

Research on Minimum Intelligent Unit for Flexible Robot

Ghasem Sahamijoo, Omid Avatefipour, Mahmoud Reza Safaei Nasrabad,
Mohammad Taghavi and Farzin Piltan

*Intelligent System and Robotic Lab, Iranian Institute of Advance Science and
Technology (IRAN SSP), Shiraz/Iran
piltan_f@iranssp.org, www.iranssp.org/english*

Abstract

Design of a robust controller for multi input-multi output (MIMO) nonlinear uncertain dynamical system can be a challenging work. This research paper focuses on the design and analysis of a high performance PID like fuzzy controller for flexible robot manipulator, in presence of uncertainties. In this research, fuzzy logic controller is a stable nonlinear controller, which selected to control of our nonlinear system. The proposed approach effectively combines of design methods from linear Proportional-Integral-Derivative (PID) controller and fuzzy logic theory to improve the performance, stability and robustness of the flexible robot manipulator. To solve system's dynamic nonlinearity, the PID fuzzy logic controller is used as a PID like fuzzy logic controller. The PID like fuzzy logic controller is updated based on gain updating factor. In this methodology, fuzzy logic controller is used to estimate the dynamic uncertainties. In this methodology, PID like fuzzy logic controller is evaluated. PID like fuzzy logic controller has three inputs, Proportional (P), Derivative (D), and Integrator (I), if each inputs have N linguistic variables to defined the dynamic behavior, it has $N \times N \times N$ linguistic variables. To solve this challenge, parallel structure of a PD-like fuzzy controller and PI-like fuzzy controller is evaluated. In the next step, the challenge of design PI and PD fuzzy rule tables are supposed to be solved. To solve this challenge PID like fuzzy controller is replaced by PD-like fuzzy controller with the integral term in output. This method is caused to design only PD type rule table for PD like fuzzy controller and PI like fuzzy controller.

Keywords: *flexible robot manipulator, artificial intelligence, fuzzy logic control, PID like fuzzy controller, system's dynamic*

1. Introduction and Theory

System or plant is a set of components which work together to follow a certain objective. Based on above definition, in this research robot manipulator is system. A robot is a machine which can be programmed to do a range of tasks. They have five fundamental components; brain, body, actuator, sensors and power source supply. A brain controls the robot's actions to best response to desired and actual inputs. A robot body is physical chasses which can use to holds all parts together. Actuators permit the robot to move based on electrical part (*e.g.*, motors) and mechanical part (*e.g.*, hydraulic piston). Sensors give robot information about its internal and external part of robot environment and power source supply is used to supply all parts of robot. Continuum robots represent a class of robots that has a biologically inspired form characterized by flexible backbones and high degrees-of-freedom structures [1-3]. Theoretically, the compliant nature of a continuum robot provides infinite degrees of freedom to these devices. However, there is a limitation set by the practical inability to incorporate infinite actuators in the device. Most of these robots are consequently under actuated (in terms of numbers of independent actuators) with respect to their anticipated tasks. In other words, they must achieve a wide

range of configurations with relatively little control inputs. This is partly due to the desire to keep the body structures (which, unlike in conventional rigid-link manipulators or fingers, are required to directly contact the environment) “clean and soft”, but also to exploit the extra control authority available due to the continuum contact conditions with a minimum number of actuators. For example, the Octarm VI continuum manipulator, discussed frequently in this paper, has nine independent actuated degrees-of-freedom with only three sections. Continuum manipulators differ fundamentally from the rigid - link and hyper-redundant robots by having an unconventional structure that lacks links and joints. Hence, standard techniques like the Denavit-Hartenberg (D-H) algorithm cannot be directly applied to developing the continuum arm kinematics. Moreover, the design of each continuum arm varies with respect to the flexible backbone present in the system, the positioning, type and number of actuators. The constraints imposed by these factors make the set of reachable configurations and nature of movements unique to every continuum robot. This makes it difficult to formulate generalized kinematic or dynamic models for continuum robot hardware. A dynamic function is the study of motion with regard to the forces. Dynamic modeling of robot manipulators is used to illustrate the behavior of robot manipulator (*e.g.*, nonlinear dynamic behavior), design of nonlinear conventional controller (*e.g.*, conventional computed torque controller, conventional sliding mode controller and conventional backstepping controller) and for simulation. It is used to analyses the relationship between dynamic functions output (*e.g.*, joint motion, velocity, and accelerations) to input source of dynamic functions (*e.g.*, force/torque or current/voltage). Dynamic functions is also used to explain the some dynamic parameter’s effect (*e.g.*, inertial matrix, Coriolis, Centrifugal, and some other parameters) to system’s behavior [1-3]. The Continuum section analytical model developed here consists of three modules stacked together in series. In general, the model will be a more precise replication of the behavior of a continuous arm to a greater of modules included in the series. However, we will show that the three modules effectively represent the dynamic behavior of the hardware, so more complex models are not motivated. Thus, the constant curvature bends exhibited by the section is incorporated inherently within the model. The model resulting from the application of Lagrange’s equations of motion obtained from this system can be represented in the form

$$F_{coeff} \underline{\tau} = D(\underline{q}) \ddot{\underline{q}} + C(\underline{q}) \dot{\underline{q}} + G(\underline{q}) \quad (1)$$

where τ is a vector of input forces and q is a vector of generalized coordinates. The force coefficient matrix F_{coeff} Transforms the input focus to the generalized forces and torques in the system. The inertia matrix, D is composed of four block matrices. The block matrices that correspond to pure linear accelerations and pure angular accelerations in the system (on the top left and on the bottom right) are symmetric. The matrix C contains coefficients of the first order derivatives of the generalized coordinates. Since the system is nonlinear, many elements of C contain first order derivatives of the generalized coordinates. The remaining terms in the dynamic equations resulting from gravitational potential energies and spring energies are collected in the matrix G . The coefficient matrices of the dynamic equations are given below [4-5],

$$F_{coeff} = \begin{bmatrix} 1 & 1 & \cos(\theta_1) & \cos(\theta_1) & \cos(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) \\ 0 & 0 & 1 & 1 & \cos(\theta_2) & \cos(\theta_2) \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1/2 & -1/2 & 1/2 & -1/2 & 1/2 + s_2 \sin(\theta_2) & -1/2 + s_2 \sin(\theta_2) \\ 0 & 0 & 1/2 & -1/2 & 1/2 & -1/2 \\ 0 & 0 & 0 & 0 & 1/2 & -1/2 \end{bmatrix} \quad (2)$$

$$D(\underline{q}) = \tag{3}$$

$$\begin{bmatrix} m_1 + m_2 + m_3 & m_2 \cos(\theta_1) + m_3 \cos(\theta_1) & m_3 \cos(\theta_1 + \theta_2) & -m_2 s_2 \sin(\theta_1) - m_3 s_2 \sin(\theta_1) - m_3 s_3 \sin(\theta_1 + \theta_2) & 0 & 0 \\ m_2 \cos(\theta_1) + m_3 \cos(\theta_1) & m_2 + m_3 & m_3 \cos(\theta_2) & -m_3 s_3 \sin(\theta_2) & -m_3 s_3 \sin(\theta_2) & 0 \\ m_3 \cos(\theta_1 + \theta_2) & m_3 \cos(\theta_2) & m_3 & m_3 s_3 \sin(\theta_2) & 0 & 0 \\ -m_2 s_2 \sin(\theta_1) - m_3 s_2 \sin(\theta_1) - m_3 s_3 \sin(\theta_1 + \theta_2) & -m_3 s_3 \sin(\theta_2) & m_3 s_2 \sin(\theta_2) & m_2 s_2^2 + I_1 + I_2 + I_3 + m_3 s_2^2 + m_3 s_3^2 + 2m_3 s_3 \cos(\theta_2) s_2 & I_2 + m_3 s_3^2 + I_3 + m_3 s_3 \cos(\theta_2) s_2 & I_1 \\ -m_3 s_3 \sin(\theta_1 + \theta_2) & -m_3 s_3 \sin(\theta_2) & 0 & I_2 + m_3 s_3^2 + I_3 + m_3 s_3 \cos(\theta_2) s_2 I & I_2 + m_3 s_3^2 + I_3 & I_1 \\ 0 & 0 & 0 & I_3 & I_3 & I_1 \end{bmatrix} \tag{4}$$

$$C(\underline{q}) = \tag{4}$$

$$\begin{bmatrix} c_{11} + c_{21} & -2m_2 \sin(\theta_1) \dot{\theta}_1 - 2m_3 \sin(\theta_1) \dot{\theta}_1 & -2m_3 \sin(\theta_1 + \theta_2) (\dot{\theta}_1 + \dot{\theta}_2) & -m_2 s_2 \cos(\theta_1) (\dot{\theta}_1) + (1/2)(c_{11} + c_{21}) - m_3 s_2 \cos(\theta_1) (\dot{\theta}_1) - m_3 s_3 \cos(\theta_1 + \theta_2) (\dot{\theta}_1) & -m_3 s_3 \sin(\theta_1 + \theta_2) & 0 \\ 0 & c_{12} + c_{22} & -2m_3 \sin(\theta_2) (\dot{\theta}_1 + \dot{\theta}_2) & -m_3 s_3 (\dot{\theta}_1) + (1/2)(c_{12} + c_{22}) - m_3 s_2 (\dot{\theta}_1) - m_3 s_3 \cos(\theta_2) (\dot{\theta}_1) & -2m_3 s_3 \cos(\theta_2) (\dot{\theta}_1) & 0 \\ 0 & 2m_3 \sin(\theta_2) (\dot{\theta}_1) & c_{13} + c_{23} & -m_3 s_3 s_2 \cos(\theta_2) (\dot{\theta}_1) - m_3 s_3 (\dot{\theta}_1) & -2m_3 s_3 (\dot{\theta}_1) - m_3 s_3 (\dot{\theta}_2) & (1/2)(c_{13} + c_{23}) \\ (1/2)(c_{11} + c_{21}) & 2m_3 s_3 \cos(\theta_2) (\dot{\theta}_1) - 2m_3 s_2 (\dot{\theta}_1) + 2m_2 s_2 (\dot{\theta}_1) & 2m_3 s_3 (\dot{\theta}_1 + \dot{\theta}_2) - 2m_3 s_2 \cos(\theta_2) (\dot{\theta}_1 + \dot{\theta}_2) & 2m_3 s_3 s_2 \sin(\theta_2) (\dot{\theta}_2) + (1^2/4)(c_{11} + c_{21}) & m_3 s_3 s_2 \sin(\theta_2) (\dot{\theta}_2) & 0 \\ 0 & (1/2)(c_{12} + c_{22}) + 2m_3 s_3 \cos(\theta_2) (\dot{\theta}_1) & 2m_3 s_3 (\dot{\theta}_1 + \dot{\theta}_2) & m_3 s_3 s_2 \sin(\theta_2) (\dot{\theta}_1) & (1^2/4)(c_{12} + c_{22}) & 0 \\ 0 & 0 & (1/2)(c_{13} - c_{23}) & 0 & 0 & (1^2/4)(c_{13} + c_{23}) \end{bmatrix} \tag{5}$$

$$G(\underline{q}) = \tag{5}$$

$$\begin{bmatrix} -m_1 g - m_2 g + k_{11}(s_1 + (1/2)\theta_1 - s_{01}) + k_{21}(s_1 - (1/2)\theta_1 - s_{01}) - m_3 g \\ -m_2 g \cos(\theta_1) + k_{12}(s_2 + (1/2)\theta_2 - s_{02}) + k_{22}(s_2 - (1/2)\theta_2 - s_{02}) - m_3 g \cos(\theta_1) \\ -m_3 g \cos(\theta_1 + \theta_2) + k_{13}(s_3 + (1/2)\theta_3 - s_{03}) + k_{23}(s_3 - (1/2)\theta_3 - s_{03}) \\ m_2 s_2 g \sin(\theta_1) + m_3 s_3 g \sin(\theta_1 + \theta_2) + m_3 s_2 g \sin(\theta_1) + k_{11}(s_1 + (1/2)\theta_1 - s_{01})(1/2) + k_{21}(s_1 - (1/2)\theta_1 - s_{01})(-1/2) \\ m_3 s_3 g \sin(\theta_1 + \theta_2) + k_{12}(s_2 + (1/2)\theta_2 - s_{02})(1/2) + k_{22}(s_2 - (1/2)\theta_2 - s_{02})(-1/2) \\ k_{13}(s_3 + (1/2)\theta_3 - s_{03})(1/2) + k_{23}(s_3 - (1/2)\theta_3 - s_{03})(-1/2) \end{bmatrix}$$

A controller (control system) is a device which can sense information from system (e.g., robot manipulator) to improve the performance of the system using actuation and

computation. According to the control theory, systems' controls are divided into two main groups: conventional control theory and soft computing control theory. Conventional control theories are work based on manipulator dynamic model. This technique is highly sensitive to the knowledge of all parameters of nonlinear robot manipulator's dynamic equation. Conventional control theory is divided into two main groups:

- Linear control theory
- Nonlinear control theory.

Soft computing (intelligent) control theory is free of some challenges associated to conventional control theory. This technique is worked based on intelligent control theory. This theory is divided into the following groups:

- Fuzzy logic theory,
- Neural network theory,
- Genetic algorithm
- Neuro-fuzzy theory

Linear control theory is used in linear and nonlinear systems. This type of theory is used in industries, because design of this type of controller is simple than nonlinear controller. However this type of controller used in many applications but it cannot guarantee performance in complex systems. Proportional (P) control is used to responds immediately to difference of control input variables by immediately changing its influences variables, but this type of control is unable to eliminate the control input difference. Figure 1 shows the block diagram of proportional controller with application to robot manipulator.

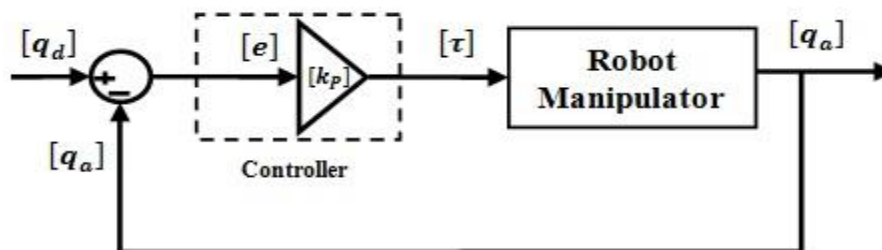


Figure 1. Block Diagram of Proportional Controller

Proportional plus Derivative (PD) controller is widely used in control process where the results are sensitive to exceeded of set point. This controller, like Proportional controller, has permanent variation in presence of self-limitation control. In mathematically, the formulation of Proportional-Derivative part calculated as follows;

$$U_{PD} = K_p \times e + K_v \left(\frac{de}{dt} \right) = K_p \times e + K_v \dot{e} \quad (6)$$

The Derivative component in this type of methodology is used to cancel outs the change process variables change in presence of quick change in controllers input. Figure 2 shows the block diagram of Proportional-Derivative (PD) control of robot manipulator.

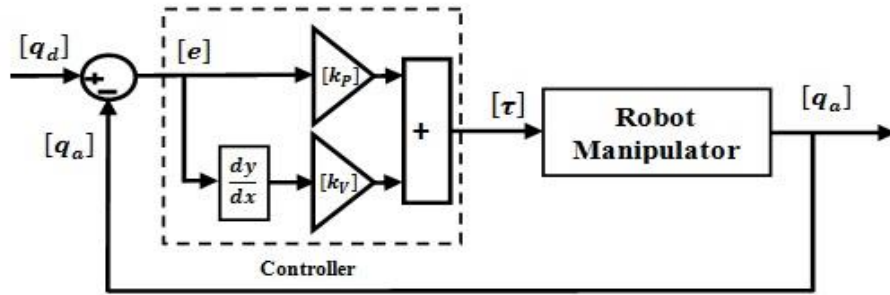


Figure 2. Block Diagram of PD Control of Robot Arm

Figure 3 shows the ramp response of PD controller.

Integral (I) control, integrate the input signal deviation over a period of time. This part of controller is used to system stability after a long period of time. Figure 4 shows the block diagram of Integral (I) controller with application to robot manipulator. In contrast of Proportional type of controller, this type of controller used to eliminate the deviation [6-7].

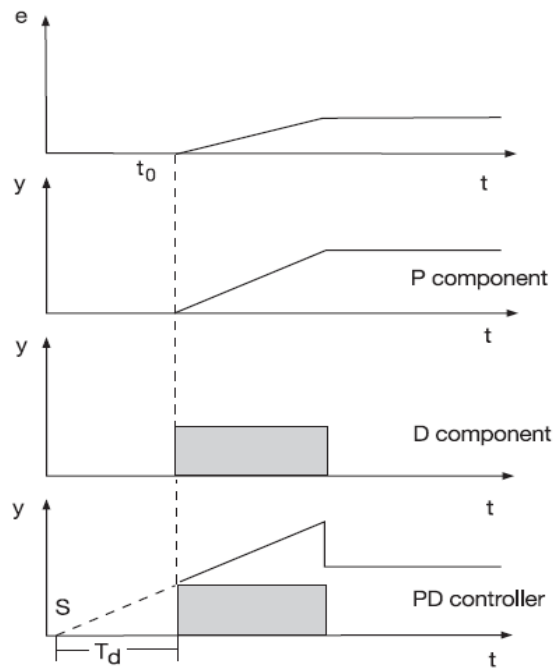


Figure 3. Ramp Response of a PD Controller

In mathematically, the formulation of integral part calculated as follows;

$$I = \frac{1}{T} \int e. dt = \sum e \quad (7)$$

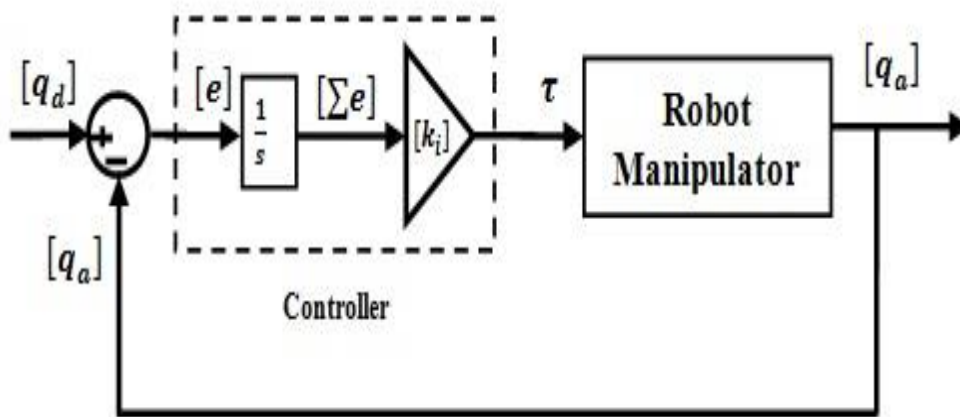


Figure 4. Block Diagram of Integral Control of Robot Manipulator

Figure 5 shows the step response of integral controller.

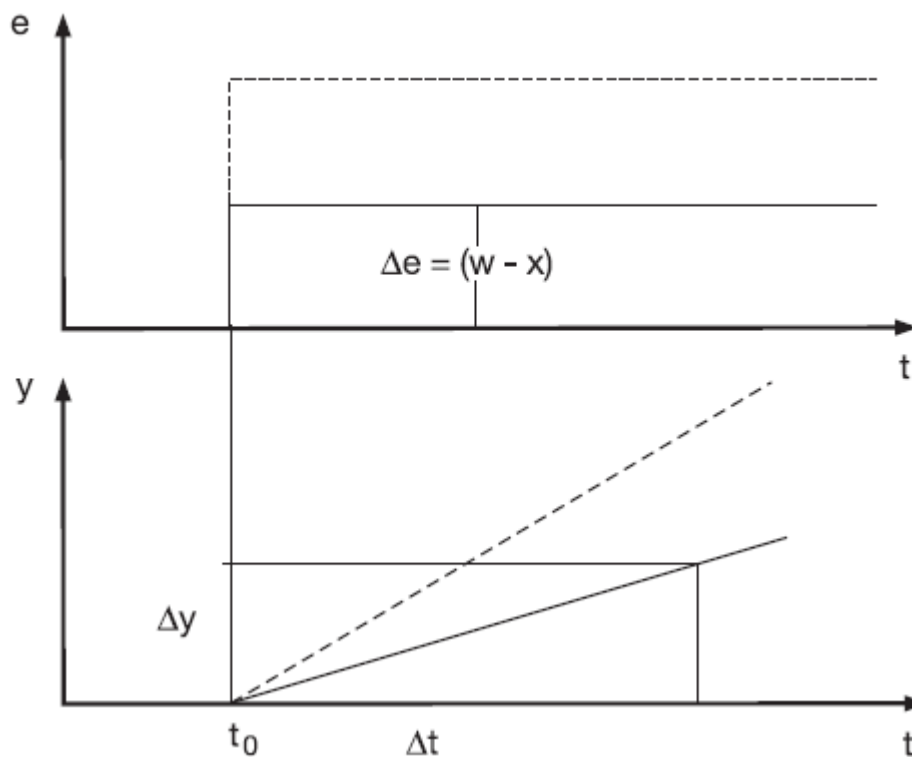


Figure 5. Step Response of an Integral (I) Controller

According to integral type of controller, it takes relatively long time. The proportional type controller used to immediately response to the input variations [8]. The proportional-integral (PI) controller has the advantages of both proportional and integral controller; it is rapid response to the input deviation as well as the exact control at the desired input. Figure 6 shows the block diagram of PI control of robot manipulator.

$$U_{PI} = K_p \times e + K_i \left(\frac{1}{T} \int e. dt \right) = K_p \times e + K_i \Sigma e \quad (8)$$

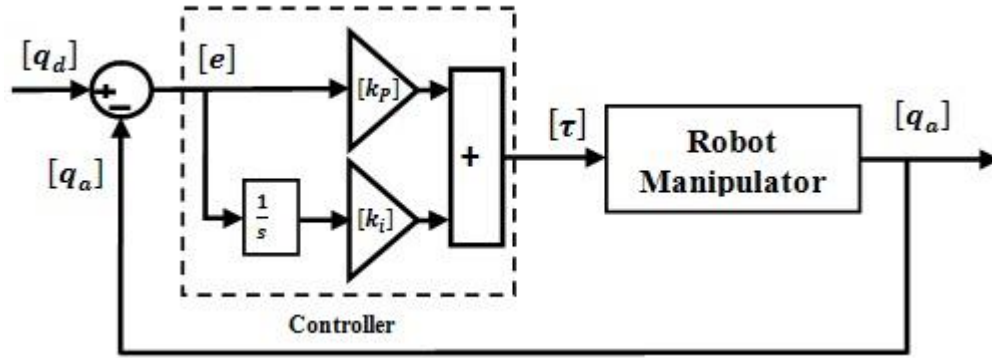


Figure 6. Block Diagram of PI Control of Robot Manipulator

Figure 7 shows the step response of PI controller.

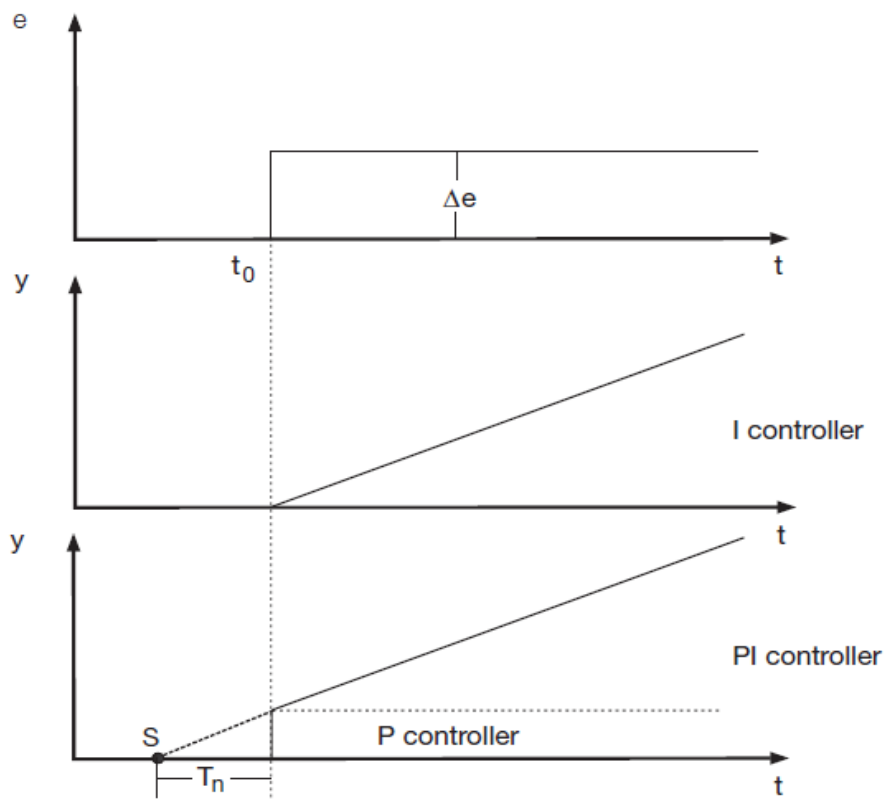


Figure 7. Step Response of a Proportional-Integral (PI) Controller

The combination of proportional (P) component, integral (I) component with a derivative (D) controller offered advantages in each case. Proportional-Integral-Derivative (PID) controller has rapid response to the input deviation, the exact control at the desired input as well as fast response to the disturbances. The PID controller takes the error between the desired joint variables and the actual joint variables to control the serial links robot manipulator. A proportional-derivative integral control system can easily be implemented. This method does not provide sufficient control for systems with time-varying parameters or highly nonlinear systems. Figure 8 shows the block diagram of PID control of robot manipulator. The formulation of PID controller calculated as follows;

$$U_{PID} = K_p \times e + K_i \left(\frac{1}{T} \int e \cdot dt \right) + K_v \left(\frac{de}{dt} \right) = K_p \times e + K_i \Sigma e + K_v \dot{e} \quad (9)$$

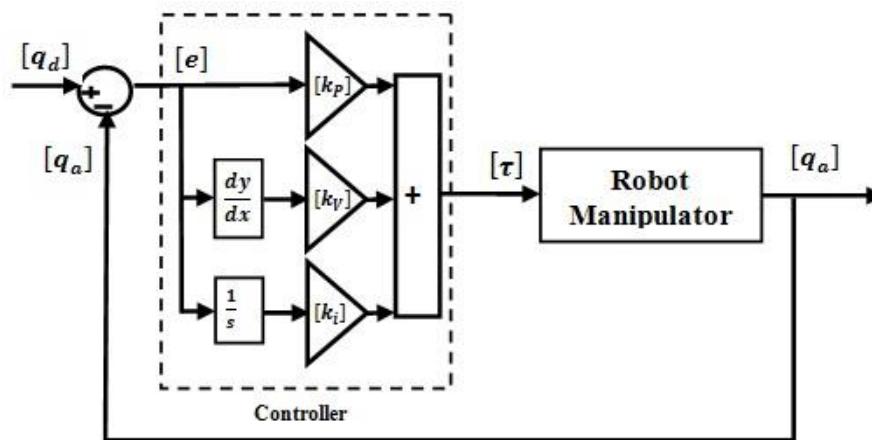


Figure 8. Block Diagram of PID Control of Robot Manipulator

Figure 9 shows the step response of PID controller.

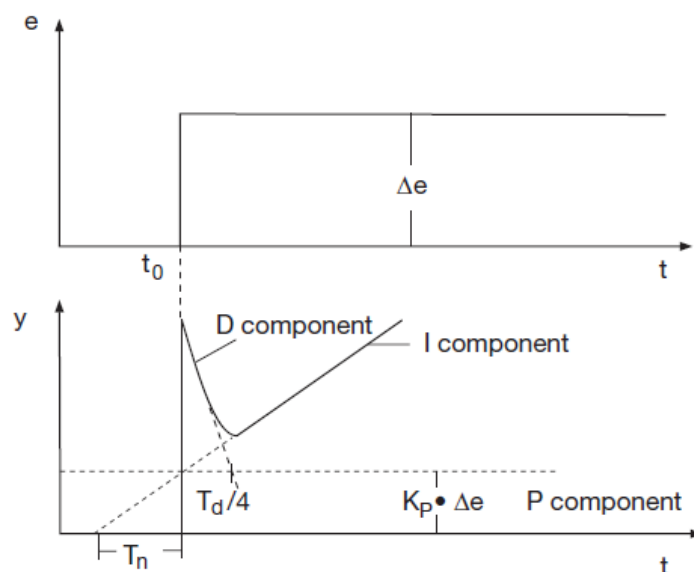


Figure 9. Step Response of a Proportional-Integral-Derivative (PID) Controller

Zadeh [7-11] introduced fuzzy sets in 1965. After 40 years, fuzzy systems have been widely used in different fields, especially on control problems. Fuzzy systems transfer expert knowledge to mathematical models. Fuzzy systems used fuzzy logic to estimate the dynamics of proposed systems. Fuzzy controllers, including fuzzy if-then rules are used to control proposed systems. Conventional control methods use mathematical models to control systems. Fuzzy control methods replace the mathematical models with fuzzy if-then-rules and fuzzy membership function to control systems. Both fuzzy and conventional control methods are designed to meet system requirements of stability and convergence. When mathematical models are unknown or partially unknown, fuzzy control models can use fuzzy systems to estimate the unknown models. This is called the model-free approach. Conventional control models can use adaptive control methods to

achieve the model-free approach. When system dynamics become more complex, nonlinear systems are difficult to handle by conventional control methods. From the universal approximation theorem, fuzzy systems can approximate arbitrary nonlinear systems. In practical problems, systems can be controlled perfectly by experts. Experts provide linguistic description about systems. Conventional control methods cannot design controllers combined with linguistic information. When linguistic information is important for designing controllers, we need to design fuzzy controllers for our systems. Fuzzy control methods are easy to understand for designers. The design process of fuzzy controllers can be simplified with simple mathematical models. Research on applied fuzzy logic methodology in inverse dynamic controller (FIDLC) to compensate the unknown system dynamics considerably improves the robot manipulator control process. The foundation and introduction of fuzzy logic theory is discussed in this section [10-11]. Supposed that U is the universe of discourse for example in this research error, change of error and torque are the sample of universe of discourse and x is the element of U it means that if error is positive and small or if change of error to be defined as positive and small, all these part of the elements of error and change of error. A crisp set to be defined as a set such as error or change of error which consists of different elements (x) such as negative and small in error will all or no membership in a defined set. In crisp set the membership degree is zero or one such as logical systems. A crisp set (C) in a universe of discourse U is defined by a following membership function (μ_C):

$$\mu_C: U \rightarrow \{0, 1\} \tag{10}$$

A fuzzy set is a set such as error or change of error that different element has different membership grade between zero to one. A fuzzy set (F) in a universe of discourse U is defined by a following membership function (μ_F):

$$\mu_F: U \rightarrow [0, 1] \tag{11}$$

The membership function ($\mu_F(x)$) of fuzzy set F has value between zero to one which each element of universe of discourse U mapped a values between zero to one and this value is called membership degree. Figure 10 shows the fuzzy membership function.

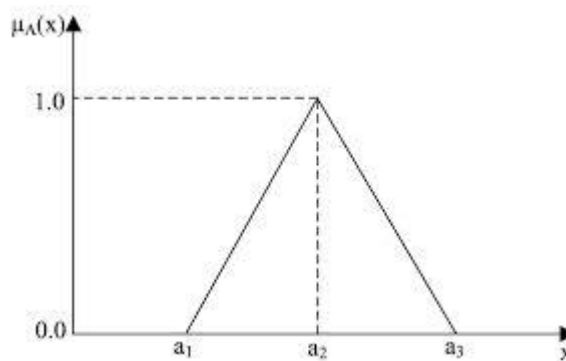


Figure 10. Fuzzy Membership Functions

If the membership functions value $\mu_C(x)$ equal to zero or one it is a crisp set. Based on literature numerical membership function and functional membership function are two important technique to define fuzzy membership function. In numerical membership function the numerator of fraction shows the degrees of membership function and the denominator of fraction shows the universe of discourses point. In functional membership function standard functions in fuzzy sets is used to define membership function and

membership degrees. According to the literature the famous functional membership function in practical applications are:

- triangular function
- trapezoidal form
- Gaussian form

According to the literature the formulation of Trapezoidal membership function is;

$$\mu_F(x) = \begin{cases} 0, & (x < a) \text{ or } (x > d) \\ \frac{x-a}{b-a}, & a \leq x < b \\ \frac{d-x}{d-c}, & c \leq x < d \\ 1, & b \leq x \leq c \end{cases} \quad (12)$$

Figure 11 shows the Trapezoidal membership function.

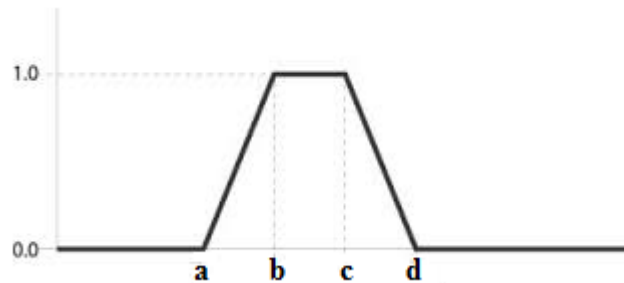


Figure 11. Trapezoidal Membership Functions

The formulation of Triangular membership function is;

$$\mu_F(x) = \begin{cases} 0, & x < a \\ \frac{x-a}{m-a}, & a \leq x < m \\ \frac{b-x}{b-m}, & m \leq x \leq b \\ 0, & x > c \end{cases} \quad (13)$$

Figure 12 shows the Triangular membership function.

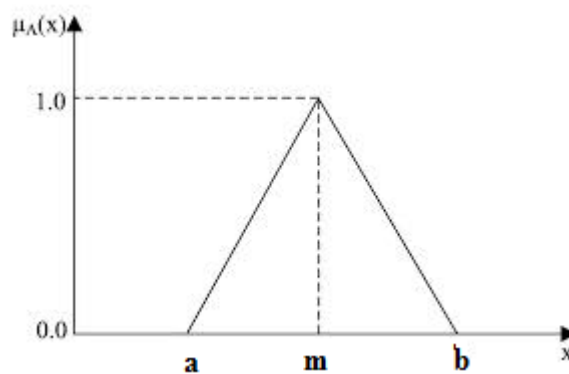


Figure 12. Triangular Membership Functions

Based on literature the Gaussian membership function is formulated by:

$$\mu_F(x) = e^{-\frac{(x-m)^2}{2K^2}} \quad (14)$$

Figure 13 shows the Gaussian membership function.

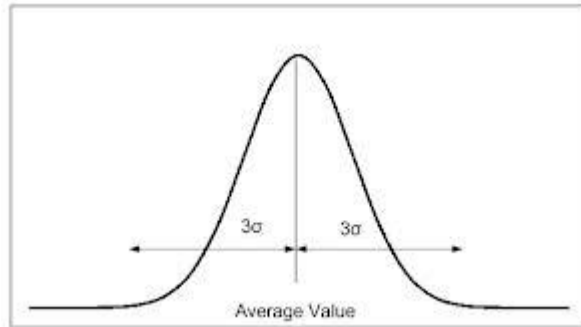


Figure 13. Gaussian Membership Functions

The variable in crisp set is numerical variable but in fuzzy logic theory the variables are linguistic variable. This item is used to describe the values of universe of discourse such as error by any words and sentences. Linguistic variable opens a window to application of fuzzy logic theory in many applications. In a fuzzy logic theory all numerical variables are replaced by words or sentences. For example various words of linguistic variables for any inputs or outputs are: Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (ZE), Positive Small (PS), Positive Medium (PM), Positive Big (PB). Figure 14 shows the membership function and linguistic variables in this graph the universe of discourse is error, the membership function is triangular and the linguistic variables are Negative Big (NB), Negative Small (NS), Zero (ZE), Positive Small (PS), Positive Big (PB).

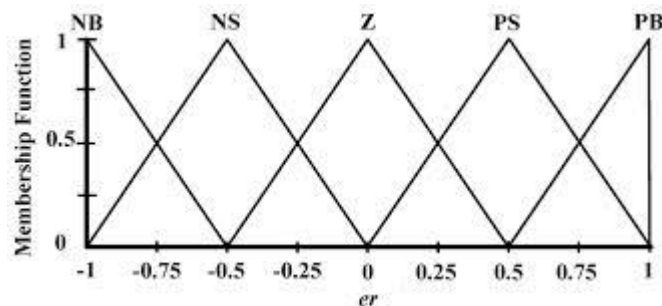


Figure 14. Membership Function and Linguistic Variables

According above discussion, fuzzy set used a membership function to support the membership value of its elements. Universe of discourse is defined by several linguistic variable and different membership functions. To define the behaviour of fuzzy logic set rule base is played important role. If-then rule is the main part to design rule base and rule table. If-then rule statements are used to formulate the condition statements in fuzzy logic theory. A fuzzy if-then rule is divided into two main groups:

- single fuzzy if-then rule
- Multiple fuzzy if-then rule

According to the literature if-then rule has two main parts:

- Antecedent part

- Consequent part

In single fuzzy if-then rule the antecedent part has one variable and it has only one part, following rule shows the single fuzzy if-then rule:

$$\mathbf{FR: If } e \text{ is } NB \text{ Then } U \text{ is } PB \quad (15)$$

In (15) e is input variable which in design controllers defined by fuzzy logic input, U is output variable, NB and PB are the Linguistic variables that can be defined by fuzzy set, the part of " e is NB " is called the antecedent part and the part of " U is PB " is called the Consequent or Conclusion part. In most of fuzzy controllers antecedent part has multiple parts; the following rule shows the multiple antecedent parts:

$$\mathbf{FR: If } e \text{ is } NB \text{ and } \dot{e} \text{ is } ML \text{ then } T \text{ is } PB \quad (16)$$

In (16) e and \dot{e} are inputs, T is output, Negative Big (NB), Medium Left (ML) and Positive Big (PB) are the linguistic variables, the part of " $If e$ is NB and \dot{e} is ML " is antecedent part and it multiple, in this state the fuzzy operations (AND/OR) is used and the part of " $then T$ is PB " is the consequent part. In most of fuzzy logic controllers, fuzzy controller inputs are used in antecedent part and the output of controller defined by consequent part.

This paper is organized as follows; Section 2 introduces and describes the methodology algorithm. Part Section 3 presents the simulation results and discussion of this algorithm applied to a continuum robot and the last part is described as conclusive.

2. Methodology

The nonlinear dynamic formulation problem in highly nonlinear system (*e.g.*, robot manipulator) can be solved by fuzzy logic theorem. Fuzzy logic theory is used to estimate the system dynamics. This type of controller is free of mathematical dynamic parameters of plant. To solve the challenge of system's dynamic especially in uncertain system, fuzzy controller is recommended.

$$\tau_{fuzzy} = U_{PID-fuzzy} \quad (17)$$

PID like fuzzy logic controller has three inputs, Proportional (P), Derivative (D), and Integrator (I), if each input defined by N linguistic variables to estimate the dynamic behavior, it has $N \times N \times N$ linguistic variables. Design fuzzy controller based on N^3 rule base for each link caused to creation lots of challenges in real timing application. To reduce the number of rule base in PID like fuzzy logic controller, parallel PD and PI strategy is recommended. According to this algorithm to design the same PID controller the number of rule base for each link is $2N^2$. According to this technique, the number of rule base is reduced with respect to have PID like fuzzy logic controller. After solve the first challenge about the number of rule base in PID like fuzzy logic controller, the second challenge is appears. To design parallel PD and PI like fuzzy controller, two types fuzzy rule table should be design. Design two types rule tables are very difficult and need to have much experience. Therefore in this research PI-like fuzzy controller is replaced by PD-like fuzzy controller with the integral term in the output. Due to this method, researcher can design PID like fuzzy logic controller based on PD rule table and $2N^2$ rule base.

According to fuzzy logic methodology definition;

$$U_{fuzzy} = (\sum_{l=1}^M \theta^T \zeta(x))_{e, \Sigma e, \dot{e}} \quad (18)$$

where θ^T is gain updating factor and $\zeta(x)$ is defined by;

$$\zeta(x) = \frac{\sum_i \mu(x_i) x_i}{\sum_i \mu(x_i)} \quad (19)$$

and the $\mu(x_i)$ parameter is membership function.

Fuzzy logic controller (FLC) is one of the most important applications of fuzzy logic theory. This controller can be used to control of nonlinear, uncertain, and noisy systems. However the application of fuzzy logic controller is really wide, all types of fuzzy logic controllers consists of the following parts;

- Choosing inputs
- Scaling inputs
- Input fuzzification (binary-to-fuzzy[B/F] conversion)
- Fuzzy rule base (knowledge base)
- Inference engine
- Output defuzzification (fuzzy-to-binary[F/B] conversion)
- Scaling output

Define the Inputs and Control Variables: In most of industrial controllers error and the functional of error are used as inputs to design controller. According to design the PD-like fuzzy controller, error and change of error are used to define as controllers' inputs. Therefore the antecedent part of rule base is comprised of two parts. In this part fuzzy controller's inputs are error (e) and change of error (\dot{e}) and the fuzzy controller output is PD fuzzy output ($U_{PD-fuzzy}$).

Scaling Inputs/Outputs: in fuzzy logic controller to define membership function, scaling the universe of discourse for all parts of rule base (consequent and antecedent part) is very important. The role of a right choice of scaling factors is obviously shown by the fact that if your choice is bad, the actual operating area of the inputs/outputs will be transformed into a saturation or narrow situation. Input scaling factors have played important role to basic sensitivity of the controller with respect to the optimal choice of the operating areas of the input signals moreover when the scale output is scaled, the gain updating factor of the controller is scaled which it is caused to modify the stability and oscillation tendency. Because of its strong impact on stability and reduce the oscillation, this factor is important factor to design fuzzy controller. In this research the scaling factor for error is $[-0.1 \text{ to } 0.1]$ and divided into eleven levels as follows:

$e = \{-0.1, -0.08, -0.06, -0.04, -0.02, 0, 0.02, 0.04, 0.06, 0.08, 0.1\}$ and the scaling factor of change of error is $[-1 \text{ to } 1]$ and divided into eleven levels as follows:

$\dot{e} = \{-1, -0.8, -0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6, 0.8, 1\}$ and at last the scaling factor of PD fuzzy output are between $[-1.5 \text{ to } 1.5]$.

Input Fuzzification (binary-to-fuzzy [B/F] conversion):

This part is divided into three main parts;

- Linguistic variables
- Scaling factor (normalization factor)
- Inputs membership function

In this research a linguistic variable is defined by;

- Symbolic name of inputs/outputs variables: *error, change of error* and *PD fuzzy output*.

- Set of linguistic values that for error can take on: Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (Z), Positive Small (PS), Positive Medium (PM), Positive Big (PB). The linguistic values for change of error are: Negative (N), Zero (Z) and Positive (P) and the linguistic variables for PD fuzzy output are: Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (Z), Positive Small (PS), Positive Medium (PM), Positive Big (PB).
- Scaling factor as actual physical domain over which the meaning of the linguistic value, based on experience knowledge this range for error is $[-0.1 \text{ to } 0.1]$, for change of error is $[-1 \text{ to } 1]$ and finally for PD fuzzy output is $-1.5 \text{ to } 1.5]$.

According to experience knowledge in this research, triangular membership function is selected for inputs and output. Figures 15 to 17 show the fuzzification part for PD like fuzzy controller system, this controller has two inputs (error and change of error) and one output (PD fuzzy output), the first input (error) is described with seven linguistic values; Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (Z), Positive Small (PS), Positive Medium (PM), Positive Big (PB) and it is quantized into eleven levels as follows: $e = \{-0.1, -0.08, -0.06, -0.04, -0.02, 0, 0.02, 0.04, 0.06, 0.08, 0.1\}$, the second input (change of error) is described with three linguistic values; Negative (N), Zero (Z) and Positive (P) and it is quantized into eleven levels as: $\dot{e} = \{-1, -0.8, -0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6, 0.8, 1\}$ and the output (PD fuzzy output) is described with seven linguistic values; Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (Z), Positive Small (PS), Positive Medium (PM), Positive Big (PB) and triangular membership functions are used for inputs and output.

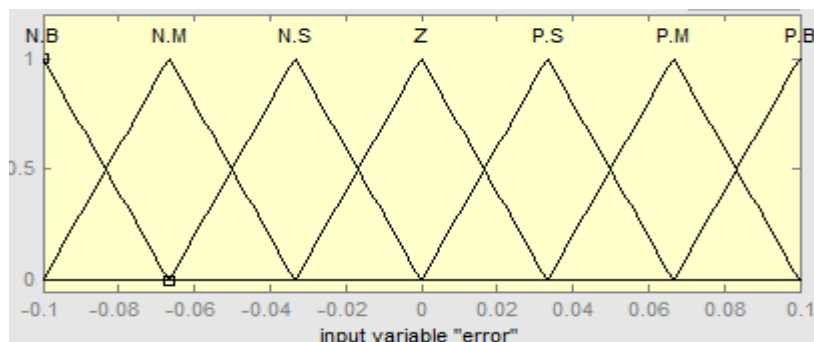


Figure 15. Membership Function, Scaling and Linguistic Variables for Error

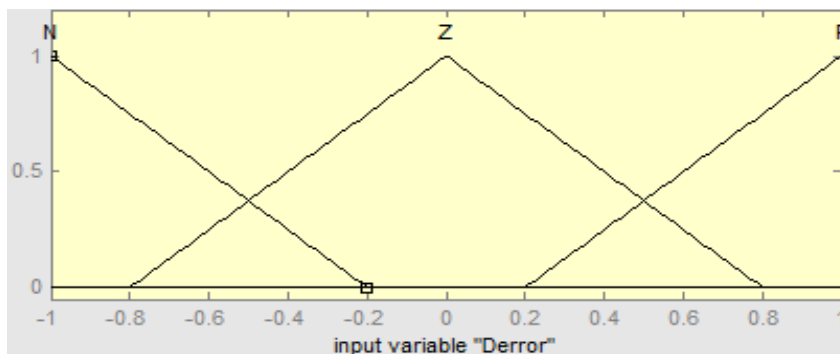


Figure 16. Membership Function, Scaling and Linguistic Variables for Change of Error

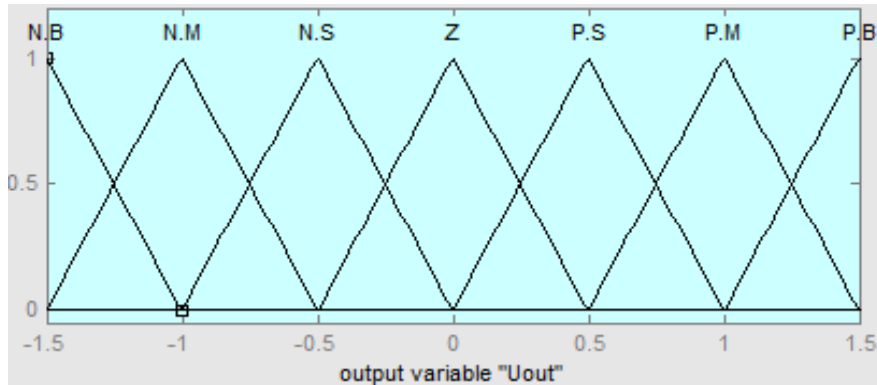


Figure 17. Membership Function, Scaling and Linguistic Variables for PD Fuzzy Output

Fuzzy rule Base: the role of the rules in fuzzy logic controller is extremely significant and the main approaches and source of fuzzy logic controller rules are;

- Expert experience and knowledge base
- Learning based on operators' control action
- Identification of fuzzy model system under control action
- The application of learning technique

According to above, the main approach comes from an expert knowledge of system because any fuzzy controller is expert system to solve the control problem. According to fuzzification the error has seven linguistic variables, the change of error has three linguistic variables and the PD fuzzy output has seven linguistic variables. Therefore PD like fuzzy controller has 21 rule-bases in five parts as follows:

Part 1:

FR^1 : IF e is PS and \dot{e} is Z then U_{PD} is NS

FR^2 : IF e is Z and \dot{e} is Z then U_{PD} is Z

FR^3 : IF e is NS and \dot{e} is Z then U_{PD} is PS

According to first three rule-base error is positive or negative small or zero and change of error is zero. In this case the system's output (U_{PD}) has close deviation around the desired level. Therefore these three rules are related to steady state system's output behavior. In this case if error is positive small to estimate it, the controllers output needs to change the direction with the same power.

Part 2:

FR^4 : IF e is PB and \dot{e} is N then U_{PD} is Z

FR^5 : IF e is PM and \dot{e} is N then U_{PD} is PS

In this part the error is Positive Big or medium, therefore based on error formulation ($e = q_d - q_a$) the desired input is considerably above the actual input. In this time the rate of error is negative, it means that actual input is moving towards to the desired input and caused to reduce the error towards to zero. The control action should to tune the rate of reduce the error. For example when error is Positive Big and change of error is Negative, no control action is recommended because the actual input will be estimate by the speed of change of error due to the desired input.

Part 3:

FR^6 : IF e is PS and \dot{e} is N then U_{PD} is PM

FR^7 : IF e is Z and \dot{e} is N then U_{PD} is PB

FR^8 : IF e is NS and \dot{e} is N then U_{PD} is PB

FR^9 : IF e is NM and \dot{e} is N then U_{PD} is PB

FR¹⁰: IF e is NB and è is N then U_{PD} is PB

In this part the actual input is near the desired input (e(t) is Positive Small, Zero or Negative Small) or the actual input is drastically above it (e(t) is Negative Medium or Negative Big) and at this time the rate of error is negative, it means the rate of actual input is greater than desired input and caused to actual input moving away from desired input. In this time, the role of controller is to reverse this trend and caused actual input start to moving toward to the desired input. According to part 3 rule bases the trend of error will be reduces.

Part 4:

FR¹¹: IF e is NM and è is P then U_{PD} is NS

FR¹²: IF e is NB and è is P then U_{PD} is Z

FR¹³: IF e is NM and è is Z then U_{PD} is PM

FR¹⁴: IF e is NB and è is Z then U_{PD} is PB

For this group the actual input is drastically below the desired input (e(t) is Negative Medium or Negative Big) and at this time the rate of error is Positive or Zero, it means the rate of actual input is lower than desired input and caused to actual input moving toward to desired input. In this time, the role of controller is to speed control to reduce the error.

Part 5:

FR¹⁵: IF e is NS and è is P then U_{PD} is NM

FR¹⁶: IF e is Z and è is P then U_{PD} is NB

FR¹⁷: IF e is PS and è is P then U_{PD} is NB

FR¹⁸: IF e is PM and è is P then U_{PD} is NB

FR¹⁹: IF e is PB and è is P then U_{PD} is NB

FR²⁰: IF e is PM and è is Z then U_{PD} is NM

FR²¹: IF e is PB and è is Z then U_{PD} is NB

This part is very similar to part 3. In this group, the actual input is near the desired input (e(t) is Negative Small, Zero or Positive Small) or the actual input is drastically below it (e(t) is Positive Medium or Positive Big) and at this time the rate of error is positive or Zero, it means the rate of actual input is lower than desired input and caused to actual input moving away from desired input. In this time, the role of controller is to reverse this trend and caused actual input start to moving toward to the desired input.

The PD like fuzzy rule table shows in Table 1.

Table 1. Rule Table in PD like Fuzzy Logic Controller

$\begin{matrix} e \\ \text{è} \end{matrix}$	PB	PM	PS	Z	NS	NM	NB
P	NB	NB	NB	NB	N M	NS	Z
Z	NB	NM	NS	Z	PS	PM	PB
N	Z	PS	PM	PB	PB	PB	PB

Inference Engine (Fuzzy rule processing): The fuzzy inference engine recommends a fuzzy method to transfer the fuzzy rule base to fuzzy set. Mamdani and Sugeno methods are two main techniques of fuzzy rule processing. In this research Mamdani fuzzy inference engine is used as fuzzy rule processing.

Defuzzification: defuzzification is the last step to design fuzzy logic controller and it is used to transform fuzzy set to crisp set. Consequently defuzzification's input is the aggregate output and the defuzzification's output is a crisp number. Centre of gravity method (*COG*) and Centre of area method (*COA*) are two types method to calculate the defuzzifications.

Figure 18 shows the output rule base viewer in PD like fuzzy controller.

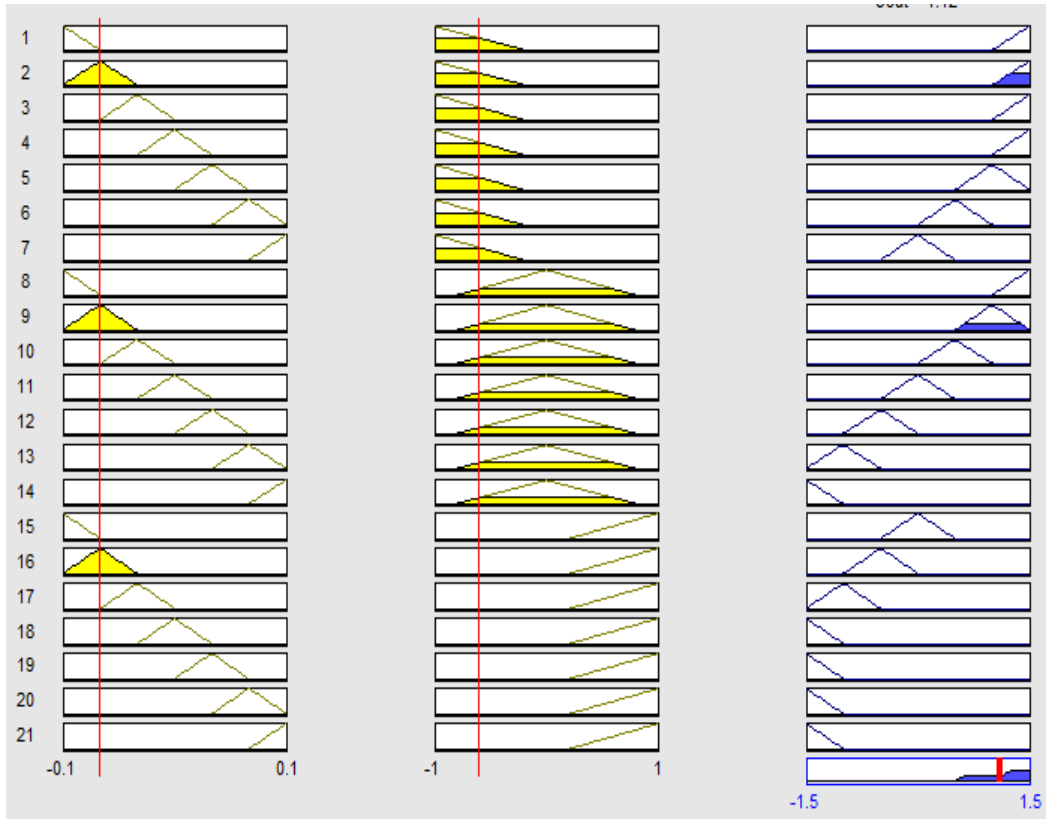


Figure 18. Rule Viewers in PD-like Fuzzy Controller

According to design fuzzy logic controller, Figure 19 shows PD like fuzzy logic controller.

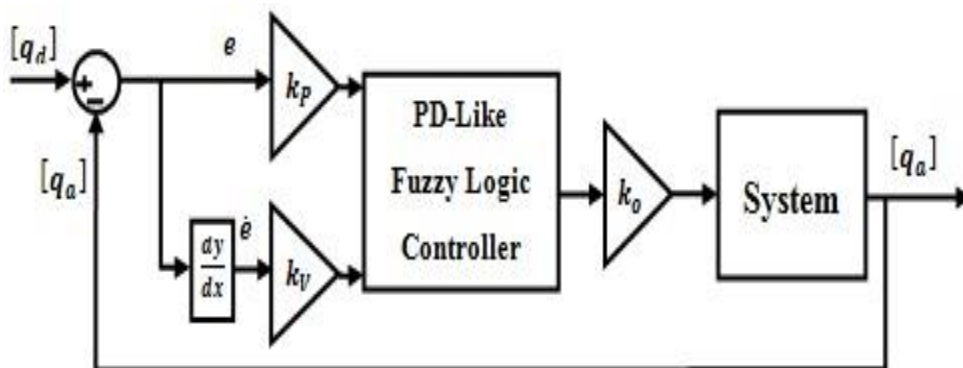


Figure 19. PD-like Fuzzy Logic Controller

Table 2 shows the PD like fuzzy logic controller lookup table.

Table 2. Lookup Table in PD like Fuzzy Logic Controller

$\begin{matrix} e \rightarrow \\ \downarrow \dot{e} \end{matrix}$	-0.1	-0.08	-0.06	-0.04	-0.02	0	0.02	0.04	0.06	0.08	1
-1	1.34	1.32	1.31	1.3	1.28	1.25	1.04	0.88	0.62	0.291	0
0.8	1.33	1.32	1.31	1.2	1.27	1.24	1.03	0.877	0.61	0.291	0
-0.6	1.31	1.12	0.923	0.888	0.635	0.619	0.464	0.222	0.1	-0.11	-0.3
-0.4	1.31	1.05	0.88	0.695	0.441	0.3	0.109	-0.074	-0.223	-0.8	-0.62
-0.2	1.28	1.04	0.877	0.622	0.291	0	-0.291	-0.623	-0.878	-1.04	-1.32
0	1.28	1.04	0.877	0.622	0.291	0	-0.291	-0.623	-0.9	-1.04	-1.32
0.2	1.28	1.04	0.877	0.622	0.291	0	-0.291	-0.623	-0.9	-1.04	-1.32
0.4	0.619	0.464	0.222	0.08	-0.11	-0.3	-0.441	-0.695	-0.9	-1.1	-1.34
0.6	0.298	0.109	-0.074	-0.222	-0.646	-0.619	-0.635	-0.89	-0.924	-1.33	-1.35
0.8	-0	-0.291	-0.622	-0.87	-1.04	-1.34	-1.32	-1.34	-1.33	-1.35	-1.37
1	0	-0.291	-0.63	-0.88	-1.04	-1.25	-1.32	-1.34	-1.35	-1.36	-1.38

In this research PI-like fuzzy logic controller is design based on PD-like fuzzy to reduce the time to design fuzzy rule base. Due to previous discussion, fuzzy rule bases are the controller's behavior and design two types of fuzzy rule bases are difficulties. Therefore design PI like fuzzy logic controller based on PD like fuzzy logic controller is introduced in this part. To design PI like fuzzy logic controller based on PD like fuzzy logic controller integral term is added to the PD controller's output. The formulation of PI controller is;

$$U_{PI} = K_p \times e + K_i \left(\frac{1}{T} \int e \cdot dt \right) = K_p \times e + K_i \sum e \quad (20)$$

If the derivative, with respect to time is taken for (20);

$$\frac{dU_{PI}}{dt} = K_p \times \frac{de}{dt} + K_i e(t) = K_p \times \dot{e} + K_i e \quad (21)$$

According to (21) to calculate the output control system integral term is used as follows;

$$\int \frac{dU_{PI}}{dt} = [K_p \times \int \left(\frac{de}{dt} \right) dt + K_i \int (e(t) dt)] = K_p \times e + K_i \times \sum e \quad (22)$$

Figure 20 shows PI controller based on PD controller. According to Figure 20 PI like fuzzy logic controller is design based on PD like fuzzy logic controller. Figure 21 shows PI like fuzzy logic controller based on PD like fuzzy logic controller.

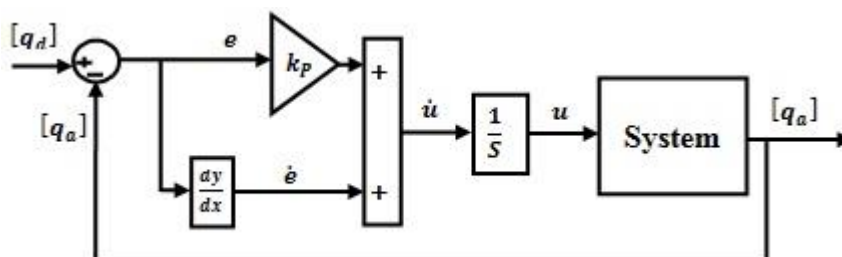


Figure 20. Design PI Controller Based on PD Controller

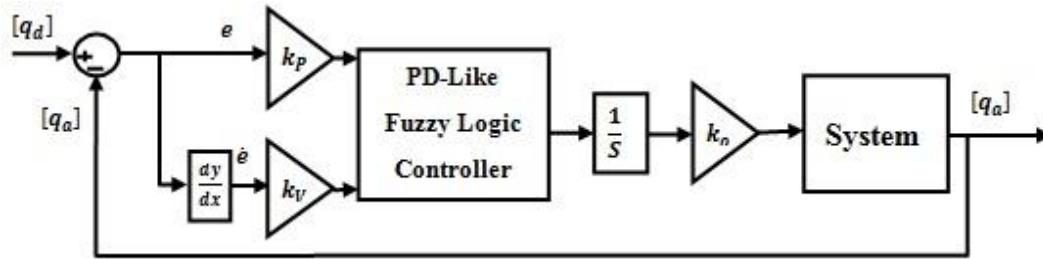


Figure 21. Design PI like Fuzzy Controller Based on PD like Fuzzy Controller

To design PI like fuzzy logic controller based on PD fuzzy logic controller following seven steps are recommended as well as design PD like fuzzy logic controller.

- Choosing inputs
- Scaling inputs
- Input fuzzification (binary-to-fuzzy[B/F] conversion)
- Fuzzy rule base (knowledge base)
- Inference engine
- Output defuzzification (fuzzy-to-binary[F/B] conversion)
- Scaling output

A fuzzy logic controller is a nonlinear controller and this type of nonlinear controller does not require the dynamic model of nonlinear system to be controlled. Therefore it can be used to nonlinear system (robot manipulator) control without finding nonlinear dynamic model and solve the complicated equations. Figure 22 illustrates the general structure of the PID like fuzzy logic controller, which consists of two main components. The PID like fuzzy logic controller is built using PD like fuzzy logic controller and PI like fuzzy logic controller. According to design PD like fuzzy controller and PI like fuzzy controller based on PD fuzzy rule base, design PID like fuzzy controller is introduced based on the following formulation;

$$U_{PID} = U_{PI} + U_{PD} = \left(\frac{K_p}{2}\right) \times e + K_i \left(\frac{1}{T} \int e. dt\right) + \left(\frac{K_p}{2}\right) \times e + K_v \dot{e} \quad (23)$$

$$U_{PIDlike\ fuzzy} = [(\sum_{l=1}^M \theta^T \zeta(x))_{e,\Sigma} e + (\sum_{l=1}^M \theta^T \zeta(x))_{e,\dot{e}} \dot{e}] \quad (24)$$

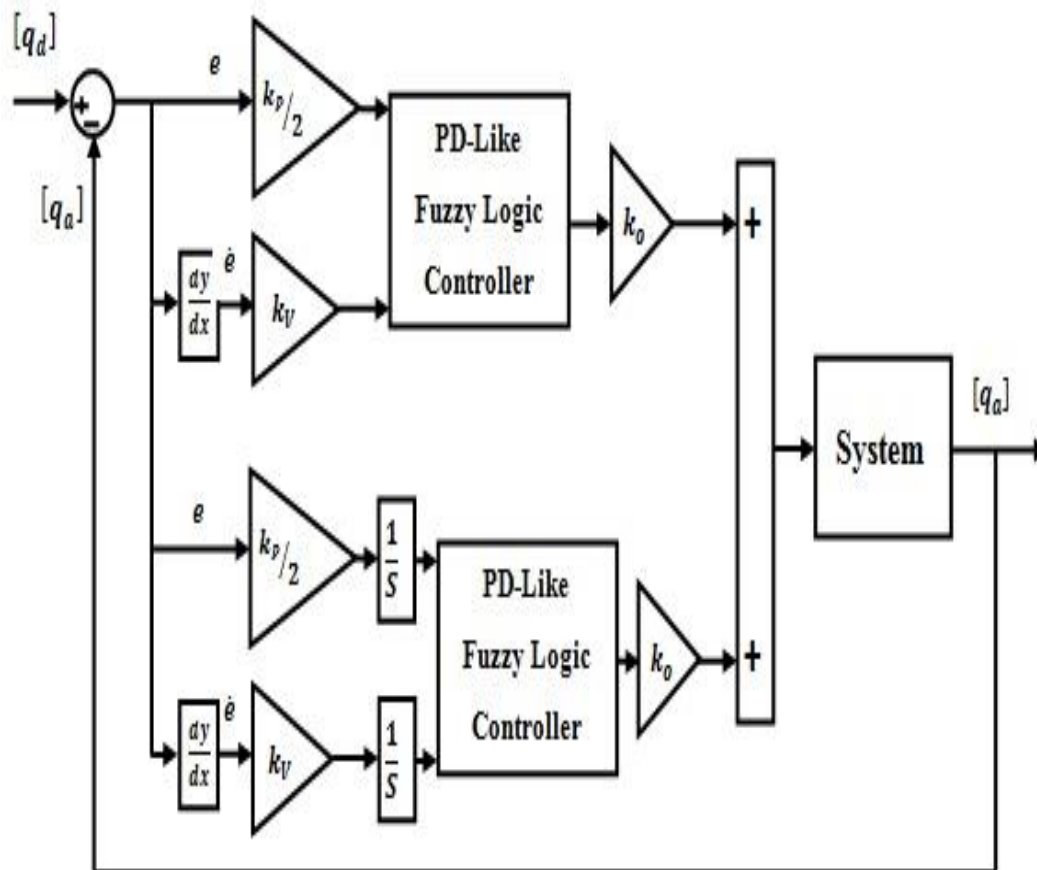


Figure 22. Design PID like Fuzzy Controller

PID like fuzzy logic controllers consists of the following parts;

- Choosing inputs
- Scaling inputs
- Input fuzzification (binary-to-fuzzy[B/F] conversion)
- Fuzzy rule base (knowledge base)
- Inference engine
- Output defuzzification (fuzzy-to-binary[F/B] conversion)
- Scaling output

Define the Inputs and Control Variables: In PID-like fuzzy controller, error and change of error are used to define as controllers' inputs. Therefore the antecedent part of rule base is comprised of two parts. In this part fuzzy controller's inputs are error (e) and change of error (\dot{e}) and the fuzzy controller output is PD fuzzy output ($U_{PD-fuzzy}$) and PI fuzzy output ($U_{PI-fuzzy}$).

Scaling Variables and Input fuzzification (binary-to-fuzzy [B/F] conversion):

Proposed PID like fuzzy logic controller has two inputs *error, change of error* and two different output; *PD fuzzy output and PI fuzzy output*. Error defined as seven linguistic variables: Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (Z), Positive Small (PS), Positive Medium (PM), Positive Big (PB). Based on experience knowledge the range of scaling factor for error is $[-0.1 \text{ to } 0.1]$ and it is quantized into eleven levels as follows: $e = \{-0.1, -0.08, -0.06, -0.04, -0.02, 0, 0.02, 0.04, 0.06, 0.08, 0.1\}$. The linguistic

values for change of error are: Negative (N), Zero (Z) and Positive (P) and the range of scaling factor for change of error is $[-1 \text{ to } 1]$ and it is quantized into eleven levels as: $\dot{e} = \{-1, -0.8, -0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6, 0.8, 1\}$. The linguistic variables for PID like fuzzy logic controller's output divided into two main parts, PD like fuzzy logic controller and PI like fuzzy logic controller which, linguistic variables for output PD like fuzzy logic controller and PI like fuzzy logic controller are: Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (Z), Positive Small (PS), Positive Medium (PM), Positive Big (PB) and the scaling factor for them are $[-1.5 \text{ to } 1.5]$.

Fuzzy rule Base: the main approach comes from an expert knowledge of system because fuzzy controller is one of the expert system to solve the control problem. According to fuzzification the error has seven linguistic variables, the change of error has three linguistic variables and the PD fuzzy output and PI like fuzzy logic controller have seven linguistic variables. Therefore PID like fuzzy controller has 42 rule-bases. The PID like fuzzy rule table shows in Table 3.

Table 3. Rule Table in PID like Fuzzy Logic Controller

$\begin{matrix} e \\ \dot{e} \end{matrix}$	<i>PB</i>	<i>PM</i>	<i>PS</i>	<i>Z</i>	<i>NS</i>	<i>NM</i>	<i>NB</i>
<i>P</i>	NB	NB	NB	NB	NM	NS	Z
<i>Z</i>	NB	NM	NS	Z	PS	PM	PB
<i>N</i>	Z	PS	PM	PB	PB	PB	PB

Inference Engine (Fuzzy rule processing): In this research 42 rule bases Mamdani fuzzy inference engine is used as fuzzy rule processing.

Defuzzification: defuzzification is the last step to design fuzzy logic controller and it is used to transform fuzzy set to crisp set. In PID like fuzzy logic controller COG method is used for defuzzification.

3. Results and Discussion

PID like fuzzy controller and PID controller are compare in this research. These controller are tested in certain and uncertain situation.

Comparison of the Tracking Data and Information: the trajectory following for PID like fuzzy controller and PID controller are compared in this section. According to Figure 23, the rise time in PID controller is better than PID like fuzzy controller but conventional PID controller has error. In error point of view, PID like fuzzy controller is better than PID controller.

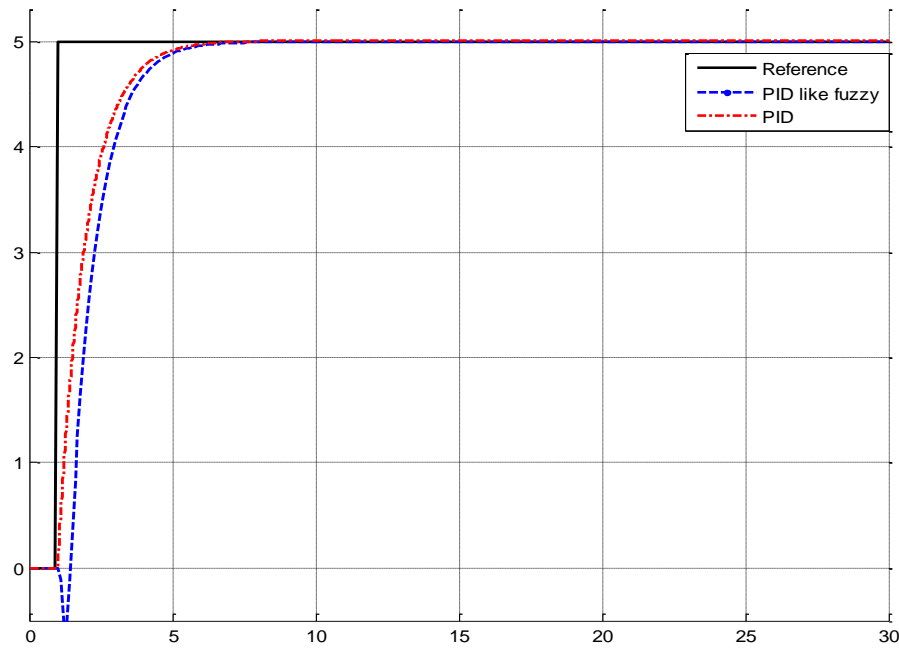


Figure 23. Tracking Data: PID like Fuzzy Controller and PID Controller

Comparison the disturbance rejection: the power of disturbance rejection is very important to robust checking in any controllers. In this section trajectory accuracy is test under uncertainty condition. To test the disturbance rejection band limited white noise with 30% amplitude is applied PID like fuzzy controller and PID controller. In Figures 24, trajectory accuracy is shown.

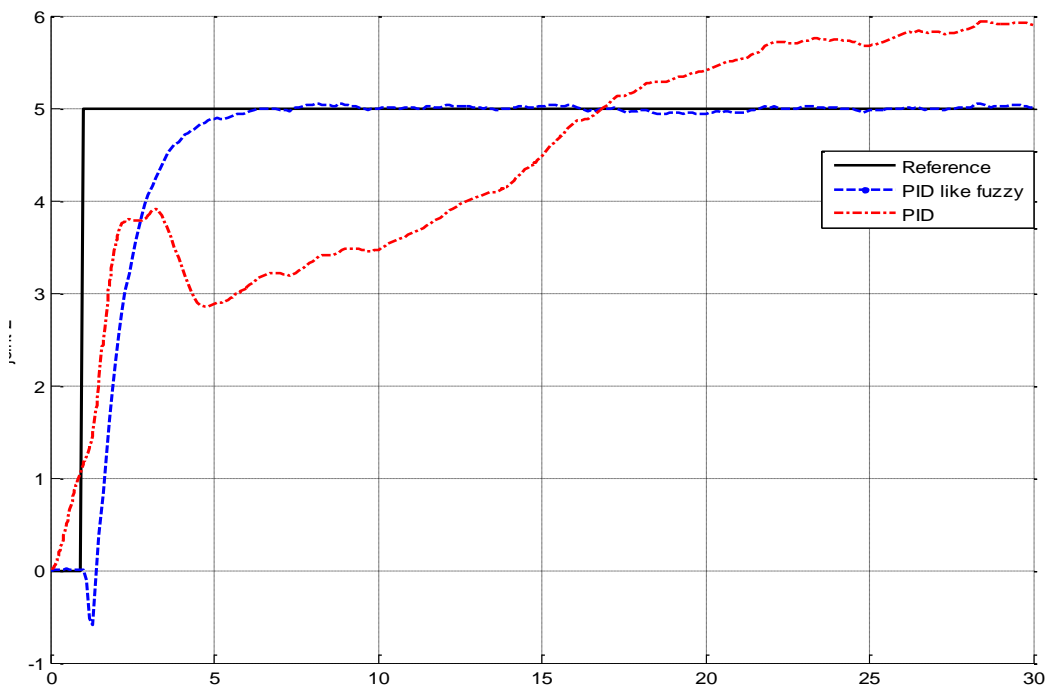


Figure 24. Tracking Data: PID like Fuzzy Controller and PID Controller in Presence of Uncertainty

According to above graph, however PID like controller has suitable oscillation in presence of uncertainty but it is more robust than PID controller. PID controller has very much fluctuation in presence of external disturbance.

4. Conclusions

Conventional nonlinear controllers are worked based on system's dynamic model. Based on equivalent part in conventional nonlinear controllers, in complex and highly nonlinear systems these controllers have many problems for accurate responses because these type of controllers need to have accurate knowledge of dynamic formulation of system. The nonlinear dynamic formulation problem in highly nonlinear system (*e.g.*, robot manipulator) can be solved by fuzzy logic theorem. Fuzzy logic theory is used to estimate the system dynamics. This type of controller is free of mathematical dynamic parameters of plant. When system works in uncertainty, the nonlinearity term of robot manipulator will be different. To reduce the role of nonlinearity term PID like fuzzy logic controller is used in this research. To design minimum rule base PID like fuzzy logic controller, parallel PD and PI strategy is recommended. According to this technique, the number of rule base is reduced with respect to have PID like fuzzy logic controller. To avoid of design two types fuzzy rule base in proposed PID like fuzzy logic controller, integral term of PI like fuzzy logic controller is replaced by PD like fuzzy logic controller. According to above algorithm to design the same PID controller the number of rule base for each link is $2 \times N \times M$. Where N is the number of linguistic variables for error and M is the number of linguistic variables for change of error.

Acknowledgment

The authors would like to thank the anonymous reviewers for their careful reading of this paper and for their helpful comments. This work was supported by the Iranian Institute of Advance Science and Technology Program of Iran under **Grant no. 2012-Persian Gulf-2A**.

Iranian center of Advance Science and Technology (IRAN SSP) is one of the independent research centers specializing in research and training across of Control and Automation, Electrical and Electronic Engineering, and Mechatronics & Robotics in Iran. At IRAN SSP research center, we are united and energized by one mission to discover and develop innovative engineering methodology that solve the most important challenges in field of advance science and technology. The IRAN SSP Center is instead to fill a long standing void in applied engineering by linking the training a development function one side and policy research on the other. This center divided into two main units:

- Education unit
- Research and Development unit

References

- [1] T. R. Kurfess, Robotics and automation handbook: CRC, (2005).
- [2] J. J. E. Slotine and W. Li, "Applied nonlinear control", Prentice hall Englewood Cliffs, NJ, vol. 461, (1991).
- [3] L. Cheng, "Multi-agent based adaptive consensus control for multiple manipulators with kinematic uncertainties", (2008), pp. 189-194.
- [4] J. J. D'Azzo, "Linear control system analysis and design with MATLAB: CRC", (2003).
- [5] B. Siciliano and O. Khatib, Springer handbook of robotics: Springer-Verlag New York Inc, (2008).

- [6] F. Piltan, N. Sulaiman, A. Zargari, M. Keshavarz and A. Badri, "Design PID-Like Fuzzy Controller with Minimum Rule Base and Mathematical Proposed On-line Tunable Gain: Applied to Robot Manipulator", International Journal of Artificial Intelligence and Expert System, vol. 2, no. 4, (2011), pp. 184-195.
- [7] F. Piltan, S. H. Tayebi Haghghi, N. Sulaiman, I. Nazari and S. Siamak, "Artificial Control of PUMA Robot Manipulator: A-Review of Fuzzy Inference Engine and Application to Classical Controller", International Journal of Robotics and Automation, vol. 2, no. 5, (2011), pp. 401-425.
- [8] L. Reznik, "Fuzzy controllers: Butterworth-Heinemann", (1997).
- [9] Z. Kovacic and S. Bogdan, "Fuzzy controller design: theory and applications: CRC/Taylor & Francis", (2006).
- [10] J. Zhou and P. Coiffet, "Fuzzy control of robots", IEE proceeding Control Theory and Applications, vol. 147, no. 2, (2002), pp. 1357-1364.
- [11] S. Banerjee and P. Y. Woo, "Fuzzy logic control of robot manipulator", Second IEEE conference on Control Applications, (2002), pp. 87-88.

Authors



Ghasem Sahamijoo has been working at "**Filtering the hand tremors in flexible surgical robot for Experimental Research and Education**" project at Iranian Institute of Advance Science and Technology, Sanaat Kade Sabz Passargad Research Center (IRAN SSP) as "**Student researcher**" of a research team composed of 34 researchers since Jan. 2013 to date.



Omid Avatefipour has been working at "**Filtering the hand tremors in flexible surgical robot for Experimental Research and Education**" project at Iranian Institute of Advance Science and Technology, Sanaat Kade Sabz Passargad Research Center (IRAN SSP) as "**Student researcher**" of a research team composed of 34 researchers since Jan. 2013 to date.



Mahmoud Reza Safaei Nasrabad has been working at "**Filtering the hand tremors in flexible surgical robot for Experimental Research and Education**" project at Iranian Institute of Advance Science and Technology, Sanaat Kade Sabz Passargad Research Center (IRAN SSP) as "**Student researcher**" of a research team composed of 34 researchers since Jan. 2013 to date.



Mohammad Taghavi has been working at "**Filtering the hand tremors in flexible surgical robot for Experimental Research and Education**" project at Iranian Institute of Advance Science and Technology, Sanaat Kade Sabz Passargad Research Center (IRAN SSP) as "**Student researcher**" of a research team composed of 34 researchers since Jan. 2013 to date.



Farzin Piltan was born on 1975, Shiraz, Iran. In 2004 he is jointed Institute of Advance Science and Technology, Research and Development Center, IRAN SSP. Now he is a dean of Intelligent Control and Robotics Lab. He is led of team (47 researchers) to design and build of nonlinear control of industrial robot manipulator for experimental research and education and published about 54 Papers in this field since 2010 to 2012, team supervisor and leader (9 researchers) to design and implement intelligent tuning the rate of fuel ratio in internal combustion engine for experimental research and education and published about 17 Journal papers since 2011 to 2013, team leader and advisor (34 researchers) of filtering the hand tremors in flexible surgical robot for experimental research and education and published about 31 journal papers in this field since 2012 to date, led of team (21 researchers) to design high precision and fast dynamic controller for multi-degrees of freedom actuator for experimental research and education and published about 7 journal papers in this field since 2013 to date, led of team (22 researchers) to research of full digital control for nonlinear systems (e.g., Industrial Robot Manipulator, IC Engine, Continuum Robot, and Spherical Motor) for experimental research and education and published about 4 journal papers in this field since 2010 to date and finally led of team (more than 130 researchers) to implementation of Project Based-Learning project at IRAN SSP research center for experimental research and education, and published more than 110 journal papers since 2010 to date. In addition to 7 textbooks, Farzin Piltan is the main author of more than 115 scientific papers in refereed journals. He is editorial review board member for ‘international journal of control and automation (IJCA), Australia, ISSN: 2005-4297; ‘International Journal of Intelligent System and Applications (IJISA)’, Hong Kong, ISSN:2074-9058; ‘IAES international journal of robotics and automation, Malaysia, ISSN:2089-4856; ‘International Journal of Reconfigurable and Embedded Systems’, Malaysia, ISSN:2089-4864. His current research interests are nonlinear control, artificial control system and applied to FPGA, robotics and artificial nonlinear control and IC engine modeling and control.

