A generalized direct approach for designing fuzzy logic controllers in Matlab/Simulink GUI environment

2 authors:

İsmail H. Altas
Karadeniz Technical University

A.M. Sharaf
SHARAF ENERGY SYSTEMS, Incorporated

Please refer this paper as:
I.H. Altas and A.M. Sharaf, "A Generalized Direct Approach for Designing Fuzzy Logic Controllers in Matlab/Simulink GUI Environment", *", International Journal of Information Technology and Intelligent Computing,* Int. J. IT&IC no.4 vol.1, 2007.

A GENERALIZED DIRECT APPROACH FOR DESIGNING FUZZY LOGIC CONTROLLERS IN MATLAB/SIMULINK GUI ENVIRONMENT

Ismail H. ALTAS[1], Adel M. SHARAF[2]

[1]*Department of Electrical and Electronics Engineering*
Karadeniz Technical University, 61080 Trabzon, Turkey
E-mail: ihaltas@altas.org

[2]*Department of Electrical and Computer Engineering*
The University of New Brunswick, Fredericton, NB, E3B 5A3 Canada
E-mail: sharaf@unb.ca

Abstract

A straightforward approach for designing fuzzy logic based controllers in Matlab/Simulink environment is presented in this paper. The approach focuses on the fuzzy rule base construction using the time varying values of control error (e) and its change (Δe) mapped on e-Δe space. In order to show the generalized model simplicity of the proposed approach, a fuzzy logic controller (FLC) is designed in Matlab/Simulink environment. For validation, the generated FLC block is simulated to control five different systems such as a PMDC motor speed control system, a position control system, a radar tracking control system, a synchronous generator voltage control system, and load-frequency control of a two-area power system. The simulation results obtained by FLC are also compared with the results obtained using PID controllers for applicability validation of the developed FLC block in Matlab/Simulink environment.

Keywords: Digital control, Control system human factors, Fuzzy control, Fuzzy logic, Fuzzy sets, Knowledge based systems

1. Introduction

Fuzzy logic (FL) controllers based on fuzzy set theory are used to represent the experience and knowledge of a human operator in terms of linguistic variables that are called fuzzy rules. Since an experienced human operator adjusts the system inputs to get a desired output by just looking at the system output without any knowledge on the system's dynamics and interior parameter variations, the implementation of linguistic fuzzy rules based on the procedures done by human operators does not also require a mathematical model of the system. Therefore a fuzzy logic controller (FLC) becomes nonlinear and adaptive in nature having a robust performance under parameter variations with the ability to get desired control actions for complex, uncertain, and nonlinear systems without the requirement of their mathematical models and parameter estimation. FL based controllers provide a mathematical foundation for approximate reasoning, which has been proven to be very successful in a variety of applications [1]. In modern control techniques, uncertainty and vagueness have a great amount of importance to be dealt with. The use of membership functions quantified from ambiguous terms in fuzzy logic control rules has given a pulse to speed up the control of the systems with uncertainty and vagueness [2-4]. Since the introduction of fuzzy set theory [5] and its application to control systems [6], it has become an important and useful tool in especially controlling non-linear systems. As the computer and chip technology developed, the applications of fuzzy logic based controllers have increased tremendously so that they have been applied in many different systems [1, 7-10].

The main purpose of this study is to develop a simple and direct approach in order to construct a fuzzy rule decision table that is suitable for modeling in Matlab/Simulink GUI environment using simple operational blocks that also works for different systems without the requirement of a redesign process. Many works based on trial and error [11], artificial neural network (ANN) [12], genetic algorithms (GA) [13] based algorithms, and clustering methods [14] have been appeared in literature concerning the rule decision table. The trial and error method uses some initial knowledge about how the system works and estimates the rest of the rule table by trying and error until a working table is obtained. The learning algorithms using ANN techniques also starts with an initial knowledge and fills in the rule decision table by estimating the rest of it using previously trained data. The algorithms using GA methods also consist of a sort of learning system, which learns a set of rules from a set of examples [15]. It has been proven that all these methods work very well. However, it should be noted that they are not just fuzzy systems. They are hybrid systems, which combine other intelligent methods such as neural networks and genetic algorithms with the fuzzy logic. Although the hybrid systems are more powerful and adaptive, they require high level algorithms with time consuming processes that are not desirable in control applications. The FL controllers appeared in literature are mostly modeled for specific applications [16-18] rather than for general cases.

In this study, a direct approach has been applied to obtain an initial three by three rule table, which is then expanded to larger dimensions depending on the number of the fuzzy sets used to partition crisp data spaces. The initial rule table is extracted directly from time response of the system control error and change in this error over one sampling period. Then the rule table is partitioned into fuzzy sublevels in a similar way done for partitioning the space of error and error change. So, the rule table is actually nothing but the partitioned version of the output space into fuzzy subsets with a symmetrical structure.

In order to validate the proposed approach, five different control schemes have been tested in Simulink environment. A permanent magnet DC (PMDC) motor speed control scheme, a PMDC motor driven position control scheme, a radar tracking control scheme for moving targets, a synchronous generator voltage control scheme, and a power system load-frequency control scheme are controlled using the developed FLC block. The results are compared with those obtained by simulating the same schemes with PID controllers.

2. The Physical Systems to Be Controlled

For a physical system operator, one of the most important knowledge is the time response of the system. An operator knows very well how the system responds for a change in reference input or system parameters in order to make necessary adjustment to keep the system going on under normal operating conditions. The time response of a physical system is important since it reflects the effects of the changes either in reference input or system's interior parameters. Therefore, the time response of the operating error and its derivation are usually used as two input parameters to the fuzzy logic controller. Since the error response includes the information about system output, it is used as a bridge connecting system's input to the output over a set of linguistic fuzzy rules.

A linear time invariant (LTI) system can be represented in different ways such as state-space model and transfer functions, which are mostly in first, second, or higher order. Depending on the order of model, the system output for a step input may vary as shown in Fig. 1, resulting in a similar response in error and error change as given in Fig. 2.

The time responses of error signals can be used to represent information related to the system output responses. As the error signals approach zero the output signals move towards reference. Depending on the performances of controllers used, the error signals may or may not become zero. The error signal of a controlled system will be sufficient to derive the controller rules since it contains the necessary information about the outputs. Therefore the error signals shown in Fig. 2 are used as the source of information for constructing the rule base systems for fuzzy logic controller. These signals may represent any types of control error with a step type reference input. In fact, the error signal of a control system with a ramp input will not be much different than that of given in Fig. 3 in spite of the differences between the outputs of the systems with step and ramp inputs.

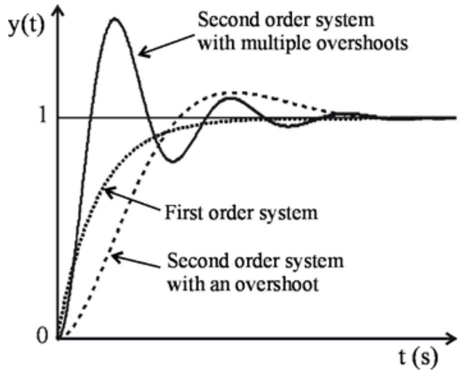


Figure 1. Step responses of the outputs of first and second order systems.

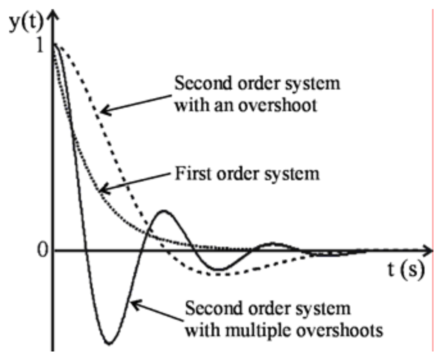


Figure 2. Step responses for the error in first and second order systems.

In a control system, the human operator makes necessary adjustments fast or slow by looking at the system output. A fast action is required if the output is away from the target, i.e. the error is large, while a slower action may be enough for the output closer to the reference target. Therefore, the information about the amount of change in error signal over a sampling period is also required. A plot of the time variation of error versus the time variation of its change, as given in Fig.3, can be used to obtain this information.

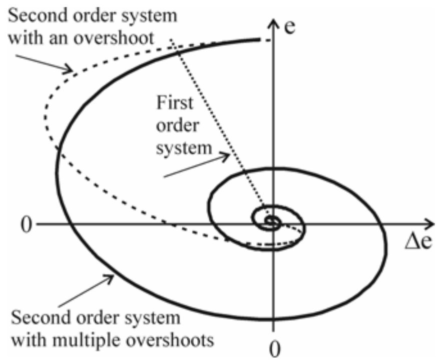


Figure. 3. e-Δe space representing the time response of error, e, versus the time response of the change in error, Δe.

The values of error, e, and its change, Δe, start with larger values and terminate at the origin or near the origin in a controlled system. As given in Fig. 3, the values of e and Δe from the first order system lie only in second quarter of e-Δe space, while the values of e and Δe from a second order oscillatory system travel all four quarters. Since higher order TI systems will also have an oscillating response similar to that of a second order TI system, the plot of e=f(Δe) on e-Δe space of the second order system given with the solid line in Fig. 3 can be used as a general case that valid for both first and higher order systems.

3. The Fuzzy Logic Controller

The studies on FL theory have increased tremendously since its development by Zadeh [5]. The application of FL theory to a control system by Mamdani and his colleges [6] has given a push to the FL real world applications. The theory of fuzzy sets and fuzzy logic is not going to be repeated here. However, the design procedure of the proposed FL based controller is given in detail.

The operation principle of a FL controller is similar to a human operator. It performs the same actions as a human operator does by adjusting the input signal looking at only the system output. A FL based controller consists of three sections namely *fuzzifier, rule base*, and *defuzzifier* as shown in Fig. 4. Two input signals, the main signal and its change for each sampling, to the FL controller are converted to fuzzy numbers first in fuzzifier. Then they are used in the rule table to determine the fuzzy number of the compensated output signal. Finally, the resultant united fuzzy subsets representing the controller output are converted to the crisp values.

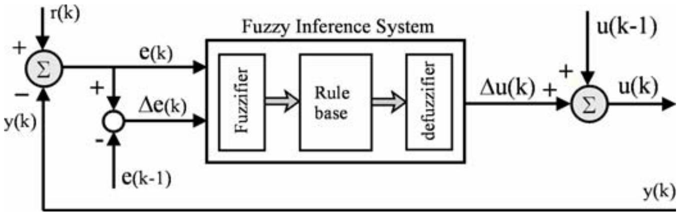


Figure 4. The basic structure of fuzzy logic based controller.

The FL based controller is designed to act as an integrator controller, such that the resultant incremental output Δu(k) is added to the previous value u(k-1) to yield the current output u(k). Recalling the digital solution of an integrator using Euler's integration as,

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

In a digital integration, the term Δu(k) is expressed as

$$\Delta u(k) = K_I T_S e(k) \tag{2}$$

Where K_I is integral constant, T_S is sampling period, and e(k) is the integrated signal. The change Δu(k) on the output of an integrator becomes zero when the input e(k) is zero. Therefore output of an integrator retains the previous value. Hence, (1) can be used for both an integrator and FL controller. The difference between an integrator and FL controller is the method that is used to obtain Δu(k), which is obtained using (2) for an integrator, and using *fuzzy inference system* shown in Fig. 4 for FL controller. The latter is the main subject here and explained next.

3.1. Fuzzy Inference System

As shown in Fig. 4, there are two inputs to the *fuzzy inference system.* One is the control error e(k), which is the difference between the reference signal r(k) and the output signal y(k), the other one is the change in this error Δe(k). These two inputs, defined as in (3) and (4), are first fuzzified and converted to fuzzy membership values that are used in the rule base in order to execute the related rules so that an output can be generated.

$$e(k) = r(k) - y(k) \tag{3}$$

$$\Delta e(k) = e(k) - e(k-1) \tag{4}$$

The fuzzy rule base, which may also be called as the fuzzy decision table, is the unit mapping two crisp inputs, e(k) and Δe(k) to the fuzzy output space defined on the universe of Δu(k). For simplification iteration counter (k) will omitted from now on as long as it is not required to be referred. The time response of the control error e for a step input can be represented by the generalized step response error of a second order system as shown in Fig. 5 where the crisp e and Δe axis are partitioned into seven fuzzy subsets as negative big (NB), negative medium (NM), negative small (NS), zero (ZZ), positive small (PS), positive medium (PM), and positive big (PB). This error signal may have a damped or an oscillatory response with a decaying exponential component. The latter one is considered for constructing the rule table since it includes overshoot effects leading the rule base to represent more generalized cases.

The fuzzy rules represent the knowledge and abilities of a human operator who makes necessary adjustments to operate the system with minimum error and fast response. In order to model the actions that a human operator would decide whether the change, Δu, in the controller output is to be increased or decreased according to the error e and its change Δe, it is necessary to observe the behaviors of the error signal e and its change Δe on different operating regions, as shown by the Roman numbers in Fig. 5. The output Δu from the FL controller is the change that is required to increase or decrease the overall control action to the controlled system. Therefore, the signs of e and Δe are used to determine the signs of Δu, which determines whether the overall control signal is to be increased. The sign of Δu should be positive if u is required to be increased and it should be negative otherwise.

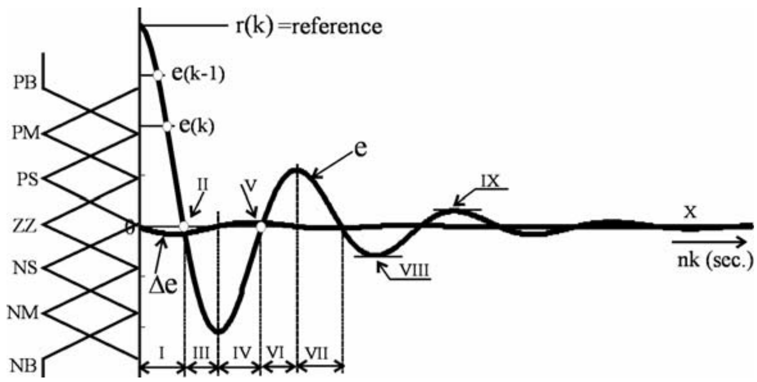


Figure 5. Operating regions and fuzzy partitioning of the time responses of error and error change for a generalized second order system.

This simple rule is applied as follows to determine the sign of Δu.

At region I : e=' + ' and Δe=' – '. The error is positive and decreasing toward zero. Therefore, Δu is set to positive to reduce the error.
At region II : e=' 0 ' and Δe=' – '. The error is zero, but is getting away and increasing in negative direction. Therefore, a negative Δu is assigned to reduce the error.
At region III : e=' – ' and Δe=' – '. The error is negative and increasing. Therefore, Δu is still required to be negative to reduce the error toward zero.
At region IV : e=' – ' and Δe=' + '. The error is still negative, but decreasing. Therefore, negative Δu is kept on to reduce the negative error.
At region V : e=' 0 ' and Δe=' + '. The error is zero, but increasing in positive direction. Therefore, a positive Δu is applied to increase the controlled output so that the voltage error is reduced.
At region VI : e=' + ' and Δe=' + '. The error is positive and increasing. A positive value for Δu will be suitable to reduce the error.
At region VII : A repeat of region I.
At region VIII : e=' – ' and Δe=' 0 '. The error is negative and constant since there is no change. Therefore a negative value for Δu should be assigned to decrease the error.
At region IX : e=' + ' and Δe=' 0 '. The error is positive and constant. Therefore a positive Δu is required to reduce the error.
At region X: e=' 0 ' and Δe=' 0 '. The error is both zero and constant. Therefore Δu is set to zero since no change is required for the control signal.

The signs of Δu in the regions described above as I to X are listed in Table I, which can be summarized as follows:

IF e is zero **THEN** Δu takes the sign of Δe, **ELSE** Δu takes the sign of e. (5)

Table 1. The signs Of basic control actions

	Operating regions									
	I	II	III	IV	V	VI	VII	VIII	IX	X
e	+	0	-	-	0	+	+	-	+	0
Δe	-	-	-	+	+	+	-	0	0	0
Δu	+	-	-	-	+	+	+	-	+	0

Both Fig. 5 and Table I show that each one of e, Δe, and Δu has three different options for the signs to be assigned. They are either positive or negative if not zero. Keeping in mind these three options, which are represented by three fuzzy sets namely positive (P), negative (N), and zero

(Z), an initial rule decision table with nine rules can be formed as shown in Table II, where the main part without shading represents the rules in terms of the signs of Δu.

Representing the input crisp variables e and Δe by three fuzzy sets, P, N, and Z means that these input spaces are partitioned into three fuzzy regions each yielding a fuzzy output space with nine rules maximum as given in Table II. A nine-rule fuzzy decision table may be sufficient for some applications. However, many applications require more rules than nine. In order to construct a fuzzy rule decision table with more than nine rules, the input spaces must be partitioned into more than three regions each.

Table 2. Initial rule table.

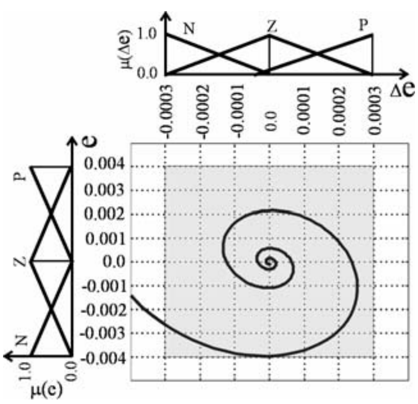
e \ Δe	**N**	**Z**	**P**
P	P	P	P
Z	N	Z	P
N	N	N	N

A plot of e(t) vs Δe(t) can be used to determine the regions for partitioning as shown in Fig. 6 where e(t) and Δe(t) axes are partitioned into three and five regions in Figs. 6a and 6b, respectively. Triangular type membership functions are used for partitioning the crisp universes into fuzzy subsets. Different membership functions such as Gaussian, trapezoidal, and bell could have also been used. Each one of these membership functions has its own effects on the FLC output [11]. However, triangular membership functions are more convenient for expressing the concept because it is easier to intercept membership degrees from a triangle. Therefore the following function is used to represent the fuzzy triangular membership functions.

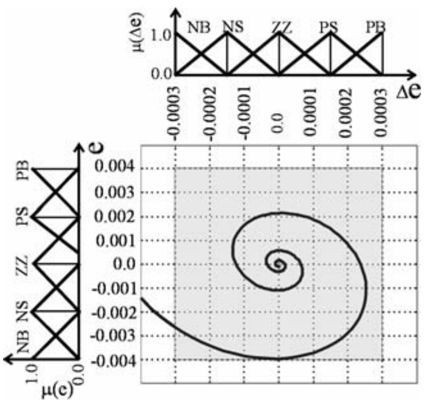
$$\mu(x) = \max\left[\min\left(\frac{x - x_1}{x_2 - x_1}, \frac{x_3 - x}{x_3 - x_2}\right), 0\right] \tag{6}$$

Where, x is the crisp values from one of the three universes e, Δe, and Δu. x_1 is the left end point on the corresponding crisp universe as the x_2 and x_3 are the crisp points corresponding peak and right end points, respectively.

The plot of e vs Δe as in Fig. 6 is useful in partitioning since the upper and lower limits of the values of e and Δe are shown separately enabling one to set different boundary limits for e and Δe. In this case the scaling of the fuzzy sets representing the partitioning will also be different for e and Δe. This is important because of the differences in minimum and maximum values of e and Δe as shown in Figs 5 and 6. If the same scaling of the fuzzy sets is used for both e and Δe, the fuzzy values of Δe do not change much compared to those of e as shown in Fig. 5 where Δe swings around zero with membership degrees higher in the fuzzy set ZZ and very lower in NS and PS, and without any membership degrees in the other fuzzy sets. Therefore Fig. 6 is more convenient to be used for fuzzy partitioning of crisp axes. Besides, Figs. 6a and 6b both have the similar appearance with Table II. If Fig. 6a and Table II are compared, it can readily be seen that horizontal and vertical axes in Fig. 6a corresponds to the first definition row and column in Table II, respectively. Since inner part of Table II contains the fuzzy rules that are represented by fuzzy sets defined on the universe of discourse Δu. The e-Δe space in Fig. 6a can be used to represent the universe of discourse Δu, which is the output space of the FL controller. Therefore, using the expression (5), the space of e-Δe representing Δu can be partitioned into the regions positive (P), negative (N), and zero (Z) as given in Figs. 7a and 7b.



(a)

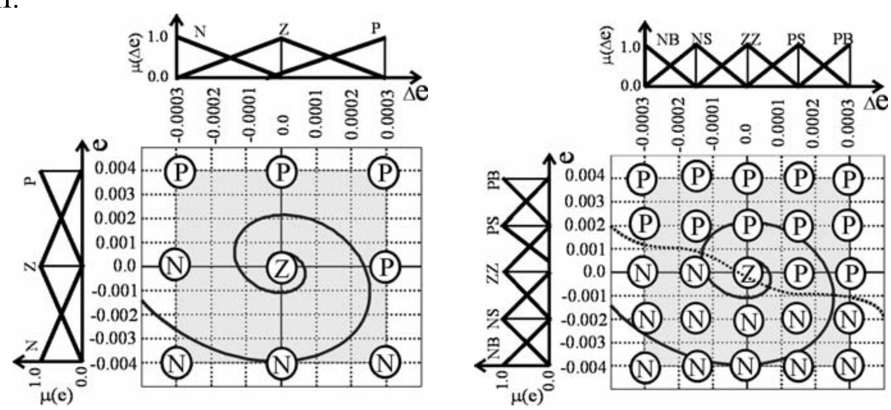


(b)

Figure 6. Partitioning input spaces into (a) three and (b) five regions.

The fuzzy rule decision table used in this study consists of twenty-five rules obtained by portioning e and Δe axes into five regions each as shown in Figs. 6b and 7b, which are going to be used to explain the construction of the rule decision table.

As it can be seen in Fig. 7b, e-Δe space representing the Δu space has been partitioned into three regions namely P, N, and Z, while each one of e and Δe axes are partitioned into five regions as NB, NS, ZZ, PS, and PB. Similar to e and Δe axes, Δu space can also be partitioned into five or more regions. A closer look at the partitioned e and Δe axes show that there is a transition region between the neighboring fuzzy sets. It can also be seen that a fuzzy set zero is used to represent the transition between positive and negative fuzzy values. This transition region is the nature of fuzzy sets, and is called as shaded region. A similar transition should also be established between the positive (P) and negative (N) regions in Δu space where the positive and negative values are mostly located toward upper right and lower left corners, respectively, as separated by a dotted line in Fig. 7b. Therefore a shaded region represented by the fuzzy set zero (ZZ) can be placed on the main diagonal separating the positive and negative parts as given in Table III.



(a) (b)

Figure 7. Initial rule assignments in the output space, Δu. (a). The input space partitioned into three regions. (b). The input space partitioned into five regions.

If the output space, Δu, is to be partitioned into more than three regions as it has been done for e and Δe axes, then the positive and negative regions should be divided into sub-regions such as NB, NS, PS, and PB. The inner partitioning of positive and negative regions should be done in such a way that the order from NB to PB becomes similar to that of e and Δe axes. Therefore, if Δu surface is partitioned into five regions consisting of the fuzzy sets NB, NS, ZZ, PS, and PB, Table III can be converted to Table IV where the regions from NB to PB have transitions as shown in Figs. 6 and 7, which e and Δe axes are partitioned into 5 subsets. A number from 1 to 25 has been assigned to each rule in Table IV.

Table 3. Initial fuzzy rule decision table

		Δe				
		NB	NS	ZZ	PS	PB
e	PB	**Z**	P	P	P	P
	PS	N	**Z**	P	P	P
	ZZ	N	N	**Z**	P	P
	NS	N	N	N	**Z**	P
	NB	N	N	N	N	**Z**

Table 4. Modified fuzzy rule decision table

		Δe				
		NB	**NS**	**ZZ**	**PS**	**PB**
e	**PB**	ZZ 1	PS 2	PS 3	PB 4	PB 5
	PS	NS 6	ZZ 7	PS 8	PS 9	PB 10
	ZZ	NS 11	NS 12	ZZ 13	PS 14	PS 15
	NS	NB 16	NS 17	NS 18	ZZ 19	PS 20
	NB	NB 21	NB 22	NS 23	NS 24	ZZ 25

3.2. The Inference and Fuzzy Reasoning

The final control action is the crisp output that is defuzzified from the resultant fuzzy values of the fuzzy rule base. The fuzzy output of the rule base is obtained by triggering the active rules for the k^{th} sampling instant corresponding to the values e(k) and Δe(k) as shown in fig. 8. For any point (e(k), Δe(k)) on the trajectory plot of e(k) vs Δe(k), there are maximum two intercepting fuzzy sets on each one of the universes e(k) and Δe(k). Thus, for any sampling instant, the value of e(k) activates only one or two fuzzy sets in the universe of e. Similarly, the value of Δe(k) for the k^{th} sampling instant also activates only one or two fuzzy sets in the universe of Δe(k).

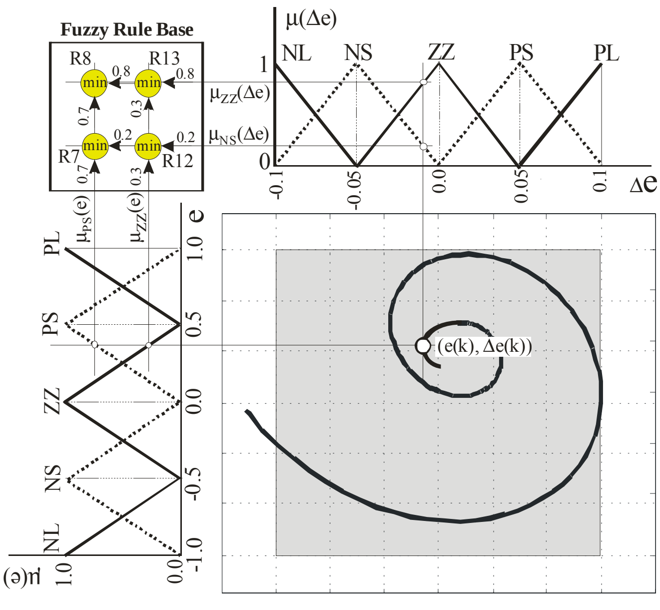


Figure 8. An illustration of fuzzy reasoning.

The point $(e(k), \Delta e(k))=(0.3, -0.01)$ on the trajectory plot shown in Fig. 8 intercepts with the fuzzy sets ZZ and PS in the universe of e and with the fuzzy sets NS and ZZ in the universe of Δe. Therefore, the following rules are activated for the given data point $(e(k), \Delta e(k))$.

Rule 7 (R7):	if e is PS and Δe is NS then Δu is ZZ
Rule 8 (R8):	if e is PS and Δe is ZZ then Δu is PS
Rule 12 (R12):	if e is ZZ and Δe is NS then Δu is NS
Rule 13 (R13):	if e is ZZ and Δe is ZZ then Δu is ZZ

The membership degrees of e in the fuzzy sets PS and ZZ are obtained as $\mu_{ps}(e)=0.7$ and $\mu_{zz}(e)=0.3$, respectively, and the membership degrees of Δe in the fuzzy sets NS and ZZ are $\mu_{ns}(\Delta e)=0.8$ and $\mu_{zz}(\Delta e)=0.2$, respectively. The triangular membership function (6) can be used to obtain these memberships. Then the application of the min operator results in the following membership values from each active rule to be used in the output space Δu.

$$\mu_{R7}(\Delta u)=\min(\mu_{PS}(e), \mu_{NS}(\Delta e))=\min(0.7, 0.2)=0.2 \tag{7}$$
$$\mu_{R8}(\Delta u)=\min(\mu_{PS}(e), \mu_{ZZ}(\Delta e))=\min(0.7, 0.8)=0.7 \tag{8}$$
$$\mu_{R12}(\Delta u)=\min(\mu_{ZZ}(e), \mu_{NS}(\Delta e))=\min(0.3, 0.2)=0.2 \tag{9}$$
$$\mu_{R13}(\Delta u)=\min(\mu_{zz}(e), \mu_{zz}(\Delta e))=\min(0.3, 0.8)=0.3 \tag{10}$$

The resultant fuzzy outputs in the universe of Δu for these four active rules are depicted in Fig. 9 with the membership degrees shown as μ_{r7}, μ_{r8}, μ_{r12},and μ_{r13}.

3.3. Defuzzification

The resultant membership values of the active rules determine the weights of the fuzzy sets in the universe of Δu as shown in Fig. 9 with the shaded parts of the fuzzy sets. Then the averaged value of the union of these shaded fuzzy sets is used to obtain final crisp output as $\Delta u(k)$. This final process is called defuzzification of the fuzzy output. Several defuzzification methods have been applied in literature. However, the method called *the center of area* is widely used in fuzzy logic control applications. The use of *the center of area* method yields the following.

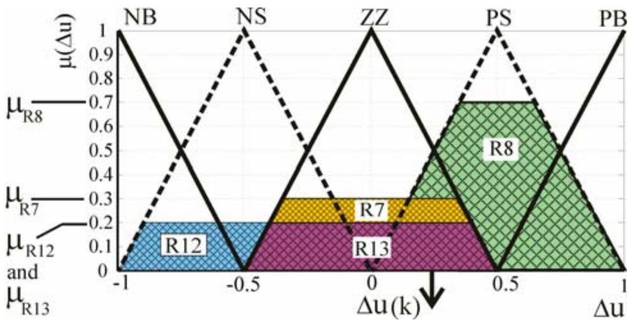


Figure 9. Membership functions used to represent fuzzy partitioning in the universe of Δu.

$$\Delta U_R(k) = \frac{\sum_{i=7,8,12,13} \mu_{Ri}(\Delta u_R)\Delta u_R(Ri)}{\sum_{i=1}^{4} \mu_i(uV_R)} \tag{11}$$

$$\begin{aligned} \Delta U_R(k) &= \frac{0.2(0.0) + 0.7(0.5) + 0.2(-0.5) + 0.3(0.0)}{0.2 + 0.7 + 0.2 + 0.3} \\ &= \frac{0.25}{1.4} = 0.17857 \end{aligned} \tag{12}$$

Where, Δu(Ri) is the crisp Δu value corresponding to the maximum membership degree of the fuzzy set that is an output from the rule decision table for the rule Ri. For example, for the rule 12, the output fuzzy set is NS and the crisp value of Δu(R12) is (-0.5), because the membership degree of (-0.5) in the fuzzy set NS is 1.0 as shown in Fig. 9.

4. Modeling the FLC for Matlab/Simulink Environment

The model development of the FLC for Matlab/Simulink environment is described in this section using the information given in previous sections. The first step here is to generate a fuzzy membership function for partitioning the input and output spaces into fuzzy subsets. As given in Fig. 4, the process of an FLC can be summarized in three steps: Fuzzification, rule based fuzzy processing, and defuzzification. Due to the reason explained right before (6), triangular fuzzy membership functions were used in both fuzzification and defuzzification stages. The Simulink model of the triangular fuzzy membership function is shown in Fig. 10 where x_1, x_2, and

x_3 are the crisp parameters used to define the location and shape of the triangle. The input x is the crisp variable whose membership value on this triangle fuzzy subset is the output mu(x) in Fig. 10.

Fuzzification stage and the first part of rule based fuzzy processing unit of the FLC are given in Fig. 11. The crisp inputs e(k) and Δe(k) are converted to fuzzy membership values on the fuzzy subsets NB, NS, ZZ, PS, and PB. Each fuzzy subset (FS) is represented by a triangular membership function as described in Fig. 10 and shown by blocks in Fig. 11. The letters E and DE in Fig. 11 are used to indicate whether the elements belong to error, e, or its change Δe, respectively.

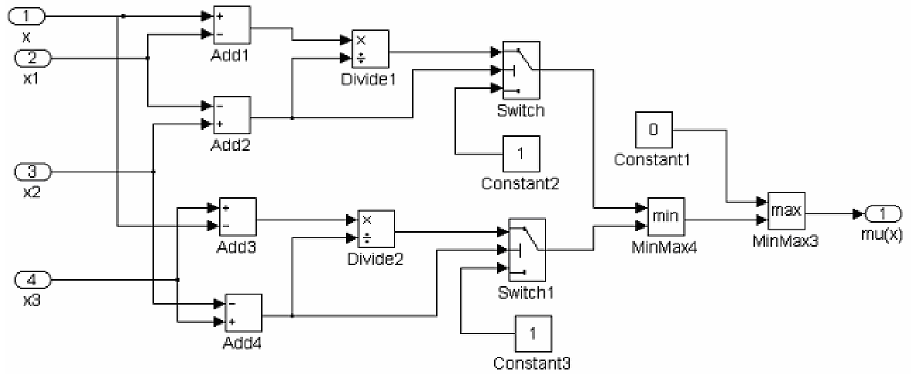


Figure 10. Simulink model of the triangular fuzzy membership function.

The Boolean operator "*min*" is used for the verbal connector "*and*" to simulate the *input space of the rules* that have the structure as in expression (12). The input space of the fuzzy rules used here has two inputs and one output as

$$\textbf{\textit{If}} \text{ e is A } \textbf{\textit{and}} \text{ } \Delta\text{e is B } \textbf{\textit{then}} \text{ } \Delta\text{u is C.} \qquad (12)$$

Where A, B, and C in (12) represent any one of the fuzzy subsets NB, NS, ZZ, PS, and PB defined before. The input space in (12) is the part that represented by the expression (e is A ***and*** Δe is B). Therefore the *min* operator in Simulink Block Library is used to model the input spaces of 25 rules used by FLC. The outputs of the "*min*" operators indicate the strength (membership degree) of the rules in the output space Δu. The implementation of the rule input space by the expression (e is A ***and*** Δe is B) is nothing but the fuzzification of the two crisp inputs e and Δe for all the rules. The process of fuzzufication of the input space with 25 rules is shown in Fig. 11.

The membership degrees obtained as depicted in Fig. 11 are multiplied by the crisp values of each corresponding fuzzy subset in the output space Δu as shown in Fig. 13. The crisp values of the fuzzy subsets used in this multiplication process are the values that have maximum membership degree of 1.0 in the corresponding fuzzy subset. In other words, these crisp values indicate the peak locations of the triangular fuzzy subsets. Actually this multiplication process represents the products in the nominator of defuzzification method called *centre of area*. Then the sum of these products is divided by the sum of membership values obtained as in Fig. 12.

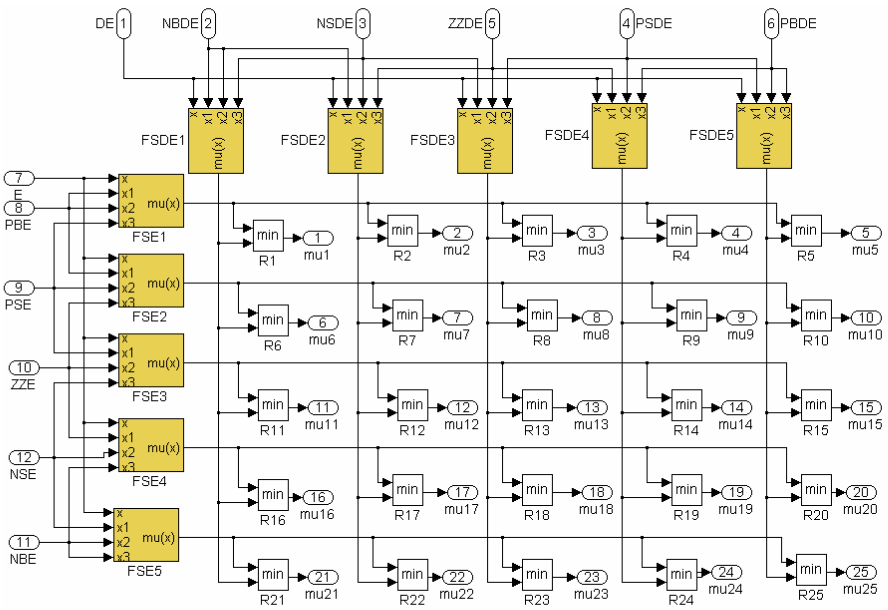


Figure 11. The Simulink model of the Fuzzification process.

Simulation model of the *centre of area* method is depicted in Fig. 13. This is the final stage of the FLC to generate the required change in control signal for the current k*th* sampling. in order to prevent zero division that causes simulation problems such as delayed simulation time and simulation hanging, a signal route has been provided.

A general overlooked view of the FLC is given in Fig. 14 where the processes from inputs e and Δe to output Δu are shown. The input data blocks to represent fuzzy membership functions for the error e, error change Δe, and the controlled output change Δu are shown in Fig. 14. The user is able to edit and change the parameters of the membership functions on this stage without going into the details of the FLC. The definition parameters of the fuzzy membership functions are problem dependable and must be set accordingly. Once the possible maximum and minimum values of the inputs and output signals assigned, the other sub values can be placed in between these limits. If the initial settings are to be used, then some gain blocks can be used to match the signals with the predefined membership parameters.

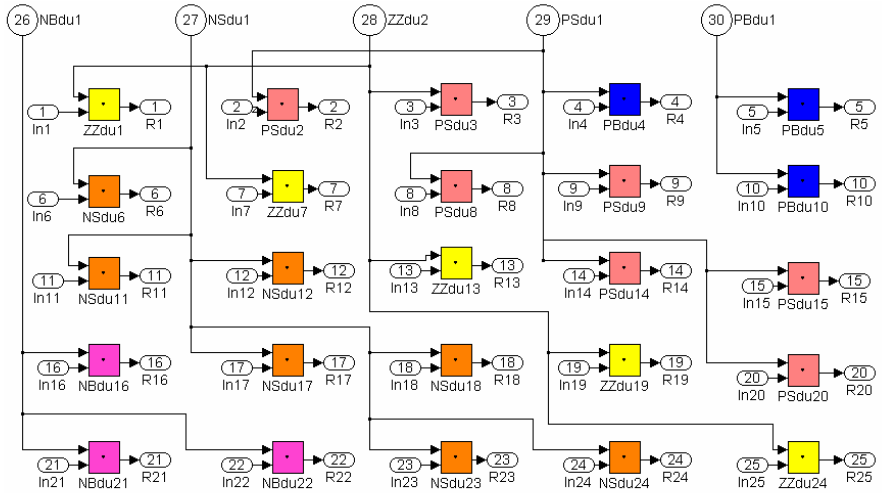


Figure 12. The Simulink model of the fuzzy rules.

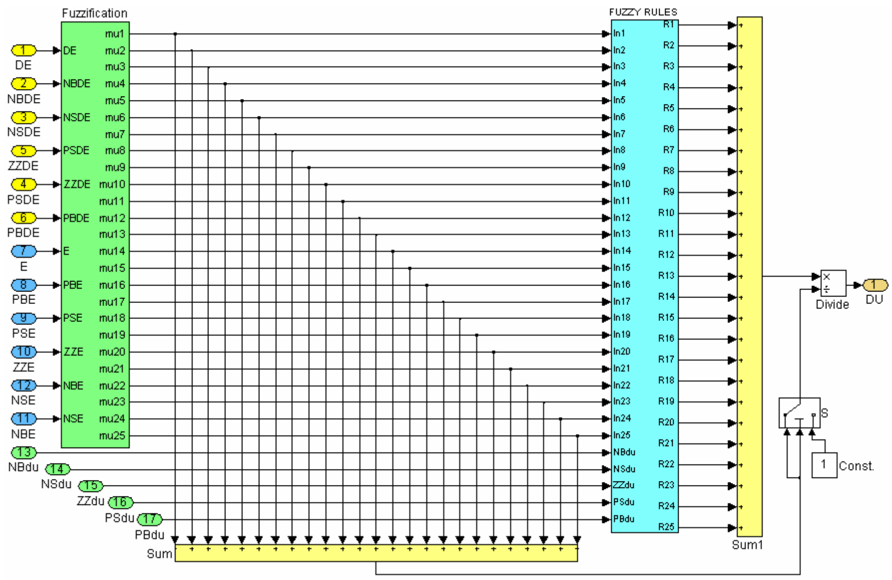


Figure 13. Fuzzy reasoning representing the process from fuzzification to defuzzification.

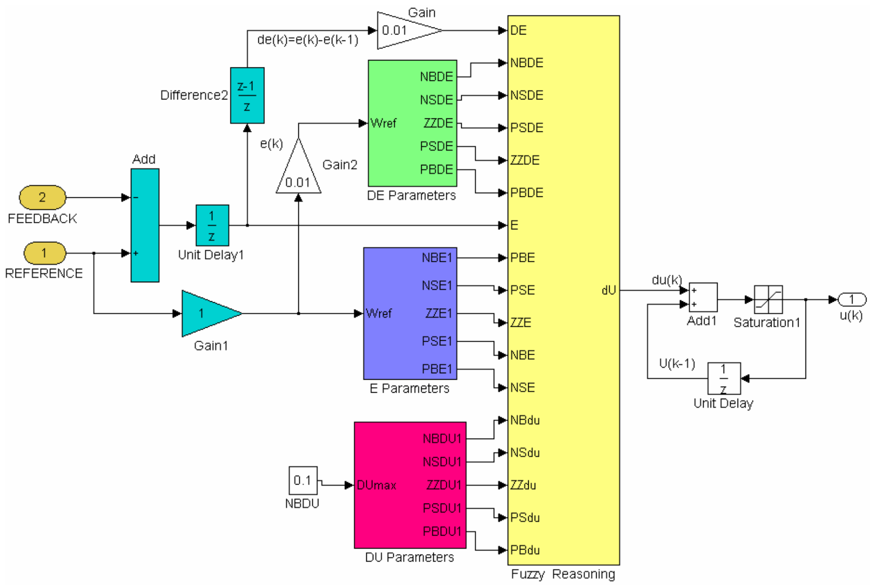


Figure 14. The input and output units of the FLC.

5. Applications and Results

This paper presents a simple approach to design FL based controllers for Matlab/Simulink environment. Since it is called *a generalized direct approach,* it should work for systems with different orders and characteristics. Therefore a FL based controller has been designed using the proposed approach and represented by a simple block with input and output connection ports so that it can be inserted into different systems as a fully functioning controller. Thus, the designed FLC block is used to simulate five different control systems namely a permanent magnet DC (PMDC) motor speed control system, a PMDC motor driven position control system, a radar tracking control system, a synchronous generator terminal voltage control system, and a two area load-frequency control power system. The modeling details of the PMDC motor speed control system are given in [8, 11] as the modeling details for position control and radar tracking systems are given in [19] and [20], respectively. The modeling details of the synchronous generator terminal voltage control and two area power system load-frequency control systems can be found in [21]. Since the modeling processes of the systems used here have been done before and given in the references as just indicated, these modeling details are not repeated in the paper.

Five different controlled systems are simulated using both FLC and PID controllers for comparison and validation purposes. Matlab/Simulink model block diagram of the PMDC motor speed control system is shown in Fig. 15 and the related simulation results are given in Figs. 16 and 17. For the simulation with PID controller, the FLC block in Fig. 15 is replaced by a PID controller block with the proportional gain (K_P) set to 0.001, integral gain (K_I) set to 0.01, and derivative gain (K_D) set to 0.

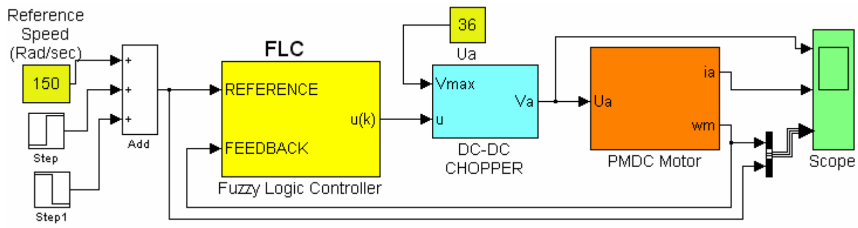


Figure 15. Using FLC block for PMDC motor speed control.

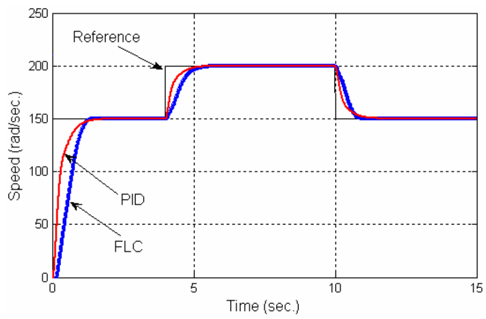


Figure 16. PMDC motor speed control responses using FL and PID controllers.

The speed control responses from both FL and PID controllers are plotted on the same graph for better comparison. Although there are some differences during the transient period of speed, both controllers gives almost the same settling time and steady-state operation by responding step changes in reference input as depicted in Fig. 16. However, each controller gives different current responses during the transients. As depicted in Fig. 17, the current has less ripple magnitudes during the transients with FLC than it has with the PID controller.

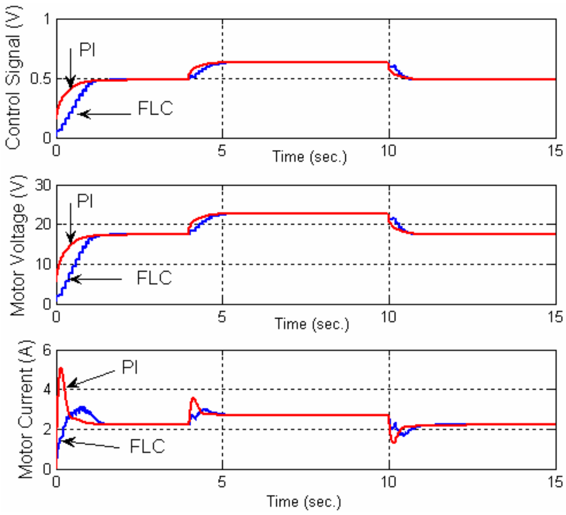


Figure 17. Responses from the PMDC motor speed control system using FL and PID controllers.

The operational block diagram of the position control system using FLC is shown in Fig. 18. The FLC block is replaced by a PID controller block with the parameters set to K_P=0.08, K_I=0.001, and K_D=0 for when PID is used. The simulation results for the position control system are given in Figs. 19 and 20, where both controller results in similar settling time and steady state responses with the differences during transients. The position response from FLC seems to be slower than that of obtained with PID when step changes occur in reference position input level. However, with the FLC controller, the control signal, motor voltage; motor current and motor speeds have considerable reduced magnitudes during transient periods.

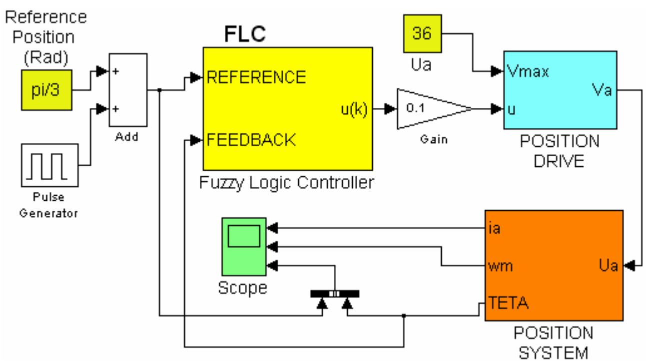


Figure 18. Using FLC block for PMDC motor driven position control.

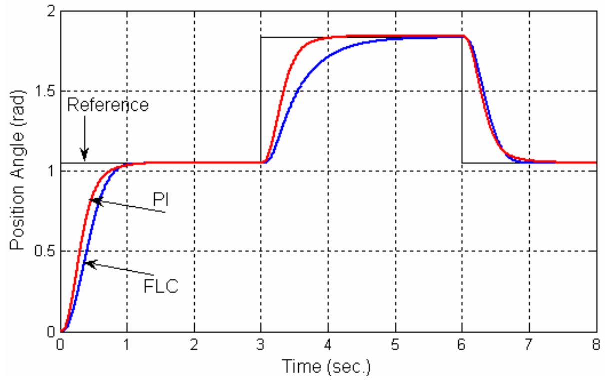


Figure 19. Position control performances of both FL and PID controllers.

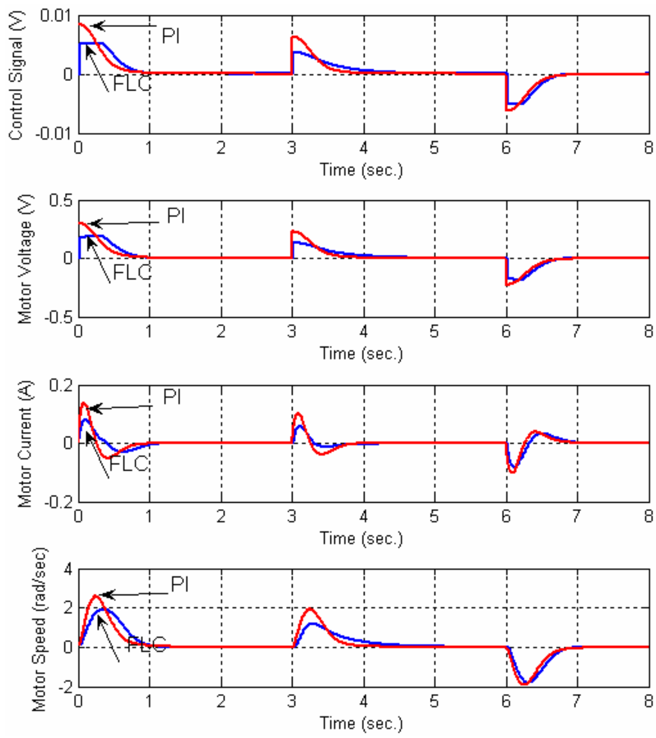


Figure 20. Responses from the position control system using FL and PID controllers.

The operational Simulink block diagram of the radar tracking system with the FL controller is given in Fig. 21. The simulation results of this system are shown in Fig. 22 for both FLC and PID controllers. In the radar tracking system, it is assumed that the radar is tracking moving objects so that after catching the object, the radar then locks and tracks it on a surface with 360° rotating ability. Therefore, the reference input to the radar tracking system is a ramp input as a function of the simulation time. It is also assumed that the moving object slows down and speeds up again during the simulation as shown in Fig. 22, where both FL and PID controllers are tracking the object. However, the FLC has a better and close tracking performance even if it has some small oscillations. In order to simulate the radar tracking system with the PID controller, the FLC block is replaced by a PID block with the parameters set to K_P=1.1, K_I=0.8, and K_D=0.

The proposed FLC block for Matlab/Simulink environment is also validated by simulating two power system examples from [21]. These examples are originally simulated in [21] using PID controllers, which are replaced by the proposed FLC here in order to show the simplicity of using the FLC block just as a simple controller that can be inserted in a system model just like it is done with the classical PID controllers.

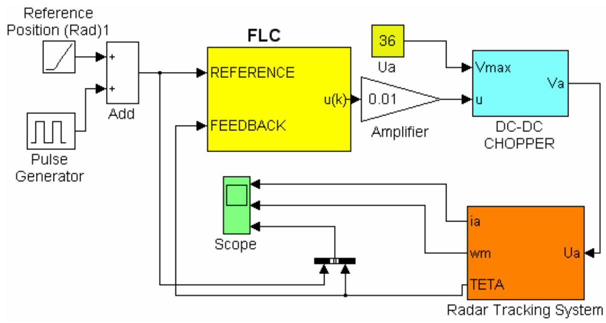


Figure 21. Using FLC block for PMDC motor driven radar control system.

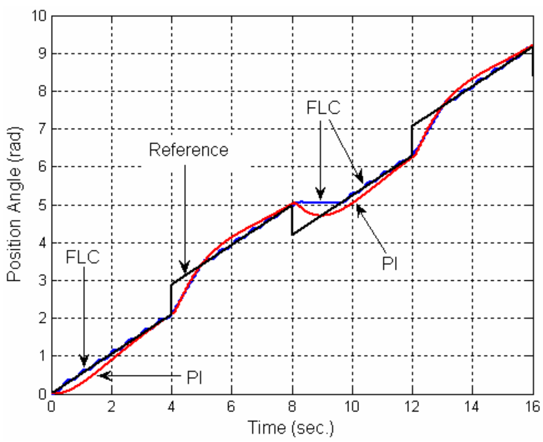


Figure 22. Radar tracking performances of both FL and PID controllers.

One of the power system control example is an automatic voltage regulator (AVR) control system used to keep the terminal voltage of the synchronous generators constant. The block diagram of the FL based voltage control system is given in Fig. 23. This is the same system used in [21], except the replacement of PID block with the parameters, K_P=1, K_I=0.25, and K_D=0.28, by the FLC. The FLC results in a slower transient response for the system used as shown in Fig. 24. However, both controllers give the same steady state operating response.

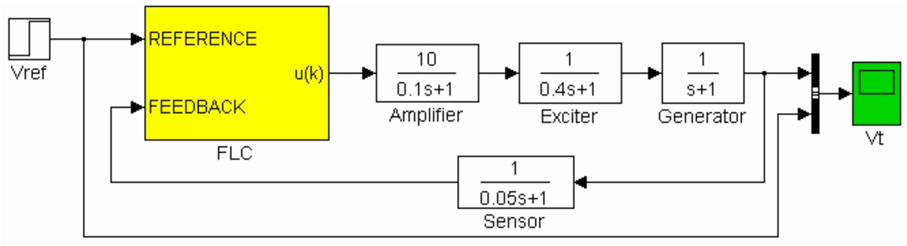


Figure 23. Using FLC block as a terminal voltage controller for a synchronous generator.

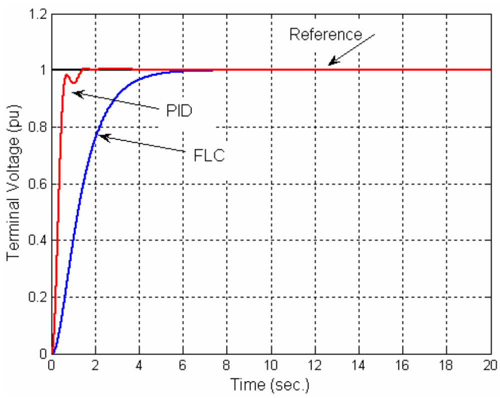


Figure 24. Terminal voltage control responses of a synchronous generator using both FL and PID controllers.

The second power system control example is a two area load-frequency control stem, which is also taken from [21] for comparison. The PID controller with the parameters K_P=0, K_I=0.3, and K_D=0, is replaced by the FLC as shown in Fig. 25, yielding the simulation results given in Fig. 26 where the FLC responses for both areas have less oscillations and shorter settling time.

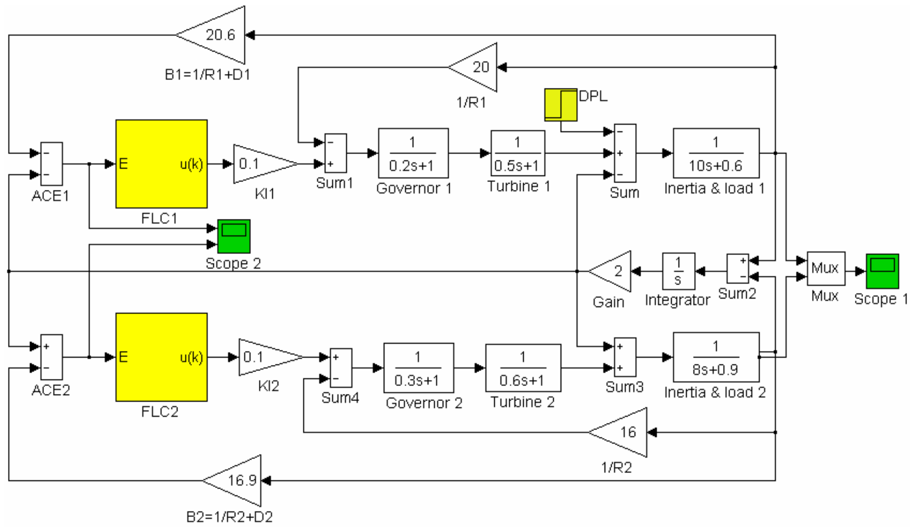


Figure 25. Using FLC blocks as AGC controller in power systems.

Actually, it should be noted that the comparison made here is not given to show which controller is better. The comparison is given to show that the proposed FLC model for Matlab/Simulink environment works and gives acceptable results without redesigning it for each separate system.

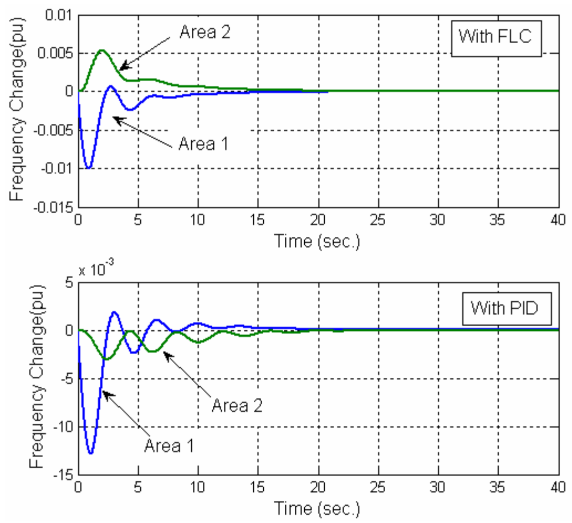


Figure 26. Load-frequency controller responses in a two area power system using both FL and PID controllers.

6. Conclusions

A generalized direct approach for designing FL based controllers for Matlab/Simulink environment is presented in this paper. The approach is based upon using e-Δe space consisting of the plot e(t) vs. Δe(t). Since the error response of an under damped second order system is used as the base system to derive the fuzzy rules, the FL controller is capable of handling different types of systems with different orders. The rule table is formed in such a way that the table itself became fuzzy with a zero main diagonal separating positive and negative fuzzy rules that are also partitioned into sub fuzzy rules among themselves. The proposed approach is used to generate a generalized FLC Simulink block that can be used in different type system control schemes by just inserting the block to the model of the system to be controlled. The generated FLC block requires only small gain adjustments to be adapted for different systems. In order to validate the developed FLC model, five different systems, which are a PMDC speed control system, a position control system, a radar tracking system, a voltage regulator system, and a two area load-frequency control system, are controlled using the proposed FLC and the results are compared with those obtained using PID controllers. The results from five simulations showed good and acceptable performances for the FLC. The purpose is not to compare the FLC with PID to see which one gives better performance, but to observe whether the FLC gives acceptable results as a PID controller does. Therefore the comparisons are not focused on to observe which one gives better response. However, it was observed that, the FLC gives lower transient magnitudes in some system variables such as current and input voltage as in position and speed control systems.

Acknowledgements

I. H. Altas thanks the Scientific and Technological Research Council of Turkey for the financial support during this work.

References

[1]. Maiers J. and Sherif Y. S., 1985, *Applications of Fuzzy Set Theory*, IEEE Trans. Systems, Man, and Cybernetics, Vol. SMC-15, No. 1, pp.175-189.

[2]. Zadeh L. A., 1973, *Outline of a New Approach to the Analysis of Complex Systems and Decision Processes*, IEEE Trans. Systems, man, and Cybernetics, Vol. SMC-3, No. 1, pp.28-44.

[3]. Lee C. C., 1990, *Fuzzy logic in control systems: fuzzy logic controller I*, IEEE Trans. Systems, man, and Cybernetics, Volume: 20 Issue:2, pp. 404 –418.

[4]. Lee C. C., 1990, *Fuzzy logic in control systems: fuzzy logic controller II*, IEEE Trans. Systems, man, and Cybernetics, Volume: 20, Issue:2, pp. 419-435.

[5]. Zadeh L. A., 1965, *Fuzzy Sets*, Information and Control 8, pp.338-353.

[6]. Mamdani E. H. and Assilian S., 1975, *An experiment in Linguistic Synthesis With a Fuzzy Logic Controller*, Int. J. Man-Machine Studies 7, pp.1-13.

[7]. Altas I. H. and Sharaf A. M., 1992, *A Fuzzy Logic Power Tracking Controller For A Photovoltaic Energy Conversion Scheme*, Electric Power Systems Research, Vol.25, No.3, pp.227-238.

[8]. Eminoglu I.and Altas I. H., 1996, *A Method To Form Fuzzy Logic Control Rules For A pmdc Motor Drive System*, Electric Power Systems Research, 39 (1996), pp. 81-87.

[9]. Altas I. H., 1997, *Stabilization of a two-link robot manipulator using fuzzy logic*, The proceedings of the 3rd International Mechatronic Design and Modeling Workshop, 15-18 September, Middle East Technical University, Ankara, Turkey, pp.41-51.

[10]. Altas I. H. and Sharaf A.M., 1994, *A Novel Fuzzy Logic Controller For Maximum Power Extraction From a PV Array Driving a Three-phase Induction Motor*, 7th Mediterranean Electrotechnical Conference (MELECON'94), Antalya, Turkey, April 12-14, pp. 853-856.

[11]. Altas I. H, 1998, *The Effects of Fuzziness in Fuzzy Logic Controllers*, 2nd International Symposium on Intelligent Manufacturing Systems, August 6-7, Sakarya University, Sakarya, Turkey, pp.211-220.

[12]. Nauck D. and Kruse R., 1993, *A Fuzzy Neural Network Learning Fuzzy Control Rules and Membership Functions by Fuzzy Error Backpropagation*, Proc. IEEE Int. Conf. Neural Networks ICNN'93, San Francisco, Mar. 28 - Apr. 1, pp. 1022-1027.

[13]. Herrera F., Lozano M., and Verdegay J. L., 1993, *Generating Fuzzy Rules from Examples using Genetic Algorithms,* Dept. of Computer Science and Artificial Intelligence, Technical Report #DECSAI-93115, October.

[14]. Klawonn F. and Keller A., 1997, *Fuzzy Clustering and Fuzzy Rules*. Proc. 7th Intern. Fuzzy Systems Association World Congress (IFSA'97) Vol. I, Academia, Prague, pp.193-198.

[15]. Gonz'alez A. and P'erez R., 1996, *A learning system of fuzzy control rules based on genetic algorithms*, Genetic Algorithms and Soft Computing, (F. Herrera, J.L. Verdegay eds.) Physica-Verlag, Studies in Fuzzyness, vol. 8, pp. 202-225.

[16]. Hugo E. M. and du Plessis J. J., 2001, *Automated Multi-mode Fuzzy Logic Controller Design*, Proc. IEEE Int. Symp. Intelligent Control, pp. 376-382.

[17]. Lai Y-S. and Lin J-C., 2003, *New hybrid fuzzy controller for direct torque control induction motor drives*, IEEE Trans. Power Electronics, Volume 18, Issue 5, Sept. pp. 1211 – 1219.

[18]. Soucy M. and Payeur P., 2005, *Flexible fuzzy logic control for collision-free manipulator operation*, IEEE International Conference on Mechatronics and Automation, Volume 2, 29 July-1 Aug. pp.723-728.

[19]. Altas I. H., 1997, *A Fuzzy Logic Controlled Tracking System For Moving Targets*, 12th IEEE International Symposium on Intelligent Control, ISIC'97, July 16-18, Istanbul, Turkey, pp.43-48.

[20]. Altas I. H., 1997, *A fuzzy logic based door position control system*, The proceedings of the 3rd International Mechatronic Design and Modeling Workshop, 15-18 September, Middle East Technical University, Ankara, Turkey, pp.107-116.

[21]. H. Saadat, 1999, “Power System Analysis” McGraw Hill Book Company, New York.