

Trajectory Tracking Control for Robot Manipulator using Fractional Order-Fuzzy-PID Controller

Reham H. Mohammed
Suez Canal University
Faculty of Engineering
Ismailia, Egypt.

Fahmy Bendary
Benha University
Faculty of Engineering
Shobra, Egypt.

Kamel Elserafi
Port Said University
Faculty of Engineering
Port Said, Egypt.

ABSTRACT

Robotic manipulator is a Multi-Input Multi-Output (MIMO), highly nonlinear and coupled system. Therefore, designing an efficient controller for this system is a challenging task for the control engineers. In this paper, the Fractional Order-Fuzzy-Proportional Integral Derivative (FO-Fuzzy-PID) controller is investigated for the first three joints of robot arm (PUMA 560) for trajectory tracking problem. To study the effectiveness of FO-Fuzzy-PID controller, its performance is compared with other three non model controllers namely Fuzzy-PID, Fractional Order PID (FOPID) and conventional PID. Genetic algorithm (GA) optimization technique was used for tuning parameters of FOPID and conventional PID controllers. Simulation results clearly indicate the superiority of FO-Fuzzy-PID controller over the other controllers for trajectory tracking, better steady state and RMS errors. All controllers were tested by simulation under the same conditions using SIMULINK under MATLAB2013a.

Keywords

PUMA560, Quantico polynomial trajectories planning, Proportional Integral Derivative (PID) controller, Fuzzy-PID, Fractional Order PID (FOPID), Fractional Order-Fuzzy-PID (FO-Fuzzy-PID).

1. INTRODUCTION

Robot manipulator is generally utilized in several applications for example welding, assembling, painting, grinding, mechanical handling and other industrial applications. These applications require path planning, trajectory generation and control design [2]. Because of highly coupled nonlinear and time varying dynamics, the robot motion tracking control is one of the challenging problems. What's more uncertainty in the parameters of both mechanical part of manipulators and the actuating systems would cause more complexity. The control design of robotic manipulators has a long history and offers an open research area to the control engineers due to the advancements in intelligent control based techniques. Several model-based controllers algorithms are utilized in position control such as computed torque method [26], optimal control [27], Variable Structure Control (VSC) [3], Neural Networks (NNs) [24, 25], Fuzzy system [2] and a model based adaptive FOPID [23]. Generally model-based controllers require the presence of an ideal mathematical model for the controlled manipulator and in this manner though to be highly complicated and computationally time consuming, particularly for higher degree of freedom manipulators. Non-model based controllers does not require an essential information of the parameters of either the manipulator or the actuators and hence no mathematical model for the manipulator is needed[24].

Regardless of the new advance in the field of control, PID control sort is still the most broadly used strategies in the industry because of the design simplicity and implementation [16, 17]. In traditional PID control, there are four

shortcomings; error computation; degradation of noise in the derivative control; over simplification, loss of performance in the control law in the form of a linear weighted sum; and complications resulting from the integral control [11]. So as to enhance the robustness and performance of PID control systems, Podlubny has proposed a generalization of the PID controllers, namely, FOPID controllers [22]. Fractional calculus is the field of mathematics that deals with integrals and derivatives using non-integer orders. FOPID control is currently an emerging technology control that its overall performance had been proven to be better than PID in numerous applications. In fractional-order PID ($PI^{\lambda}D^{\mu}$) controller modeling process, the five parameters (k_p , k_d , μ , k_i and λ) need to be selected based on some design specifications, so there is a need for an efficient global approach to optimize these parameters automatically. One of evolutionary optimization techniques is Genetic algorithm (GA) used to optimize the five parameters of the FOPID controller [7].

Fuzzy supervisory provide nonlinear action for the output controller using fuzzy reasoning where the PID gains are tuned based on a fuzzy inference system rather than the traditional approaches, this type of controller is called Fuzzy-PID controller [19]. Effort to merge fuzzy with FOPID control was recently taken place. The adaptive mechanism provided by fuzzy logic can minimize the trade-off between PID parameters tuning and its fractional-order terms when either term can be selected in an adaptive manner.

The aim of this work is to introduce FO-Fuzzy-PID controller to control the position of the first three joint of PUMA 560 robot manipulator in order to obtain fine quintic polynomial trajectory with minimum steady state and RMS errors. The controller should grantee excellent joint space tracking to a given desired trajectory by providing stability, and small tracking errors [12, 14]. At the last the performance of the proposed controller FO-Fuzzy-PID compared with the other three non model controllers namely Fuzzy-PID, fractional-order $PI^{\lambda}D^{\mu}$ and conventional PID for the trajectory tracking task. Moreover, Root Mean Square (RMS) and Steady State Errors (SSE) are discussed to witness the effectiveness of the proposed FO-Fuzzy-PID.

The organization of this paper: The dynamic model of robot manipulator is presented in Section 2. Section 3 introduces the quintic polynomial trajectories planning for the first three joint. Trajectory tracking control of the of robot arm using classical PID, fractional-order $PI^{\lambda}D^{\mu}$, Fuzzy-PID and FO-Fuzzy-PID controllers are introduced in Sections 4, 5, 6 and 7 respectively. Simulation results for all developed controllers are illustrated in Section 8, followed by the concluding remarks in Section 9.

2. DYNAMIC MODEL OF ROBOT MANIPULATOR

Dynamic modeling is vital stage in order to mechanical design, control, and simulation the robot manipulator. It is used to describe dynamic parameters and also to describe the relationship between displacement, velocity and acceleration to torque/ force acting on the joints of the robot manipulator [1]. The joint space dynamic model of a robot manipulator is usually described as in equation 1 [4, 18]:

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) \quad (1)$$

Where, τ is a $n \times 1$ vector of joint torques and /or forces, depending on whether the joint is revolute or prismatic respectively., $M(q)$ is a $n \times n$ symmetric and positive definite inertia matrix, $C(q, \dot{q})\dot{q}$ is a $n \times 1$ vector of centrifugal and Coriolis torques, and $G(q)$ is a $n \times 1$ vector of gravitational torque, q : is a $n \times 1$ vector of joint displacements, \dot{q} : is a $n \times 1$ vector of joint velocities, \ddot{q} : is a $n \times 1$ vector of joint accelerations and n corresponds to the number of degrees of freedom of the robot [4]. The direct dynamic model describes the joint accelerations in terms of the joint positions, velocities and applied torques. It is represented by equation 2:

$$[\ddot{q}]^T = M^{-1}(q). \{ \tau - C(q, \dot{q})\dot{q} - G(q) \} \quad (2)$$

3. TRAJECTORY PLANNING

Actuators working to move the robot arm in certain trajectories based on a pre-programmed routine. A path for the robot arm is a set of positions in joint space and a trajectory is movement over this path in a particular time. Fifth order polynomial or quintic polynomial trajectories approximations are natural choices for providing smoothing, continuous movement where position, velocity and acceleration are given in equations 3, 4 and 5 respectively below [5]:

$$q(t) = q = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \quad (3)$$

$$\dot{q}(t) = v = a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4 \quad (4)$$

$$\ddot{q}(t) = \alpha = 2a_2 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3 \quad (5)$$

This can be written as:

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5 \\ 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4 \\ 0 & 0 & 2 & 6t_0 & 12t_0^2 & 20t_0^3 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 0 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} q_0 \\ v_0 \\ \alpha_0 \\ q_f \\ v_f \\ \alpha_f \end{bmatrix} \quad (6)$$

The robot arm will move from initial position $q(t_0) = 0$ to the final position $q(t_f) = 1$ for joint 1, for joint 2 from initial position $q(t_0) = 0$ to the final position $q(t_f) = 2$, for joint 3 from initial position $q(t_0) = 0$ to the final position $q(t_f) = 3$, initial and final velocities and accelerations = zero. When that happens we see the quintic trajectory curve as shown in Figure 1.

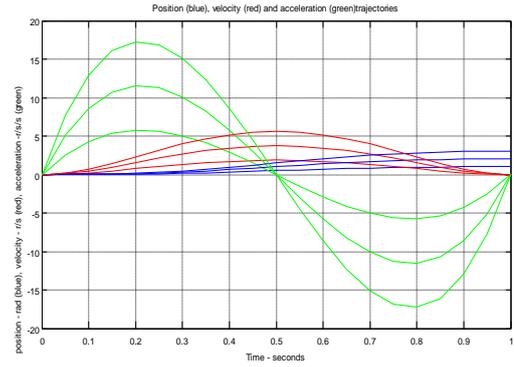


Fig. 1: The corresponding quintic polynomial trajectories for the three joint.

This figure is divided into three parts for each joint to show the relation between the position (blue), velocity (red) and acceleration (green) with time [13]. From trajectory planning generation in this section, the desired values of each joint were obtained, referred to q_d for desired position vector, \dot{q}_d for desired velocity vector and \ddot{q}_d for desired acceleration vector. Since the manipulator like any other machine is affected by internal disturbances and dynamics, the desired joint value and the actual joint value will differ and produce an error. For this reason, development of a controller is required to reduce an error tends to zero.

4. ROBOT ARM TRAJECTORY TRACKING USING PID CONTROLLER

Practically, the block diagram of such a control scheme in the joint space is shown in Figure 2. The control law is given by:

$$\tau = k_p(q_d - q_a) + k_d(\dot{q}_d - \dot{q}_a) + k_i \int (q_d - q_a) \quad (7)$$

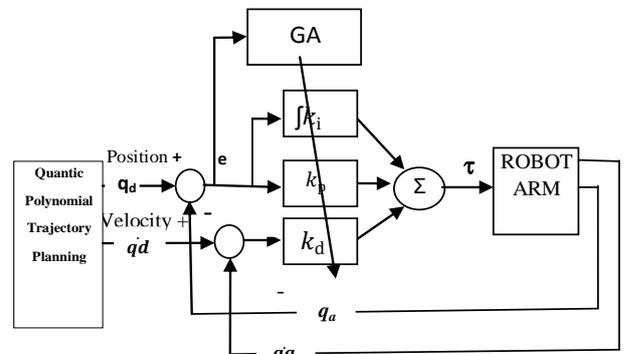


Fig.2. The overall block diagram of the system.

Where $q_d(t)$ and $\dot{q}_d(t)$ denote the desired joint positions and velocities; $q_a(t)$ and $\dot{q}_a(t)$ denote the actual joint positions and velocities; k_p , k_d and k_i represent the proportional, integral, and derivative gains, respectively.

The purpose of PID control is to design a position controller of a robot arm by selecting the PID parameters gains k_p , k_d and k_i using genetic algorithm GA[6]. GA is a useful optimization method to be used in situations involving non-linearities and local minima, consisting essentially of a refined trial-and-error that imitates the evolutionary principle of the survival of the fittest [8]. GA has been applied for tuning the PID position controller gains k_p , k_d and K_i by minimizing the objective function that represents Integral Square-Error (ISE) to ensure optimal control performance at nominal operating conditions, GA

parameter $[k_{p1} k_{i1} k_{d1} k_{p2} k_{i2} k_{d2} k_{p3} k_{i3} k_{d3}]$ are set with lower bounds $= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$ and upper bounds $= [250 \ 250 \ 250 \ 250 \ 250 \ 250 \ 250 \ 250 \ 250]$. The three gains of PID controller after tuning for joint1 ($k_{p1}=77.054, k_{d1}=169.295$ and $k_{i1}=62.318$), for joint2 ($k_{p2}=56.706, k_{d2}=200.787$ and $k_{i2}=45.5$) and for joint3 ($k_{p3}=71.8, k_{d3}=141.088$ and $k_{i3}=39.774$). This control input will force the system to track as close as possible the reference level. As the difference between reference input and instantaneous output reaches zero the system is driven under control. The PID controller demonstrates its breaking points in control because of its weaknesses. So as to upgrade the robustness and performance of PID control systems, Podlubny has proposed a generalization of the PID controllers, specifically, FOPID controllers [22].

5. ROBOT ARM TRAJECTORY TRACKING USING FOPID

5.1 .Principles of FOPID

Fractional-order calculus (FOC) is a generalization of the traditional differentiation and integration that include non integer orders. Essential operator representing the fractional-order integration and differential is given in (8) where α is a real number [21].

$${}_a D_t^\alpha = \begin{cases} \frac{d^\alpha}{dt^\alpha}, & \alpha > 0 \\ 1, & \alpha = 0 \\ \int_a^t (dt)^\alpha, & \alpha < 0 \end{cases} \quad (8)$$

D was a linear operator interpreted as an integrator when α is negative and a differentiator when α positive. Otherwise, D is a unity when α is zero [21].

The most widely recognized sort of a fractional order PID controller is the $PI^\lambda D^\mu$ controller. Including an integrator of order λ and a differentiator of order μ where λ and μ can be any real numbers. The transfer function of such a controller has the structure shown in equation 9 [9]:

$$G_c = \frac{U(s)}{E(s)} = k_p + k_i \frac{1}{s^\lambda} + k_d s^\mu, (\lambda, \mu > 0) \quad (9)$$

Where $G_c(s)$ is the transfer function of the controller, $E(s)$ is the error, and $U(s)$ is controller's output. The control signal $u(t)$ can then be defined in the time domain as:

$$u(t) = k_p e(t) + k_i D_t^{-\lambda} e(t) + k_d D_t^\mu e(t) \quad (10)$$

Figure 3 is a block-diagram arrangement of FOPID. Plainly, selecting $\lambda = 1$ and $\mu = 1$, a classical PID controller can be recuperated. The determinations of $\lambda = 1, \mu = 0$, and $\lambda = 0, \mu = 1$ respectively corresponds traditional PI & PD controllers. All these traditional sorts of PID controllers are the unique instances of the fractional $PI^\lambda D^\mu$ controller given by [9]:

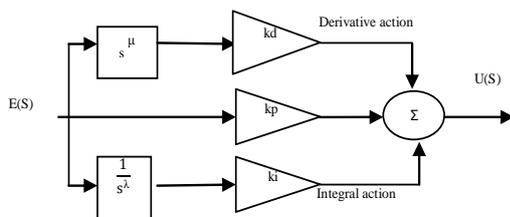


Fig. 3: Block-diagram configuration of FOPID.

It can be expected that the $PI^\lambda D^\mu$ controller may improve the systems performance. One of the most essential points of interest of the $PI^\lambda D^\mu$ controller is the better control of

dynamical frameworks, which are described by fractional order mathematical models. Another favorable position lies in the way that the $PI^\lambda D^\mu$ controllers are provides more flexibility in the controller designing compared with the traditional PID controller; additionally FOPID less sensitive to changes of parameters of a controlled system [9].

5.2 Structure of Robot Arm Based on FOPID

A block diagram of the robot system controlled with the FOPID controllers is shown in Figure4. All of the simulations are performed here using MATLAB 2013b. The block structure of the FOPID controller optimized with GA using Integral Square-Error (ISE) cost function to ensure optimal control performance at nominal operating conditions. Since each FOPID controller has 5 parameters, there are a total of 15 parameters to be optimized with GA. All of the parameters of the FOPID controllers are updated at every simulation time, where GA parameters $[k_{p1} k_{i1} k_{d1} \lambda \mu k_{p2} k_{i2} \lambda \mu k_{d2} k_{p3} k_{i3} k_{d3} \lambda \mu]$ with lower bounds $= [0 \ 0 \ 0 \ 0.01 \ 0.01 \ 0 \ 0 \ 0 \ 0.01 \ 0.01 \ 0 \ 0 \ 0 \ 0.01 \ 0.01]$ and upper bounds $= [400 \ 200 \ 200 \ 2 \ 2 \ 200 \ 200 \ 200 \ 2 \ 2 \ 200 \ 200 \ 2 \ 2]$.

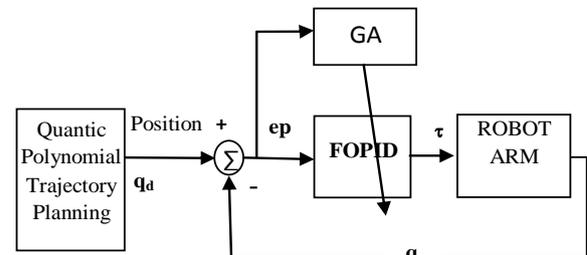


Fig.4. The overall block diagram of the system.

The five gains of FOPID controller after tuning for joint1 ($k_{p1}=385, k_{d1}=43.854, k_{i1}=22.378, \lambda_1=0.05$ and $\mu_1=2$), for joint2 are ($k_{p2}=22.729, k_{d2}=24.831, k_{i2}=8.155, \lambda_2=0.133$ and $\mu_2=1.984$) and for joint3 are ($k_{p3}=16.24, k_{d3}=6.586, k_{i3}=4.35, \lambda_3=0.81$ and $\mu_3=1.712$).

6. ROBOT ARM TRAJECTORY TRACKING USING FUZZY-PID

The fuzzy logic programming has been become broadly utilized in industry. Broad number of researches were developed using fuzzy logic technique [20]. Fuzzy-PID controllers are arranged into two sorts: the direct action fuzzy control and the fuzzy supervisory control. The direct action sort replaces the PID control with a feedback control loop to compute the action through fuzzy reasoning where the control actions are resolved directly by means of a fuzzy inference. These sorts of fuzzy controllers are also called PID-like controllers. On the other hand, the fuzzy supervisory type attempts to give nonlinear action for the controller output utilizing fuzzy reasoning where the PID gains are tuned based on a fuzzy inference system rather than the traditional methodologies. The design process of the fuzzy controller is described as follows [19]:

- Define the input and output variables of FLC. In this work, there are two inputs of FLC, the error $e(t)$ and its rate of change of error $\dot{e}(t)$ and three outputs K_P, K_I and K_d are respectively as shown in Figure 5.

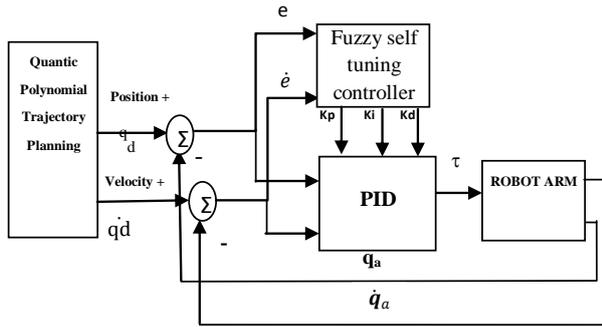


Fig. 5: Block diagram of a fuzzy-PID controller.

- Fuzzify the input and output variables by defining the fuzzy sets and membership functions. Every variable of fuzzy control inputs has seven fuzzy sets running from negative big (NB) to positive big (PB) as shown in Figure 6 for the two inputs e and \dot{e} , and the output of FLC has the following membership function as shown in Figure 7 for the three outputs K_p , K_i , and K_d .
- Outline the inference mechanism rule to get the input-output relation. This work utilized Mamdani (max-min) inference mechanism where, Tables (1), (2), and (3) show the control rules that utilized for fuzzy self tuning of PID controller.
- Defuzzify the output variable. Here, the center of gravity (COG) method, the most frequently utilized method, is utilized. The control activity is[20]:

$$COG = \frac{\sum_{i=1}^m \mu(f_i) \cdot f_i}{\sum_{i=1}^m \mu(f_i)} \quad (11)$$

Now the control activity of the PID controller after self tuning can be depicted as:

$$U_{PID} = K_{p2} * e(t) + K_{i2} \int edt + K_{d2} \frac{de(t)}{dt} \quad (12)$$

Where K_{p2} , K_{i2} , and K_{d2} are the new gains of PID controller and are equivalents to: $K_{p2}=K_{p1} * K_p$, $K_{i2}=K_{i1} * K_i$, and $K_{d2}=K_{d1} * K_d$. Where, K_{p1} , K_{i1} and K_{d1} are the output gains of fuzzy control that are differing online with the output of the system under control. And K_p , K_i , and K_d are the initial values gains of the traditional PID controller.

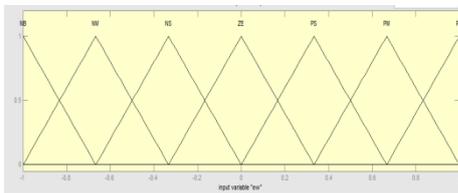


Fig. 6: Memberships function of inputs (e) and (\dot{e}).

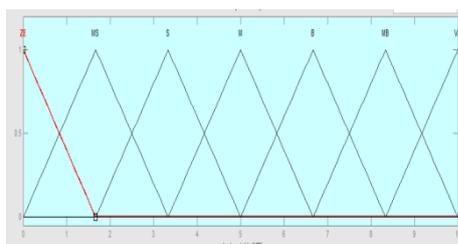


Fig.7: Memberships function of outputs (K_p , K_i , and K_d).

Table 1: Rule bases for determining the gain K_p

\dot{e} / e	NB	NS	ZE	PS	PB
NB	M	M	M	M	M
NS	S	S	S	S	S
ZE	MS	MS	ZE	MS	MS
PS	S	S	S	S	S
PB	M	M	M	M	M

Table 2: Rule bases for determining the gain K_i .

\dot{e} / e	NB	NS	ZE	PS	PB
NB	VB	VB	VB	VB	VB
NS	B	B	B	MB	VB
ZE	ZE	ZE	MS	S	S
PS	B	B	B	MB	VB
PB	VB	VB	VB	VB	VB

Table 3: Rule bases for determining the gain K_d .

\dot{e} / e	NB	NS	ZE	PS	PB
NB	ZE	S	M	MB	VB
NS	S	B	MB	VB	VB
ZE	M	MB	MB	VB	VB
PS	B	VB	VB	VB	VB
PB	VB	VB	VB	VB	VB

7. ROBOT ARM TRAJECTORY TRACKING USING FO-FUZZY-PID

The proposed FO-Fuzzy-PID uses a two dimensional linear rule base for the error, fractional rate of variety of error and the FLC output with standard triangular membership functions and Mamdani type inference as the same with Fuzzy-PID. But in the FO-Fuzzy-PID the integer order rate of the error at the input to the FLC is replaced by its fractional order counterpart (μ) with membership function. Additionally the order of the integral is replaced by a fractional order (λ) with membership function at the output of the FLC. A block diagram of the robot system controlled with the FO-Fuzzy-PID controllers is shown in Figure 8.

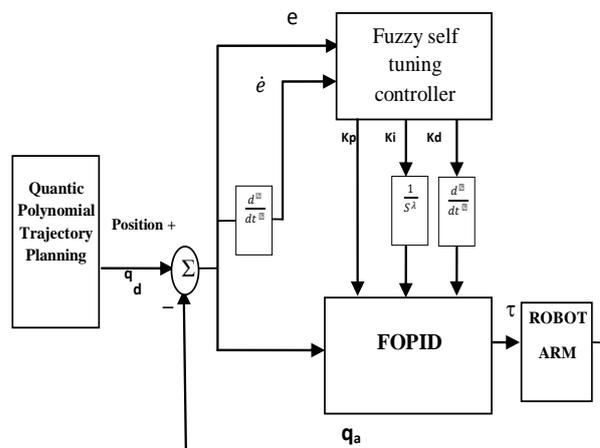


Fig. 8: Structure of the FO-Fuzzy-PID controller.

8. SIMULATION RESULTS.

The simulation has been performed for the first three degrees of freedom of PUMA 560 using MATLAB 2013b by considering the PUMA 560 robot manipulator dynamics from [4, 10]. All information about inertial and gravitational constants are given in Appendix [3] based on the studies carried out by Armstrong and Corke[10]. This simulation is implemented for showing the efficiency of the suggested FO-Fuzzy-PID position controller compared with other three non model controllers namely Fuzzy-PID, Fractional Order PID (FOPID) and conventional PID where, all controllers tested to quintic polynomial trajectories. Starting from random initialized parameters, GA progressively minimizes different integral performance indices iteratively while finding optimal set of parameters for the FOPID and PID controller. The desired and actual position for the first 3 joints of PUMA 560 robot arm controlled using PID controller tuned based on genetic algorithm are shown in Figures 9, 10 and 11 respectively where, GA reaches to the values of the nine PID parameters after 700 epochs with fitness value 0.00207247. The desired and actual position for joints 1, 2 and 3 of puma 560 robot arm controlled using FOPID controller are given in Figures 12, 13 and 14 respectively where, GA reaches to the values of the 15 FOPID parameters after 450 epochs with fitness 4.600648×10^{-5} .

The fuzzy supervisory controller tries to vary the PID parameters during process operation to enhance the system response and eliminate the disturbances. This technique can be used to minimize energy consumption in distributed environmental control systems while maintain a high occupant comfort level where, the desired and actual position for the first 3 joints of puma 560 robot arm controlled using fuzzy-PID controllers with respect to quintic polynomial trajectory planning are shown in Figures. 15, 16 and 17. With results obtained from simulation for desired and actual position as shown in Figures 18,19 and 20 for the same joints controlled using the proposed FO-Fuzzy-PID controller, it is clear that for the same operation condition the position control using FO-Fuzzy-PID controller technique had better performance than the others controllers.

Table.4 shows a comparison between all types of controllers (PID tuned using GA, FOPID tuned using GA, Fuzzy-PID and Fuzzy-FOPID) implemented to control the position angle of the first three joints of puma 560 robot arm.

Table.4: Comparison results of PID, FOPID, Fuzzy-PID and FO-Fuzzy-PID

Controller type	RMS error	S.S. error for joint 1 position	S.S. error for joint 2 position	S.S. error for joint 3 position
PID	0.04676	-0.013	-0.076	-0.008
FOPID	0.001799	0.0007	-0.004	0.005
Fuzzy-PID	0.02278	0.003	0.007	0.005
FO-Fuzzy-PID	0.0008419	0.001	0.001	-0.003

From Table.4 the position control using FO-Fuzzy-PID has better steady state and RMS errors than classical PID, FOPID tuned using GA and Fuzzy-PID. By comparing SSE and RMS error in a system it was found that the FO-Fuzzy-PID's

errors (SSE for joint1= 0.001, joint2=0.001, joint3=-0.003 and RMS error=0.0008419) are less than the other controllers where PID's errors (SSE for joint1=- 0.013, joint2=-0.076, joint3=-0.008 and RMS error=0.04676), FOPID's errors (SSE for joint1= 0.0007, joint2=0.004, joint3=0.005 and RMS error=0.001799) and Fuzzy-PID's errors (SSE for joint1= 0.003, joint2=0.007, joint3=0.005 and RMS error=0.02278). Figs. 21, 22 and 23 give complete comparisons between the all controllers for joint 1, 2 and 3 errors respectively. From this comparison it was observed that the errors of the first three joints converge to zero after the robot is controlled using FO-Fuzzy-PID. These results show that FO-Fuzzy-PID controller has better and fast response and small errors for quintic polynomial trajectory control of robot arm compared to the other controllers.

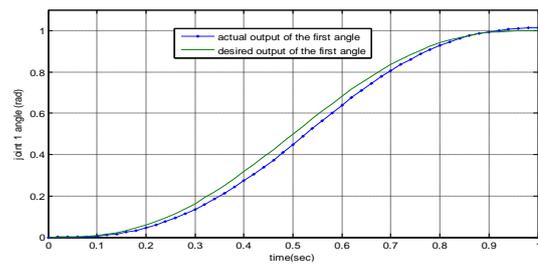


Fig. 9: Desired and actual position for joints 1 controlled using PID controller.

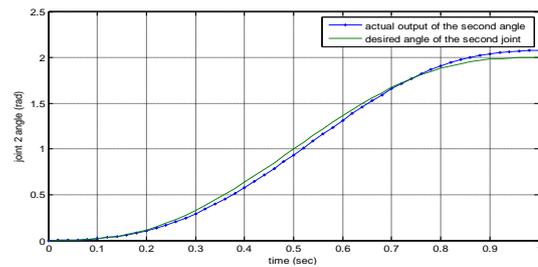


Fig. 10: Desired and actual position for joints 2 controlled using PID controller

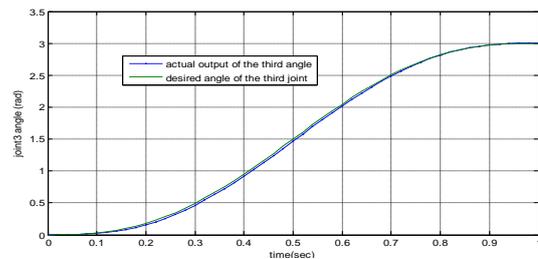


Fig. 11: Desired and actual position for joints 3 controlled using PID controller

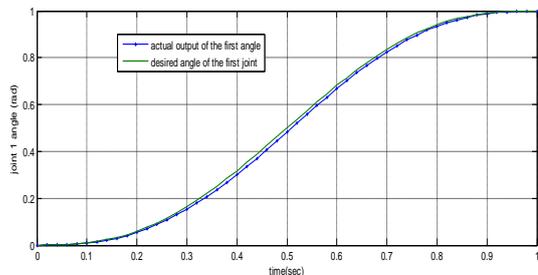


Fig. 12: Desired and actual position for joints 1 controlled using FOPID controller

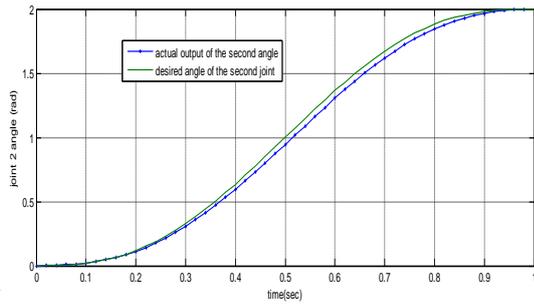


Fig. 13: Desired and actual position for joints 2 controlled using FOPID controller.

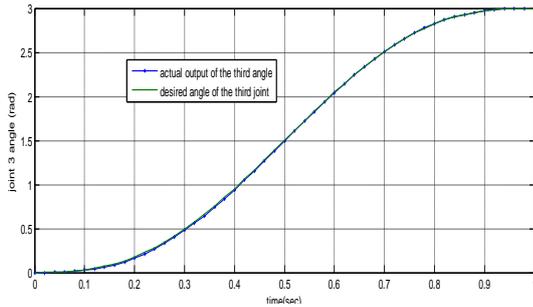


Fig. 14: Desired and actual position for joints 3 controlled using FOPID controller.

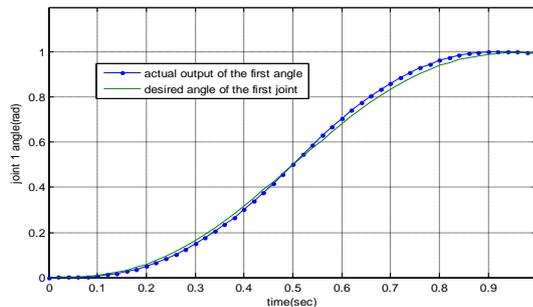


Fig. 15: Desired and actual position for joints 1 controlled using Fuzzy-PID controller.

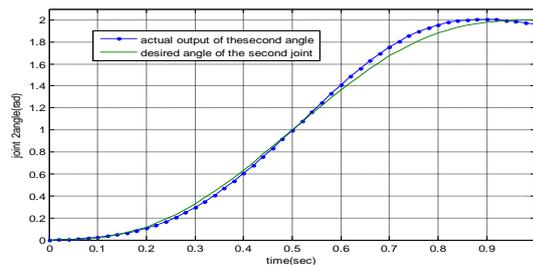


Fig. 16: Desired and actual position for joints 2 controlled using Fuzzy-PID controller.

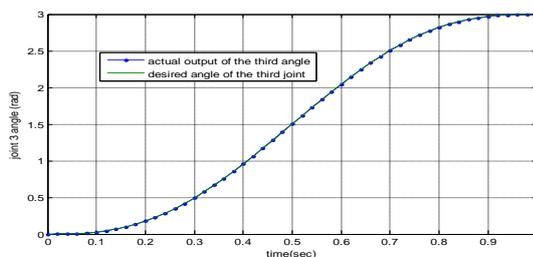


Fig. 17: Desired and actual position for joints 3 controlled using Fuzzy-PID controller.

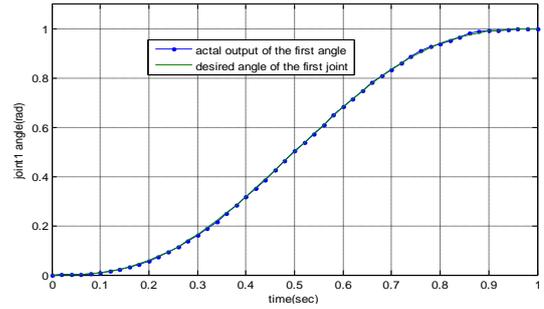


Fig. 18: Desired and actual position for joints 1 controlled using FO-FUZZY-PID controller.

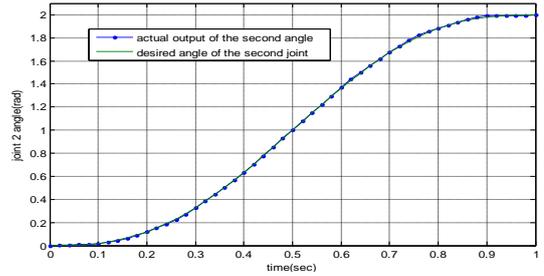


Fig. 19: Desired and actual position for joints 2 controlled using FO-FUZZY-PID controller.

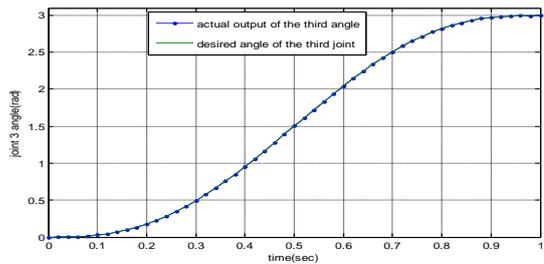


Fig. 20: Desired and actual position for joints 3 controlled using FO-FUZZY-PID controller.

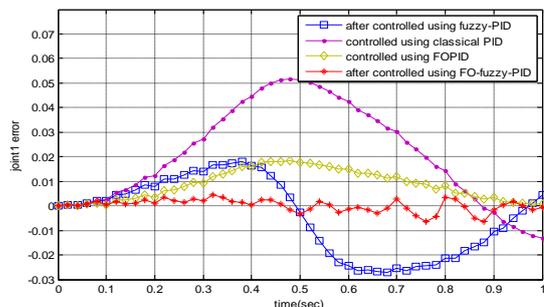


Fig. 21: Comparison between joint1 errors after controlled using PID, FOPID, Fuzzy-PID and FO-Fuzzy-PID controller.

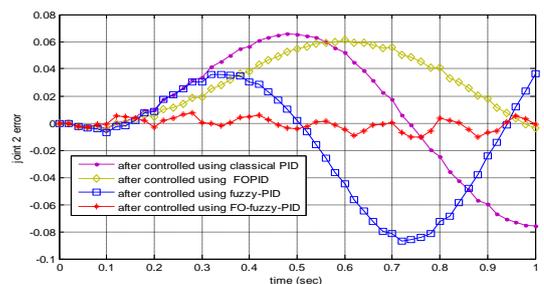


Fig. 22: Comparison between joint2 errors after controlled using PID, FOPID, Fuzzy-PID and FO-Fuzzy-PID controllers.

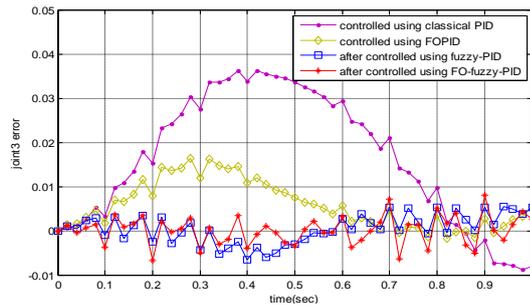


Fig. 23: Comparison between joint3 errors after controlled using PID, FOPID, Fuzzy-PID and FO-Fuzzy-PID controllers.

9. CONCLUSION

In this study, FO-FUZZY-PID controller has been applied to control the position of the first three joints of the PUMA 560 robot arm in order to obtain fine quintic polynomial trajectory with minimum error. Results have been compared with other three non model controllers namely Fuzzy PID (FPID), Fractional Order PID (FOPID) and conventional PID.

From the simulation results it was concluded that:

- The position control of the first three joints controlled using FO-FUZZY-PID has better steady state and RMS errors than controlled with other three non model controllers namely Fuzzy PID, FOPID and PID tuned by GA.
- FOPID converges with a smaller number of iteration and minimum fitness value compared with PID.
- The system responses have showed that the FO-FUZZY-PID controller has much faster response than FOPID, Fuzzy PID and PID tuned by GA.

10. REFERENCES

- [1] S. Yadegar an Azura binti Che Soh , "Design Stable Robust Intelligent Nonlinear Controller for 6- DOF Serial Links Robot Manipulator", International Journal of Intelligent Systems and Applications (IJISA), MECS, July 2014, pp.19-38.
- [2] Ch. R. Kumar, K. R. Sudha, D. V. Pushpalatha and Ch. V. N. Raja , "Fuzzy C-Means Controller for a PUMA-560 Robot Manipulator", IEEE Workshop on Computational Intelligence: Theories, Applications and Future Directions, IIT Kanpur, India, July 2013, pp. 57-62 .
- [3] F. Piltan, S. Emamzadeh, Z. Hivand, F. Shahriyari and M. Mirzaei," PUMA-560 Robot Manipulator Position Sliding Mode Control Methods Using Matlab / Simulink and Their Integration into Graduate/Undergraduate Nonlinear Control, Robotics and MATLAB Courses", International Journal of Robotic and Automation, Volume (6) : Issue (3) , 2012, pp.167-191.
- [4] Siciliano, Bruno, and Oussama Khatib, Springer handbook of robotics: Springer Science & Business Media, 2008.
- [5] D. Breen, D. Kennedy and E. Coyle, "Thermal Robotic Arm Controlled Spraying via Robotic Arm and Vision System", PhD Thesis, School of Electrical Engineering Systems Dublin Institute of Technology Ireland, January 2010.
- [6] M. D. Youns, S. M. Attya and A. I. Abdulla, "Position Control of Robot Arm Using Genetic Algorithm Based PID Controller", Al-Rafidain Engineering, Mosul, Iraq, Vol.21, No. 6, December 2013, pp. 19-30.
- [7] Z. Bingul and O. Karahan," Fractional PID controllers tuned by evolutionary algorithms for robot trajectory control", Turkish Journal of Electrical Engineering & Computer Sciences, Vol.20, No.Sup.1, 2012, pp. 1123-1136.
- [8] D. Valerio and J. Sa da Costa," Optimisation of non-integer order control parameters for a robotic arm", In 11th International Conference on Advanced Robotics, Coimbra, 2003.
- [9] M. R. Faieghi and A. Nemati, "On Fractional-Order PID Design, Applications of MATLAB in Science and Engineering, ISBN: 978-953-307-708-6, InTech, 2011, Available from: <http://www.intechopen.com/books/applications-of-matlab-in-science-and-engineering/on-fractional-order-pid-design>.
- [10] P.I. Corke and B. Armstrong-Helouvy, "A search for consensus among model parameters reported for the PUMA 560 robot," Proceedings of the 1994 IEEE International Conference on Robotics and Automation, Vol. 2, 1994, pp. 1608-1613.
- [11] J. Han, "From PID to active disturbance rejection control," IEEE Trans. Ind. Electron., vol. 56, no. 3, Mar. 2009, pp. 900–906.
- [12] J. J. D'Azzo, C. H. Houpis and S. N. Sheldon, "Linear control system analysis and design with MATLAB", CRC, 2003.
- [13] M. R. Abu Qassem, H. Elaydi and I. Abuhadrous, " Simulation and Interfacing of 5 DOF Educational Robot Arm", MS.c, Islamic University of Gaza ,Electrical Engineering Department June 2010.
- [14] Z. Alassar, H. Elaydi and I. Abuhadrous "Modeling and Control of 5DOF Robot Arm Using Supervisory Control", MS.c, Islamic University of Gaza , Electrical Engineering Department, March 2010.
- [15] Potkonjak V., "Thermal criterion for the selection of DC drives for industrial robots", *Proc. 16th Int. Symp. On Industrial Robots*, Brussels, Belgium, September-Oct.1986, p. 129-140.
- [16] S. E. Hamamci, "An algorithm for stabilization of fractional-order time delay systems using fractional-order PID controllers," *IEEE Trans. Autom. Control*, vol. 52, no. 10, Oct. 2007, pp. 1964–1969.
- [17] M. Ö. Efe, "Neural network assisted computationally simple PI^λD^μ control of a quadrotor UAV", *IEEE Trans. Ind. Informat.*, vol. 7, no. 2, May 2011, pp. 354–361.
- [18] Chedmail P., Gautier M. "Optimum choice of robot actuators", *Trans. of ASME, J. of Engineering for Industry*, Vol. 112 , No. 4, 1990, p. 361-367.
- [19] G.U.V.Ravi Kumar and Mr.Ch.V.N.Raja," Control of 5 DOF Robot Arm using Fuzzy Supervisory Control", International Journal of Emerging Trends in Engineering and Development Vol.2, No. 4, ,March 2014, Available

online on http://www.rpublication.com/ijeted/ijeted_index.htm ISSN 2249-6149, pp. 294-306.

- [20] T. pattaradej, G. Chen and P. Sooraksa, "Design and Implementation of Fuzzy PID Control of a bicycle robot", Integrated computer-aided engineering, Vol.9, No.4, 2002.
- [21] D. Xue and Y. Chen, "A Comparative Introduction of Four Fractional Order Controllers," in Proceedings of the 4th World Congress on Intelligent Control and Automation, 2002, pp. 3228–3235.
- [22] Podlubny, "Fractional-order systems and $PI^{\lambda}D^{\mu}$ controller," IEEE Trans. Automatic Control, vol. 44, no. 1, pp. 208–214, 1999.
- [23] H. Delavari, R. Ghaderi, A. Ranjbar N., S.H. HosseinNia and S. Momani, "Adaptive Fractional PID Controller for Robot Manipulator", The 4th IFAC Workshop Fractional Differentiation and its Applications. Badajoz, Spain, October 18-20, 2010, pp. 1-7.
- [24] Mahmoud M. Al Ashi, H. Elaydi and I. Abu Hadrous, "Trajectory Tracking Control of A 2-DOF Robot Arm Using Neural Networks", MS.c, Islamic University of Gaza, Electrical Engineering Department, Feb. 2014.
- [25] K. A. EL Serafi, K. Z. Moustafa, and A. A. Sallam "An Adaptive Neuro Controller for Robotic Manipulator", IEEE Int. Conf. on Electronics, Circuits, and Systems, Cairo, December 15-18, 1997.
- [26] Spong, Mark W. "Vidyasagar, Robot Dynamics and Control." John and Wiley and Sons , 1989.
- [27] Green,A.,and Sasiadek, J.Z. , "Dynamics and trajectory tracking control of a two-link robot manipulator" , J. Vibr. Control10, Vol.10, 2004, PP.1415–1440.

11. APPENDIX

Table 5: Inertial constant reference (Kg.m²)

$I_1=1.43\pm 0.05$	$I_2=1.75\pm 0.07$
$I_3=1.38\pm 0.05$	$I_4=0.69\pm 0.02$
$I_5=0.372\pm 0.031$	$I_6=0.333\pm 0.016$
$I_7=0.298\pm 0.029$	$I_8=-0.134\pm 0.014$
$I_9=0.0238\pm 0.012$	$I_{10}=-0.0213\pm 0.0022$
$I_{11}=-0.0142\pm 0.0070$	$I_{12}=-0.011\pm 0.0011$
$I_{13}=-0.00379\pm 0.0009$	$I_{14}=0.00164\pm 0.00007$ 0
$I_{15}=0.00125\pm 0.0003$	$I_{16}=0.00124\pm 0.0003$
$I_{17}=0.0006642\pm 0.0003$	$I_{18}=0.000431\pm 0.0001$ 3
$I_{19}=0.0003\pm 0.0014$	$I_{20}=-0.000202\pm 0.0008$
$I_{21}=-0.0001\pm 0.0006$	$I_{22}=-0.000058\pm 0.000015$
$I_{23}=0.00004\pm 0.00002$	$I_{m1}=1.14\pm 0.27$
$I_{m2}=4.71\pm 0.54$	$I_{m3}=0.827\pm 0.093$
$I_{m4}=0.2\pm 0.016$	$I_{m5}=0.179\pm 0.014$
$I_{m6}=0.193\pm 0.016$	

Table 6: Gravitational constant (N.m)

$g_1=-37.2\pm 0.5$	$g_2=-8.44\pm 0.20$
$g_3=1.02\pm 0.50$	$g_4=0.249\pm 0.025$
$g_5=-0.0282\pm 0.0056$	