

Rule 3: IF error *e* (*k*) is positive THEN control action *U* (*k*) is positive.

To design the PD-like FLC parallel with I-like FLC proposed by Kwok et al. [19], the rule-base matrix shown in Table III is used as rule-base for PD-like FLC [12]. The following set of rules were used as I-like FLC rule-base:

Rule 1: IF sum of error *se* (*k*) is negative, THEN control action *U* (*k*) is zero.

Rule 2: IF sum of error *se* (*k*) is zero, THEN control action *U* (*k*) is positive.

Rule 3: IF sum of error *se* (*k*) is positive, THEN control action *U* (*k*) is zero.

To design the *PD-like FLC parallel with PI-like FLC,* the simplified method proposed by Li and Gatland [9] was employed, with the rule-base matrix shown in Table III as rule-base for both controllers.

To implement all the previous design methods with the proposed *rule-based performance model*, a 5-second desired rise-time for the armature-controlled DC motor was chosen for threshold 1 and a 6 % percentage overshoot was chosen for threshold 2.

TABLE III. RULE-BASE MATRIX FOR PD AND PI FLC

Error

Change of the error

	Negative	**Zero**	**Positive**
Negative	Negative	Negative	Zero
Zero	Negative	Zero	Positive
Positive	Zero	Positive	Positive

Sum of the error

Having defined the fuzzy linguistic control rules, the membership functions corresponding to each element in the linguistic set must be defined as well. The optimal membership functions method described in the previous chapter was employed to define the membership functions for e, de, se, and U linguistic variables. These membership functions used for the DC motor systems have universe of discourse of *e, de, se,* and *U* as [-50 50], [-40 40], [-100 100], and [-40 40] respectively.

For all the methods of designing PID-like FLCs for armature-controlled DC motor, the desired angular displacement of the motor shaft was 50 radians and the sampling interval time T was 1 second.

D. Simulation results

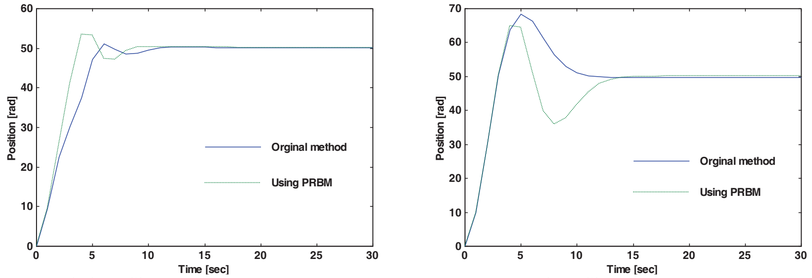
The performance of the PID-like FLC design methods is investigated in the following subsection by studying the transient response, error integral criteria, and accuracy, while robustness is analyzed in subsection 2.

1) Transient response, error integral criteria, and accuracy

Comparisons between the step responses of the armature-controlled DC motor system using various methods of designing PID-like FLCs and the proposed PRBM are shown in Fig. 8, Table IV, and Table V. These results indicate that using the proposed PRBM with all design methods generates both a faster rise-time and a faster settling-time. The only exception is the break-up method due to the separation of the proportional, integral, and derivative parts where they need independent tuning.

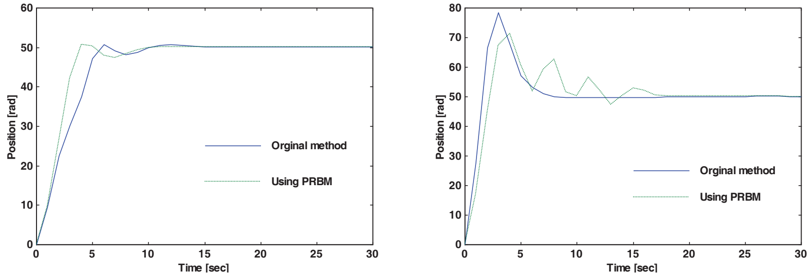
In case of the incremental PID-like FLC, the motor response curve shows a high overshoot and deviation from the reference response and then returns to the desired response. It is unrealistic to expect that an operator or expert can determine reasonable control rules, considering second and higher differences of the error.

When considering the average values of the results, the use of the proposed PRBM with all design methods reduces both rise-time and settling-time by 2 seconds and, at the same time, decreases the overshoot by 1.7% compared to the original design methods.



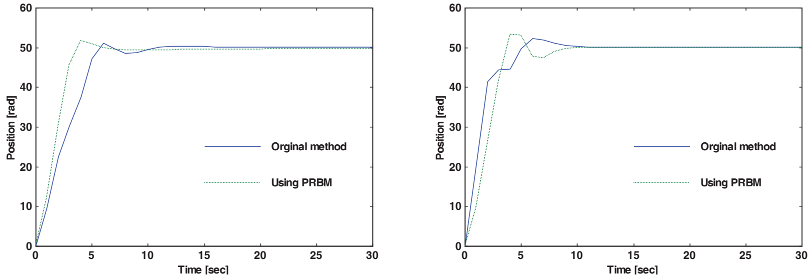
A- Ideal PID-like FLC.

B- Incremental PID-like FLC.



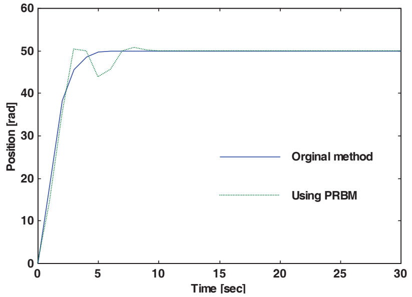
C- Velocity algorithm of PID-like FLC.

D- Break-up PID-like FLC.



E- PD-like parallel with I-like FLC.

F- PD-like parallel with PI-like FLC.



G- VSPID-like FLC

Fig. 8. Comparisons between step responses of armature-controlled DC motor system using various methods of designing PID-like FLC and PRBM

TABLE IV. PERFORMANCE OF ARMATURE-CONTROLLED DC MOTOR SYSTEM USING VARIOUS METHODS OF DESIGNING PID FLC

	Tr (Sec)	Ts (Sec)	Os (%)	IAE	ITAE
Ideal PID-like FLC	7	13	2.29	160.48	631.84
Incremental PID-like FLC	4	13	36.97	186.86	1210.7
Velocity algorithm FLC	7	13	1.39	161.20	644.62
Break-up PID-like FLC	3	10	56.66	150.00	1356.5
PD-like parallel with I-like FLC	7	12	2.29	160.48	631.83
PD-like parallel with PI-like FLC	5	11	4.759	107.98	391.51
VSPID-like FLC	6	7	0	101.29	311.31

TABLE V. COMPARISON OF PERFORMANCE OF ARMATURE-CONTROLLED DC MOTOR SYSTEM USING PROPOSED PRBM WITH VARIOUS METHODS OF DESIGNING PID-LIKE FLC

	Tr (Sec)	Ts (Sec)	Os (%)	IAE	ITAE
Ideal PID-like FLC	4	8	6.887	142.20	592.91
Incremental PID-like FLC	3	13	29.80	196.673	1337.0
Velocity algorithm FLC	4	8	1.578	133.725	652.22
Break-up PID-like FLC	3	17	42.73	184.415	1133.2
PD-like parallel with I-like FLC	4	5	3.464	126.939	590.69
PD-like parallel with PI-like FLC	4	8	6.769	138.250	518.99
VSPID-like FLC	3	7	1.520	116.066	444.23

TABLE VI. PERFORMANCE OF ARMATURE-CONTROLLED DC MOTOR SYSTEM USING VARIOUS METHODS OF DESIGNING PID FLC

	Tr (Sec)	Ts (Sec)	Os (%)	IAE	ITAE
Ideal PID-like FLC	8	13	2.269	175.672	928.253
Incremental PID-like FLC	3	10	43.58	195.701	1338.1
Velocity algorithm FLC	8	15	6.698	182.226	1176.3
Break-up PID-like FLC	2	15	55.74	165.804	1055.4
PD-like parallel with I-like FLC	9	13	2.269	175.672	928.253
PD-like parallel with PI-like FLC	4	12	7.451	129.820	757.073
VSPID-like FLC	5	5	0	122.415	807.124

2) *Robustness test*

Fig. 9, Table VI, and Table VII show the analysis of the robustness of the PID-like FLC design methods when changing the defuzzification method from center of area (COA) to bisector of area (BOA). These results indicate that using the proposed PRBM with all design methods generates a faster rise-time and a faster settling-time. Therefore, it can be stated that using PRBM with various designing methods of PID-like FLCs achieves more robustness than the original design methods alone.

When considering the average values of the results, the use of the proposed PRBM with all design methods reduces rise-time by 2 seconds, settling-time by 2.6 seconds and, at the same time, it decreases the overshoot by 3.4% compared to the original design methods.

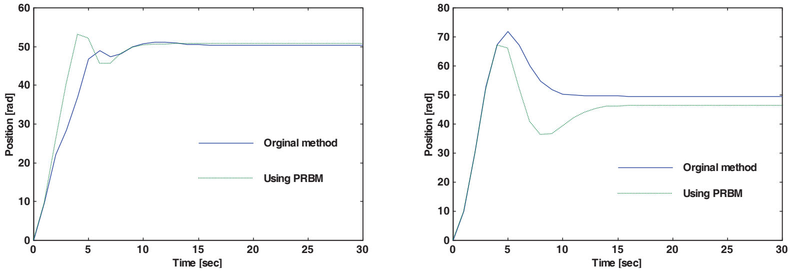
TABLE VII. COMPARISON OF PERFORMANCE OF ARMATURE-CONTROLLED DC MOTOR SYSTEM USING PRBM WITH VARIOUS METHODS OF DESIGNING PID-LIKE FLC

	Tr (Sec)	Ts (Sec)	Os (%)	IAE	ITAE
Ideal PID-like FLC	4	8	6.255	159.112	984.245
Incremental PID-like FLC	3	10	34.02	302.510	4037.9
Velocity algorithm FLC	4	8	1.926	147.744	987.704
Break-up PID-like FLC	3	13	41.96	176.111	1007.3
PD-like parallel with I-like FLC	4	6	3.982	134.079	712.61
PD-like parallel with PI-like FLC	4	8	5.657	159.995	952.76
VSPID-like FLC	3	7	0.306	130.831	739.32

IV. CONCLUSION

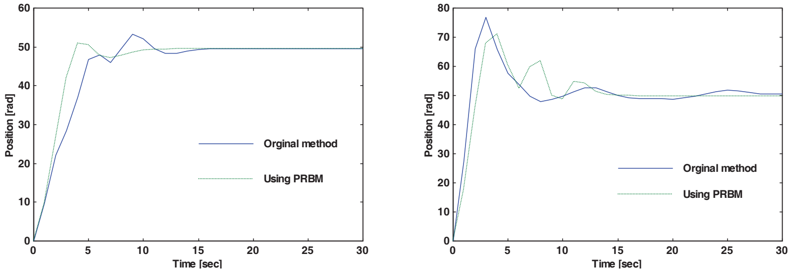
A servomotor is an AC or DC electric motor with a feedback loop to control its speed and position. Armature-controlled DC motors are generally used for positioning applications due to their low cost and the characteristic of the motor speed remaining substantially constant with changes in torque. The DC motor is fitted with an optical encoder for sensing the rotor position. Signals from the optical encoder are fed back to a PID controller. The PID controller sends signals to the motor to rotate to achieve the set final position. The PID controller is programmed with fuzzy logic to improve the speed of response and to control the motor hunting between overshoot and undershoot positions before achieving the set final position. The main issue with various methods of designing PID-like FLCs is that they do not take into account desired performance criteria. Our taxonomy of design methods was used to derive a novel design method, that is, a *performance rule-*

based model. The accuracy analysis shows that the proposed design method is 2.0 seconds faster in rise-time, 2.0 seconds faster in settling-time and, at the same time, it decreases the overshoot by 1.7% compared to the original design methods. Meanwhile, the robustness analysis shows that the proposed design method is 2.0 seconds faster in rise-time, 2.6 seconds faster in settling-time and, at the same time, it decreases the overshoot by 3.4% compared to the original design methods.



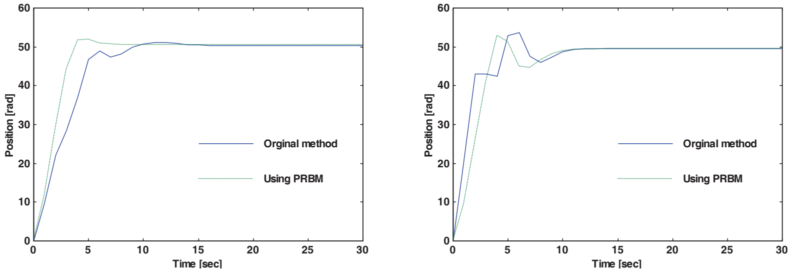
(A) Ideal PID-like FLC.

(B) Incremental PID-like FLC.



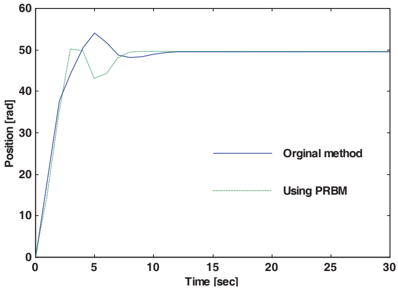
(C) Velocity algorithm of PID-like FLC.

(D) Break-up PID-like FLC.



(E) PD-like parallel with I-like FLC.

(F) PD-like parallel with PI-like FLC.



(G) VSPID-like FLC

Fig. 9. Comparisons between step responses of armature-controlled DC motor system using various methods of designing PID-like FLC and proposed PRBM

REFERENCES

[1] J. Jantzen, "Foundations of Fuzzy Control: A Practical Approach", 2nd Edition, Wiley, 2013.

[2] C. C. Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller-Part I", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 20, No. 2, pp. 404-418, 1990.

[3] C. C. Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller-Part II", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 20, No. 2, pp. 419-435, 1990.

[4] T. E. Marlin, "Process Control: Designing Process and Control Systems for Dynamic Performance", McGraw-Hill, 1995.

[5] J. Nie and D. A. Linkens, "Fuzzy-Neural Control: Principles, algorithms and applications", Prentice Hall, 1995.

[6] 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.

[7] 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.

[8] C. Liaw and J. Wang, "Design and Implementation of a Fuzzy Contrller for a High Performance Induction Motor Drive", IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-21, No. 4, pp. 921-929, 1991.

[9] H. X. Li and H. B. Gatland, "A New Methodology for Designing a Fuzzy Logic Controller", IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-25, No. 3, pp. 505-512, 1995.

[10] K. L. Tang and R. J. Mulholland, "Comparing Fuzzy Logic with Classical Controller Designs", IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-17, No. 6, pp. 1085-1087, 1987.

[11] S. Chiu, S. Chand, D. Moore, and A. Chaudhary, "Fuzzy Logic for Control of Roll and Moment for a Flexbile Wing Aircraft", IEEE Control Systems Magazine, June 1991.

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

[13] J. Kim, J. Park, S. Lee, and E. K. Chong, "A Two-Layered Fuzzy Logic Controller for Systems with Deadzones", IEEE Transactions on Industrial Electronics, Vol. 41, No.2, pp. 155-161, 1994.

[14] Z. Zhao, M. Tomizuka and S. Isaka, "Fuzzy Gain Scheduling of PID Controllers", IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-23, No. 5, pp. 1392-1398, 1993.

[15] H. X. Li and H. B. Gatland, "Conventional Fuzzy Control and Its Enhancement", IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics, Vol. SMC-26, No. 5, pp. 791-797, 1996.

[16] R. Dorf and R. Bishop, "Modern Control Systems", 13th Edition, Pearson, 2016.

[17] F. Golnaraghi and B. C. Kuo, "Automatic Control Systems", 10th Edition, McGraw-Hill Education, 2017.

[18] K. Ogata, "Modern Control Engineering", 5th edition, Pearson India, 2015.

[19] D. P. Kwok, P. Tam, C. K. Li, and P. Wang, “linguistic PID controllers”, in Proc. IFAC 11th Triennial World Congr., Vol. 4, pp. 205-210. 1990.

[20] 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.