Mansoura Engineering Journal

Volume 40 | Issue 4 Article 1

7-9-2020

Comparative Study Methods of Trajectory Tracking Control for Robot Manipulator.

R. Hassan
Assistant lecturer, Electrical Engineering Department, Faculty of Engineering, Suez Canal University, Ismailia, Egypt, riry4mody@yahoo.com

F. Bendary
Professor, Electrical Engineering Department, Faculty of Engineering, Benha University, Shobra , Egypt, fahmybendary10@gmail.com

K. Elserafi
Associate Professor, Electrical Engineering Department, Faculty of Engineering, Port Said University, Port Said, Egypt, relserafi@hotmail.com

A. Ghanem
Doctorate Engineer, Electrical Engineering Department, Faculty of Engineering, Suez Canal University, Shobra, Ismailia, Egypt, amghanem@gmail.com

M. Soliman
Doctorate Engineer, Electrical Engineering Department, Faculty of Engineering, Benha University, Shobra, Egypt, msoliman_28@yahoo.com

Recommended Citation

Hassan, R.; Bendary, F.; Elserafi, K.; Ghanem, A.; and Soliman, M. (2020) "Comparative Study Methods of Trajectory Tracking Control for Robot Manipulator.," *Mansoura Engineering Journal*: Vol. 40 : Iss. 4 , Article 1.
Available at: https://doi.org/10.21608/bfemu.2020.101868

Comparative Study Methods of Trajectory Tracking Control for Robot Manipulator

دراسة مقارنة لطرق التحكم في تتبع مسار روبوت مناول

Hassan, R.[1], Bendary, F.[2],Elserafi, K. [3],Ghanem,A.[4] and Soliman, M. [5]

[1] Assistant lecturer, Electrical Engineering Department, Faculty of Engineering, Suez Canal University, Ismailia, Egypt, E-mail:riry4mody@yahoo.com

[2] Professor, Electrical Engineering Department, Faculty of Engineering, Benha University, Shobra , Egypt, E-mail:fahmybendary10@gmail.com

[3] Associate Professor, Electrical Engineering Department, Faculty of Engineering, Port Said University, Port Said, Egypt, E-mail:relserafi@hotmail.com

[4] Doctorate Engineer, Electrical Engineering Department, Faculty of Engineering, Suez Canal University, Shobra, Ismailia, Egypt, E-mail: amghanem@gmail.com

[5] Doctorate Engineer, Electrical Engineering Department, Faculty of Engineering, Benha University, Shobra, Egypt, E-mail: msoliman_28@yahoo.com

الخلاصة

فى هذا البحث تمت دراسة مقارنة لثلاثة من الحاكمات المستخدمة فى التحكم فى الروبوت، بهدف تتبع مسار معين.
رغم أن الحاكم الذى يعتبر الموديل الرياضى للروبوت جزء منه للتخلص من الاخطية و الارتباط الشديد بين المتغيرات،يعتبر مثاليا الا أنة يفقد هذه المثالية في حالة عدم التعرف علي قيم البرامترات سواء الهندسية أو الديناميكية. لذا يعتبر الحاكم من النوع PID الأبسط و الأكثر استخداما فى الربوتات الصناعية. لذا يتم فى هذا البحث مقارنة الأداء لثلاث حاكمات من هذا النوع، الأول هو PID المعتمد على المنطق الغيمى، الثانى عبارة عن PD تم توليف ثوابت التغذية العكسية باستخدام الخوارزميات الجينية، أما الثالث فهو توافقى PID يعتمد على التكامل بين الشبكات العصبية و المنطق الغيمى. ولاتمام المقارنة تم تطبيق كل من هذة الحاكمات على الثلاث مفاصل الأولى لروبوت صناعى شائع الاستخدام في الصناعة (**PUMA560**) بهدف تتبع مسار معين. دراسة المقارنة تمت من خلال المحاكاة باستخدام الماتلاب وقد أظهرت الدراسة تفوق الحاكم الأول PID المعتمد على المنطق الغيمى علي ألآخرين من حيث قيمة الخطأ أثناء التتبع وعند الاستقرار.

Abstract

In this article a three non-model based trajectory tracking controllers for rigid simple open chain robot manipulator are investigated. The first is a Fuzzy-PID controller, it is considered as a reference benchmark to compare its results with the others which are a proportional Derivative (PD) tuned using genetic algorithm (GA) and an Adaptive Neuro Fuzzy Inference System (ANFIS). The simulation is carried out for the first three joints of robot arm (PUMA560) aiming to track aquintic polynomial trajectory with minimum errors, and good disturbance rejection. Simulation results, shows that using Fuzzy-PID has better steady state error and RMS error than the ANFIS and PD tuned using GA. The three controllers are tested by simulated under the same conditions using SIMULINK under MATLAB2013a.

Keywords

PUMA560**,** Trajectories planning, Proportional Derivative (PD) controller, Adaptive Neuro Fuzzy Inference System (ANFIS), Fuzzy-PID controller**.**

Received: 10 June, 2015 – Accepted: 15 October, 2015

1. Introduction

Robotics is a special engineering science which deals with designing, modelling, controlling and robot's utilization [2].The system that we are working on is PUMA 560 which has 6 degrees of freedom (6 DOF) and it's joints are revolute[15]. These arms are widely used in applications like welding, assembling, painting, grinding, mechanical handling and other industrial applications. These applications may require path planning, trajectory generation and control design [2]. Due to highly coupled nonlinear and time varying dynamic, the robot motion tracking control is one of the challenging problems. In addition uncertainty in the parameters of both mechanical part of manipulators and the actuating systems would cause more complexity [11]. Many modelled based controllers algorithm such as computed torque method [20], Variable Structure Control (VSC) [3], Neural Networks (NNs) [21], Fuzzy system [2]. Generally model-based controllers required the presence of an ideal mathematical model for the controlled manipulator and therefore considered to be highly complicated and computationally time consuming, especially for higher degree of freedom manipulators. Non-model based controllers did not require a prerequisite knowledge of the parameters of either the manipulator or the actuators and hence no mathematical model for the manipulator was needed [19].

The main objective is concerned with designing a controller for the motion of the robot manipulator to meet the requirement of the desired quintic polynomial trajectory input with stability, good disturbance rejection, and small tracking error[12],[14].

Various joint space controllers have been designed and applied feedback controller that allows the actual motion $q_a(t)$ tracking of the desired motion $q_d(t)$[3].

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 PD control tuned using GA , but in classical PD controller, there exist four weaknesses such as error computation; noise degradation in the derivative loop; oversimplification and the loss of performance in the control law in the form of a linear weighted sum; and complications brought by the integral control[6]. To overcome these problems and creates more appropriate solution to trajectory tracking control of the robot manipulator, artificial intelligent controllers have been proposed such as Fuzzy-PID and a hybrid combined between Fuzzy Inference Systems (FIS) and Neural network controllers to design ANFIS.

The organization of the rest of this paper can be summarized as follows. Dynamic model of robot manipulator is presented in Section 2. Section 3 introduces quintic polynomial trajectories planning for the 3 joints. Position controller strategies of the robot arm using classical PD tuned using GA; ANFIS and Fuzzy-PID controllers are summarized in Sections 4, 5 and 6 respectively. Simulation results for all cases are illustrated in Section 7, followed by the concluding remarks in Section 8.

2. Dynamic model of robot manipulator

Dynamic modelling is vital for control, mechanical design, and simulation. It is used to describe dynamic parameters and also to describe the relationship between displacement, velocity and acceleration to torque force acting on robot manipulator joints [1]. The joint space dynamic model of a robot manipulator is usually described by the following matrix equation 1 [4], [10]:

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

Where, τ is a n × 1 vector of joint torques and/or forces, depending on whether the joint is revolute or prismatic respectively ,M(q) is a n × n symmetric and positive define inertia matrix, $C(q,\dot{q})\dot{q}$ is a n ×1vector of centrifugal and Coriolis torques, and G (θ) is a n ×1 vector of gravitational torque, q: is a n × 1 vector of joint displacements, $\dot{q}$: is a n × 1 vector of joint velocities, $\ddot{q}$: is a n × 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:

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

3. Trajectory planning

Actuators must move the robot arm in particular 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 profile. Quintic polynomial trajectories or fifth order polynomial approximations are natural choices for providing smoothing, continuous motion where position, velocity and acceleration are given in equations3, 4and 5respectively below [5]:

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

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

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

This can be written as:

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

The robot arm will be moved 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 , final velocities and accelerations = zero. When that happens we see the quintic trajectory curve as shown in Fig. 1. 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].

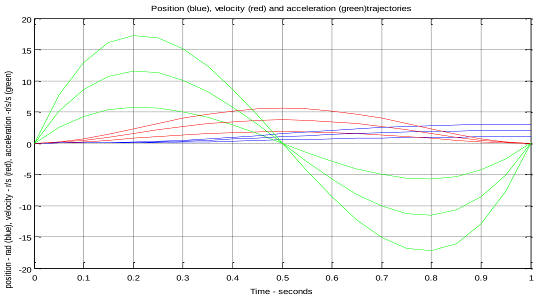


Fig. 1: The corresponding quintic polynomial trajectories for the three joint.

From trajectory planning generation in this section, the desired values of each joint were obtained, referred to as 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, controller is needed to reduce an error tends to zero.

4. Robot arm trajectory tracking using PD controller

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

$$\tau = K_P(q_d - q_a) + k_d(\dot{q}_d - \dot{q}_a) \qquad (7)$$

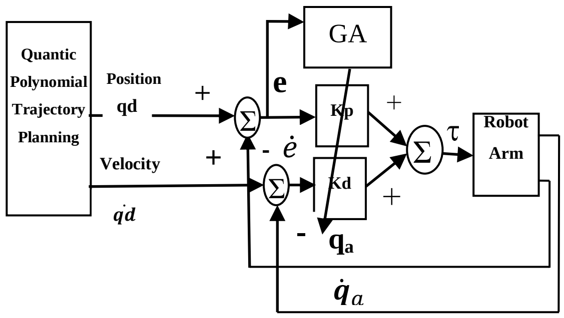


Fig.2: The overall block diagram of the robot arm based on PD controller.

Where q_d (t) and $\dot{q}_d$(t) denote the desired joint positions and velocities; qa (t) and $\dot{q}_a$(t) denote the actual joint positions and velocities; K_P and K_d are (nxn) positive definite diagonal matrices.

The aim of PD is to design a position controller of a robot arm by selection of a PD parameters gains k_p, k_d using genetic algorithm, where GA is a stochastic global adaptive search optimization technique based on the mechanisms of natural selection [6].GA applied to the tuning PD position controller gains k_p and k_d for the three joints using Integral Square-Error (ISE) to ensure optimal control performance at nominal operating conditions.

The Two gains of PD controller after tuning for joint1 k_{p1}=41.032 and k_{d1}=100.991, for joint2 k_{p2}=27.959 and k_{d2}=128.169 and for joint3 k_{p3}=51.264 and k_{d3}=94.943 then modify this error signal to produce control input for system. This control input then forces the system to produce output as close as possible to the desire trajectory.

The solution we propose to use artificial intelligent such as ANFIS and Fuzzy-PID controllers.

5. Robot arm trajectory tracking using ANFIS

5.1. Principles of ANFIS

Neuro_fuzzy network systems combine the advantageous of neural network and fuzzy logic system. Neural network provides connectionist structure and learning abilities to the fuzzy logic systems, and the fuzzy logic systems provide neural networks with a structural framework with high-level fuzzy IF-THEN rule of thinking and reasoning. Neural network-based fuzzy systems, NF have the learning ability of neural networks to realize the fuzzy logic inference system, are gained popularity in the control of nonlinear systems [7]. The adaptive NF inference system (ANFIS) is one of the proposed methods to combine Fuzzy logic and artificial neural networks. Fig.3 shows the adaptive NF inference system structure. It is composed of five functional blocks (rule base, database, a decision making unit, a fuzzyfication interface and a defuzzyfication interface) which are generated using five network layers:

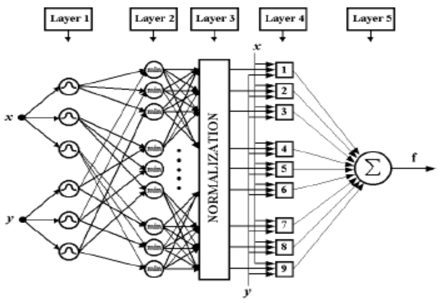


Fig. 3: Two-input NF controller structure.

Layer 1: This layer is composed of a number of computing nodes whose activation functions are fuzzy logic membership functions.

Layer 2: This layer chooses the minimum value of the inputs.

Layer 3: This layer normalizes each input with respect to the others (The *i*th node output is the i^{th} input divided by the sum of all the other inputs).

Layer 4: This layer's i^{th} node output is a linear function of the third layer's i^{th} node output and the ANFIS input signals.

Layer 5: This layer sums all the incoming signals. The ANFIS structure can be tuned automatically by a least-square estimation (for output membership functions) and a back propagation algorithm (for output and input membership functions) [8].

5.2. Structure of robot arm Based on ANFIS controller

The adaptive neural fuzzy inference system (ANFIS) method is chosen to design the Neuro-Fuzzy Controller. The overall block diagram of the system under ANFIS control is shown in Fig. 4. The system consists of a forward path controller in addition to a feedback path controller. The forward path controller is ANFIS and dynamics model for the robot arm. The feedback path consists of the actual position angle (q_a) and actual velocity ($\dot{q}_a$).

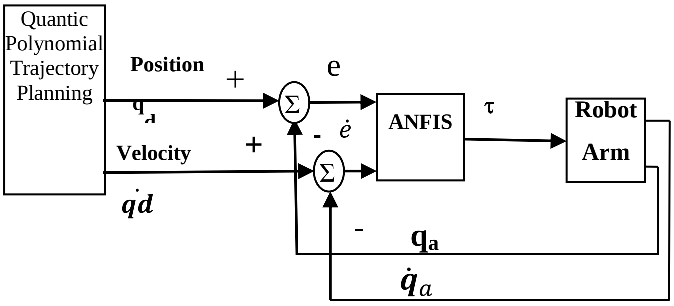
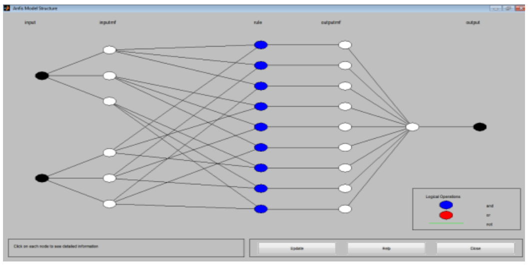


Fig.4: The overall block diagram of the robot arm based on ANFIS controller.

The ANFIS controller developed consists of two inputs, position error (e) and velocity error ($\dot{e}$). This work considers the ANFIS internal structure for the three joints as the same with first order sugeno model as shown in Fig. 5 where, the first, second and third joints contain 9 rules with triangular membership function. The membership functions with product inference rule are used at the fuzzification level. Hybrid learning algorithm that combines least square method with gradient decent method is used to adjust the parameter of membership function.



Click on each node to see detailed information

Fig. 5: Structure of ANFIS for the three joint.

In the first layer of the NF structure, sampled position error e and velocity error$\dot{e}$, multiplied byrespective weights, are each mapped through three fuzzy logic membership functions.

The second layer calculates the minimum error value of two input weights by determining the firing strengths of the rules which is given as:

$$\mu_i = \min(\mu^j_{A1}(e), \mu^j_{A2}(e^\bullet)) \tag{8}$$

The third layer calculates the weight which is normalized. Normalized value of the firing strengths is defined as the ratio of firing strength of the n rule to the sum of the firing strengths.

$$\overline{\mu_i} = \frac{\mu_i}{\sum_{i=1}^{9} \mu_i} \tag{9}$$

Where n=9 the number of rules for each joint.

The fourth layer containing adaptive nodes is the defuzzyfication (weighted average method) layer. The output from this layer is: $\overline{\mu_k}$ ($p_i x + q_i y + r_i$), where p_i, q_i,m_i and r_i are the consequent parameters of the node. The inputs(x=e, y=$\dot{e}$) and output

$O_{4,i}$ relationship in this layer can be defined as:

$$O_{4,i} = \overline{\mu_i} f_i = \overline{\mu_i}(p_i x + q_i y + r_i) \quad i=1,2\ldots9 \qquad (10)$$

Where i is the 4th layer output.

For a zero-order Sugeno model, the output level *O* is a constant *(p=q=0) so the relationship between inputs and output is:*

$$O_{4,i} = \overline{\mu_i} f_i = \overline{\mu_i}(r_i) \ i=1,2\ldots9 \qquad (11)$$

The fifth layer consists of a single fixed node; it is the summation of the weighted output of the consequent parameters in layer 4. The output layer is given by:

$$O_5 = \sum_{i=1}^{9} \overline{\mu_i} f_i = \frac{\sum_{i=1}^{9} \mu_i f_i}{\sum_{i=1}^{9} \mu_i} \qquad (12)$$

6. Robot arm trajectory tracking using Fuzzy-PID

The fuzzy logic programming have been become widely used in industry. Extensive number of researches were developed using fuzzy logic technique [17].Fuzzy PID controllers are classified into two types: the direct action fuzzy control and the fuzzy supervisory control. The direct action type replaces the PID control with a feedback control loop to compute the action through fuzzy reasoning where the control actions are determined directly by means of a fuzzy inference. These types of fuzzy controllers are also called PID-like controllers. On the other hand, the fuzzy supervisory type attempts to provide nonlinear action for the controller output using fuzzy reasoning where the PID gains are tuned based on a fuzzy inference system rather than the conventional approaches. The design process of the fuzzy controller is described as follows [18]:

- Define the input and output variables of FLC. In this work, there are two inputs of FLC, the error e (t) and it`s rate of change of error $\dot{e}$(t) and three outputs K_P ,K_I and K_d are respectively as shown in Figure (6).

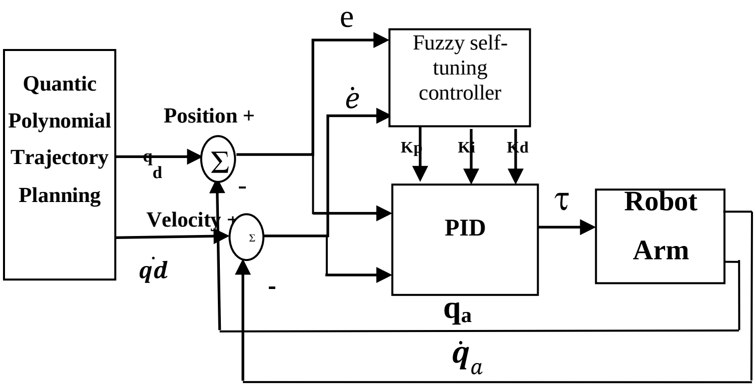


Fig. 6: Fuzzy self-tuning proposed.

- Fuzzify the input and output variables by defining the fuzzy sets and membership functions. Each variable of fuzzy control inputs has seven fuzzy sets ranging from negative big (NB) to positive big (PB) as shown in Fig. 7 for the two inputs e and $\dot{e}$, and the output of FLC has the following membership function as shown in Fig. 8 for the three outputs K_p, K_i , and K_d.

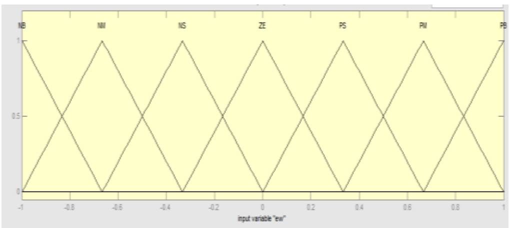


Fig. 7: Memberships function of inputs (e) and ($\dot{e}$).

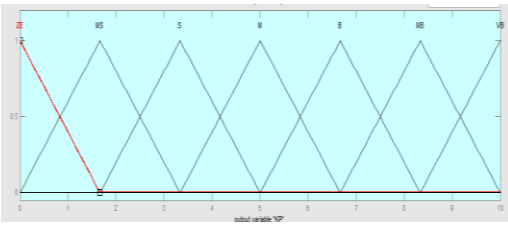


Fig.8: Memberships functions of outputs (K_{p1}, K_{i1}, and K_{d1}).

- Design the inference mechanism rule to find the input-output relation. This work uses Mamdani (max-min) inference mechanism where, Tables (1), (2), and (3) show the control rules that used for fuzzy self-tuning of PID controller.
- Defuzzify the output variable. Here, the center of gravity (COG) method, the most frequently used method, is used. The control action is[18]:

$$COG = \frac{\sum_{i=1}^{m} \mu(f_i).f_i}{\sum_{i=1}^{m} \mu(f_i)} \qquad (13)$$

Now the control action of the PID controller after self-tuning can be describing as:

$$U_{PID} = K_{p2} * e(t) + K_{i2} \int edt + K_{d2} \frac{de(t)}{dt} \qquad (14)$$

Where K_{P2}, K_{I2}, and K_{d2} are the new gains of PID controller and are equals 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 gains outputs of fuzzy control that are varying online with the output of the system under control. K_p, K_i, and K_d are the initial values of the conventional PID.

Table 1: Rule bases for determining the gain K_{P1}.

ė/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_{i1}.

ė/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_{d1}.

ė/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. Simulation results

The simulation has been performed for the first three degrees of freedom of PUMA560 using MATLAB 2013a by considering the PUMA-560 robot manipulator dynamics from [4], [15], information about inertial constant and gravitational constant are given in the Appendix A [3] based on the studies carried out by Armstrong and Corke[15],for showing the efficiency of the suggested Fuzzy-PID position controller than PD tuned by GA and ANFIS where, all controllers tested to quintic polynomial trajectories. Desired and actual position for joints 1, 2 and 3 of puma 560 robot arm controlled using PD controller tuned by GA are shown in Figs. 9, 10 and11 respectively where, GA reaches to the values of the 6 PD parameters after 450 epochs with fitness value 0.0105411. ANFIS editor GUI is available in Fuzzy Logic Toolbox [9]. Using a given input/output data set, the toolbox constructs a fuzzy inference system (FIS) whose membership function parameters are adjusted using either a back propagation algorithm alone, or in a combination with a least squares type of method. We used A hybrid method which employs for updating membership function parameters which consisting of back propagation for the parameters associated with the input membership functions, and least squares estimation for the parameters associated with the output membership functions[16].This allows the fuzzy systems to learn from the data they are modelling[9].

ANFIS control provides the robot arm joints with minimum error between desired and actual position for joints 1, 2 and 3 respectively with minimum number of iteration= 51 epochs compared with PD controller As shown in Figs.12, 13 and 14.

The fuzzy self-tuning PID controller is applied to position control of the first 3 joint of puma560 robot arm. The simulation results were obtained using SIMULINK-MATLAB 2013a as shown in Figs. 15, 16and 17 show the desired and actual position for joints 1, 2 and 3 of puma 560 robot arm controlled using fuzzy-PID controllers with respect to quintic polynomial trajectory planning.

The fuzzy supervisory tries to vary the PID parameters during process operation to enhance the system response and eliminates the disturbances. The gradient-based optimization technique determines search directions for minimization of an objective (or error) function. This technique can be used to minimize energy consumption in distributed environmental control systems while maintain a high occupant comfort level.

Table4 show a comparison between RMS error, steady state error for joint 1, and 3 for all types of controllers (PD tuned by GA, ANFIS and Fuzzy-PID implemented to control the position angle of puma 560 robot arm θ1, θ2, θ3.

Table4: The comparison results of PD, ANFIS and 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**
PD tuned using GA	**0.05932**	**-0.012**	**-0.097**	**-0.008**
ANFIS	**0.04878**	**-0.011**	**-0.077**	**-0.011**
Fuzzy-PID	**0.02278**	**0.003**	**0.007**	**0.005**

From Table4 position control using Fuzzy-PID has better steady state error and RMS error than controlled based on PD tuned by GA and ANFIS. By comparing steady state and RMS error in a system it was found that the Fuzzy-PID's errors (Steady State error for joint1= 0.003, joint2=0.007, joint3=0.005 and RMS error=0.02278) than ANFIS's errors (Steady State error for joint1= -0.011, joint2=-0.077, joint3=-0.011 and RMS error=0.04878) and PD's errors (Steady State error for joint1= -0.012, joint2=-0.097, joint3=-0.008and RMS error=0.05932). Fuzzy-PID controller has fast response and small errors for quintic polynomial trajectory control of robot arm. Figs.18, 19 and 20 give complete comparisons between the three controllers for joint 1, 2 and 3 errors respectively.

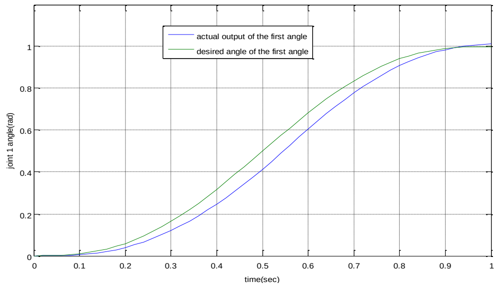


Fig. 9: desired, actual position for joints 1 controlled using PD tuned using GA

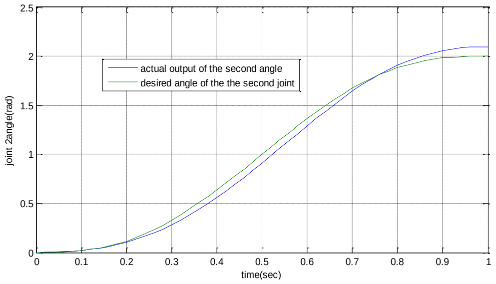


Fig. 10: desired, actual position for joints 2 controlled using PD tuned using GA.

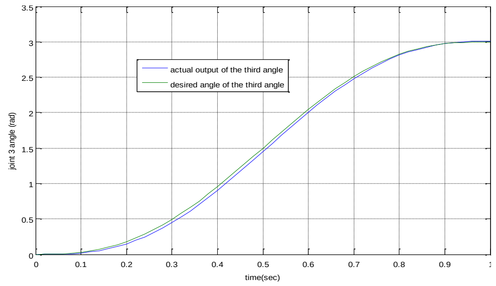


Fig. 11: desired, actual position for joints 3 controlled using PD tuned using GA.

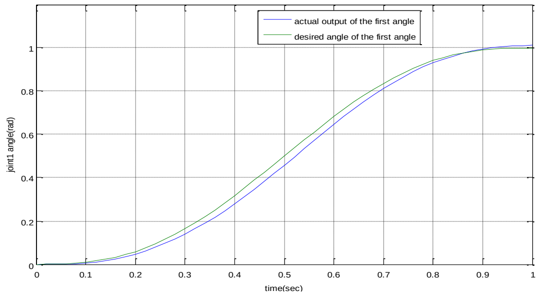


Fig. 12: desired, actual position for joints 1 controlled using ANFIS controller.

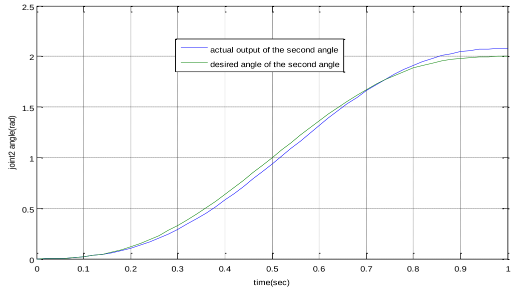


Fig. 13: desired, actual position for joints 2 controlled using ANFIS controller.

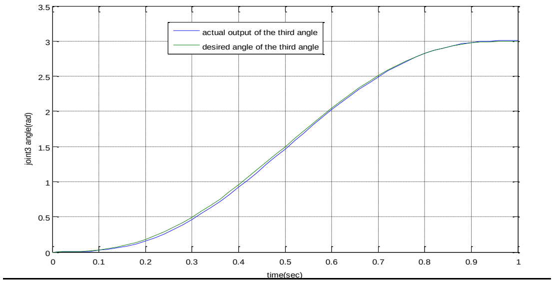


Fig. 14: desired, actual position for joints 3 controlled using ANFIS controller.

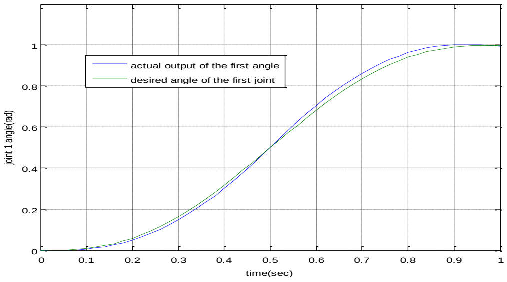


Fig. 15: desired, actual position for joints 1 controlled using Fuzzy-PID controller.

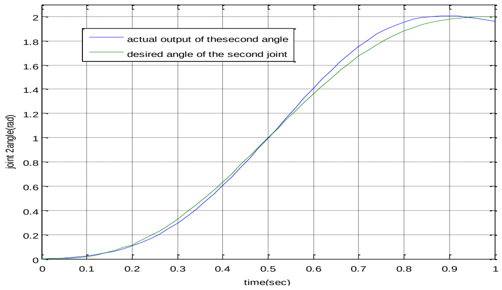


Fig. 16: desired, actual position for joints 1 controlled using Fuzzy-PID controller.

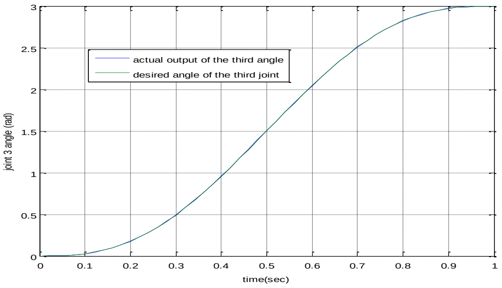


Fig. 17: desired, actual position for joints 3 controlled using Fuzzy-PID controller.

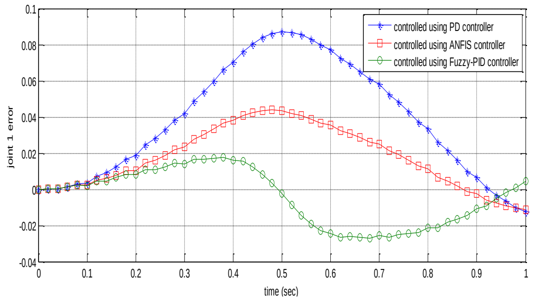


Fig. 18: comparison between joint1 errors after controlled using PD and ANFIS.

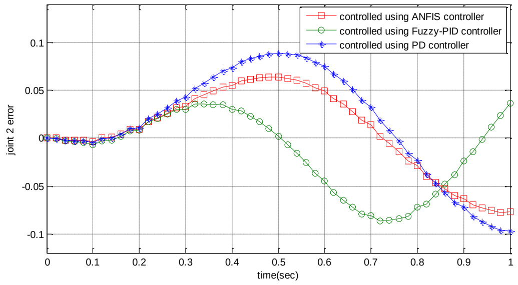


Fig. 19: comparison between joint2 errors after controlled using PD and ANFIS.

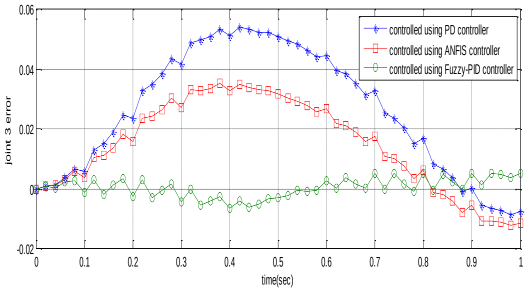


Fig. 20: comparison between joint3 errors after controlled using PD and ANFIS.

8. Conclusion

In this study, Fuzzy-PID controller has been applied to position control of the first three joints of the PUMA 560 robot arm in order to obtaining fine quintic polynomial trajectory with minimum error and good disturbance rejection. Results have been compared with PD tuned using GA and ANFIS, from the simulation results it was concluded that:

- By comparing steady state and RMS error the position control of the three joints controlled using Fuzzy-PID has better steady state error and RMS error than controlled using PD tuned by GA and ANFIS.
- ANFIS converges with a smaller number of iteration steps with the hybrid learning algorithm compared with PD controller tuned by GA.
- The responses had showed to us that the designed based on Fuzzy-PID controller has much faster response than using the other controllers.

References

[1.] S. Yadegar an Azurabinti CheSoh,"Design Stable Robust Intelligent Nonlinear Controller for 6- DOF Serial Links Robot Manipulator", I.J. Intelligent Systems and Applications, 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 Matlap / Simulink and Their Integration into Graduate/Undergraduate Nonlinear Control, Robotics and MATLAB Courses", International Journal of Robotic and Automation, Vol. 6,No.3, 2012, pp.167-191.

[4.] B. Siciliano and O. Khatib, "Springer handbook of robotics", Springer-Verlag New York Inc, 2008.

[5.] D. Breen and D. Kennedy,"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.] J. Kim and N. Kasabov: "HyFIS, Adaptive neuro-fuzzy inference systems and their application to nonlinear dynamical systems", Elsevier ,Neural Networks, vol. 12, No.9, 1999, pp. 1301–1319.

[8.] J.S.R. Jang, " ANFIS: Adaptive-network-based fuzzy inference system, " IEEE Trans Syst. Man. Cybernet, vol.23, No.3, 1993, pp. 665-685.

[9.] Jang, J.-S. R. and C.-T.Sun, "Neuro-fuzzy modeling and control", Proceedings of the IEEE, March 1995.

[10.] Chedmail P., Gautier M. "Optimum choice of robot actuators", Trans, ofASME, J. of Engineering for Industry, Vol. 112 , No. 4, 1990, p. 361-367.

[11.] K. Ogata, "Modern control engineering", Prentice Hall, 2009.

[12.] J. J. D'Azzo and C. H. Houpis, "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.] A. 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.] P.I. Corke and B. Armstrong-Helouvry, “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.

[16.] Jang J.," ANFIS: Adaptive network-based fuzzy inference systems", IEEE Transactions on Systems, Man, and Cybernetics 23, 1993, pp.665-685.

[17.] 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.

[18.] G.U.V.Ravi Kumar and Mr.Ch.V.N.Raja," Control of 5DOF Robot Arm using Fuzzy Supervisory Control", International Journal of Emerging Trends in Engineering and Development Vol.2, No. 4, ,March 2014.

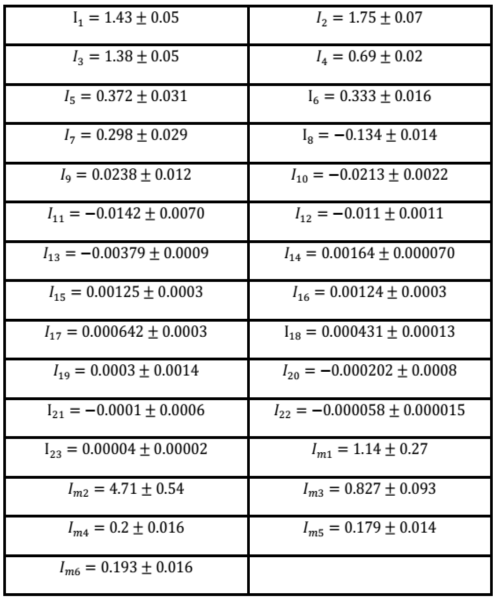
[19.] Al Ashi, Mahmoud M. et al.,"Trajectory Tracking Control of A 2-DOF Robot Arm Using Neural Networks", MS.c, Islamic University of Gaza, Electrical Engineering Department, Feb. 2014.

[20.] Spong, M.W., and Vidyasagar, M.," Robot Dynamics and Control", Wiley, New York, 1989.

[21.] 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.

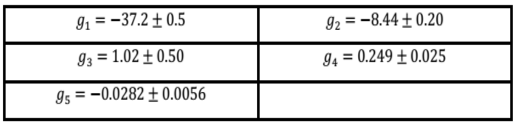
Appendix a

Table 5: Inertial constant reference (Kg.m^2)



$I_1 = 1.43 \pm 0.05$	$I_2 = 1.75 \pm 0.07$
$I_3 = 1.38 \pm 0.05$	$I_4 = 0.69 \pm 0.02$
$I_5 = 0.372 \pm 0.031$	$I_6 = 0.333 \pm 0.016$
$I_7 = 0.298 \pm 0.029$	$I_8 = -0.134 \pm 0.014$
$I_9 = 0.0238 \pm 0.012$	$I_{10} = -0.0213 \pm 0.0022$
$I_{11} = -0.0142 \pm 0.0070$	$I_{12} = -0.011 \pm 0.0011$
$I_{13} = -0.00379 \pm 0.0009$	$I_{14} = 0.00164 \pm 0.000070$
$I_{15} = 0.00125 \pm 0.0003$	$I_{16} = 0.00124 \pm 0.0003$
$I_{17} = 0.000642 \pm 0.0003$	$I_{18} = 0.000431 \pm 0.00013$
$I_{19} = 0.0003 \pm 0.0014$	$I_{20} = -0.000202 \pm 0.0008$
$I_{21} = -0.0001 \pm 0.0006$	$I_{22} = -0.000058 \pm 0.000015$
$I_{23} = 0.00004 \pm 0.00002$	$I_{m1} = 1.14 \pm 0.27$
$I_{m2} = 4.71 \pm 0.54$	$I_{m3} = 0.827 \pm 0.093$
$I_{m4} = 0.2 \pm 0.016$	$I_{m5} = 0.179 \pm 0.014$
$I_{m6} = 0.193 \pm 0.016$	

Table 6: Gravitational constant (N.m)



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