A Performance Comparison for Various Methods to Design the Three-Term Fuzzy Logic Controller

Essam Natsheh† and Khalid A. Buragga†

† College of Computer Sciences & IT, King Faisal University, Hofuf 31982, Saudi Arsabia

Summary
A three-term (*proportional plus integral plus derivative*, PID) fuzzy logic controller (FLC) algorithm have been and continue to be a very active and fruitful research field since early pioneered work on fuzzy controller. In this study, we propose taxonomy of various methods to design three-term FLC. Two applications representing second-order systems and third-order systems were used to analyze the performance of the various design methods and compare their performance. The analysis shows that the *PD-like FLC parallel with PI-like FLC* design method are quite efficient and superior to other design methods with respect to transient response, accuracy, and robustness to varying of defuzzification methods.

Key words:
Three-term controller, PID controller, fuzzy systems, fuzzy logic controller

1. Introduction

Fuzzy control is a control method based on *fuzzy logic*. Just as fuzzy logic can be described simply as "computing with words rather than numbers"; fuzzy control can be described simply as "control with sentences rather than equations". A fuzzy controller can include empirical rules, and that is especially useful in operator controlled plants. A comprehensive review of the classical design and implementation of the fuzzy logic controller can be found in the literature [1], [2], [3], [4], [5].

A three-term (*proportional plus integral plus derivative*, PID) fuzzy logic controller (FLC), or simply PID-like FLC, algorithms have been and continue to be a very active and fruitful research field since Mamdani and Assilian pioneering work on fuzzy controller in 1974 [3]; the controller is shown in Fig. 1. The impetus behind this continuity of the subject lies largely in the fact the conventional PID algorithms has been successfully used in the process industries since 1940s [7] and remains the most commonly used algorithm today, while numerous application of fuzzy logic control (FLC) have immerged covering a wide range of practical areas [8] and that many software and hardware products for fuzzy control have been commercialized during the last few years.

Because designing methods of three-term FLC emulates human control strategy, their principles are easy to understand for non-control specialists. During the last two decades, designing methods of conventional three-term controller have been using more and more advanced mathematical tools. This is needed in order to solve difficult problems in a rigorous fashion. However, this also results in fewer and fewer practical engineers who can understand these design methods. Therefore, practical engineers who are on the front line of designing consumer products tend to use the approaches that are simple and easy to understand. The three-term FLC is just such approach. Actually, many design methods for the three-term FLC have been developed during the last two decade. Choosing the best algorithm for the process is dependent on process control needs and objectives. The objective of this study is to identify, study and taxonomies these design choices for the three-term FLC and compare between their performance.

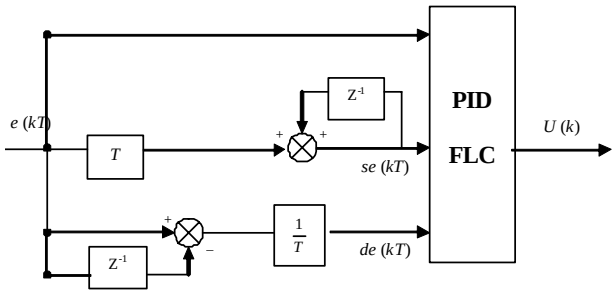

Fig. 1 PID-like fuzzy logic controller.

The remaining of this paper is organized as follows: Section 2 is a survey of various methods used for designing three-term FLC. Comparing the performance of these design methods using second-order armature-controlled DC motor and third-order field-controlled DC motor as a case studies are presented in section 3. Finally, section 4 presents the conclusion and some possible future extensions.

2. Taxonomy of Various Methods to Design the Three-Term FLC

There are two techniques to design PID-like FLC. First technique uses PD-like FLC in parallel with some

Manuscript received May 5, 2010
Manuscript revised May 20, 2010

deterministic gains, we call this method *hybrid PID-like FLC*. The second technique doesn't use any deterministic gains in designing PID-like FLC; we call it *pure fuzzy PID-like FLC*. This section surveys and classifies all PID-like FLC design methods, as shown in Fig. 2.

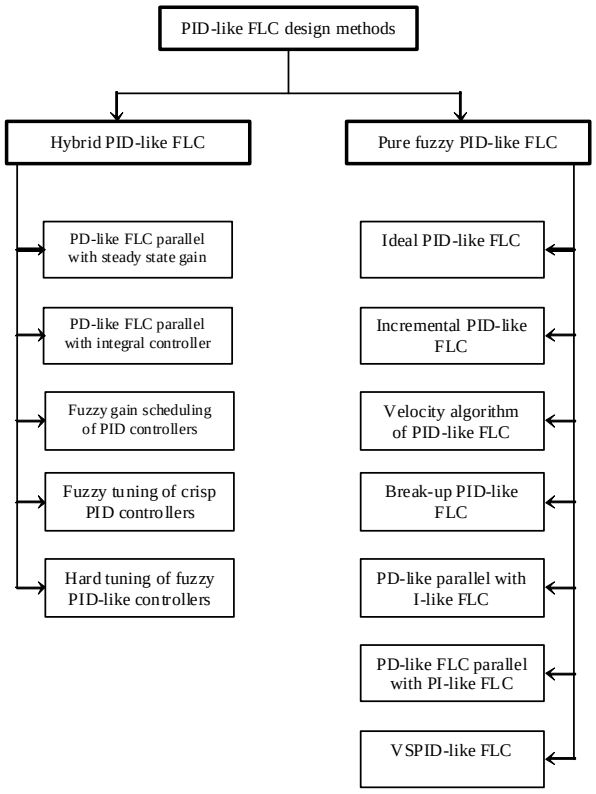

Fig. 2 Taxonomy of various methods to design PID-like FLC.

2.1 Hybrid PID-like FLC

The hybrid fuzzy controller takes advantage of nonlinear characteristics of the fuzzy controller (which are particularly important in ensuring suitable dynamic properties of the system under control) and the accuracy around a set point that is guaranteed by the conventional PID controller. We can think of blending these two controllers so as to bring their advantageous features into a single structure [9].

2.1.1 PD-like FLC parallel with Steady State Gain

Kwok *et al*. [10] proposed this method, and applied it to a process whose steady state gain is known or can be measured easily as K_S. In this design the integral action is omitted. The system is shown in Fig. 3.

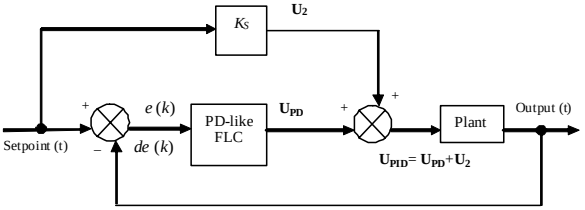

Fig. 3 PD-like FLC parallel with steady state gain.

The PD-like FLC uses the error $e(k)$ and change of error $de(k)$ (proportional-derivative) as inputs, much like a conventional PD controller. The feed forward gain K_S is needed in order to eliminate the steady state error. Nominally, the value of K_S is set to the reciprocal of the static gain of the plant–this results in a zero steady state error.

Kim *et al*. [11] applied this controller to a system with a deadzone, the system suffers from poor transient performance and a large steady state error. They propose a novel two-layered FLC for controlling systems with deadzones. The two-layered control structure consists of a fuzzy logic-based precompensator followed by PD-like FLC parallel with steady state gain (PID-like FLC).

2.1.2 PD-like FLC parallel with Integral Controller

Kwok *et al*. [10] proposed that if the steady state gain, K_S, is not known, integral action is necessary. This can be achieved by placing a conventional integral controller in parallel with the PD-like FLC. The controller is shown in

$$U_{PID} = U_{PD} + U_I = U_{PD} + K_i \times \sum_{n=1}^{k} e(n) \quad (1)$$

Fig. 4, where $se(k)$ is sum of the error. The output of the PID-like FLC will be:

K_i is some integral gain that has to be determined. Index n refers to the time instant.

2.1.3 Fuzzy Gain Scheduling of PID Controllers

Fig. 5 shows the PID control system with a fuzzy gain scheduler proposed by Zhao *et al*. [12]. In the proposed scheme, proportional (*Kp*) and derivative (*Kd*) gains are in prescribed ranges [*Kp*, min, *Kp*, max] and [*Kd*, min, *Kd*, max], respectively. The appropriate ranges are determined experimentally as described in [12].

The parameters K_P', K_d', and α are determined by a set of fuzzy rules of the form:

IF $e(k)$ is *Ai* **AND** $de(k)$ is *Bi* **THEN** K_P' is *Di* **AND** K_d' is *Ei* **AND** $\alpha = \alpha_i$

Here, *Ai*, *Bi*, *Di* and *Ei* are fuzzy sets on the corresponding supporting sets; α_i is a constant. Once $K_P^{'}$, $K_d^{'}$, and α are obtained, the PID controller parameters are calculated from the following equations:

$$K_P = \left(K_{P,\max} - K_{P,\min}\right) K_P^{'} + K_{P,\min}$$
$$K_d = \left(K_{d,\max} - K_{d,\min}\right) K_d^{'} + K_{d,\min}$$
$$K_i = \frac{K_P^2}{\alpha\, K_d} \qquad (2)$$

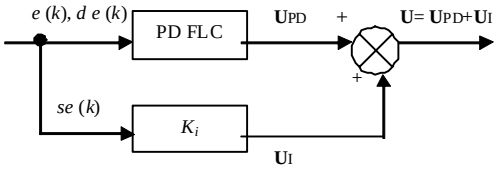

Fig. 4 PD-like FLC parallel with integral controller.

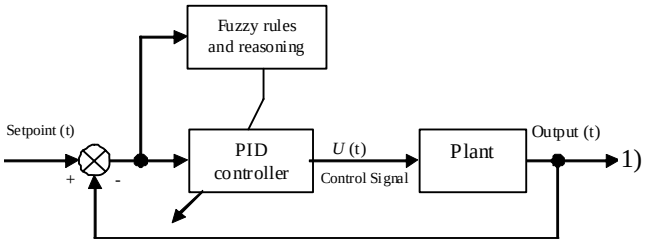

Fig. 5 PID control system with a fuzzy gain scheduler.

2.1.4 Fuzzy Tuning of Crisp PID Controllers

Tzafestas and Papanikolopoulos [13] propose this method and assume that one has available nominal controller parameter settings through some classical tuning technique (Ziegler-Nichols, Kalman, etc.). By using an appropriate fuzzy rule-base, the authors determine small changes of these values during the system operation, and this lead to improved performance of the transient and steady behavior of the closed-loop system.
The expressions of this controller are given by the following equations:

$$P(k) = P(k-1) + F(e(k), de(k)) \times k_1 \text{ (proportional)} \quad (3)$$
$$I(k) = I(k-1) + F(e(k), de(k)) \times k_2 \text{ (integral)} \quad (4)$$
$$D(k) = D(k-1) + F(e(k), de(k)) \times k_3 \text{ (derivative)} \quad (5)$$

The function F is the control law described by a rule-base. The values of the parameters k_1, k_2, and k_3 are determined from both the stability analysis and particular characteristics of the closed-loop response. In general, these parameters provide large flexibility and can be used in conjunction with all available PID tuning algorithms. The fuzzy inference system is used to ensure greater stability around the setpoint and smoother transition from one value to another.
Tzafestas and Papanikolopoulos [13] give three examples that show the result of the application of this fuzzy approach to the Ziegler-Nichols, the analytical, and the Kalman PID tuning techniques.

2.1.5 Hard Tuning of Fuzzy PID-like Controllers

Jantzen [14] proposes a tuning procedure that carries tuning rules from the PID domain over to fuzzy single-loop controllers. The idea is to start with a tuned, conventional PID controller, replace it with an equivalent PID-like FLC (Jantzen use PD-like FLC parallel with integral controller that was proposed in subsection 2.1.2), and eventually fine-tune the nonlinear FLC. This is relevant whenever a PID controller is possible or already implemented.
The output of the PID-like FLC is a nonlinear function:

$$U(k) = [F(GE \times e(k), GDE \times de(k)) + GSE \times se(k)] \times GU \qquad (6)$$

The gains GE, GDE, GSE, and GU are mainly for tuning the response but they can also be used for scaling the input/output signals onto the input/output universes. The linear approximation of (6) is:

$$U(k) = [GE \times e(k) + GDE \times de(k) + GSE \times se(k)] \times GU \qquad (7)$$

$$U(k) = GE \times GU \times \left[e(k) + \frac{GDE}{GE} \times de(k) + \frac{GSE}{GE} \times se(k) \right] \qquad (8)$$

In the last line we have assumed a nonzero GE. The

$$U(k) = K_P \left(e(k) + \frac{1}{T_i} \times se(k) + T_d \times de(k) \right)$$

conventional continuous PID controllers have the form: (9)

where the constant K_P is the *proportional gain*, T_i is the *integral time*, and T_d the *derivative time*. Comparing Eq. (8) and Eq. (9) the gains are related in the following way:

$$GE \times GU = K_P \quad (10)$$

$$\frac{GDE}{GE} = T_d \quad (11)$$

$$\frac{GSE}{GE} = \frac{1}{T_i} \quad (12)$$

In summary, we can form a tuning procedure for PID-like FLC proposed by Jantzen as follows:

1. Insert a crisp PID controller, and tune it (use Ziegler-Nichols, optimization, hand-tuning, or another method).
2. Insert a PID-like FLC (PD-like FLC parallel with integral controller).
3. Transfer KP, Td and 1/ Ti to GE, GDE, GSE, and GU using Eq. (10) – Eq. (12).
4. Insert a nonlinear rule base.
5. Fine-tune using hand-tuning; use GE to improve the rise time, GDE to dampen overshoot, and GSE to remove any steady state error.

2.2 Pure fuzzy PID-like FLC

The *hybrid* types of PID-like FLC are not true fuzzy PID controllers as they include deterministic controls as well [10]. In next subsections we will classify well-known methods to design a pure fuzzy PID-like FLC.

2.2.1 Ideal PID-like FLC

A comprehensive review of the classical design and implementation of the *ideal* PID-like FLC can be found in [3], [6]. The control law can be realized by:

$$U(k) = F(e(k), se(k), de(k)) \quad (13)$$

where $e(k)$, $se(k)$ and $de(k)$ are error, integral of error and rate of change in error, respectively; $U(k)$ is the control action and the function F is the control law that is described by a rule-base. The reasoning algorithm translates the set of vague rules into a mapping:

$$U(k) = f(e(k), se(k), de(k)) \quad (14)$$

that is similar to the ideal PID control algorithm:

$$U(k) = K_P \times e(k) + K_i \times se(k) + K_d \times de(k) \quad (15)$$

The major problem to this design method is that it is difficult to write rules for the integral action [14]. Another problem is *integrator windup*. Windup occurs when the actuator has limits, such as maximum speed for a motor. When the actuator saturates, the control action stays constant, but the error will continue to be integrated, the integrator winds up. The integral term may become very large and it will then take a long time to wind it down when the error changes sign. Large overshoots may be the consequence. In next subsections we will discuss other design methods, in which the authors tried to avoid these two problems.

2.2.2 Incremental PID-like FLC

Yager and Filev [6] proposed to configure PID-like FLC as an incremental controller to be described by the control law:

$$dU(k) = F(e(k), e(k-1), e(k-2)) \quad (16)$$

where $dU(k)$ is the change in control action. By application of the reasoning algorithm it is translated into a mapping:

$$dU(k) = f(e(k), e(k-1), e(k-2)) \quad (17)$$

that is similar to the incremental PID control law:

$$dU(k) = K_0 \times e(k) + K_1 \times e(k-1) + K_2 \times e(k-2) \quad (18)$$

Further extension of the input set (differences of the error) of the PID-like FLC is theoretically possible; it leads to an increasing complexity of the internal structure of this controller. The ability of an expert to formulate a clear and reasonable control strategy becomes a rather unrealistic assumption. It is rather unrealistic to expect that an operator or expert can determine reasonable control rules, considering second and higher differences of the error.

The advantage of this design method is that the controller output is the change in control action not the control action itself. Then it is easy to deal with windup and noise. A disadvantage is that it doesn't include derivative action well.

2.2.3 Velocity algorithm of PID-like FLC

To overcome disadvantage of the last method, Abdel-Nour *et al*. [15], [16] proposed the following design method that is described by the control law:

$$dU(k) = F(e(k), de(k), d2e(k)) \quad (19)$$

where $d2e(k)$ is the acceleration of the error:

$$d2e(k) = de(k) - de(k-1) = e(k) - 2e(k-1) + e(k-2) \quad (20)$$

Disadvantage of this method is that the acceleration error term has little influence on the performance because of limited measurement and computer resolution [17]. On the

other hand, as with the previous two design methods this method has three inputs, so the rule-base is three-dimensional and should have n^3 rules (n is the number of membership functions) for a complete rule base. A large number of rules make the knowledge base design more difficult. Good solution to this problem is to break-up the rule-base. We will describe different choices to this technique in the following subsections.

2.2.4 Break-up PID-like FLC

Golob [18] presents a concept called the decomposition of multivariable control rules. This concept shows how the inference of the rule-base with complex rules can be reduced to the inference of a number of rule-bases with simple rules. We apply this concept to PID-like FLC. The controller has proportional, integral, and derivative separate parts, which are tuned independently. This means that all parts have their own rule-bases as shown in Fig. 6. Advantages of this structure are:

1. Decreasing number of rules.
2. Simplifying the evolution of the rule base by decomposition of multivariable control rules into multi-sets of one-dimensional rules for each variable.
3. A direct relationship between a fuzzy control and a conventional control.
4. Easy connection between fuzzy parameters and operation of the controller.

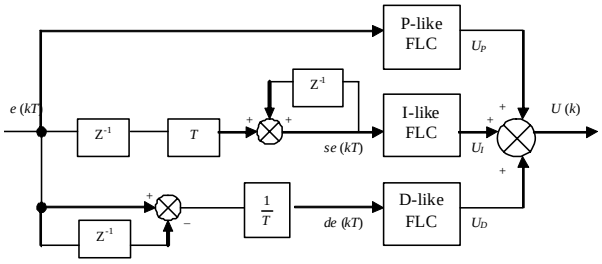

Fig. 6 Break-up PID-like FLC.

2.2.5 PD-like FLC parallel with I-like FLC

It is straightforward to invent a PID-like FLC with three input terms: *error*, *integral error*, and *derivative error*. A rule base with three inputs, however, easily becomes rather big and, as mentioned earlier, rules concerning the integral action are troublesome. Therefore it is common to separate the integral action as in the design method proposed by Kwok *et al*. [10] that use PD-like FLC in parallel with I-like FLC. The control algorithm is:

$$U_{PID}(k) = U_{PD}(k) + U_I(k) \tag{21}$$

This controller provides all the benefits of PID-like FLC, but it also suffers from the integrator windup.

2.2.6 PD-like FLC parallel with PI-like FLC

To overcome disadvantage of the last method, Kwok *et al*. [10] proposed this method that uses PD-like FLC in parallel with PI-like FLC as shown in Fig. 7.
The reasoning mechanisms of the PD-like and PI-like FLC work formally in the same manner. The only difference is in the output variable [6]. The output variable of PI-like FLC is the change of control action $dU\,(k)$ while the output variable of PD-like FLC is the control value $U(k)$ itself.

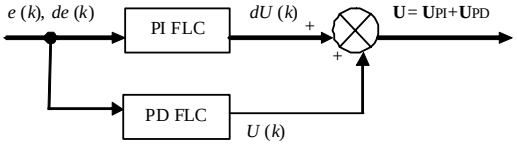

Fig. 7 PD-like FLC parallel with PI-like FLC.

To further reduce the complexity of the rule-base design and increase efficiency, Li and Gatland [17] proposed a simplified *PD-like FLC in parallel with PI-like FLC* by sharing a common rule-base for both PD-like FLC and PI-like FLC; as shown in Fig. 8. Practically, this simplified method can achieve the similar performance as the actual method shown in Fig. 7. The simplified *PD-like FLC parallel with PI-like FLC* is simple in structure, easy in implementation and fast in computation.

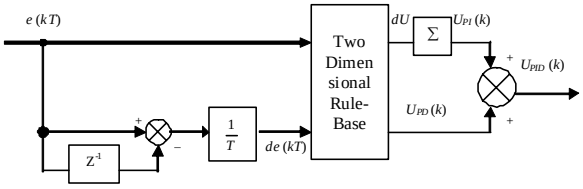

Fig. 8 Structure of the simplified PD-like FLC parallel with PI-like FLC.

2.2.7 Variable structure PID-like FLC (VSPID)

Essentially, the FLC is a variable structure system [6]. The basic concepts of the theory of the variable structural FLC can be found in [19], [20].
The fuzzy variable structure PID-like controller (VSPID-like FLC) has a structure similar to that of the conventional PID. In this controller, the PD mode is used in the case of large errors to speed up response, whereas

the PI mode is applied for small error conditions to eliminate the steady-state offset.
Chen and Chang [21] proposed the design methodology of VSPID-like FLC for nonlinear processes that is depicted in Fig. 9, where the controller output is:

$$U_{PID} = U_P + (1-\alpha(t))U_I + \alpha(t)U_D \tag{22}$$

where $\alpha(t) \in [0, 1]$. The controller would run out to be either a PD or a PI controller if $\alpha(t)$ is either 1 or 0. Instead of a drastic change of $\alpha(t)$ value, it could be reasonably defined as:

$$\alpha(t) = \tanh(\eta\,\beta(t)) \tag{23}$$

where

$$\beta(t) = \begin{cases} |e(t)| - \varepsilon & if \mid e(t) \mid \geq \varepsilon > 0 \\ 0 & if \mid e(t) \mid \leq \varepsilon \end{cases} \tag{24}$$

α (t) is an increasing function of $|e(t)|$, and converges to either 1 or 0 if $|e(t)|$ approaches infinity or $|e(t)|$ enters the tube $\varepsilon \geq |e(t)| \geq 0$, i. e.

$$\alpha(t) = \begin{cases} 1 & if \mid e(t) \mid >> \varepsilon \\ 0 & if \mid e(t) \mid \leq \varepsilon \end{cases} \tag{25}$$

The η value in Eq. (23) determines how quickly $\alpha(t)$ change between zero and one. For reasonable η values, the VSPID-like FLC would behave from the PD controller in the case of large error to the PID case and the PI case, i.e.

$$U(t) = \begin{cases} U_p + U_D & for \mid e(t) \mid >> \varepsilon \\ U_p + \alpha U_D + (1-\alpha)U_I & for \mid e(t) \mid \approx \varepsilon \\ U_p + U_I & for \mid e(t) \mid \leq \varepsilon \end{cases} \tag{26}$$

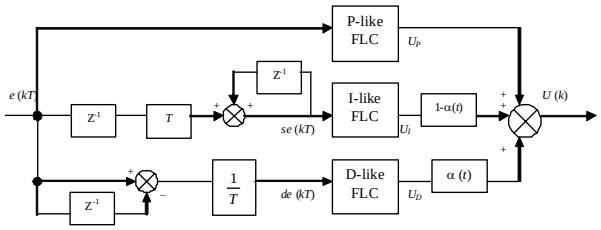

Fig. 9 Variable structure PID-like FLC.

3. Performance Comparison for Various Methods to Design the Three-Term FLC

Having designed a controller, it is important to validate its performance and compare it with other types of design methods. Analyzing time responses usually makes such evaluations. But simply examining the behavior of a controller is not enough to validate its performance or prove that it is better or worse than other controllers are. In the next section, we suggest using three performance measures with two simulated systems. The objectives of the simulation are to compare the behavior of various methods to design three-term FLC when applied to second order systems and third order systems.
In the following section, we present the performance measures that will be used during this study. In section 3.2, we present the applications that will be used in testing and analyzing the performance. Section 3.3 presents implementation technique for all design methods of pure fuzzy three-term FLC. The simulation results of these methods presented in section 3.4.

3.1 Performance Measures

To test all design methods of pure fuzzy three-term FLC, we choose to use three performance measures. They are:

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[22]:
   - *The swiftness of response*, as represented by the rise-time (*Tr*).
   - *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* as defined as follows:
   - *Integral of the absolute of the error* (*IAE*) defined by:
   - *Integral of time multiplied by the absolute of the error* (*ITAE*) defined by:

$$IAE = \int_0^\infty \mid e(t) \mid dt \tag{27}$$

   where $e(t)$ is the measured error. The calculation in

$$ITAE = \int_0^\infty t \mid e(t) \mid dt \tag{28}$$

   the studies was implemented by substituting an algebraic sum for the integrals [8]. *IAE* accounts mainly for error at the beginning of the response and to a lesser degree for the steady state duration. *ITAE* keeps account of errors at the beginning but also emphasizes the steady state [17].
3. *Robustness*: A robust controller is capable of dealing with significant parameter variations. Examining its performance for parameter values different from the designed values usually assesses controller robustness. Analysis of the effects of parameter variations on

PID-like FLC design methods provides useful quantitative, albeit empirical, measure of robustness.
For measuring robustness, we suggest varying defuzzification method parameter. During designing of the PID-like FLC, center of area (COA) was chosen as defuzzification method. We suggest measuring robustness of this controller by using bisector of area (BOA) as defuzzification method [2], [23] that is defined by:

$$U = \left\{ x \mid \int_{Min}^{x} \mu(x)\,dx = \int_{x}^{Max} \mu(x)\,dx \right\} \quad (29)$$

Here U is the control action, x is the running point in the universe, $\mu(x)$ 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.

3.2 Description of the Applications used in our Study

Two types of direct current (DC) motors are studied to examine the performance correspond to various design methods to design PID-like FLC.
DC motors are classified into several broad categories. They are described in [24]. The DC motors have 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 [25], as shown in Fig. 10.

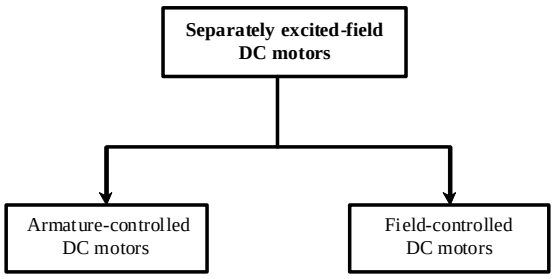

Fig. 10 Separately excited-field DC motors.

DC motors whether armature-controlled or field-controlled, are used in our simulation. The block diagram of such systems is shown in Fig. 11. The control objective for both types of DC motors is to reach a specified motor position using an appropriate input drive voltage.
A zero-order holder device is used to keep a constant controller output during each interval. The PID controller inputs are defined as follows:

$$e(kT) = \{setpoint(t) - position(t)\}|_{t=kT} \quad (30)$$

$$de(kT) = \frac{e(kT) - e((k-1)T)}{T} \quad (31)$$

$$se(kT) = se((k-1)T) + T \times e((k-1)T) \quad (32)$$

here T is sampling interval time; *setpoint*(t) and *position*(t) are reference and process output that is the angular displacement of the motor shaft.

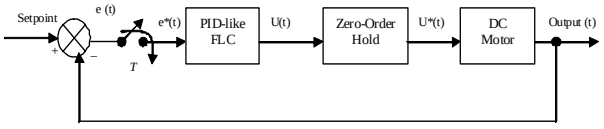

Fig. 11 DC Motor system with PID-like FLC.

3.2.1 Armature-controlled DC Motors

Ogata [25] gives the transfer function between the output angular displacement of the motor shaft $\theta(t)$ and the input control action $U(t)$:

$$\frac{\theta(s)}{U(s)} = \frac{K_m}{s(T_m s + 1)} \quad (33)$$

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

3.2.2 Field-controlled DC Motors

The transfer function between the output angular displacement of this motor shaft $\theta(t)$ and its input control action $U(t)$ is given by [25]:

$$\frac{\theta(s)}{U(s)} = \frac{K_m}{s(T_f s + 1)(T_m s + 1)} \quad (34)$$

where K_m is motor gain constant, T_f is time constant of field circuit and T_m is time constant of inertia-friction element. For simplicity, we assume that K_m = 1 rad/volt-sec, T_f = 0.1 second and T_m = 1 second.

3.3 Implementation of PID FLC Designing Methods

The *hybrid* types of PID-like FLC are not true fuzzy PID controllers as they include deterministic controls as well [10]. So, the only *pure* PID-like FLC types were used in our simulation.
For simulating all design methods of PID-like FLC, we use *MATLAB* version 7.10 with *Fuzzy Logic Toolbox* version 2.2.11 developed by the *Mathworks*. In the

implementation, *MIN* operator is chosen as *AND* connective between the antecedents of the rules of the FLC and as the fuzzy implication operation, while *MAX* operator is chosen as *OR* connective between the individual rules. Center of area (COA) is chosen as defuzzification method [2], [23].

To design *ideal PID-like FLC*, we propose three-dimensional rule-base matrix described in Table 1, 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 1: 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
	N	**Z**	**P**	**N**	**Z**	**P**	**N**	**Z**	**P**

Sum of the error

For designing *incremental PID-like FLC* as proposed by Yager and Filev [6], the same rule-base matrix in Table 1 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 also is used to design *velocity algorithm* of PID-like FLC where the inputs of the controller are replaced to be *e* (*k*), *de* (*k*), and *d2e* (*k*).

For designing *break-up PID-like* FLC proposed by Golob [18], we use three rule-bases: First rule-base for error, the second for the *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 *U* (*k*) is negative.
Rule 2: IF error *e* (*k*) is zero THEN *U* (*k*) is zero.
Rule 3: IF error *e* (*k*) is positive THEN *U* (*k*) is positive.

For designing *PD-like FLC parallel with I-like FLC* proposed by Kwok et al. [10], we use the rule-base matrix shown in Table 2 to be rule-base of PD-like FLC [6]. For I-like FLC we use the following set of rules to be its rule-base:

Rule 1: IF *se* (*k*) is negative THEN *U* (*k*) is zero.
Rule 2: IF *se* (*k*) is zero THEN *U* (*k*) is positive.
Rule 3: IF *se* (*k*) is positive THEN *U* (*k*) is zero.

To design *PD-like FLC parallel with PI-like FLC*, we use simplified method proposed by Li and Gatland [17]. We use the rule-base matrix shown in Table 2 to be rule-base of both controllers.

Having defined the fuzzy linguistic control rules, the membership functions corresponding to each element in the linguistic set must be defined. The proposed membership functions are chosen due to their economic nature of the parametric and functional descriptions. These membership functions mainly contain the *triangular* shaped membership to represent the whole set of *Zero* values, *Z*-shaped membership to represent the whole set of *Negative* values and *S*-shaped membership to represent the whole set of *Positive* values. These membership functions used for both DC motor systems types, have universe of discourse of *e*, *de*, *se* and *U* as [-50 50], [-40 40], [-100 100], and [-40 40] respectively.

Table 2: 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

For all methods of designing PID-like FLC for both DC motor types, 50 radian were used as desired angular displacement of the motor shaft and sampling interval time *T* were used as 1 second.

3.4 Simulation Results

In the following subsection we will investigate the performance of the armature-controlled DC motor, while in subsection 3.4.2 we will investigate the performance of the field-controlled DC motor.

3.4.1 Performance of Armature-controlled DC Motor

Comparison between step responses of armature-controlled DC motor system using various methods of designing PID-like FLC are shown in Table 3.

VSPID-like FLC have the best performance than other designing methods. Followed by *PD-like FLC parallel with PI-like FLC* and then *Ideal PID-like FLC*. The same results are obtained with robustness test. Table 4 compares the robustness of the PID-like FLC design methods when varying defuzzification method from center of area (COA) to bisector of area (BOA) of armature-controlled DC motor system.

3.4.2 Performance of Field-controlled DC Motor

Comparisons between step responses of field-controlled DC motor system using various methods of designing PID-like FLC are shown in Table 5.

Ideal PID-like FLC and *PD-like FLC parallel with PI-like FLC* showed the best performance than other designing methods. Actually, the performance of these two methods almost the same. On other hand, while *VSPID-like FLC* tried to minimize the percentage of overshoot (*Os*), it spent a lot of time to reach the steady state resulting the worst rise-time (*Tr*) and settling-time (*Ts*).

The same results mentioned above are obtained with robustness test. Table 6 compares the robustness of the PID-like FLC design methods when varying

defuzzification method from center of area (COA) to bisector of area (BOA).

Table 3: 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	4.75	107.98	391.51
VSPID-like FLC	6	7	None	101.29	311.31

Table 4: Robustness test 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	2.26	175.67	928.25
Incremental PID-like FLC	3	10	43.58	195.70	1338.1
Velocity algorithm FLC	8	15	6.69	182.22	1176.3
Break-up PID-like FLC	2	15	55.74	165.80	1055.4
PD-like parallel with I-like FLC	9	13	2.26	175.67	928.25
PD-like parallel with PI-like FLC	4	12	7.45	129.82	757.07
VSPID-like FLC	5	5	None	122.41	807.12

4. Conclusion

Three-mode FLC provides a systematic and efficient framework to incorporate linguistic fuzzy information from human experts. In this paper, taxonomy of its design methods was proposed. The performance of all pure fuzzy designing methods of the three-mode FLC was compared. Two simple applications were used in the performance tests to make it easy to validate the results. The performance analysis showed that using *PD-like FLC parallel with PI-like FLC* design method of the three-mode FLC can generally satisfy the desired rise-time and percent overshoot than other designing methods of this controller. Also, testing various designing methods of the three-mode FLC for varying of defuzzification method show that *PD-like FLC parallel with PI-like FLC* generally had better robustness than other methods.

Table 5: Performance of field-controlled DC motor system using various methods of designing PID-like FLC

	Tr (Sec.)	***Ts*** (Sec.)	***Os*** (%)	***IAE***	***ITAE***
Ideal PID-like FLC	6	9	2.68	175.30	863.89
Incremental PID-like FLC	7	23	27.92	314.28	2585.9
Velocity algorithm FLC	7	14	13.99	261.63	1644.9
Break-up PID-like FLC	11	20	13.52	339.12	2753.9
PD-like parallel with I-like FLC	8	14	7.02	246.59	1490.1
PD-like parallel with PI-like FLC	6	9	2.23	174.26	889.78
VSPID-like FLC	16	16	0.06	293.48	1957.9

Table 6: Robustness test of field-controlled DC motor system using various methods of designing PID-like FLC

	Tr (Sec.)	***Ts*** (Sec.)	***Os*** (%)	***IAE***	***ITAE***
Ideal PID-like FLC	6	8	1.51	172.88	924.75
Incremental PID-like FLC	7	23	27.72	297.20	2177.3
Velocity algorithm FLC	7	None	17.22	337.90	3631.3
Break-up PID-like FLC	11	None	12.83	456.64	5728.6
PD-like parallel with I-like FLC	8	13	9.847	258.91	1965.9
PD-like parallel with PI-like FLC	5	8	3.53	170.20	912.56
VSPID-like FLC	16	16	None	311.80	2422.1

The main problem of various designing methods of the three-mode FLC is that they not take desired performance criteria in their design. In our future work, we will propose performance optimization design method to handle this problem. This method will depends on using *performance rule-based model* in the design of PID FLC.

Acknowledgment

The authors would like to express their appreciation to Deanship of Scientific Research at King Faisal University for supporting this research.

References

[1] J. Jantzen, "Tutorial on Fuzzy Logic", Technical University of Denmark: Department of Automation, Technical report number 98-E 868, Aug 1998, URL: http://www.iau.dtu.dk/

[2] J. Jantzen, "Design of Fuzzy Controllers", Technical University of Denmark: Department of Automation, Technical report number 98-E 864, http://www.iau.dtu.dk/

[3] 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.

[4] 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.

[5] R. Isermann, "On Fuzzy Logic Applications for Automatic Control, Supervision, and Fault Diagnosis", IEEE Trans. on Systems, Man, and Cybernetics–Part A: Systems and Humans, Vol. SMC-28, No. 2, pp. 221-235, 1998.

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

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

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

[9] W. Pedrycz, "Fuzzy Control and Fuzzy Systems", Research Studies Press, 1993.

[10] 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.

[11] 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.

[12] 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.

[13] S. Tzafestas and N. P. Papanikolopoulos, "Incremental Fuzzy Expert PID Control", IEEE Transactions on Industrial Electronics, Vol. 37, No.5, pp. 365-371, 1990.

[14] J. Jantzen, "Tuning of Fuzzy PID Controllers", Technical University of Denmark: Department of Automation, Technical report number 98-H 871, 30 Sep 1998, URL: http://www.iau.dtu.dk/

[15] G. M. Abdel-Nour, C. Chang, F. Hung and J. Y. Cheung, "Design of a Fuzzy Controller Using Input and Output Mapping Factors", IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-21, No. 5, pp. 952-960, 1991.

[16] G. M. Abdel-Nour, C. Chang, J. Y. Cheung and G. Tinetti, "Application of Describing Functions in the Transient Response Analysis of a Three-Term Fuzzy Controller", IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-23, No. 2, pp. 603-606, 1993.

[17] 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.

[18] M. Golob, "Decomposition of a Fuzzy Controller Based on the Inference Break-up Method", URL: http://www.au.feri.uni-mb.si/~marjan/

[19] H. X. Li, H. B. Gatland, and A. W. Green, "Fuzzy Variable Structure Control", IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics, Vol. SMC-27, No. 2, pp. 306-312, 1997.

[20] A. Suyitno, J. Fujikawa, H. Kobayashi, and Y. Dote, "Variable-Structured Robust Controller by Fuzzy Logic for Servomotors", IEEE Transactions on Industrial Electronics, Vol. 40, No. 1, pp. 80-88, 1993.

[21] C. L. Chen and F. Y. Chang, "Design and analysis of neural/fuzzy variable structural PID control systems", IEE Proc. Control Theory Appl., Vol. 143, No. 4, pp. 200-208, July 1996.

[22] R. Dorf and R. Bishop, "Modern Control Systems", Addison-Wesley, 1995.

[23] J. Jantzen, H. Verbruggen and J. Ostergaard, "Fuzzy Control in the Process Industry: Common Practice and Challenging Perspectives", Technical University of Denmark: Department of Automation, URL: http://www.iau.dtu.dk/

[24] B. C. Kuo, "Automatic Control Systems", Prentice-Hall, 1987.

[25] K. Ogata, "Modern Control Engineering", Prentice-Hall, 1970.

Essam Natsheh obtained his PhD in Communications and Networks Engineering from University Putra Malaysia in 2006. Currently, he is an Assistant Professor at the Computer Information Systems Department, College of Applied Studies and Community Services, King Faisal University (Saudi Arabia). Essam has more than ten years of teaching and research experiences in Malaysia and Saudi Arabia. Also, he has more than 15 publications in refereed journals at international level. His research interest is mobile ad-hoc networks, in particular, the development of a new routing algorithm for this type of networking.

Khalid Buragga received the B.Sc. in Computer Information Systems (CIS) from King Faisal University, Saudi Arabia, and the M.Sc. in CIS from University of Miami, USA, and the Ph.D. in Information Technology from George Mason University, USA. He is currently an assistant professor at college of Computer Sc. & IT, King Faisal University. His general research interests span the areas of software design, development, quality, and reliability; business process re-engineering, integrating systems; communications, networking and signal processing.