Received: August 20, 2020. Revised: September 21, 2020. 487

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Design Stable Controller for PUMA 560 Robot with PID and Sliding Mode Controller Based on PSO Algorithm

Hanan A. R. Akkar[1] Suhad Qasim G. Haddad[2]*

[1]*Department of Electrical Engineering, University of Technology, Baghdad, Iraq*
[2]*Department of Computer Engineering, University of Technology, Baghdad, Iraq*
* Corresponding author's Email: 120007@uotechnology.edu.iq

Abstract: The most significant challenge facing the researcher in the field of robotics is to control the robot manipulator with appropriate overall performance. This paper focuses mainly on the novel Intelligent Particle Swarm Optimization (PSO) algorithm that was used for optimizing and tuning the gain of conventional Proportional Integral Derivative (PID), and improve the parameters of dynamic design in Sliding Mode Control (SMC), which is considered a strong nonlinear controller for controlling highly nonlinear systems, particularly for multi-degree serial link robot manipulator. Additional modified Integral Sliding Mode Controller (ISMC) was implemented to the design of dynamic system with high control theory of sliding mode controller. Intelligent Particle Swarm Optimization (PSO) algorithm was introduced for developing the nonlinear controller. The algorithm demonstrates superior performance in determining the appropriate gains and parameters value in harmony with robot scheme dynamic layout in order to achieve suitable and stable nonlinear controller, besides reduce the chattering phenomenon. PUMA robot manipulator that was used as study case in this work, shows perfect result in step response, with acceptable steady state, and overshoot, besides, eliminating the disadvantage of chattering in conventional SMC. Matlab / Simulink presents to increase the speed of matrix calculation in forward, inverse kinematics and dynamic model of manipulator. Comparison was made between the proposed method with existing methods. Result shows that integral sliding mode with PSO (ISMC/PSO) gave best result for stable step response, minimum mean square error with best objective function, and stable torque.

Keywords: Proportional integral derivative, Integral sliding mode controller, PUMA robot manipulator, Dynamic design, Matlab/simulink, Intelligent particle swam optimization.

1. Introduction

Robot manipulator is a rigid framework that contains many joints, and can be described as a set of components that work together to achieve specific goals. In addition, it can be designed to perform a series of tasks by collecting environmental data using some sensors and making automated decisions. The robotic institution in the United States describes robot as "a reprogrammable, multifunctional manipulator design for moving material, parts, tools or specialized devices through various programmed motions to perform various tasks" [1].

In this work the PUMA 560 robot, which has six degrees of freedom, all are rotary joints with serial connections, was used as a study case. The first three joints are used to control the robot's handle position. The second three joints are to obtain the orientation of the robot's wrist locus. PUMA robot manipulator is widely used in medical, automotive, education and other important applications, which are considered to be a challenging job for people to operate. The parameters and dimensions of this robot were all known and documented in different literatures [2]. Studying the robot manipulator can be divided into two key parts, the first part is kinematic and the second part is dynamic. Robot kinematics, which is an important part of controller design, can be described as a non-force analysis relating the robot 's rigid bodies to the base of the effector. Kinematic consists of two main parts, forward and inverse

Kinematic. Forward kinematic are to calculate the final effector position for different joints, while inverse kinematic are to calculate the joint angle for the specified base effector position. The kinematic problem, which is an integral part of controller design, is to discover the transformation concerning Cartesian space and joint space, where robots are controlled within the joint space. The dynamic robot model is used to define robot performance as linear or non-linear. Dynamic simulation describes the relation between joint motion, speed, accelerations, torque with current or voltage, and can also be employed to define the particular dynamic properties associated with the system behaviour, such as inertia, Coriolis, centrifugal and other related parameters[1, 3].Controlling robot manipulators is challenging because of the multi-input with multi-output and non-linear design, so it was a huge challenge to the specialists and researchers to build stable and reliable controller supporting the robot, where location and orientation analytics are crucial for the suitability of robotic system in different environments [4, 5]. Designing a stable and strong controller is an important part for sensitive and various applications for robot manipulator. The control of robot arms can be categorized into two specific categories: first, traditional control strategies with linear and nonlinear controls. second, intelligent control methods. this work will concentrate on linear and nonlinear controller with intelligent swarm optimization. Due to its simple basic configuration and robustness under diverse operating environments, PID controllers are highly effective for the linear system and are used widely in industrial applications and robotic controllers. Nonetheless, precise PID tuning is difficult because most system are very complex and have certain challenges, such as non-linearity, and time delay in response. Modification of PID parameters are the most important components of the PID controller design, which is a crucial optimization [6]. The most common nonlinear model-based controller is Sliding Mode Controller SMC which has been properly applied in various applications such as motor control, space system, automatic flight control and finally robot control. It is considered a powerful advance stable robotic manipulator control system which can achieve asymptotic reactivity and stability. Adjusting the sliding mode control parameters is an important part of reducing the chattering disadvantage and developing stable coefficient for nonlinear controllers. Several techniques have been suggested to reduce chattering in sliding mode controller, to minimize the output error and increased the system dynamic response. The Integral sliding mode controller ISMC was suggested in this work to boost the manipulator output to accomplish the desired tasks with high stability [7]. With the emergence of smart algorithm and optimization theories, modern smart optimization techniques have played an increasingly important role in tuning and modification of robot manipulator parameters. Intelligent particle swarm optimization (PSO) is considered as a soft computing framework that provides high prepared attributes as a good output optimizer. In this work, PSO was suggested to adjust the nonlinear coefficient for SMC, ISMC, and to obtain tuning of PID parameters, it is basically based on the simulation of the birds and bees' social behaviour, and the concept of fish swarm operation. This algorithm can create the perfect solution, with a high quality and a lower computing time than other techniques, with progressively stable assembly characteristics, so it tends to be used in numerous controller tuning systems and methods. The algorithm efficiently changes the parameters with sufficient convergence to achieve impressive acceleration of the system. In addition, feature optimization with PSO was implemented to obtain the minimum necessary error for the location in the joints, and to improve the performance quality in the final of end effector response and the robot stability. Simulations and testing have been used to demonstrate that PSO can effectively determine the coefficients for the switching sliding control and PID control [8].

This paper has been arranged accordingly. Some related works are presented in the second section. The third section deals with the analysis, modelling and simulation of the kinematics of PUMA robot and its dynamics. Section four concentrates on the strategy of controller model, the methodology that used first with conventional PID, sliding mode control and, ultimately, the integral part of SMC for controlling the dynamic part of the robot. Section 5 discusses (PSO) algorithm as well as how the control of PID and SMC parameters can be set. The outcome of the debate and simulation is shown in the sixth section. The last section in the paper introduce on conclusion and future work.

2. Related work

The tuning of the controller and the dynamic model is considered an attractive exploration and technology researchers' zone. Many controller designs with different algorithms were suggested. A concise summary of previous works is given below.

[3] An adaptive sliding-mode controller was proposed with PSO algorithm for optimizing PUMA

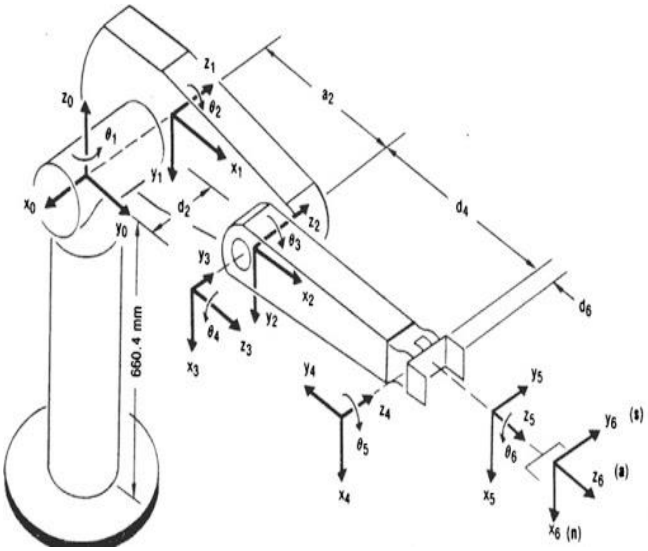


Figure. 1 Details of coordinates for PUMA robot

560 robot manipulator, this controller can respond and adapt itself to changes system parameters according to the external intervention. PSO algorithm was used to enhance the parameters of the sliding function and minimize the chattering action. [9] Developed a stable adaptive Particle Swarm Optimizer PSO for automatic and systematic tuning of PID gains for two degrees of freedom robot arm manipulator as a study case, thus optimizing cost function. The disadvantage of this method was the stability and robustness of control. [10] Simple controller PD has been implemented with a non-linear robotic arm dynamic system to compensate for uncertainties of the model system. PSO algorithm has been used as a powerful method to find the optimal value of the PD parameters with minimum cost function. The simulation results show very satisfactory results of monitoring difficult trajectory with various initial conditions for the proposed controller. [11] PID controller auto-tuning system was implemented for Puma robot manipulators. Two methods of multi-objective optimization were tested, namely multi objective cuckoo scan (MOCS) and multi objective particle swarm optimization (MOPSO). Comparison was made between the results of the two algorithms in the case of achieving a predefined trajectory with a reasonable tracking accuracy. Statistics taken from a simulation of the robot clearly indicate that (MOCS) performs significantly better than (MOPSO) with respect to all the parameters. [12] Sliding mode controller with boundary layer was presented for dynamic design with control of a six-degree robot arm manipulator type IRB-120. The chattering that generates unreliable signal with high frequency oscillation in the sliding mode control was eliminated by using the boundary layer which provides the controller with stabilization and improve the total performance. [13] Nonlinear PD with terminal SMC was proposed for designing PUMA robot control. Screw theory was used to overcome the calculation of robot manipulator kinematics. Device efficiency for linear tracking and non-linear tracking was testing with many trajectory problems. [14] The robotic device stability was demonstrated using the Lyapunov impedance-based technique, with Particle Swarm algorithm PSO for optimizing the parameters of control. The PSO algorithm was constructed and developed through the use of different index in joint and Cartesian spaces. A robot manipulator fixed to a circular trajectory was used to test the efficiency of the proposed process. [15] New technique for robot manipulators with a robust high-order composite was introduced, with super-twisting sliding mode controller (HOSTSMC) .The suggested approach improves the conventional sliding mode controller (TSMC) and the approximate state of sliding mode (ESMC) to ameliorate the chatter.

3. Kinematics and Dynamic of PUMA robot

The most critical element of robotic manipulator is the end effector 's final correct location without any disturbance that would influence the robot's final performance. Robot kinematics is to design a geometry representation to study the multi-degree robot manipulator movement which forms the final robot scheme. Kinematic problem includes solution for both forward and inverse kinematic.

3.1 Forward kinematic

For robot manipulator forward kinematics can be measured in four parts as follows: 1) analyse the descriptions of the link. 2) Determine the convention matrix of Denavite– Hartenberg (DH). 3) Attachment of frames.4) Computed forward kinematics. First part we have to analyse the link descriptions, where there are four parameters as shown: link length (a_i) : length distance among Z_i and Z_{i+1}taken along the X_i . Link offset (d_i) :distance between X_{i+1} and X_i taken along the Z_i Joint angle (θ_i) : angle between X_i and X_{i+1} taken along Z_i.Link twist (α_i) : angle between Z_i and Z_{i+1}taken along X_i. Fig. 1 shows the details of coordinates of the six joints for PUMA 560 robot manipulator.

There will be three parameters set and one is variable. For example the joint angle (θ_i) would be variable in rotating joints. Whereas the offset of the link (d_i) is variable if the joint is prismatic [16,17]. Second, the standard Hartenberg (DH) Convention is

to be found. It is a style used to pick the reference frame for the robotic arm and to measure the forward kinematics for the robot manipulator depending on the relation between the joints. Under this convention, the coordination of each frame between two joints will determine the position of each joint relative to its preceding joint [1,2]. Third part is calculating the matrix of the frame connection. Consider that the rotation matrix ${}^{i-1}_{i}[R]$,which define the orientation of link i according to the link $i-1$ is obtain by multiplying two matrices , the rotation of angle α_{i-1} about X and the rotation of angle θ_i about Z. So we have:

$$ {}^{i-1}_{i}[R] = [\, R\,(\mathrm{X}_{i-1}, \alpha_{i-1})]\,[R\,(\mathrm{Z}_i, \theta_i)] \quad (1)$$

Where:

$$X_{i(\alpha i-1)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_{\alpha i-1} & -s_{\alpha i-1} \\ 0 & s_{\alpha i-1} & c_{\alpha i-1} \end{bmatrix} \quad (2)$$

$$\mathrm{Z}_{i(\theta i)} = \begin{bmatrix} c_{\theta i} & -s_{\theta i} & 0 \\ s_{\theta i} & c_{\theta i} & 0 \\ 0 & o & 1 \end{bmatrix} \quad (3)$$

The matrix ${}^{i-1}_{i}[T]$ for a particular link can be obtained as follows :

$$ {}^{i-1}_{i}[T] = \hat{R}_{\mathrm{x}}(\alpha_{i-1})\breve{D}_{\mathrm{x}}(\alpha_{i-1})\hat{R}_{\mathrm{z}}(\theta_i)\,\breve{Q}_{\mathrm{i}}(d_i) \quad (4)$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha i-1} & -s_{\alpha i-1} & 0 \\ 0 & s_{\alpha i-1} & c_{\alpha i-1} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & \alpha_{i-1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x$$

$$\begin{bmatrix} c_{\theta i} & -s_{\theta i} & 0 & 0 \\ s_{\theta i} & c_{\theta i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & di \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$$ {}^{i-1}_{i}[T] = \begin{bmatrix} c_{\theta i} & -s_{\theta i} & 0 & \alpha_{i-1} \\ s_{\theta i}\, c_{\alpha i-1} & c_{\theta i} c_{\alpha i-1} & -s_{\alpha i-1} & -d_i s_{\alpha i-1} \\ s_{\theta i} s_{\alpha i-1} & c_{\theta i} s_{\alpha i-1} & c_{\alpha i-1} & d_i c_{\alpha i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

Where $\hat{R}_{\mathrm{x}}$ and $\hat{R}_{\mathrm{z}}$ states to rotation matrix while $\breve{D}_{\mathrm{x}}$ and $\breve{Q}_{\mathrm{i}}$ states to translation matrix, and $c_{\theta i}$, $s_{\theta i}$ shorthand of $\cos\theta$,$\sin\theta$ respectively. The next step is the estimation of total forward kinematics, the final 4x4 homogeneous transformation for the link(i)in relation to the link $(i-1)$, for example, forward kinematics of the end effector (i) , in relation to the base position$(i-1)$will be calculated by multiplying the all of ${}^{i-1}_{i}[T]$ matrices as shown below:

Table. 1 DH for PUMA 560 robot manipulator [17]

Link 1	$\theta_i\ (rad)$	$\alpha_i(rad)$	$\mathrm{a_i}(m)$	$\mathrm{d_i}(m)$
1	θ_1	$-\pi/2$	0	0
2	θ_2	0	0.4318	0.14909
3	θ_3	$\pi/2$	0.0203	0
4	θ_4	$-\pi/2$	0	0.43307
5	θ_5	$\pi/2$	0	0
6	θ_6	0	0	0.05625

$$[{}^{0}_{6}T] = [{}^{0}_{1}T][{}^{1}_{2}T][{}^{2}_{3}T][{}^{3}_{4}T][{}^{4}_{5}T][{}^{5}_{6}T] \quad (7)$$

Depending on the above formulation the $[{}^{0}_{6}T]$ can be determined as following [5 ,17]:

$$[{}^{0}_{6}T] = \begin{bmatrix} a_x & b_x & o_x & p_x \\ a_y & b_y & o_y & p_y \\ a_z & b_z & o_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

All the elements in matrix of Eq. (8) are calculated depending on [5], where the first three columns are for orientation and the last column is for position of the final matrix. Table 1 shows the basic DH for PUMA 560 robot manipulator.

3.2 Inverse kinematics

It can be solved in several forms, such as geometric or algebraic analysis, considering robotic arm arrangement. The procedure for solving robot's inverse kinematics is to multiply the ${}^{i-1}_{i}[\,T]$ inverse matrix on both sides of Eq. (8) and make the matrix's corresponding elements equal on both sides such that mutual variables are obtained , as follows:

$$[{}^{0}_{1}T]^{-1}[{}^{0}_{6}T] = [{}^{1}_{2}T][{}^{2}_{3}T][{}^{3}_{4}T][{}^{4}_{5}T][{}^{5}_{6}T] \quad (9)$$

$$[{}^{1}_{2}T]^{-1}[{}^{0}_{1}T]^{-1}[{}^{0}_{6}T] = [{}^{2}_{3}T][{}^{3}_{4}T][{}^{4}_{5}T][{}^{5}_{6}T] \quad (10)$$

$$[{}^{2}_{3}T]^{-1}[{}^{1}_{2}T]^{-1}[{}^{0}_{1}T]^{-1}[{}^{0}_{6}T] = [{}^{3}_{4}T][{}^{4}_{5}T][{}^{5}_{6}T] \quad (11)$$

$$[{}^{3}_{4}T]^{-1}[{}^{2}_{3}T]^{-1}[{}^{1}_{2}T]^{-1}[{}^{0}_{1}T]^{-1}[{}^{0}_{6}T] = [{}^{4}_{5}T][{}^{5}_{6}T] \quad (12)$$

$$[{}^{4}_{5}T]^{-1}[{}^{3}_{4}T]^{-1}[{}^{2}_{3}T]^{-1}[{}^{1}_{2}T]^{-1}[{}^{0}_{1}T]^{-1}[{}^{0}_{6}T] = [{}^{5}_{6}T] \quad (13)$$

From the above process, an undefined quantity will be moved from the right part of the equation to the left part to separate it from the other undefined and then solving it. Then the same process can repeat to the others quantity to solve all the unknown variables [18]. In this work all the forward and inverse kinematics are designed and solved with matrix based

on Matlab/Simulink model to increase the speed of matrix computation [5, 19].

3.3 Dynamic model of PUMA 560 manipulator

The dynamic equation is about the analysis and study of motion concerning forces. Dynamic modelling is required for mechanical part design, control and finally in simulation. It is used to define dynamic parameters and also to describe the relationship between displacement, distance, acceleration and force acting on the manipulator of the robot. Equation of a multi degree of freedom for robot manipulator can be calculate as follow [1, 20]:

$$A\ (\vartheta\ )\ \ddot{\vartheta} + N\ (\vartheta\ \dot{\vartheta}) = \tau \qquad (14)$$

Dynamic equation for the robot manipulator as a result can be written as follows:

$$\mathrm{N}\ (\vartheta\ \dot{\vartheta}) = V\ (\vartheta\ \dot{\vartheta}) + G\ (\vartheta) \qquad (15)$$

$$V(\vartheta\ \dot{\vartheta}) = B\ (\vartheta)\ [\dot{\vartheta}\ \dot{\vartheta}] + C(\vartheta)\ \left(\dot{\vartheta}\right)^2 \qquad (16)$$

$$\tau = A\ (\vartheta)\ \ddot{\vartheta} + B\ (\vartheta)\ [\dot{\vartheta}\ \dot{\vartheta}] + C(\vartheta)\ \left(\dot{\vartheta}\right)^2 + G\ (\vartheta) \qquad (17)$$

Where: $A\ (\vartheta\ )$: Symmetric positive matrix is considered for kinetic energy and inertia matrix , with n x n dimension.$B\ (\vartheta)$: is for Coriolis torques matrix, with n x n (n-1)/2 dimension. $C(\vartheta)$: is for centrifugal torques, with n x n dimension. $G(\vartheta)$: is for gravity torques, with nx1 dimension. ϑ: is the joint position (or joint angle), for $\vartheta = [\vartheta_1\ ,\ \vartheta_2\ , \ldots \vartheta_n\ ]$,$\dot{\vartheta}$: is consider as n- vector for joint velocities, $\ddot{\vartheta}$: is consider as n- vector of accelerations. And τ: is consider as the joint force vector (torque). $[\dot{\vartheta}^2\ ]$: that can a vector given by $[\ \dot{\vartheta_1^2}, \dot{\vartheta_2^2}, \ldots, \dot{\vartheta_n^2}\ ]^{\mathrm{T}}$, $[\ \dot{\vartheta}\ \dot{\vartheta}]$: that can a vector given by$[\ \dot{\vartheta}_1\ \dot{\vartheta}_2\ ,\ \dot{\vartheta}_1\ \dot{\vartheta}_3\ \ \dot{\vartheta}_1\ \dot{\vartheta}_n\ ,\ \dot{\vartheta}_2\ \dot{\vartheta}_3, \ldots\ ]^{\mathrm{T}}$.

The input of the dynamic system is torque matrix in the robotic manipulator arrangement while the outputs are real variables displacement and joints, as result it can be written as follows:

$$\ddot{\vartheta} = A^{-1}\ (\vartheta).\left[\tau - \mathrm{N}\ (\vartheta\ \dot{\vartheta})\right] \qquad (18)$$

All the parameters and matrix have been computed as mentioned in [5, 16 ,17]. In this work only the first, second and third links will be taken into consideration. Fig. 2 illustrate the Block diagram for dynamic and kinematics model for Puma 560 robot manipulator.

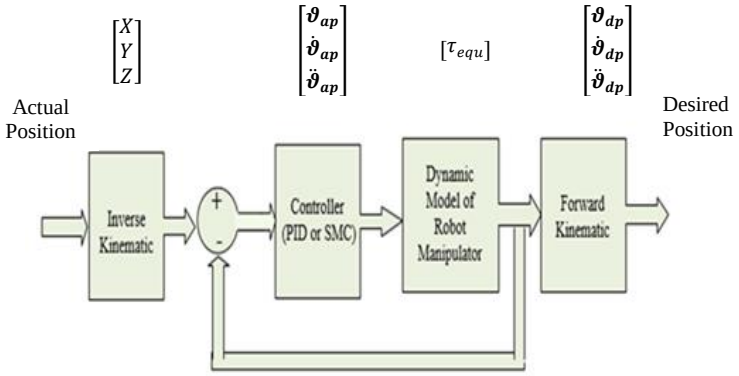


Figure. 2 Block diagram for dynamic and kinematics model for proposed design

4. Strategy of the controller model

4.1 Design PID controller

The Proportional Integral Derived PID controller is the most commonly used controller in a large range of applications, because of the simple nature of this technique and satisfactory results when specifications for outcomes are appropriate and changes in parameters are minimal. So the, PID controller is seen as one of the common controllers used in robot controlling. The PID controller can be expressed as follows [6, 9]:

$$u_{pid}\ (t) = kp\ e\ (t) + ki\ \int e(t)dt + kd\ \dot{e}(t) \qquad (19)$$

Where u_{pid} (t) is the control signal that will be as the final trajectory position of the robot controller, e (t) is the error and $\dot{e}$ (t) is the rate of change in error. Where $e\ (t)$ is the difference between desired error and actual position error, which is regarded as the final position of the end effector of the robot. The PID controller constants (kp , ki and kd) can described as: (kp) is the proportional gain and gives control action commensurate with the error signal e (t). The purpose of integral term (ki) in the PID controller is to reduce the steady state error by integrating the error signal e (t) continuously. (kd) is the derivative term that provides a control signal proportional to the change rate of error ($\dot{e}$) , resulting in the damping of the output overshoot and hence an improved transient response. The tuning of PID equation can be done as minimizing the mean square error between the desired error (e^d) and actual position error (e^a),as follows [6 ,14]:

$$e\ (t) = e^d(t) - e^a(t) \qquad (20)$$

$$Mse = \frac{1}{N}\sum_{i=1}^{N}\left(\mathrm{e}^d - \mathrm{e}^a\right)^2 \qquad (21)$$

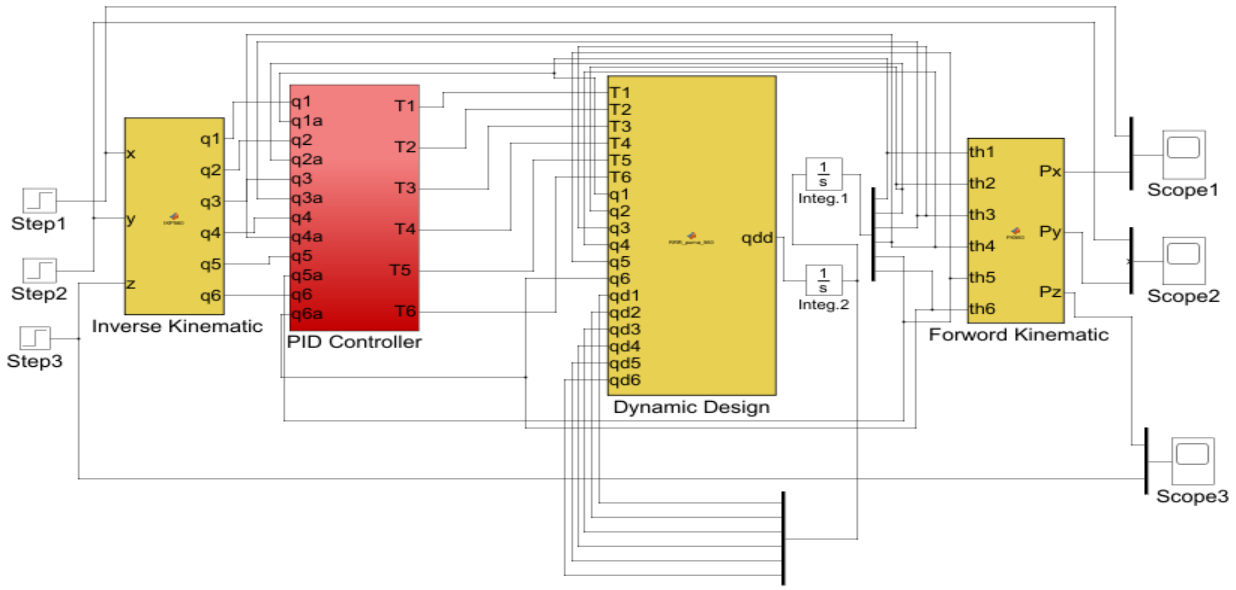


Figure. 3 Simulink design for dynamic model of PUMA robot with PID controller

Tuning the PID parameters, which is basically an optimization problem, is the most important element in the design of PID control by selecting the correct values of kp , ki and kd to improve tuning for the gains. Otherwise an insufficient selection for these values can result in the response being degraded rather than improved. Nonetheless, PID does not give maximum efficiency with nonlinear elements of robot manipulator [6]. PUMA 560 contains a separate controller for each joint. Therefore, 18 values for PID parameters will be used for six joints. In this work, these parameters will be adjusted on the basis of minimizing the mean square error of the joint depending on PSO algorithm. Fig. 3 shows the total design of Simulink diagram for PUMA Robot that was proposed in this work, the figure illustrates the block diagram for inverse, forward kinematics ,dynamic model and PID Controller, in addition blocks of step input function for each link of the robot was shown ,and finally the scopes that can describe and analyze the response of the links.

4.2 Design of mathematical model of SMC

The SMC is applied to force a system state path to pass through the sliding surface, then imposes the state's system path to slide along the switching surface till it stays at the origin. Thus, the fundamental goal of designing the sliding mode controller is to perform a high speed sequence and greatest precision for joint tracking. This controller can achieve good stability of the system beside quick dynamic response. The controller can be divided into two main parts, which are the discontinues part (τ_{dis}) that used to design suitable tracking performance based on linear methodology requiring very fast switching. The second part is equivalent controller (τ_{eq}) which is the effect of nonlinear terms that induced reliability and used to fine-tune the sliding surface slopes , sometimes causes system instability and chattering phenomenon [7]. Let us define the nonlinear input single for dynamic system as follow:

$$\mathrm{X}_n = f\ (\mathrm{X}) + b(\mathrm{X})v(t) \tag{22}$$

where (v) is considered as control input vector, X_n is n_{th} derivative for state vector X , f is nonlinear function for uncertainty dynamic , b(X) function of identified switching [$SIGN$] . The main objective of designing SMC is to train the appropriate state desire position (X_d) according to the variables in actual joint , the vector of tracking error will be defined as follows [21] :

$$e = \tilde{\mathrm{X}} = \mathrm{X} - \mathrm{X}_d \tag{23}$$

where: X is for real and actual position for state vector, X_d is the desired position, and $\tilde{\mathrm{X}}$: is for estimated tracking error vector. According to the theory of the SMC, sliding surface is the key essential important part to design this controller, calculation of varying in time for sliding surface $s(x,t)$, and the integral part will give as follows:

$$s(x,t) = \left(\frac{d}{dt} + \mu\right)^{n-1} \left(\tilde{\mathrm{X}}\right) = 0 \tag{24}$$

$$s(x,t) = \left(\frac{d}{dt} + \mu\right)^{n-1} \left(\int_0^t \tilde{\mathrm{X}}\ dt\right) = 0 \tag{25}$$

μ it is the coefficient of slope of sliding surface and is positive constant, in this methodology the main target is to keep the slope of sliding surface $s(x,t)$ close to zero. Thus, best strategies to achieve this is to find the input control v outside the sliding surface s (x, t) and remains on it. To keep the $S(e,\dot{e})$ close to zero, control law is design to satisfy the sliding condition of Lyapunov function, as follows:

$$V = \frac{1}{2} S^2 \geq 0 \tag{26}$$

Its time derivative becomes $\dot{V} = S^{\cdot} S$ and the control u is chosen such that

$$S^{\cdot} S \leq \eta |S| \tag{27}$$

Where η is consider appositive constant [16,17], so the sliding surface s (x, t) will computed as follows:

$$\frac{1}{2}\frac{d}{dt} S^2(x,t) \leq \eta |S(x,t)| \tag{28}$$

When the surface (S) ≈ Zero
So, error $e = \hat{X} = X - X_d \approx$ Zero. Let consider that:

$$S = \dot{e} + \mu e \tag{29}$$

$$\dot{e} = \frac{de}{dt} = \dot{X} - \dot{X}_d \tag{30}$$

$$\text{So : } S = (\dot{X} - \dot{X}_d) + \mu (X - X_d) \tag{31}$$

Derivative of Eq. (31) will consider as change in sliding surface, as follows:

$$\dot{S} = \frac{ds}{dt} = (\ddot{X} - \ddot{X}_d) + \mu (\dot{X} - \dot{X}_d) \tag{32}$$

$$\text{Since: } \ddot{X} = f + v \tag{33}$$

$$\text{So : } \dot{S} = f + U - \ddot{X}_d + \mu (\dot{X} - \dot{X}_d) \tag{34}$$

By imposing $S \to 0$ also $\dot{S} \to 0$
If we put $\dot{S} = 0$,in Eq. (34) , as follows:

$$0 = f + U - \ddot{X}_d + \mu (\dot{X} - \dot{X}_d) \tag{35}$$

Where f is the uncertainty of dynamic, under this hypothesis we get the best approximation for control ($\widehat{U}$) which can be defined as follows:

$$\widehat{U} = -\hat{f} + \ddot{X}_d - \mu (\dot{X} - \dot{X}_d) \tag{36}$$

Using the uncertainty switching control low is the best method to control the dynamic parameters of sliding mode [22] :

$$U_{dis} = \widehat{U} - K(x,t) S_{Sign} \tag{37}$$

K : is a positive constant function of (x,t), and S_{Sign} is the switching function which define as follows:

$$\begin{cases} S_{Sign} = 1 & \text{if} \quad S > 0 \\ S_{Sign} = 0 & \text{if} \quad S = 0 \\ S_{Sign} = -1 & \text{if} \quad S < 0 \end{cases} \tag{38}$$

keep the $S(e,\dot{e})$ close to zero in order to satisfy the sliding condition of Lyapunov function. So Eq. (25) and Eq. (32) can be arranged as follows:

$$\frac{1}{2}\frac{d}{dt} S^2(x,t) = \dot{S}\, S = \left[f - \hat{f} - K\, S_{SIGN}\right].S = \left(f - \hat{f}\right).S - K|S| \tag{39}$$

$$s(x,t) = \left(\frac{d}{dt} + \mu\right)^2 \left(\int_0^t \tilde{X}\, dt\right) = \left(\dot{X} - \dot{X}_d\right) + 2\mu(\dot{X} - \dot{X}_d) + \mu^2(X - X_d) \tag{40}$$

So, the approximation of $\widehat{U}$ will be computed as follows:

$$\widehat{U} = -\hat{f} + \ddot{X}_d - 2\mu (\dot{X} - \dot{X}_d) + \mu^2 (X - X_d) \tag{41}$$

The sliding mode control law for robot manipulator with multi degrees of freedom can be written as follows [17, 21]:

$$\tau_{Total} = \tau_{eq} + \tau_{dis} \tag{42}$$

Dynamic model of PUMA robot manipulator as mentioned in Eq. (17) and Eq. (18) can be computed for equivalent part for the first three links of PUMA as follows:

$$\tau_{eq} = [\, A^{-1} (B + C + G) + \dot{S}\,]\, A \tag{43}$$

Where. A: is for inertia matrix. B:is for Coriolis torques matrix. C: is centrifugal torques .and, G: gravity torques.

$$\tau_{dis} = K\, S_{Sign} \tag{44}$$

$$\tau_{Total} = \tau_{eq} + K\, S_{Sign} \tag{45}$$

$$\tau_{Total} = [\, A^{-1} (B + C + G) + \dot{S}\,]\, A + K\, S_{Sign} \tag{46}$$

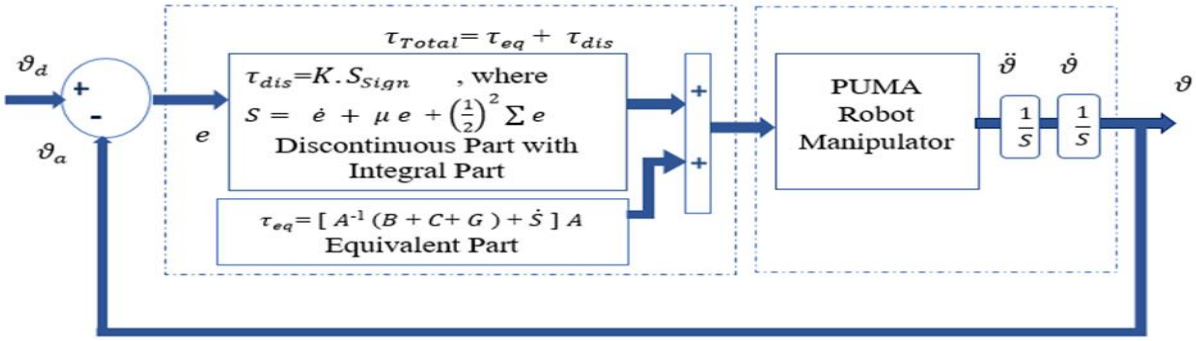


Figure. 4 Block diagram of integral sliding mode controller (ISMC) for PUMA robot manipulator

To reduce the sliding controller's steady condition error and to improve stability and minimize total error, the integral part of the sliding mode surface (ISMC) will be used, our goal is to keep the sliding surface s (x, t) close to zero all the time. The formula that define the integral sliding mode control surface is shown:

$$S_{PID} = \mu e + \dot{e} + (\tfrac{1}{2})^2 \sum e \qquad (47)$$

According to Eq. (46) and Eq. (47) two parameters (K, μ) have to be adjusted in the SMC design , if these parameters are correctly modified, the controller will reject external disturbances affecting the tracking trajectory and increase the torque of the robot joints [21, 22]. Fig. 4 shows block diagram of integral sliding mode controller for PUMA robot. Particle Swarm Optimization algorithm is the technique used in this work to pick the best values for these deterministic coefficients to achieve high performance power. This algorithm changes the gains and sets the correct values for these parameters in accordance with the system implemented.

5. Optimization with PSO algorithm

Particle swarm optimization PSO is the most efficient optimization technique. It was motivated by a basic social behavior such as fish schooling or bird flock, and found successful in solving nonlinear optimization systems with High-grade solution between all intelligent techniques and consistent convergent properties in faster computation time. PSO is a very simple algorithm for understanding and implementing, it requires less computing memory and less lines of code compared with other algorithms. As with other population-based algorithms PSO is using the initial random solutions called particles, generations of updates can create the best search-space solution. Such particles travel through the search space of the system, following the optimum particle obtained and their own experiences. The principle of this optimization is to use its best known positions of particles to converge the swarm population in the solution space to a single optimal. The PSO algorithm requires, at each point, changing the location and velocity of the particle to its P_{best} and G_{best} .For each particle, the velocity is modified iteratively by its personal best position, which is found by the particle, and also by the best position found by the particles in its vicinity as shown [23]:

$$V_{i,k}^{(t+1)} = W \cdot V_{i,k}^{(t)} + c_1 \varphi_1 (Pbest_{i,k} - X_{i,k}^{(t)}) + c_2 \varphi_2 (Gbest_{i,k} - X_{i,k}^{(t)}) \qquad (48)$$

$$X_{i,k}^{(t+1)} = X_{i,k}^{(t)} + V_{i,k}^{(t+1)} \qquad (49)$$

Where: i = is for particles number. k = for dimensions' number. X_i is for k- dimensional position vector for ($X_{i1}, X_{i2}, .., X_{in}$). V_i= velocity of the particle for (V_{i1} , V_{i1} , ….., V_{id}). $Pbest$ = best visited position for the particles. $Gbest$ = best position explored in the population. φ_1 φ_2 = random numbers between 0 and 1. W = inertia weight. c_1 c_2 = positive constants .and (t) for iteration pointer .Coefficients $c_1 and\ c_2$ include the relative weight of the probabilities that accelerate each particle in $Pbest$ and $Gbest$ position . The Small values allow particles to migrate before removal from target zones. In addition, large values result in a rapid movement to, or prior to, target areas. As, with previous research, the constant factors of acceleration. $c_1 and\ c_2$ between (1-2). The weight of inertia is a crucial factor controlling the impact of each particle in the previous velocity. Sufficient choice of inertia weight (W) will allow for a balance between global and local exploration, which enables reduced aggregate iteration in order to find a fairly optimum

solution. So, the initially enhanced W often decreases from approximately (0.9) to (0.4) sequentially over time [23, 24]

6. Simulation results

In this work, in order to analyze the performance of the proposed controller design, Matlab/Simulink framework and M-File were used to implement the design strategy of the controller. The PSO algorithm is used to inspect and optimize the parameters of the controller, by performing a basic objective fitness function. The optimization process is aimed to minimize the fitness of objective function

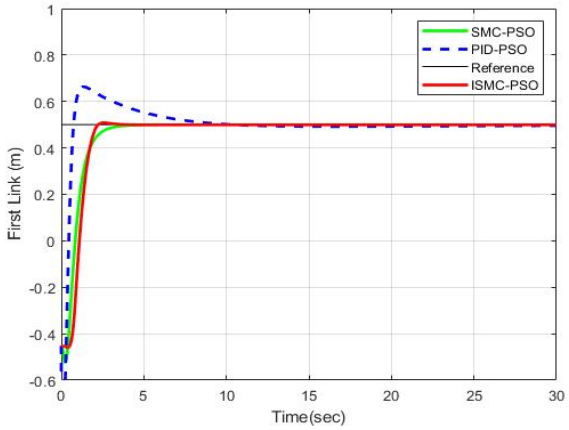
Traditional proportional integral derivative with PSO (PID/ PSO) was first introduced, then nonlinear sliding mode controller (SMC/ PSO), which is an appropriate substitute for conventional linear PID controllers, nonlinear SMC is efficient and robust in resolving system fluctuations and disturbances. The integral sliding mode controller with optimization algorithm method (ISMC /PSO) was suggested to solve the chattering problem in SMC.

The PSO algorithm will take the (X_{desire}) as a reference and check each particle in (X_{actual}) by using the mean square error between them until it reaches the best fitness function with smallest mean square error, as follows:

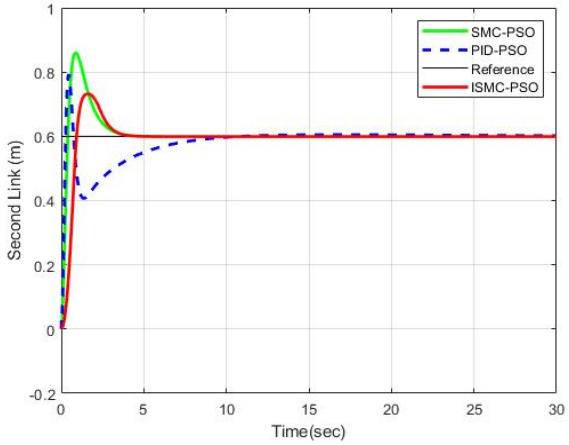
$$Mse = \frac{1}{N}\sum_{i=1}^{N}\left(X^{d} - X^{a}\right)^{2} \qquad (50)$$

Where: N: is number of random samples that were used. X^{d} : is for desire position. X^{a}: is for actual position. So, for PID there are 18 parameters for optimization. And for SMC and ISMC there are two parameters (K, μ) . PSO algorithm is initialized with the following parameters: N=20 birds, Iteration = 100, c_1= c_2=2, W= 0.5, and the k (dimension vector) for PID = 18 parameters, and for SMC = 2 parameters.

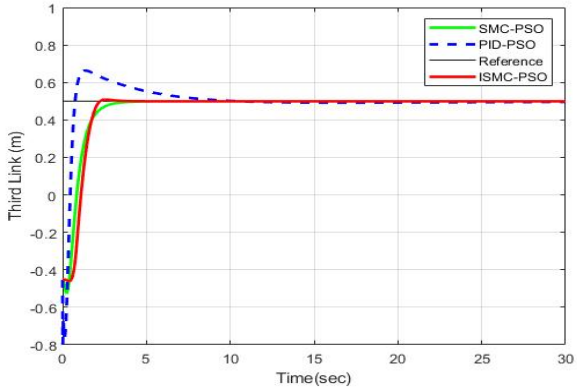
As it was mentioned, only the first three links of PUMA robot were being considered in this work. Fig. 5 (a-b-c) shows the step response of the first, second and third links for the robot, from which it can be seen that ISMC/PSO has the best response with nearly zero maximum overshoot and stable steady state, minimum Settling time, and near zero mean square error comparing with the PID/PSO ,and SMC/PSO . Fig. 6 (a-c) shows the final torque in the first, second and third links for SMC/PSO , and ISMC/PSO , it can be noticed that the torque response in ISMC/PSO has become more stable with ratio between (-10,10) N.m, besides eliminating the chattering phenomena in SMC/PSO design.



(a)



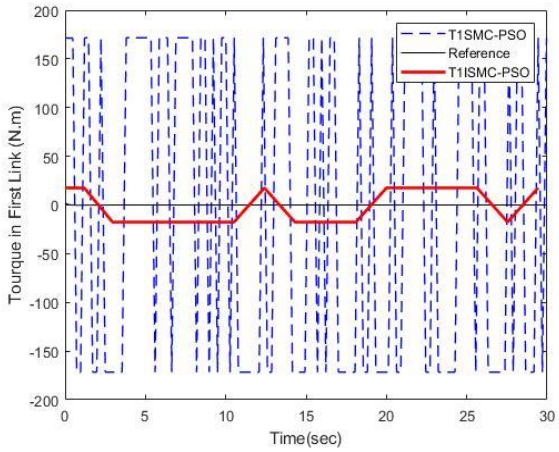
(b)



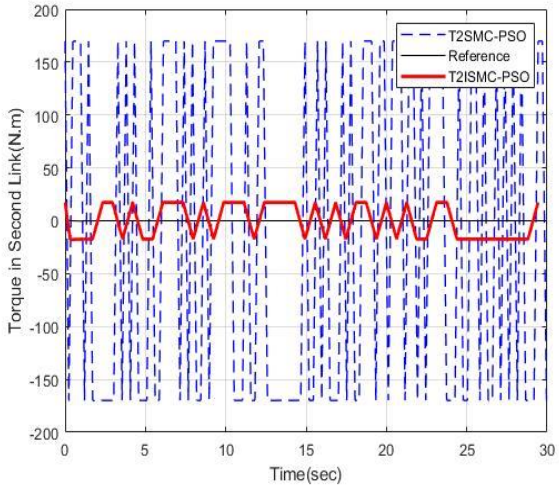
(c)

Figure. 5 Step response performance for PID/PSO, SMC/PSO, and ISMC/PSO in: (a) first link, (b) second link, and (c) third link

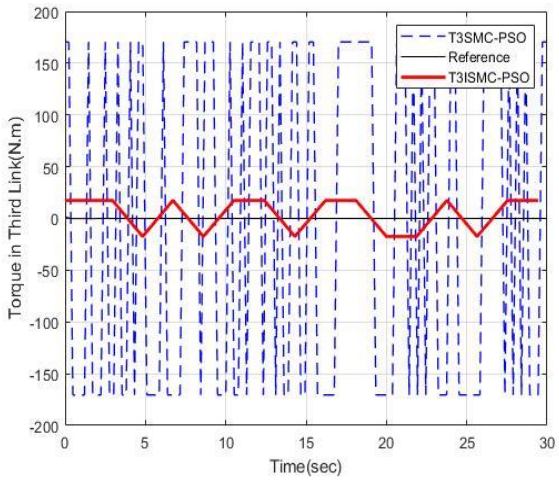
Fig. 7 (a-c) illustrate the rate of error for the three links, it can be noticed that ISMC/PSO has the minimum actual error in all links. Fig. 8 shows the iteration of PSO algorithm with mean square error and objective function, from which it can be seen that PSO successively iterates until it reaches the optimal value for parameters ,with best fitness = $4.4876\ e^{-6}$



(a)

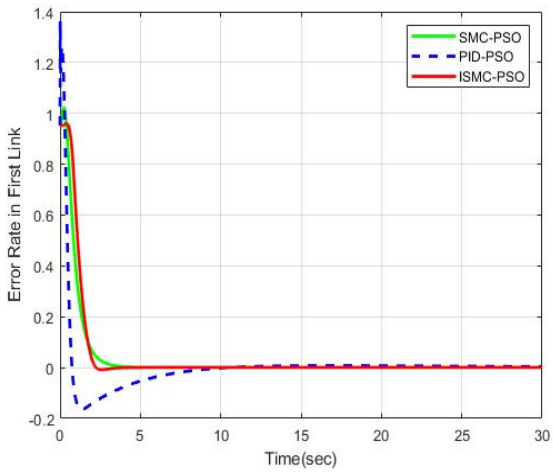


(b)

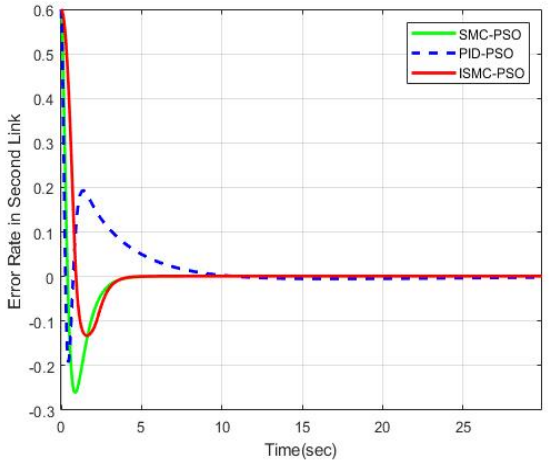


(c)

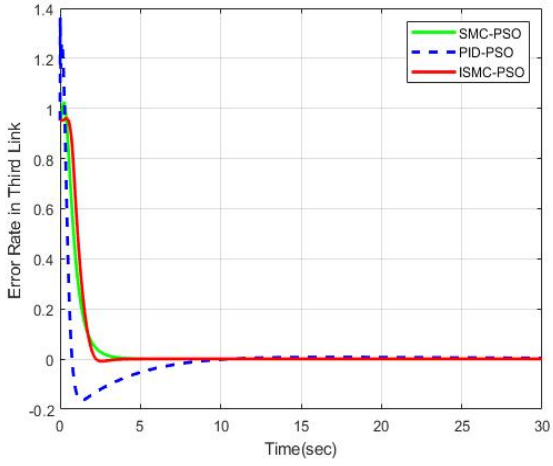
Figure. 6 Torque performance for SMC/PSO and ISMC/PSO in: (a) first link (b) second link, and (c) third link



(a)



(b)



(c)

Figure. 7 Rate of actual error for PID/PSO, SMC/PSO, and ISMC/PSO in: (a) first link and (b) second link, and (c) third link

Table 2, Table 3, and Table 4 shows a comparison between the three proposed techniques that were used for the performance criteria of maximum overshoot, rise time, settling time, mean square error, and actual error, from which it can be seen that for the first second and third links, best result was shown in

Table 2. Comparison between the three proposed technique in first link

Criteria	**PID-PSO**	**SMC-PSO**	**ISMC-PSO**
Overshoot M_p %	10.26%	1.23%	1.04%
Rise time t_r (Sec.)	0.468	0.729	1.089
Settling time t_s (Sec.)	8.079	3.405	2.551
Mean square error (Mse)	0.7043	0.0474	0.0025
Actual error	0.00394	0.00069	0.000062

Table 3. Comparison between the three proposed technique in second link

Criteria	**PID-PSO**	**SMC-PSO**	**ISMC-PSO**
Overshoot M_p %	20.33%	22.01%	17.13%
Rise time t_r (Sec.)	0.186	0.186	1.501
Settling time t_s (Sec.)	9.471	3.076	3.784
Mean square error (Mse)	0.1794	0.0105	0.0062
Actual error	0.00188	0.00279	0.000356

Table 4. Comparison between the three proposed technique in third link

Criteria	**PID-PSO**	**SMC-PSO**	**ISMC-PSO**
Overshoot M_p %	11.33%	1.29%	1.09%
Rise time t_r (Sec.)	0.458	0.605	1.294
Settling time t_s (Sec.)	8.072	3.038	2.805
Mean square error (Mse)	0.1378	0.0322	0.0042
Actual error	0.00694	0.00075	0.000225

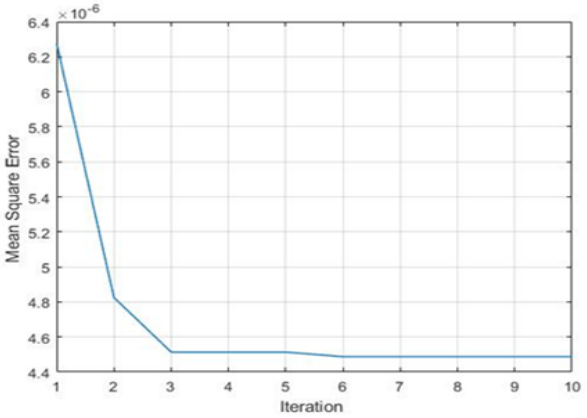


Figure. 8 Iteration and objective function of PSO algorithm

Table 5. Comparison between proposed method and existing methods

The Technique	Mean Square Error
Proposed Method ISMC/PSO	**0.0025**
AFSMC/PSO	0.0022
ETSMC	0.0027
Impedance Con. /PSO	0.0029

the first link with ISMC/PSO as: M_p = 1.04% , t_r =1.089 Sec. t_s =2.551 Sec. , M_{SE} = 0.0025,and finally the actual error = 0.000062.

In order to verify the validation of the proposed controller techniques in robustness and stability, a comparison was done with existing methods that were used to control robot manipulator in specifically. These techniques are as follows, adaptive fuzzy sliding mode controller with particle swarm for optimization (AFSMC/PSO) [3], estimated twisting sliding mode controller (ETSMC) [15], and impedance controller tuned by particle swarm optimization (Imp. Control /PSO) [14] the comparing criteria was done by using Mean Square Error. Table 5 shows the comparison of the proposed technique with existing methods. The findings demonstrate how the results of the proposed technique in mean square error equal (M_{se} = 0.0025) converge with best data that obtained from existing methods.

7. Conclusion

This paper introduces a technique for tuning three types of dynamic control strategies for PUMA 560 robot manipulator, which are Proportional Integral Derived PID, Sliding Mode Control SMC, and Integral Sliding Mode Control ISMC. To boost the parameters in the proposed strategies, Intelligent Particle Swarm Optimization PSO has been suggested to achieve the optimum parameters, this algorithm relies on analysing the system's behaviour. The parameters of each controller were improved by PSO optimization method according to the initial values of the particles such as their swarm size and initial velocity. Afterwards, the best location was determined for each particle, which gave the minimum value of the objective function that was chosen. PSO points to a strong optimization rule that eliminates inefficient particles which produce undesired performance that can collapse the stability criteria, for PID 18 parameters were improved, in SMC and ISMC two parameters were improved.

Results of simulations demonstrate the efficacy of the proposed tuning method by performing a high degree of stability, with excellent implementation of the proposed technique for PUMA robot particularly

with ISMC/PSO which gave best result in step response, steady state, overshoot, and ultimately the oscillation response of the torque to become more reliable and stable (M_p = 1.04%, t_r =1.089 Sec.t_s =2.551 Sec.),besides eliminating the disadvantage of chattering in conventional SMC and reduce the mean square error in addition to actual error (M_{SE} = 0.0025, actual error = 0.000062) which had a perfect result comparing with existing techniques.

The iteration and objective fitness function of PSO algorithm was illustrated to show the best optimum value and excellent performance of the presented algorithm, with best fitness $= 4.4876\ e^{-6}$.

Results proved that the proposed algorithm is an efficient method for optimization, and that another kind of intelligent swarm algorithm for optimization can be added to control the robot manipulator in the future. Besides, disturbance will be added to check the response of the links during noise.

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationship that could have appeared to influence the work reported in this paper

Author Contributions

Conceptualization, Hanan, and Suhad.; methodology, Hanan; software, Suhad; validation, Hanan, and Suhad; formal analysis, Hanan, and Suhad.; investigation, Hanan, and Suhad; resources, Suhad; data curation, Suhad; writing—original draft preparation, Suhad; writing—review and editing, Suhad; visualization, Hanan and Suhad; supervision, Hanan; project administration, Hanan.

Acknowledgments

Acknowledgments are to show that the article is supported by what organization. For example, “This work was supported by the National Nature Science Foundation under Grant No. 405”.

References

[1] M. Spong and M. Vidyasagar, Robot Dynamics and Control, John Wileys & Sons, United State of America, 2008.

[2] C. S. G. Lee and M. Ziegler, “Geometric Approach in Solving Inverse Kinematics of Puma Robots”, *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 20, No. 6, pp. 695-706, 1984.

[3] A. Jalali, F. Piltan, A. Gavahian, M. Jalali, and M. Adibi, “Model-Free Adaptive Fuzzy Sliding Mode Controller Optimized by Particle Swarm for Robot Manipulator”, *International Journal Information Engineering Electronic Business*, Vol. 5, No. 1, pp. 68–78, 2013.

[4] J. Angeles, *Fundamentals of Robotic Mechanical Systems*, Springer, International Publishing Switzerland, 2002.

[5] B. Armstrong, O. Khatib, and J. Burdick, “The Explicit Dynamic Model and Inertial Parameters of the PUMA 560 Arms”, In: *Proc. of IEEE International Conf. on Robotics and Automation*. Vol. 3, pp. 510-518, 1986.

[6] S. A. Mazhari and S. Kumar, “PUMA 560 Optimal Trajectory Control using Genetic Algorithm, Simulated Annealing and Generalized Pattern Search Techniques”, *World Academy of Science, Engineering and Technology*, Vol. 17, pp. 800–809, 2008.

[7] A. Oumar, R. Chakib, M. Labbadi, and M. Cherkaoui, “Robust Nonlinear Controller of the Speed for Double Star Induction Machine in the Presence of a Sensor Fault”, *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 3, pp. 124–133, 2020.

[8] S. Mahfoudhi, M. Khodja, and F. Mahroogi, “A Second-Order Sliding Mode Controller Tuning Employing Particle Swarm Optimization”, *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 3, pp. 212–221, 2020.

[9] O. Boundjou, X. Xu, and R. Ordonez, “Automated Particle Swarm Optimization Based PID Tuning for Control of Robotic Arm”, In: *Proc. of IEEE National Aerospace and Electronics Conf. (NAECON) and Ohio Innovation Summit (OIS)*, pp. 164–169, 2016.

[10] S. M. Swadi, A. I. Majeed, M. A. shuriji, and A. A. Okla, “Design and Simulation of Robotic Arm PD Controller Based on PSO”, *University of Thi-Qar Journal for Engineering Sciences*, Vol. 10, No. 1, pp. 18-24,2019.

[11] A. Zidan, S. Tappe, and T. Ortmaier, “A Comparative Study on the Performance of MOPSO and MOCS as Auto-Tuning Methods of PID Controllers for Robot Manipulators”, In: *Proc. of 15th International Conf. on Informatics in Control, Automation and Robotics(ICINCO)*, Vol. 1, pp. 240–247, 2018.

[12] M. H. Barhaghtalab, V. Meigoli, M. Haghighi, S. A. Nayeri, and A. Ebrahimi, “Dynamic Analysis, Simulation, and Control of a 6-DOF IRB-120 Robot Manipulator Using Sliding Mode Control and Boundary Layer Method”, *Journal of*

Central South University, Vol. 25, No. 9, pp. 2219–2244, 2018.

[13] M. Kuang, P. Ouyang, W. Yue, and H. Kang, "Nonlinear PD Terminal Sliding Mode Control for PUMA Robotic Manipulator", In: *Proc. of 14th IEEE Conf. on Industrial Electronics and Applications, (ICIEA)*, pp. 250–255, 2019.

[14] H. Mehdi and O. Boubaker, "Impedance Controller Tuned by Particle Swarm Optimization for Robotic Arms", *International Journal of Advanced Robotic Systems*, Vol. 8, No. 5, pp. 93-103, 2011.

[15] S. T. Haghighi, F. Piltan, J. Kim, "Robust composite high-order super-twisting sliding mode control of robot manipulators", *Robotics*, Vol. 7, No. 1, 2018.

[16] 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 Robotics and Automation*, Vol. 3, No. 3, pp. 106-150, 2012.

[17] S. Yadegar and A. Soh, "Design Stable Robust Intelligent Nonlinear Controller for 6- DOF Serial Links Robot Manipulator", *International Journal of Intelligent System and Applications*, Vol. 6, No. 8, pp. 19 –38, 2014.

[18] T. Mei, Y. Yang, J. Chen, Z. Guihong, Z. Jiao, L. Gao, X. Ren, Q. Li, "Simulation Research on Motion Trajectory of PUMA 560 Manipulator Based on MATLAB", In: *Proc. of 31st IEEE Chinese Control Decision Conf. (CCDC)*, pp. 4857–4862, 2019.

[19] P. Corke, *Robotics, Vision and Control Fundamental Algorithms in MATLAB*, Second Completely Revised, Springer Tracts in Advanced Robotics, Vol. 118, 2017.

[20] T. Dewi, P. Risma, Y. Oktarina, L. Prasetyani, and Z. Mulya, "The Kinematics and Dynamics Motion Analysis of a Spherical Robot", In: *Proc. of IEEE 6th International Conf. on Electrical Eng. Computer Science and Informatics, (EECSI)*, Indonesia, pp. 101–105, 2019.

[21] C. Soon, R. Ghazali, H. I. Jaafar, and S. Hussien, "Sliding Mode Controller Design with Optimized PID Sliding Surface Using Particle Swarm Algorithm", *Procedia Computer Science, Elsevier*, Vol. 105, pp. 235–239, 2017.

[22] F. E. Lamzouri, E. Boufounas, A. Brahmi, and A. Amrani, "Optimized TSMC Control Based MPPT for PV System under Variable Atmospheric Conditions Using PSO Algorithm", *Procedia Computer Science, Elsevier*, Vol. 170, pp. 887–892, 2020.

[23] F. Hasan, L. Rashad, and A. Humod, "Integrating Particle Swarm Optimization and Routh-Hurwitz's Theory for Controlling Grid-Connected LCL-Filter Converter", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 4, pp. 102-113, 2020.

[24] C. Ahmed, M. Cherkaoui, and M. Mokhlis, "PSO-SMC Controller Based GMPPT Technique for Photovoltaic Panel Under Partial Shading Effect", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 2, pp. 307–316, 2020.