

Performance Study of PID and Fuzzy Controllers for Position Control of 6 DOF arm Manipulator with Various Defuzzification Strategies

Ahmed Kharidege , Ding Jianbiao, Yajun Zhang

College of Mechanical and Electrical Engineering, Beijing University of Chemical Technology, Beijing 100029, China

Abstract. Today's arm manipulators are more and more demanding in terms of productivity. Conventional controllers are not always able to provide good and accurate results. To complete a position movement of the manipulator's end-effector, a set of joint angles of manipulator first required to be converted to the position coordinates by using the forward kinematics method, and each joint rotation is executed using a servomotor feedback control. The kinematic model has been validated using MATLAB® robotics toolbox. An end-effector based 6 degree of freedom (6-DOF) platform is proposed in this work which uses DC servomotor for actuation of the three revolute joints. PID controller is used as a reference benchmark. And FLC controller with different defuzzification strategies was employed. Results were compared in terms of time response criteria. Simulation results using MATLAB are demonstrated that PID has superior performance in terms of transient parameters. In Steady state response, both PID and FLC manage to converge to the desired output but in terms of overshoot FLC is outperformed.

1 Introduction

Arm manipulators are the most commonly used robots in industrial applications. Arm manipulator consists of a series of links, connected by joints, which can be moved translationally or rotationally. Parts of all of the joints are actuated and extend from a base to an end-effector. Arm manipulator is analogous to the human hand, and the end-effector can be designed to perform any desired task [1]. To successfully fulfil the desired task, the kinematics and dynamics of a manipulator must be studied first, and then the controllers also need to be designed and implemented in the mechanical manipulator system.

Various controllers have been designed and applied in the robot manipulator. Different types of these controllers in terms of performance index will be evaluated. Proportional Integral Derivative (PID) controller may be the most widely used controller in the industrial and commercial applications for the early decades. Due to its simplicity of designing and implementation, so the first attempt is to apply PID control.

Due to high reliabilities, flexibilities and low costs, DC motors are widely used in industrial applications, robot manipulators and home appliances where speed and position control of motor are required. Numerous papers about DC motor fuzzy control system design. In [2] authors reviewed fuzzy-logic applications for robot systems and flexible links manipulators. Delibasi and Türker presented the position control of DC motor using FLC and PID control algorithms. The two controllers were designed based on LabVIEW program. Results

showed that the fuzzy controller better than using PID controller for achieving the desired position. They featured the influence of fuzzy logic controller upon the performance of robot movement simulation [3]. Lin *et al.* compared PID and FLC for position control and observed that FLC performed better than PID [4]. Azevedo *et al.* have shown that FLC is less sensitive than PID to load variations [5]. Sreentha and Pradhanb designed fuzzy controller for the position control of single flexible link revolute. In this controller, time rate of change and error in joint angular displacement as input variables were used. Both theoretical and experimental results showed that the angular displacement at the base joint is highly oscillatory and settling time extremely large [6]. Mishra *et al.*, made a comparison between PID and FLC for servomotor control and described that PID parameters had to be tuned again under variations of plant parameters external disturbance or noise [7]. Young and Shih designed Fuzzy-PID controller of a six axis serial manipulator. Results compared with PID and two fuzzy controllers, Study shown that Fuzzy-PID has smaller steady state errors [8].

In this work, 6 DOF robot Manipulator proposed the robotic arm has five revolute joints include tool, wrist, elbow, shoulder, waist or base joints as shown in Figure 1. An important contribution is the use of servomotors for actuation of the revolute joints. The advantages of the servomotors is that they have a high accuracy and efficiency and reliability, low maintainability, and good dynamic control for variable speed applications. The rest of the paper are organized as follows: section 2 describes the kinematic model of the proposed platform, section 3

describes the DC servo motor design and value of transfer function. Different control schemes were introduced in section 4, simulations carried out in section 5 and the last section is the result analysis.

2 The Robot Model

In this work, 6 DOF robot Manipulator proposed the robotic arm has five revolute joints include tool, wrist, elbow, shoulder, waist or base joints as shown in Fig. 1. This robot is available for studies in machinery design and theory Laboratory in Beijing University of Chemical Technology, and used by students of Robotics.

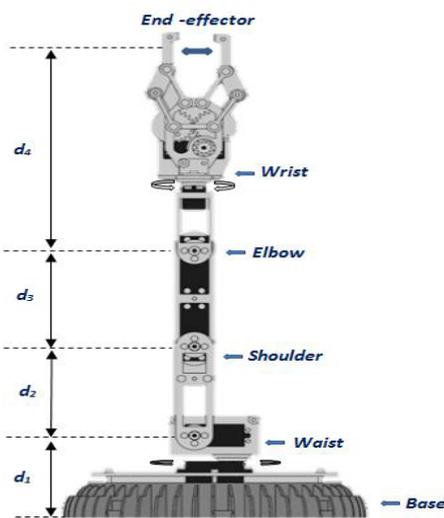


Figure 1. Arm Manipulator Model.

2.1 The kinematic model

Kinematic problem is associated with the motion of robots. It does not deal with the forces that cause the motion. It is further divided into two categories, a) Forward Kinematic, b) Inverse Kinematics, forward kinematic deals with the computation of end-effector coordinates knowing the achieved joint angles, the most commonly used methods for kinematic modelling are Denavit-Hartenberg (D-H) and successive screw displacement, In D-H formulation approach, first D-H parameters are defined. Having the knowledge of these parameters, the kinematic model can be described for any robot [9]. In contrast, inverse kinematic model computes the required joint angles for the given coordinates and is more complex than forward kinematic [10].

The relationship between the individual joints of the rehabilitation device and the position and orientation of the robot's end-effector is expressed concisely by the four D-H parameters given in Table.1. The four parameters a_i , α_i , d_i , θ_i are generally known as the link length, link twist, link offset, and joint angle respectively [11].

Table 1. The link parameters of Manipulator (D-H parameters).

No	$\alpha_{i-1}(\circ)$	$a_{i-1}(mm)$	$d_{i-1}(mm)$	$\theta_{i-1}(\circ)$
1	90	0	0	θ_1
2	0	0	d_2	θ_2
3	0	0	d_3	θ_3
4	-90	0	d_4	θ_4

5	0	0	0	θ_5
6	0	0	0	End-effector

Table 2. The link lengths of Manipulator.

joint	waist	shoulder	elbow	wrist
symbol	d_1	d_2	d_3	d_4
Link length (MM)	93	80	81	172

Based on these parameters, the transformation matrix T_6^0 includes the overall rotation and translation of tool frame {6} with respect to base frame {0}. The transformation matrix is given by (1):

$$T_6^0 = \begin{bmatrix} C_1(C_{234}C_5 - S_{234}S_5) & C_1(-C_{234}S_5 - S_{234}C_5) & -C_1(C_2S_4 + S_2C_4) & C_1(C_{234}d_4 + C_2d_3 + C_2d_2) \\ S_1(C_{234}C_5 - C_{234}S_5) & S_1(-C_{234}S_5 + C_2C_5) & -S_1(C_2S_4 + S_2C_4) & S_1(C_{234}d_4 + C_2d_3 + C_2d_2) \\ S_{234}C_4 + C_{234}C_4 & -S_{234}S_4 & -S_4(S_{234} + C_2C_{23}) & S_{234}d_4 + S_2d_3 + S_2d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where:

$$C_i = \cos \theta_i, S_i = \sin \theta_i, S_{12} = S_1C_2 + C_1S_2, C_{12} = C_1C_2 - S_1S_2$$

$$C_{234} = C_2(C_3C_4 - S_3S_4) - S_2(S_3C_4 + C_3S_4), S_{234} = S_2(C_3C_4 - S_3S_4) + C_2(S_3C_4 + C_3S_4)$$

Based on the user-defined desired angles, the developed forward kinematic model computes the corresponding positions. These positions are calculated using MATLAB©, Fig. 2. shows the side and top view.

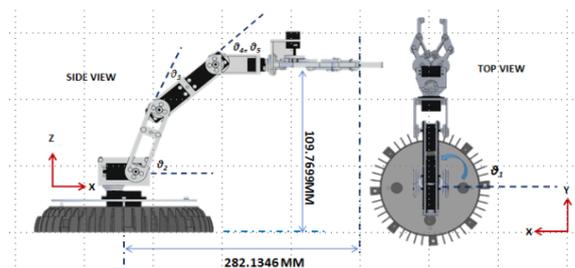


Figure 2. Side and top view.

2.2 Forward kinematic model validation

Going one step forward, for well visualizing, the derived forward kinematic model has been validated using MATLAB® environment, The Toolbox developed by Peter Corke provides functions for manipulating and converting between datatypes e.g (vectors, homogeneous transformations, roll-pitch-yaw and Euler angles) which are necessary to represent 3-Dimensional position and orientation [12]. Visual plot of a robot gives clear insight of the kinematic behaviour of a robot. Given various angle set as input to the developed forward kinematics model (1) and MATLAB toolbox, corresponding results have been compared and plotted.

Considering joint angles configuration $[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5]$ as $[0^\circ, -60^\circ, -30^\circ, -30^\circ, 0^\circ]$, the position and orientation of the end-effector expressed in the base coordinates is:

$$T_6^0 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.9999 & 0 & 0.0000 & 282.1346 \\ 0.9999 & 0 & 0.6250 & 109.7699 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix} \quad (2)$$

Using the command 'fkine' in MATLAB Toolbox for Robotics, the same result has been obtained. The corresponding MATLAB plot for this joint configuration is illustrated in Figure 3.

Considering joint angles configuration $[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5]$ as $[90^\circ, -60^\circ, -30^\circ, -30^\circ, 0^\circ]$, the position and orientation of the end-effector expressed in the base coordinates is:

$$T_6^0 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.9999 & 0 & 0.0000 & 282.1346 \\ 0.9999 & 0 & 0.6250 & 109.7699 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix} \quad (3)$$

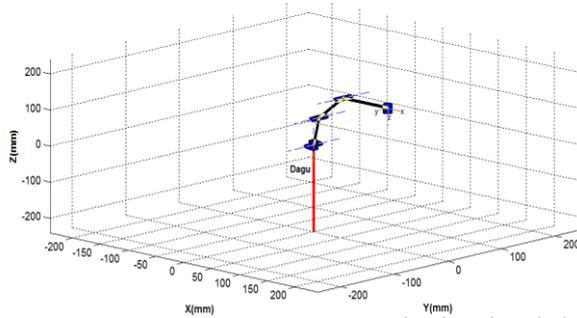


Figure 3. joint angle configuration $[0^\circ, 60^\circ, -30^\circ, -30^\circ, 0^\circ]$.

Using the command 'fkine' in MATLAB Toolbox for Robotics, the same result has been obtained. The corresponding MATLAB plot for this joint configuration is illustrated in Fig. 4.

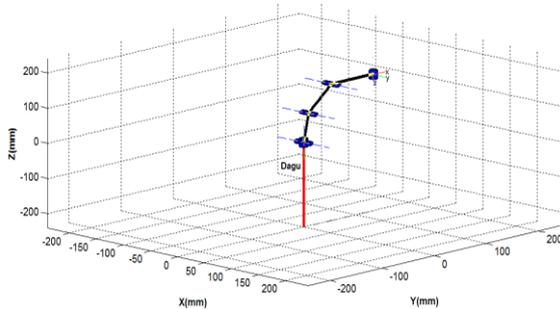


Figure 4. joint angle configuration $[90^\circ, -60^\circ, -30^\circ, -30^\circ, 0^\circ]$.

3 DC Servomotor Modeling

DC servomotor is a common actuator found in many mechanical systems and industrial applications such as industrial and educational robots [13].

Using block diagram open loop system could be represented and the motor has built-in feedback EMF, which tends to reduce the current flow.

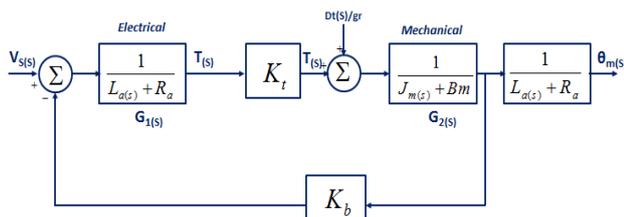


Figure 5. Block diagram of DC servo motor.

The advantage of using the block diagram gives a clear picture of the transfer function relation between each block of the system.

Using SIMULINK, the model of the motor was created. Table 3 shows DC servomotor parameters and values chosen for motor simulation.

Table 3. DC servo Motor Parameters.

Parameters	Values	Unit
Moment of inertia (B_m)	0.00003	kg.m ²
Friction coefficient (J_m)	0.196	N.ms
Back EMF constant (K_b)	0.04	v/ms-1
Torque constant (K_t)	0.7	Nm/A
Electric resistance (R_a)	2	Ohm
Electric inductance (L_a)	0.015	H

The overall transfer function of the system is given below:

$$\frac{\theta_m(s)}{v_i(s)} = \frac{0.7}{0.000000045s^3 + 0.00036s^2 + 0.424s} \quad (4)$$

4 Controller Design

To be able to control a motion process, the precise position of object needs to be measurable and maintained. The designed system should respond to the applied input with a suitable overshoot a settling time and a zero steady state error as possible. To simplify the process of robot and control design and analysis, two suggested MATLAB function block models with its function block parameters for arm robot will be proposed.

4.1 PID Controller

PID controller [14] uses the Error $E_{(t)}$ between the reference input $R_{(t)}$ and the output $U_{(t)}$ as input, and then generates a control signal $U_{(t)}$ for the controlled system. The transfer function of the PID controller has the following form:

$$G_{pid}(s) = K_p + K_i / s + K_d s \quad (5)$$

where K_p , K_i and K_d are the proportional, integral, and derivative gains respectively. Another useful equivalent form of the PID controller is in the form:

$$G_{pid}(s) = K_p (1 + 1 / (T_i s) + T_d s) \quad (6)$$

where $T_i = K_p / K_i$ and $T_d = K_d / K_p$ are known as integral and derivative time constant respectively.

For Tuning PID parameters there are some rules should be considered as:

- If the error $E_{(t)} = R_{(t)} - Y_{(t)}$ is positive high then the proportional gain K_p must be high, integral term K_i low and the derivative term K_d is low. Therefore this will speed the system output.

- If the current error is very small the PID parameters will have to be smaller value for proportional gain, larger value of integral time constant and larger value of derivative gain. So the speed of the system response will be small to reduce the overshoot of the output.

Parameters of PID controller were tuned using a Simulink block instead of conventional tuning methods. Initial values of parameters were defined as and the new parameter values were adjusted in just a few iterations as

shown in Fig. 6. Thus the time and effort for tuning parameters decreased considerably.

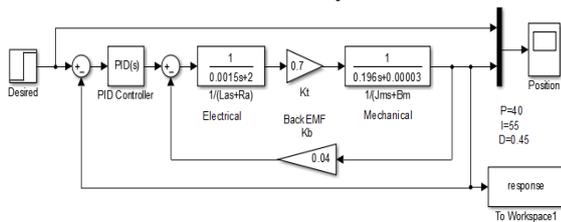


Figure 6. Block Diagram of Servomotor with PID Controller.

4.2 Fuzzy Logic Controller

In the design of fuzzy logic controllers there are four main components: fuzzifier, knowledge base, inference mechanism and Defuzzifier [15]. Fuzzifier converts a crisp input signal into a fuzzified signals identified by membership functions into fuzzy sets. The knowledge base consists of rule base and the data base. The inference mechanism evaluates which control rules are relevant at the current time and then decides what the input to the plant should be. Finally the defuzzification process converts the fuzzy output into crisp controlling signal.

Designing procedure of the fuzzy controller [16] is described as follows:

- Define the input and output of FLC. There are two inputs of FLC, the error $E_{(t)}$ and change of error $\Delta_{(t)}$ and One output is a control signal to the plant.
- Fuzzifying input and output variables. Each variable of fuzzy control inputs has five fuzzy sets ranging from negative high (NH) to positive high (PH).
- Design the inference mechanism rule to find the input-output relation. This paper uses Mamdani (Pro-Max) inference mechanism.
- Defuzzifying the output variable of fuzzy mechanism. Different Defuzzification methods were used and compared to obtain the control signal.

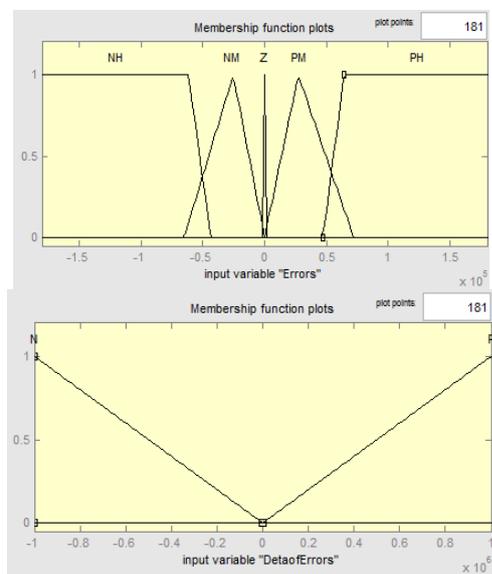


Figure 7. Membership Functions of $E_{(t)}$ and $\Delta E_{(t)}$.

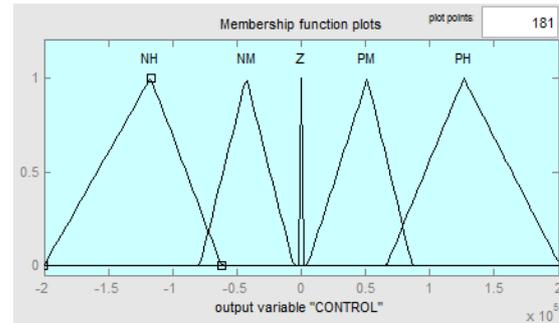


Figure 8. Membership functions of output.

The fuzzy membership functions for the two input parameters are shown in Fig. 7, and the membership function for the output is shown in Fig. 8. For error (E) as input and control as output, here NH means Negative High, NM means Negative Medium, Z means Zero, PM means Positive Medium, and PH means Positive High. For change of errors (Delta DE) as input, here N means Negative and P means Positive.

Table 4. Rule base.

No	Rule Description
1	If E is PH then control is PH
2	If E is NH then control is NH
3	If E is Z and DE is N then control is NM
4	If E is Z and DE is P then control is PM
5	If E is Z then control is Z
6	If E is NM then control is NM
7	If E is PM then control is PM

Rule base and membership functions were designed. Fig. 9 shows the Simulink model for FLC controller.

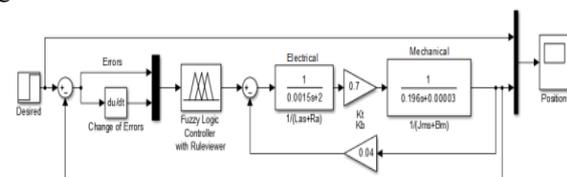


Figure 9. Block diagram of servomotor with fuzzy controller.

4.2.1 Defuzzification strategies

Defuzzification is a process with the aim to produce a non-fuzzy action; it transforms a union of fuzzy sets into a crisp value. The most common methods are as follows:

- 1- Centre of Gravity (COG). 2- Bisector of Area (BOA). 3- Mean of maximum (MOM). 4- Smallest of Maximum (SOM). 5- Largest of Maximum (LOM).

An idea of Centre of Gravity (COG) where the crisp control value u_{COG} is the abscissa of the centre of gravity of the fuzzy set, u_{COG} is calculated as follows:

$$u_{COG} = \frac{\sum_i \mu_c(x_i)x_i}{\sum_i \mu_c(x_i)} \quad (7)$$

where x_i is a point in the universe of the conclusion ($i=1,2,\dots$) and $\mu_c(x_i)$ is the membership value of the resulting conclusion set. For continuous sets summations are replaced by integrals.

The (BOA) defuzzification method calculates the abscissa of the vertical line that divides the area of the resulting membership function into two equal areas.

$$\left| \sum_{i=1}^j \mu_c(x_i) - \sum_{i=j+1}^{i_{\max}} \mu_c(x_i) \right|, i \leq j < i_{\max} \quad (8)$$

Here i_{\max} is the index of the largest abscissa $x_{i_{\max}}$, BOA is considered a computationally complex method.

Another approach to obtain the crisp value is to choose the point with the highest membership. There may be several points in the overall implied fuzzy set which have maximum membership value. Therefore it's a common practice to calculate the mean value of these points. This method is called (MOM) and the crisp value is calculated as follows:

$$u_{MOM} = \frac{\sum_{i \in I} x_i}{|I|}, I = \{i | \mu_c(x_i) = \mu_{\max}\} \quad (9)$$

Here I is the (crisp) set of indices i where $\mu_c(x_i)$ reaches its maximum μ_{\max} , and $|I|$ is its cardinality (the number of members).

One can also choose the leftmost point among the points which have maximum membership to the overall implied fuzzy set. This method is (SOM) defuzzification method. Crisp value is calculated as follows:

$$u_{SOM} = x_{\min(I)} \quad (10)$$

Another possibility is to choose the right most point among the points which have maximum membership to the overall implied fuzzy set. This method is called (LOM) defuzzification method where crisp value is calculated as:

$$u_{LOM} = x_{\max(I)} \quad (11)$$

These strategies proposed in the literature. Four of them will be adopted in this work.

5. Simulation Results

The simulation is carried out with the various Controllers. In order to assess the performance of the controller, PID and Fuzzy Controllers were designed and simulations are presented. In this work the first joint (Base θ_1) sought to move 90° in 8 second for the step time.

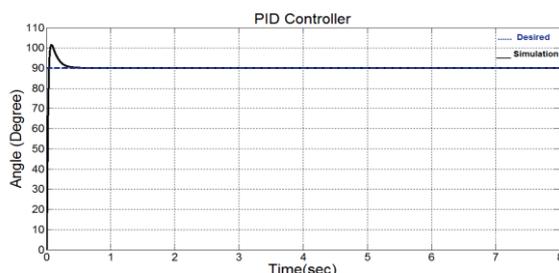


Figure 10. Position response of DC servo motor using PID Controller.

5.1 PID Controller

Using a PID controller, the position time response of the DC servomotor is shown in Fig. 10, where the dashed line refers to a desired signal, the solid line refers to a simulation result.

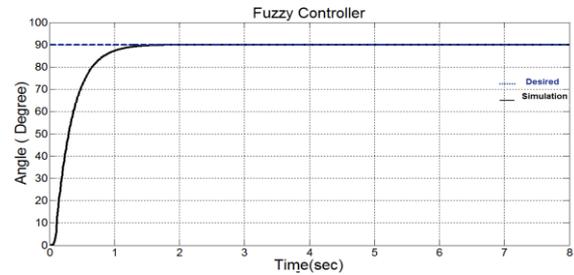


Figure 11. Response of DC servo motor using Fuzzy Controller.

5.2 Fuzzy controller

Using a fuzzy logic controller, the position time response of the DC servomotor is shown in Fig. 11, where the dashed line refers to a desired signal, the solid line refers to a simulation result.

6. Result Comparison

The model has been subjected to reference step inputs signal. Simulation results in Fig. 10.11 show the output response performance of PID and fuzzy Controller respectively. The results show that all Controllers can complete the desired motion of the Manipulator's Servomotor.

The time response parameters, including rise time (t_r), percent overshoot (Mp), settling time (t_s) and steady state error (E_{ss}) of the PID Controller and the fuzzy logic Controller (FLC) for the higher order system transfer function of Manipulator 's servo motor are obtained. PID controller produces faster rise time, settling time, whereas FLC producing lower steady state error with zero overshoot. The comparison between two controllers is summarized in Table (5).

Table 5. Comparison between PID and FLC Controllers.

Controller Type	System Characteristics			
	Rise time (t_r)	Settling time (t_s)	Steady State Errors (E_{ss})	Overshoot ($Mp\%$)
PID	0.04	2	1	10.72
FLC	0.2667	4	0.8	0

Different defuzzification strategies were investigated and results are shown in Table 6.

Table 6. Comparison between various Defuzzification strategies of FLC Controller.

Defuzzification Methods	System Characteristics		
	Rise time (t_r)	Steady State Errors (E_{ss})	Overshoot ($Mp\%$)
Center of Gravity	0.4000	0.8	0
Smallest of Maximum	0.2667	0.86	0
Mean of Maximum	0.2667	0.85	0
Bisector of Area	0.2667	0.86	0

The results obtained using the four Defuzzification strategies have been shown in Table 6. From this table, noticed that Bisector of Area, Mean of Maximum and Smallest of Maximum strategy are giving approximately the same results, whereas for the Centre of Gravity approach there is wide variations in the results that are obtained. Due to complex operations as fuzzification and particularly Defuzzification, implementing simplified Defuzzification strategy leads to optimize the system. That means recommended avoiding Centre of Gravity strategy.

7. Conclusion

One of the most problems with fuzzy controller is that the computing time is longer than for PID. The comparative analysis based on the simulation for Manipulator's servomotor is tabulated which shows the performance index of the two proposed controllers. This work presents design and performance comparison of PID and FLC strategies. Given step input signal to characterize the response, simulation results demonstrate that PID has superior performance in terms of transient parameters. In Steady state response, both PID and FLC manage to converge to the desired output but in terms of overshoot FLC outperformed with zero value. Different defuzzification strategies were employed. Results shown that Bisector of Area, Mean of Maximum and Smallest of Maximum strategies are outperformed as compared with Centre of Gravity.

Acknowledgment

This work was financially supported by the Fundamental Research funds for the Central Universities (YS1403, ZY1521).

References

1. L. W. Tsai, Robot Analysis: The Mechanics of Serial and Parallel Manipulators, New York: Wiley, (1999).
2. M. Nazemizadeh1, M. Taheri and Sh Nazeri. "The application of fuzzy-logic method to control of robots: review study". International Journal of Mechanical Engineering and Robotics Research, Vol. 3, No. 2, April 2014.
3. A. Delibasi, T. Turker, and G. Cansever. Real-Time DC motor position control by fuzzy logic and PID controllers using LabVIEW, IEEE International Conference on Mechatronics and Robotics (MechRob2004), September 13-15 Aachen, Germany.
4. Lin, P.H. Hwang, S. and Chou, J., Comparison on Fuzzy Logic and PID Controls for a DC Motor Position Controller. Conference Record of the 1994 IEEE Industry Applications Society Annual Meeting, 1930-1935, (1994).
5. Azevedo, H.R., Brandao, S.F.M. and Mota Alves, J.B., A Fuzzy Logic Controller for dc Motor Position Control. IEEE 2nd International Workshop on Emerging Technologies and Factory Automation. Design and Operations of Intelligent Factories. Workshop Proceedings, 18-27, (1993).
6. Sreenatha. A. G. Makarand. P., "Fuzzy Logic Controller for Position Control of Flexible Structures," Acta Astronaut journal, Vol. 50, No.11. pp.665-671, (2002).
7. Mishra, M.K., Kothari, A.G., Kothari, D.P. and Ghosh, A., Development of a Fuzzy Logic Controller for Servo Systems. IEEE Region 10 International Conference on Global Connectivity in Energy, Computer, Communication and Control (TENCON '98), 204-207, (1998).
8. Yong and Shih, "Position Control of a Serial Manipulator Using Fuzzy-PID Controllers" International Journal of Automation and Smart Technology, doi: 10.5875/ausmt.v5i1.832, ISSN: 2223-9766.
9. J.J. Craig, (2004). Introduction to robotics: mechanics and control.
10. M.W. Spong, S. Hutchinson and M. Vidyasagar, (2006). Robot modeling and control, New York, John Wiley & Sons.
11. JJ Crage. Introduction to Robotics Mechanics and Control, 3rd Edition, Prentice Hall. (2005).
12. Corke, Peter (2011) Robotics, Vision and Control: Fundamental Algorithms in MATLAB, ISBN 978-3-642-20144-8, Springer Tracts in Advanced Robotics, 73. Springer, Berlin.
13. M. W. Spong, S. Hutchinson, and M. Vidyasagar, Robot Modeling and control: John Wiley & Sons New York, (2006).
14. R.P. Copeland and K.S. Rattan, "A Fuzzy Logic Supervisor for PID Control of Unknown Systems," IEEE International Symposium on Intelligent Control. pp. 22-26, Aug.(1994).
15. G. Feng, "A Survey on Analysis and Design of Model-Based Fuzzy Control Systems," IEEE Trans. on Fuzzy Sys, Vol. 14, No. 5, pp. 676-697, Oct. (2006).
16. 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, March/April (1990).