Optimal PID Controller Design for 3-DOF Robot manipulator

Dr. Suhad Qasim G. Haddad[1]
suhad.qasim@muc.edu.iq

Ass.Prof. Enass H. Flaieh[2]
Enass.h.flaieh@uotechnology.edu.iq

Prof.Dr. Hanan A.R. Akkar[3]
Hanan.a. akkar@ uotechnology.edu.iq

Abstract: The most difficult task for a robotics researcher is to control the robot manipulator with adequate overall efficiency. This work presents accurate and fast techniques for optimizing the parameters of a 3-DOF robot manipulator, this kind of robot is intended for academic and educational work and it is also used in the medical field. The links of the 3- DOF are all rotary or revolute joints with serial connections.

The optimization technique focuses on the Intelligent Swarm algorithms that are used for optimizing and tuning the gain of the proposed controller, which is the conventional Proportional Integral Derivative (PID) is used.

The Intelligent Particle Swarm Optimization (PSO) and Social Spider Optimization (SSO) have been used for obtaining the best values for the parameters of the controller, to achieve consistency, stability, and robustness. MATLAB/Simulink is used to speed up the computing of FK, IK, and DM of the 3-DOF robot manipulator. The results show that modified PID /SSO gives the best results in terms of stable phase response, minimum Root Means Square Error with the best objective function, stable control signal and reasonable steady-state with perfect trajectory tracking for circular path.

Keywords: PID Controller, PSO Algorithm, SSO Algorithm, 3- DOF Robot Manipulator, Kinematics and Dynamic of Robot manipulator.

1. Introduction

This work presents accurate and fast techniques for optimizing the parameters of a 3-DOF robot manipulator. Proportional-integral-derivative (PID) controllers are

[1] Ph D. Al-Mansour University College, Director of the Laboratories Department, Baghdad, Iraq
[2] Associate Professor, University of Technology, Artificial Intelligence Engineering College.
[3] Professor, University of Technology, Electrical Engineering College, Baghdad, Iraq

widely used in robotics due to their simplicity, robustness, and ease of implementation. However, traditional PID controllers often struggle with real-time adaptation in dynamic environments, leading to suboptimal performance in robotic applications. To address this limitation, intelligent optimization algorithms have been increasingly integrated into PID tuning strategies, allowing for improved adaptability and control precision. Among these algorithms, Particle Swarm Optimization (PSO) and Social Spider Optimization (SSO) have demonstrated significant potential in fine-tuning PID parameters dynamically, ensuring optimal system response under varying conditions.

This research focuses on developing an adaptive PID control framework for robotic applications, leveraging PSO and SSO for real-time parameter tuning. By combining these nature-inspired algorithms, we aim to enhance the controller's ability to adapt to changing environments, reducing overshoot, steady-state error, and response time. The proposed approach is validated through simulations and experimental evaluations, demonstrating its efficacy in improving robotic control performance.

2. Related Work

Several studies have explored optimization-based approaches for PID tuning in robotics. Traditional methods such as Ziegler-Nichols and Cohen-Coon provide static tuning solutions but lack adaptability in real-time applications [1]. To overcome this limitation, researchers have employed heuristic and metaheuristic optimization techniques to optimize PID gains dynamically. integrated with PID control in robotic applications [2]. Particle Swarm Optimization (PSO) has been extensively applied in control systems due to its ability to efficiently search large solution spaces. PSO-based PID tuning has been used in mobile robotics [3], and industrial automation, showing improved control performance compared to classical methods. Social Spider Optimization (SSO) is a relatively newer approach inspired by the cooperative behavior of social spiders. Studies have shown that SSO can outperform traditional optimization methods in complex search spaces [4]. Its application in PID tuning has demonstrated promising results in reducing control errors and enhancing system stability [5]. Hybrid approaches combining multiple optimization techniques have also been explored. For instance, a combination of Genetic Algorithms (GA) and PSO has been proposed for self-tuning controllers [6], while Artificial Bee Colony (ABC) algorithms have been Dynamic and kinematic formulation of 3-DOF articulated robotic manipulator [7]. Intelligent PID Controller for Vibration Suppression of Horizontal Flexible Plate Based on Social Spider Optimization: This study presents an intelligent PID controller optimized using the Social Spider Optimization (SSO) algorithm to suppress vibrations in a horizontal flexible plate. The research demonstrates that the SSO-optimized PID controller effectively reduces vibration amplitudes, leading to improved system stability and performance. (8).Speed Control of Wheeled Mobile Robot by Nature-Inspired Social Spider Optimization Algorithm: The paper proposes using the SSO algorithm to tune PID controllers for the speed control of wheeled mobile robots. The results indicate that the SSO-optimized PID controller achieves precise speed control, even in the presence of disturbances and uncertainties, showcasing the potential of SSO in robotic applications. (9)

PID Controller with Novel PSO Applied to a Joint of a Robotic Manipulator: This research presents a novel PSO algorithm applied to the speed control of a robotic manipulator joint driven by a three-phase induction motor. The study demonstrates that the PSO-optimized PID controller effectively improves the dynamic response and precision of the manipulator joint (10)

Despite these advancements, limited research has focused on integrating PSO and SSO for adaptive PID control in robotics. This study aims to bridge this gap by developing a hybrid optimization framework that leverages the strengths of both algorithms for enhanced real-time adaptability.

3. Kinematics and Dynamic of Robot manipulator:

The paramount component of the robotic manipulator is the precise ultimate position of the end effector, devoid of any disturbances that could affect the robot's overall performance. The study of robot manipulators can be categorized into two primary components: kinematics and dynamics. Kinematics pertains to the relationships among the locations, velocities, and accelerations associated with the movements of a robotic arm. The objective of kinematics is to explain the position of a frame in relation to its initial coordinates. Dynamic simulation defines the relationship among joint motion, velocity, acceleration, torque, and electrical parameters such as current or voltage. It may also be used to describe specific dynamic qualities pertinent to system behavior, including inertia, Coriolis effects, centrifugal forces, and other associated factors [11]. Figure (1) illustrates the general configuration of the 3-DOF robotic arm manipulator in spherical coordinates, featuring several joints and their corresponding linkages. This robot will serve as the case example for this effort.

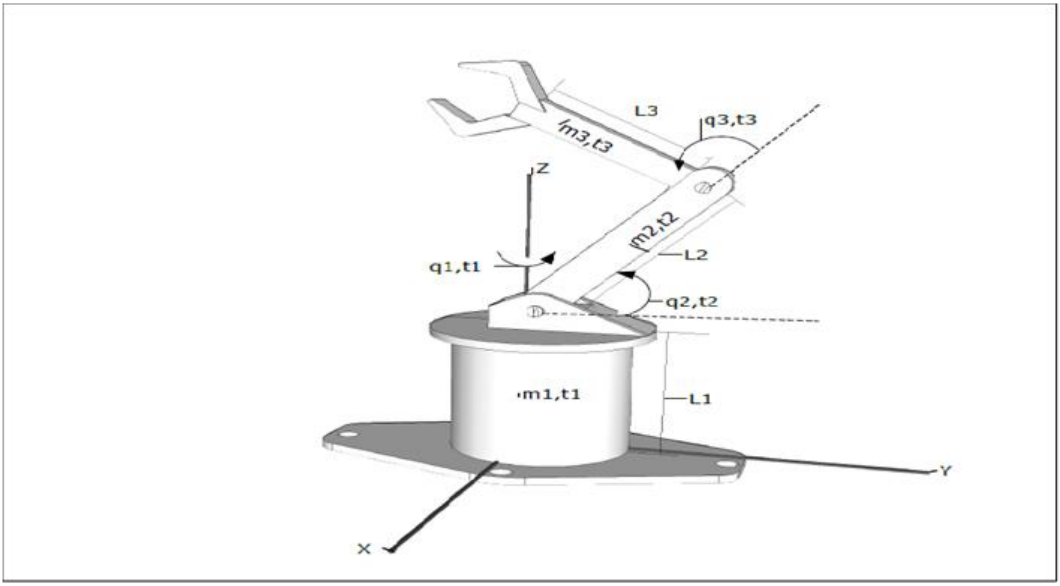


Figure (1): The three-DOF robot in spherical organize [11]

3.1 Forward Kinematic:

Typically, the forward kinematics (FK) of a robotic manipulator can be computed in four phases, as outlined below [12]:

1. Examine the descriptions of the link.
2. Identify the parameters of the Denavit-Hartenberg (D-H) convention.
3. Attachment of frames.
4. Calculated FK.

The Denavit-Hartenberg (D-H) conference is to be located next. This framework was introduced by Jacques Denavit and Richards S. Hartenberg. This style is employed to select the reference frame for the robotic arm and to evaluate the forward kinematics of the robot manipulator based on the relationship between the joints. According to this convention, the alignment of each frame between two joints will establish the position of each joint in relation to its preceding joint, as shown in table (1) [12]. The third component involves computing the matrix of the frame connection. The rotation matrix ${}^{i-1}_{\ \ i}R$, which delineates the orientation of link i relative to link i-1, is derived by multiplying two matrices: the rotation of angle α_(i-1) about the X-axis and the rotation of angle θ_i about the Z-axis, [12]. Consequently:

$$ {}^{i-1}_{\ \ i}R \ = [\ \ (X_{i-1}, \alpha_{i-1})]\,[R\ (Z_i, \theta_i)] \qquad \ldots\ (1) $$

Where:

$$X_{i(\alpha i-1)}=\begin{bmatrix}1 & 0 & 0\\ 0 & C\alpha_{i-1} & -S\alpha_{i-1}\\ 0 & S\alpha_{i-1} & C\alpha_{i-1}\end{bmatrix} \qquad \dots (2)$$

$$Z_{i(\theta i)} \quad =\begin{bmatrix}C\theta_i & -S\theta_i & 0\\ S\theta_i & C\theta_i & 0\\ 0 & o & 1\end{bmatrix} \qquad \dots (3)$$

So, the final matrix for the particular link will be as follow:

$$ {}^{i-1}_{\ \ i}T=\begin{bmatrix}C\theta_i & -S\theta_i & 0 & \alpha_{i-1}\\ S\theta_i\, C\alpha_{i-1} & C\theta_i C\alpha_{i-1} & -S\alpha_{i-1} & -S\alpha_{i-1}d_i\\ S\theta_i S\alpha_{i-1} & C\theta_i S\alpha_{i-1} & C\alpha_{i-1} & C\alpha_{i-1}d_i\\ 0 & 0 & 0 & 1\end{bmatrix} \qquad \dots (4)$$

Where, $C\theta_i$, $S\theta_i$ shorthand of cos θ, sin θ respectively.

The next step is the estimation of total FK, the final 4x4 homogeneous transformation for the link (i) in relation to the link $(i-1)$. For example, FK of the end effector (i), in relation to the base position $(i-1)$ will be calculated by multiplying all of ${}^{i-1}_{\ \ i}T$ matrices as follows [12]:

$$ {}^{\ \ \ \ base}_{end-effector}T={}^{i-1}_{\ \ i}T={}^0_1T\;{}^1_2T\;{}^2_3T \qquad \dots (5)$$

Figure (2)depicts the three-dimensional location chart of the robotic manipulator with three degrees of freedom that will serve as the case model for this investigation.[13].

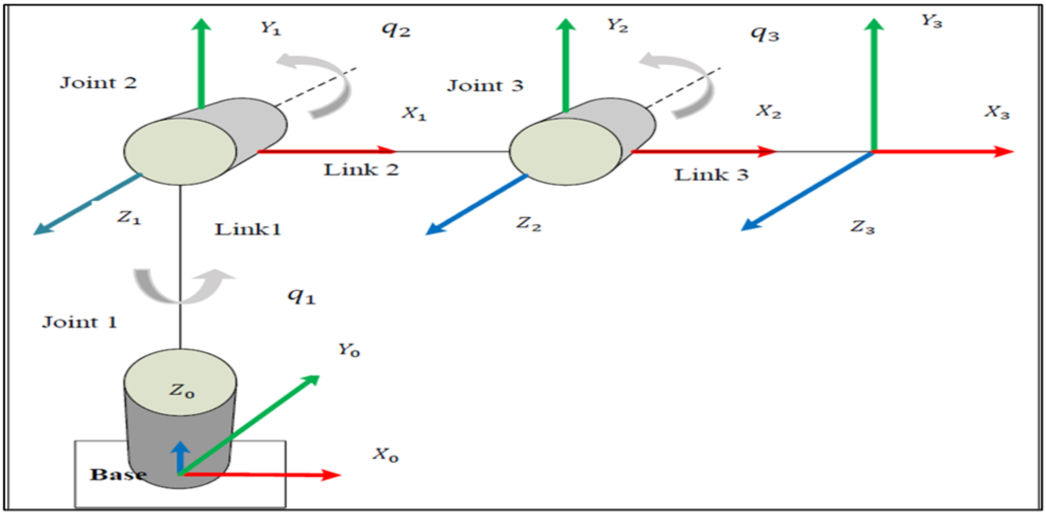


Figure (2): Joint frames of the 3-DOF robotic articulated support [13]

Table (1): D-H for 3-DOF robot manipulator arm [13]

Joint i	θ_i (rad)	α_{i-1}(red)	a_{i-1}(m)	d_i(m)
1	θ_1	$^{\pi}/_{2}$	0	L_1
2	θ_2	0	L_2	0
3	θ_3	0	L_3	0

According to the D-H convention of the 3-DOF articulated robot manipulator and based on Equation (4), the general homogeneous transformation matrix for the location and orientation of the end effector that is needed will be defined as:

$$ {}_{0}^{3}T = \begin{bmatrix} C\theta_1 C\theta_{23} & -C\theta_1 S\theta_{23} & S\theta_1 & L_3 C\theta_1 C\theta_{23} + L_2 C\theta_1 C\theta_2 \\ S\theta_1 C\theta_{23} & -S\theta_1 S\theta_{23} & -C\theta_1 & L_3 S\theta_1 C\theta_{23} + L_2 S\theta_1 C\theta_2 \\ S\theta_{23} & C\theta_{23} & 0 & L_3 S\theta_{23} + L_2 S\theta_2 + L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \ldots (6) $$

The homogeneous transformation matrix specified in Equation (6) delineates the forward kinematics of the 3-DOF robotic manipulator illustrated in Figure (1). The location and orientation of end effectors are non-linear functions of joint variables derived from this matrix. Upon deriving the forward kinematics of Figure (1), it is feasible to determine the position and orientation of the end-effector based on the individual joint angles (q1, q2, and q3). The orientation matrix of the end-effector is denoted by the initial three rows and columns of the transformation matrix in Equation (7) [13,14], as follows:

$$ \left\{ \begin{aligned} P_x &= L_3 C\theta_1 C\theta_{23} + L_2 C\theta_1 C\theta_2 \\ P_y &= L_3 S\theta_1 C\theta_{23} + L_2 S\theta_1 C\theta_2 \\ P_z &= L_3 S\theta_{23} + L_2 S\theta_2 + L_1 \end{aligned} \right\} $$

$$ R_d = \begin{bmatrix} C\theta_1 C\theta_{23} & -C\theta_1 S\theta_{23} & S\theta_1 \\ S\theta_1 C\theta_{23} & -S\theta_1 S\theta_{23} & -C\theta_1 \\ S\theta_{23} & C\theta_{23} & 0 \end{bmatrix} \quad \ldots (7) $$

Where:

$C\theta_i = Cos\,\theta_i$; $S\theta_i = Sin\,\theta_i$; $C\theta_{ij} = Cos(\theta_i + \theta_j)$; $S\theta_{ij} = Sin(\theta_i + \theta_j)$.

3.2 Inverse Kinematic IK

The inverse kinematics problem is employed to ascertain the sequence of necessary joint angles to achieve a specific ideal end-effector position. Determining the inverse kinematics solution for the entire manipulator can be a genuinely tough endeavor, as the inverse kinematics are nonlinear solutions. the algebraic approach is use in this work where with some operation related to the laws of (sin θ , and cosθ), the joints angles will be considered. the final Equations for the three angles are shown below [15]:

$$\theta_1 = 2\, tan^{-1}\left[\frac{\left(-x \pm \sqrt{x^2+y^2}\right)}{y}\right] \qquad \ldots (8)$$

$$\theta_3 = 2\, tan^{-1}\left[\frac{\left(\sqrt{L_3^2-k^2}\right)}{k+L_3}\right] \qquad \ldots (9)$$

$$\theta_2 = 2\, tan^{-1}\left[\frac{\left(-L_2-L_3C\theta_3 \pm \sqrt{L_2^2+L_3^2+2L_2L_3C\theta_3-z^2}\right)}{z-L_3S\theta_3}\right] \qquad \ldots (10)$$

$$k = \frac{1}{2L_2}\left[x^2+y^2+z^2-L_2^2-L_3^2\right] \qquad \ldots (11)$$

Where:

1) θ_1 Is defined only when $\sqrt{x^2+y^2} > 0$.

2) θ_2 is between [π to $-\pi$].

The value of all the physical practical limits constraints of the manipulator is itemized in Table (2).

Table (2): List of the constraints for the 3-DOF robot [13]

Parameters	Symbol	Value
Length of the first relation	L_1	0.15 m
Length of the second relation	L_2	0.5 m
Length of the third relation	L_3	0.5 m
Mass of the first relation	m_1	0.5 kg
Mass of the second relation	m_2	0.5 kg
Mass of the third relation	m_3	0.5 kg
Unit of gravity	g	9.81 m / s^2

3.3 Dynamic Model DM:

The dynamic robot model DM is utilized to characterize the robot's performance as either linear or non-linear. The dynamics of an n-link robotic manipulator are characterized by a set of highly nonlinear second-order differential equations. The dynamic manipulator motion equation defines the system reaction and dynamic behavior of the manipulator, encompassing mechanical force, torque, position, velocity, and acceleration. System stability is a primary objective and a significant area of focus in medical applications, evaluated by dynamic equations.

Figure (3) shows the block diagram for the dynamic and kinematics model that was designed in this work for controlling the 3-DOF robot manipulator.

For 3- degree of freedom (3-DOF) robot manipulator the dynamic Equations in a vector format, are obtained as follows [14,16]:

$$A\,(\,q\,)\,\ddot{q} + N\,(q\,\dot{q}) = \tau \qquad \ldots (12)$$

DM equation for the robot schemer as a result can be printed as follows:

$$N\,(q\,\dot{q}) = V\,(q\;\;\dot{q}) + g\,(q) \qquad \ldots\ (13)$$

$$V(q\;\;\dot{q}) = B\;(q)\,[\dot{q}\;\dot{q}] + E(q)\,[\dot{q}]^2 \qquad \ldots\ (14)$$

$$\tau = \;A\;(q)\;\ddot{q} + B(q)\;[\dot{q}\,,\;\dot{q}] + E(q)\;[\dot{q}]^2 + g\;(q) \qquad \ldots\ (15)$$

Where:

A (q): Symmetric the positive matrix, which is considered for kinetic energy and inertia matrix, with $[n * n]$ dimension. N(q) : is for the term of the nonlinearity vector. B (q): is for Coriolis torques matrix, with $[n * n(n-1)/2\,]$ dimension. E (q) : is for centrifugal torques, with $[n * n]$ dimension. g (q): is for gravity torques, with $[n * 1]$ dimension. q: is the joint position (or joint angle), for q = [q_1 , q_2 , ... q_n] . $\dot{q}$: is considered as n- vector for joint velocities. $\ddot{q}$: is considered as n- vector of accelerations. τ: is considered as the joint force vector (torque). $[\dot{q}]^2$: can be a vector given by [$\dot{q}_1^2$, $\dot{q}_2^2$ …, $\dot{q}_n^2$]^T. [$\dot{q}$ $\dot{q}$]: can be a vector given by [$\dot{q}_1$ $\dot{q}_2$, $\dot{q}_1$ $\dot{q}_3$.. $\dot{q}_1\dot{q}_n$, $\dot{q}_2\dot{q}_3$,$\dot{q}_{n-2}\dot{q}_n$]^T. According to the basic information all the functions are derived as follows:

$$outputs = function\;(inputs) \qquad \ldots (16)$$

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 inferred, as follows:

$$q = \text{function}\;(\tau) \qquad \ldots (17)$$

$$\ddot{q} = A^{-1}(q).\{\tau - N\;(q,\dot{q})\} \qquad \ldots (18)$$

$$q = \iint A^{-1}(q).\{\tau - N\;(q,\dot{q})\} \qquad \ldots (19)$$

So, the DM formulation for the 3-DOF robotic schemer with serial links robot schemer that is shown in Figure (3), is calculated as follows:

$$A\;(\ddot{q})\begin{bmatrix}\ddot{q}_1\\ \ddot{q}_2\\ \ddot{q}_3\end{bmatrix} + B\;(q)\begin{bmatrix}\dot{q}_1 & \dot{q}_2\\ \dot{q}_1 & \dot{q}_3\\ \dot{q}_2 & \dot{q}_3\end{bmatrix} + E(q)\begin{bmatrix}\dot{q}_1^2\\ \dot{q}_2^2\\ \dot{q}_3^2\end{bmatrix} + \;g\;(q) = \begin{bmatrix}\tau_1\\ \tau_2\\ \tau_3\end{bmatrix} \qquad \ldots (20)$$

The Lagrange-Euler approach is simpler than the Newton-Euler method for calculating the dynamic equation of a robot manipulator because it is based on nonlinear and quadratic differential equations that provide an energy difference between the kinetic energy and potential.

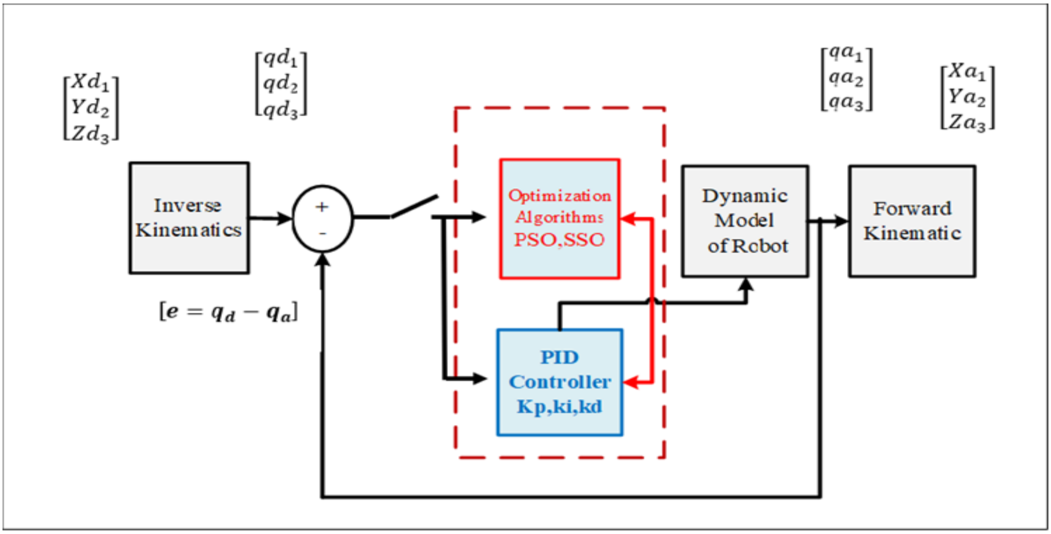

$\begin{bmatrix} Xd_1 \\ Yd_2 \\ Zd_3 \end{bmatrix}$
$\begin{bmatrix} qd_1 \\ qd_2 \\ qd_3 \end{bmatrix}$
$\begin{bmatrix} qa_1 \\ qa_2 \\ qa_3 \end{bmatrix}$
$\begin{bmatrix} Xa_1 \\ Ya_2 \\ Za_3 \end{bmatrix}$
Inverse Kinematics
+
-
$[e = q_d - q_a]$
Optimization Algorithms PSO,SSO
PID Controller Kp,ki,kd
Dynamic Model of Robot
Forward Kinematic

Figure (3): Block diagram of dynamic and kinematics model for the 3-DOF robot

Each link in the 3-DOF robot manipulator is a rigid body, and it may be assumed that the mass of each link is concentrated at the midpoint of its respective center. The parameters of all physical practical limits of the manipulator are presented in Table 2. The Euler-Varangian equations are derived for the three torques (τ_1, τ_2, τ_3), are as follows [13]:

$$\boldsymbol{\tau_1} = \left(\frac{1}{2} m_1 r_1^2 + \frac{1}{4} m_2 l_2^2 C_{23}^2 + \frac{1}{4} m_3 l_3^2 C_{23}^2 + m_3 l_2^2 C_2^2 + m_2 l_2 l_{23} C_2 C_{23}\right) \ddot{q}_1 + \left(-\frac{1}{4} m_2 l_2^2 \sin(2q_2) - m_3 l_2^2 \sin(2q_2) - m_3 l_2 l_3 \sin(2q_2 + q_3) - m_3 l_{c2}^2 \sin\left(2(q_2 + q_3) - m_3 l_{c2}^2 \sin\left(2(q_2 + q_3)\right)\right) \dot{q}_1 \dot{q}_2 + \left(-m_3 l_2 l_3 C_2 S_{23} - m_3 l_{c3}^2 \sin\left(2(q_2 + q_3)\right)\right) \dot{q}_1 \dot{q}_3 \quad \ldots (21)$$

$$\boldsymbol{\tau_2} = \left(\frac{1}{4} m_2 l_2^2 + m_3 (l_2^2 + l_{c3}^2 + l_2 l_3 C_3)\right) \ddot{q}_2 - m_2 l_2 l_3 S_3 \dot{q}_2 \dot{q}_3 + m_3 (l_2 l_{c3} C_3 + l_{c3}^2) \ddot{q}_3 - m_3 l_2 l_{c3} S_3 \dot{q}_3 + \left(\frac{1}{2} m_2 l_{c2}^2 \sin(2q_2) + m_3 \frac{l_2^2}{2} \sin(2q_2) + l_2 l_{c3} m_3 \sin(2q_2 + q_3) + \frac{1}{2} l_{c3}^2 sin2(q_2 + q_3)\right) \dot{q}_1^2 + (l_{c2} m_2 g C_2 + l_2 m_3 g C_2 + l_{c3} m_2 g C_{23}) \quad \ldots (22)$$

$$\boldsymbol{\tau_3} = \frac{1}{3} m_3 l_3^2 \ddot{q}_3 - m_3 (l_2 l_{c3} C_3 + l_{c3}^2) \ddot{q}_2 + m_3 l_{c3} S_{23} (l_2 C_2 + l_{c3} C_{23}) \dot{q}_2^2 + m_3 (l_2 l_{c3} S_3 \dot{q}_2 \dot{q}_3 + (g l_{c3} C_{23}) \quad \ldots (23)$$

4. General Model of the System with Actuators:

How quickly a robot can do a task is determined by the actuators and the loads that each joint is able to handle. The matching circuit for the PMDC Permanent Magnet DC Motor is shown in Figure (4).

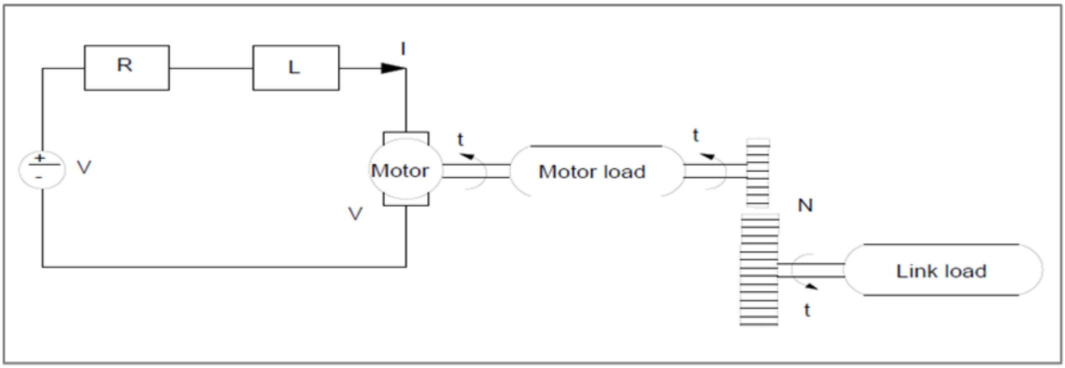


Figure (4): Circuit of the DC motor [17]

The PMDC motor is utilized because to its superior performance, strength, and adjustable speed control. The mathematical equation representing the electrical and mechanical dynamics of a permanent magnet DC motor is derived using Kirchhoff's voltage law, applied to the armature windings, resulting in [36]:

$$V_a = Ri_a + L\frac{di_a}{dt} + K_b\frac{dq_m}{dt} \qquad \dots (24)$$

Where:

V_a : is the armature voltage of the motor. R and L: are armature equivalent resistance and inductance, respectively. K_b : is the back electromotive force constant. i_a : is the armature current. $\frac{dq_m}{dt}$: denotes the angular position of the motor.

$$E_b = K_b\frac{dq_m}{dt} \qquad \dots (25)$$

The dynamics of the 3-DOF robotic manipulator, as described by Euler-Lagrange's equation, are represented by three coupled nonlinear differential equations for motion, as indicated in Equation (20)20. It is important to note that the armature voltage of the DC motor serves as the control input signal for operating the 3-DOF robot manipulator. Consequently, owing to the minimal inductance L, it may be disregarded. Table (3), illustrated the mechanical and electrical characteristics of the motor [17]. Finally, for the 3-DOF robotic arm, the complete nonlinear DM is represented as follows:

$$\begin{bmatrix}\ddot{q}_1\\ \ddot{q}_2\\ \ddot{q}_3\end{bmatrix}_{3x1} =[AI]_{3x3}\left[\begin{bmatrix}Va_1\\ Va_2\\ Va_3\end{bmatrix}_{3x1} - B(q)_{3x3}\begin{bmatrix}\dot{q}_1 & \dot{q}_2\\ \dot{q}_1 & \dot{q}_3\\ \dot{q}_2 & \dot{q}_3\end{bmatrix}_{3x1} - E(q)_{3x3}\begin{bmatrix}\dot{q}_1^2\\ \dot{q}_2^2\\ \dot{q}_3^2\end{bmatrix}_{3x1} - G(q)_{3x1}\right]$$
.. (26

Table (3): List of the main characteristics of the PMDC motor [17]

Description of parameters	Value and units
The Motor inertia J_m	$43 * 10^{-4}$ Kg.m^2
The torque constant of the Motor K_m	1.73 Nm A^{-1}
The voltage Constant of the Motor K_b	1.06 V rad^{-1}
The Winding of the Resistance R_a	1.88 Ω
The Winding of the inductance L_a	$15.5 * 10^{-3}$ H
Continuous stall torque T_s	12 N m
Theoretical maximum torque T_p	43.5 N m
Speed reducer reduction ratio N	119
Speed reducer efficiency γ	0.8

5. Proportional Integral Derived PID Controller:

PID controllers are highly effective for linear systems and are used widely in industrial applications and robotic controllers. Nonetheless, precise PID tuning is difficult because most systems are very complex and have certain challenges, such as non-linearity, and time delay in response. So, this approach has some drawbacks when the robot manipulator deals with different environments and has complexity in DM [18].

Modification of PID parameters is the most important component of the PID controller design, which is a crucial optimization. The PID controller can be expressed as follows [18]:

$$u_{pid}\,(t) = kp\; e\,(t) + ki\int e(t)dt + kd\;\dot{e}(t) \qquad \ldots (27)$$

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. $e\,(t)$: is the variance between desired error and real position error, which is regarded as the final location of the effector of the robot arm. The PID controller constants (p , ki and kd) can be described as: (kp): is the proportional gain and gives control action commensurate with the error signal e (t). (ki): The purpose of the integral term in the PID controller is to reduce the steady-state error by integrating the error signal e (t) continuously. (d) : 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 [18].

Figure (5) shows the global design of the PID controller. The tuning of the PID equation can be done by minimizing the Root Mean Square Error $RMSE$ between the desired position (X_d) and actual position (X_a), as follows:

$$e\ (t) = X_d(t) - X_a(t) \quad \ldots\ (28)$$

The tuning of the PID controller, as delineated by Equation (27), can be achieved by minimizing the root-mean-square error between the planned joint position and the actual joint position. To establish the upper and lower thresholds for the signal control of the DC motor connected to the joints of the articulated robot arm, a saturation limit has been implemented prior to DM being used to regulate the torque limits.

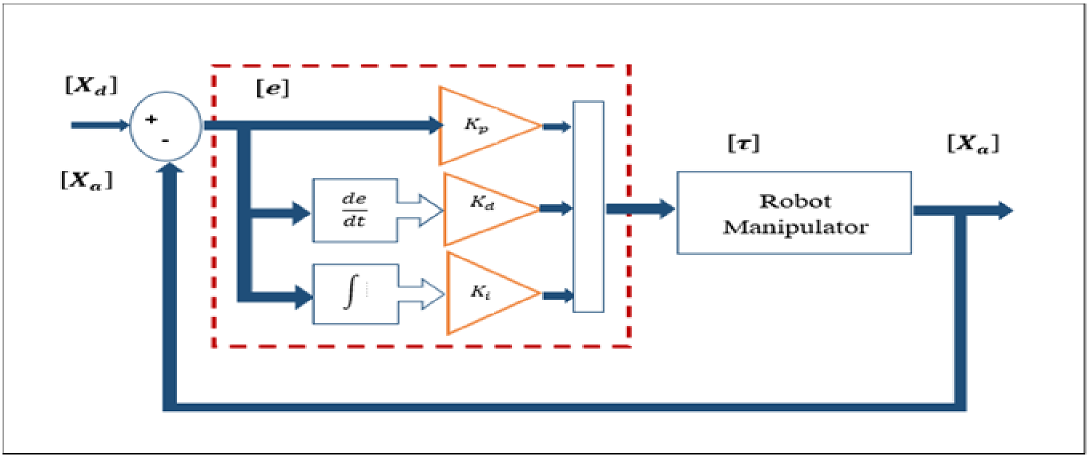


Figure (5): Global design of PID controller

6. Optimized Control of PID with Intelligent Swarm Algorithms:

Intelligence Swarm (IS) optimizing algorithm strategies significantly boost the performance of robotic arm manipulator control. These strategies iteratively employ random units to transform one candidate solution into a more robust one, based on a specified fitness function. This study employs Particle Swarm Optimization (PSO) and Social Spider Optimization (SSO) algorithms to calibrate and improve the parameters of the proposed nonlinear control devices, aiming for stability and resilience in the robot arm's final trajectory with minimal tracking error.

The PSO and SSO algorithms are dependent on fitness function and performance index.

Root Mean Square Error ($RMSE$) = $\sqrt{\frac{\sum_{i=1}^{N}(X_d - X_a)^2}{N}}$ ... (29)

Where: X_d: is for the desired position. X_a: is for the actual position. N: the number of elements in the colony. In this work, the performance index in Equation (29) is used to calculate the fitting of each solution in population size for both PSO and $\mathcal{SSO}$ algorithms.

6.1 Particle Swarm Algorithm for Optimization PSO

The algorithm of Particle Swarm Optimization PSO was initially inspired by fish schooling or bird flocks' social behavior. The optimization concept includes using the most effective placements of particles to direct the swarm population towards the single optimal solution inside the solution space. The PSO method demands, at each iteration, the modification of the particle's position and velocity towards its Pbest and Gbest. Each particle's velocity is continuously adjusted based on its personal optimum position, identified by the particle itself, as well as the optimal position discovered by adjacent particles. as seen in the following equations [20, 23]:

$$\zeta_{\mathcal{p},\mathcal{g}}^{(t+1)} = \Phi \,.\, \zeta_{\mathcal{p},\mathcal{g}}^{(t)} + c_1\, \varphi_1\, (Pbest_{\mathcal{p},\mathcal{g}} - \mathcal{M}_{\mathcal{p},\mathcal{g}}^{(t)}) + c_2\, \varphi_2 (Gbest_{\mathcal{p},\mathcal{g}} - \mathcal{M}_{\mathcal{p},\mathcal{g}}^{(t)}) \quad \text{... (30)}$$

$$\mathcal{M}_{\mathcal{p},\mathcal{g}}^{(t+1)} = \mathcal{M}_{\mathcal{p},\mathcal{g}}^{(t)} + \zeta_{\mathcal{p},\mathcal{g}}^{(t+1)} \quad \text{... (31)}$$

Where:

$\mathcal{p}$: is for particles number. $\mathcal{g}$: for dimensions number. $\mathcal{M}_{\mathcal{p}}$:is for $\mathcal{g}$- dimensional position matrix vector as ($\mathcal{M}_{\mathcal{p}1}$,$\mathcal{M}_{\mathcal{p}2}$,..,$\mathcal{M}_{\mathcal{p}\mathcal{g}}$). $\zeta_{\mathcal{p}}$: velocity of the particle for ($\zeta_{\mathcal{p}1}$, $\zeta_{\mathcal{p}1}$, …., $\zeta_{\mathcal{p}\mathcal{g}}$).

$Pbest$: best visited position for the particles. $Gbest$: best position explored in the population.

φ_1 φ_2: random numbers between 0 and 1. Φ : inertia weight. c_1 c_2: positive constants.

t: for iteration pointer.

Coefficients $c_1 and\, c_2$ include the qualified weight of the probabilities that accelerate each particle in $Pbest$, and $Gbest$ position. So, the minor values allow particles to migrate before elimination from target zones. In addition, large values result in a fast movement to, or previous to, target areas. Appropriate choice of inertia weight (Φ) allows for a balance between global and local exploration, which enables reduced aggregate iteration to find a fairly optimum solution [24]. Table (4) illustrates the global pseudo-code for PSO.

Table (4): Global pseudo-code for PSO algorithm

Step No.	Description
1	Generate random initial population and initial parameters (initialization of individuals).
2	While not (The population converges towards the desired and optimum solution or the ultimate iteration obtained) or each particle in the whole population
3	For any dimension in the particle
4	Update the velocities of the particle by using Equation (3.41)
5	Update the positions of particles by using Equation (3.42)
6	Evaluate the value of $fit(\mathcal{M}_{pg}^{(t)})$
7	If $fit(\mathcal{M}_{p,g}^{(t)}) < fit(Pbest_{p,g})$ then $(Pbest_{p,g}) \leftarrow \mathcal{M}_{p,g}^{(t)}$
8	End If
9	If $fit(\mathcal{M}_{p,g}^{(t)}) < fit(Gbest_{p,g})$ then $fit(Gbest_{p,g}) \leftarrow \mathcal{M}_{p,g}^{(t)}$
10	End If
11	End For
12	Loop (*Next generation*)

6.2 Social Spider Optimization SSO

The SSO algorithm recognizes two search agents or two sexes, male and female, in spider webs. Each entity is equipped with a unique array of evolutionary operators, determined by gender, which replicates various cooperative behaviors typically observed within the colony. Social spiders create mutual web colonies, positioning themselves in proximity to their friends, with females exceeding males by approximately 70%. Dominant males (D_m) mate with their female neighbors within a specific close proximity whereas non-dominant males (ND_m) remain near other males in the network and depend on the resources available to them. Spiders communicate using vibrations, which are affected by two critical factors: the spider's weight and the distance between the conversing spiders. The male and female spiders serve as the search agents, and their number can be quantified as follows. [25]:

$$\mathcal{N}_{\mathrm{f}} = \left[(0.9 - \mathrm{rand} * 0.25) * \mathcal{N}_{\mathrm{p}}\right] \quad \ldots (32)$$

$$\mathcal{N}_{\mathrm{m}} = \mathcal{N}_{\mathrm{p}} - \mathcal{N}_{\mathrm{f}} \quad \ldots (33)$$

Where:

$\mathcal{N}_\mathrm{p}, \mathcal{N}_\mathrm{f}$, and $\mathcal{N}_\mathrm{m}$ Denote the total population of all spiders in the communal web, reflecting the number of the female and the male spiders respectively. To find the specific weight that should represent the heaviness of each spider in the population, is calculated as follows:

$$\mathbb{W}_{\mathbb{i}} = \frac{fitness_i - worst}{best - worst} \quad \text{For} \quad 0 \leq \mathbb{i} \leq \mathcal{N}_\mathrm{p} \qquad \dots (34)$$

Where:

$\mathbb{W}_{\mathbb{i}}$: represents the weight of the spider in the population.

$\mathbb{i}$: represents the population of the web.

$\mathcal{N}_\mathrm{p}$: The number of total populations.

$fitness_i$, $best$ and $worst$ are objective function values.

In the common popular web, the spiders connect with each other's by using the vibrations $\mathbb{V}_{\mathbb{i}\mathbb{j}}$, which is calculated as follows:

$$\mathbb{V}_{\mathbb{i},\mathbb{j}} = \mathbb{W}_{\mathbb{j}} * \mathrm{e}^{-\mathrm{d}^2_{\mathbb{i}\mathbb{j}}} \qquad \dots (35)$$

$\mathbb{W}_{\mathbb{j}}$: is related to the heaviness of the spider $\mathbb{j}$ that transmits the vibration.

$\mathrm{d}_{\mathbb{i},\mathbb{j}}$: is considered as the distance of the Euclidean computed between two interacting spiders.

The location of the female and the male spiders is modified at each iteration ($\mathscr{k} = iteration$) of the optimization algorithm using the formula as follows:

$$f_{\mathbb{i}}^{\mathscr{k}+1} = \begin{cases} f_{\mathbb{i}}^{\mathscr{k}} + \alpha \mathbb{V}_{n,\mathbb{i}} \left(\mathcal{S}_n - f_{\mathbb{i}}^{\mathscr{k}}\right) + \beta \mathbb{V}_{b,\mathbb{i}} \left(\mathcal{S}_b - f_{\mathbb{i}}^{\mathscr{k}}\right) + \gamma \left(rand - \frac{1}{2}\right) & if\ r_m < \mathcal{PF} \\ f_{\mathbb{i}}^{\mathscr{k}} - \alpha \mathbb{V}_{n,\mathbb{i}} \left(\mathcal{S}_n - f_{\mathbb{i}}^{\mathscr{k}}\right) - \beta \mathbb{V}_{b,\mathbb{i}} \left(\mathcal{S}_b - f_{\mathbb{i}}^{\mathscr{k}}\right) + \gamma \left(rand - \frac{1}{2}\right) & if\ r_m \geq \mathcal{PF} \end{cases}$$

... (36)

Where: $\alpha, \beta, and\ \gamma$ are considered as random numbers, that are between [0 or 1]. $\mathcal{PF}$ is regarded as the threshold of probability factor that is matched with all the randomly produced numbers, while the spiders will be travelled in the space. So, the transition is randomly regulated by the $\mathcal{PF}$ factor of likelihood and the motion is generated in relation with the other spiders depending on their vibration through the space of search. In the optimization procedure the male spider $m_{\mathbb{i}}^{\mathscr{k}+1}$, will be operated as shown in the following equation:

$$m_{\mathbb{i}}^{\mathscr{k}+1} = \begin{cases} m_{\mathbb{i}}^{\mathscr{k}} + \alpha \mathbb{V}_{f,\mathbb{i}} \left(\mathcal{S}_f - m_{\mathbb{i}}^{\mathscr{k}}\right) + \gamma \left(rand - \frac{1}{2}\right) & if\ \ m_{\mathbb{i}}^{\mathscr{k}} > \mathcal{D}_m \\ m_{\mathbb{i}}^{\mathscr{k}} + \alpha \left(\frac{\sum_{hi=1}^{N_m} m_h^{\mathscr{k}} W_h}{\sum_{hi=1}^{N_m} W_h} - m_{\mathbb{i}}^{\mathscr{k}}\right) & if\ \ m_{\mathbb{i}}^{\mathscr{k}} \leq N\mathcal{D}_m \end{cases} \qquad \dots (37)$$

Where:

$\left(\frac{\sum_{hi=1}^{N_m} m_h^{k} \mathbb{w}_h}{\sum_{hi=1}^{N_m} \mathbb{w}_h} - m_{\mathbb{i}}^{k}\right)$ is considered as the weighted average of the male spider in the web population,

The dominant males, as described earlier, will mate with the females in the mating range, which is given as follows:

$$r_m = \frac{\sum_{\mathbb{j}=1}^{n}(p_{\mathbb{j}}^{high} - p_{\mathbb{j}}^{low})}{2n} \qquad \ldots (38)$$

Where: $p_{\mathbb{j}}^{high}$ *and* $p_{\mathbb{j}}^{low}$ are assumed to be as the higher and lower initial limits. [25, 26]. Table (5) illustrates the pseudo-code for the SSO algorithm as follows:

Table (5): Pseudo-code for SSO algorithm

Step No.	Description
1	Assign the value of the initial parameters, of SSO algorithm
2	Create the population of spiders and assign memory for them
3	Initial the first position for both male spider and female spider
4	Counting the k iterations in the population ($\mathcal{N}_p$)
5	While $\mathbb{i} \le \mathcal{N}_p$
6	Calculate the mating radius for both female and male spiders as in Equation (3.49)
7	Calculating the spiders' weight as in Equation (3.45)
8	Calculate the passage of the female and the male spider's dependent on mutual the female and the male operators as shown in Equation (3.47) and (3.48)
9	Conduct mating between dominant males and females
10	Update solutions if there are heavier spider progenies
11	End of the loop

7. Circular Path Optimization PID Controller Simulation Outcomes Based on PSO and SSO Algorithms

This section presents the design of the Simulink diagram for the proposed controller. This illustrates the simulation of a 3-DOF robot manipulator for the control of the robot's joints. Additionally, the simulation results and analysis for the suggested controller, together with its optimization methodologies and characterization, are presented. The PID controller is derived from the mathematical model, which regulates the DM of the 3-DOF robot to achieve a stable response with precise final positioning at the end-effector. There are nine PID parameter values for the three joint controllers, with an initial value designated for each. Table (6) delineates the values of the fundamental parameters for non-optimal PID settings, which will serve as a benchmark to enhance controller performance.

Table6: Values of basic parameters for a PID controller that is not ideal

	Number of linked		
Parameters in PID	L_1	L_2	L_3
K_p	10	10	10
K_i	3	3	3
K_d	5	5	5

The simulations utilized MATLAB/Simulink, following to the equations governing circular trajectories for P_x, P_y), as demonstrated:

$$P_X = x_c + r\cos(wt) \quad \dots (39)$$

$$P_y = y_c + r\sin(wt) \quad \dots (40)$$

Where: $r = 0.05\ (m)$ it is the radius of the circular. $w = \frac{\pi}{4}$ (Rad/S), which is the angular frequency, and, $x_c = y_c = 0.45\ (m)$ is the center of the initial point.

The PSO method utilizes X_ (desire) as a reference and evaluates each particle in X_(actual) by calculating the Root Mean Square Error (RMSE) between them until it attains the optimal fitness function with the minimal RMSE. The performance of the suggested PID controller, based on the PSO algorithm, was examined, with the PSO algorithm parameters detailed in Table7.

Table (7): Configuration of parameters for the PSO algorithm

Intelligent swarm optimization algorithm PSO	Selected parameters
No. of birds	20
No. of iterations	100
Inertia weight factor (Φ)	0.5
Positive constants ($c_1\ c_2$)	2

Figure (6) displays the iterative process of PSO algorithms along with the target function and Root Mean Square Error (RMSE). It is evident that the PSO method iteratively approaches the ideal parameter value with the best fitness = 3.7930e^{-3}

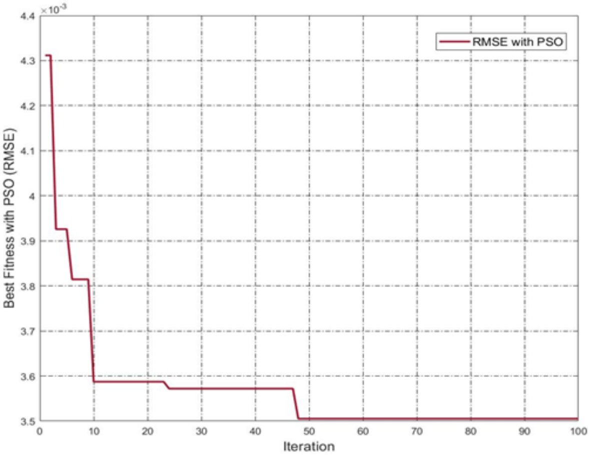


Figure (6): Reiteration and objective function for PSO algorithm

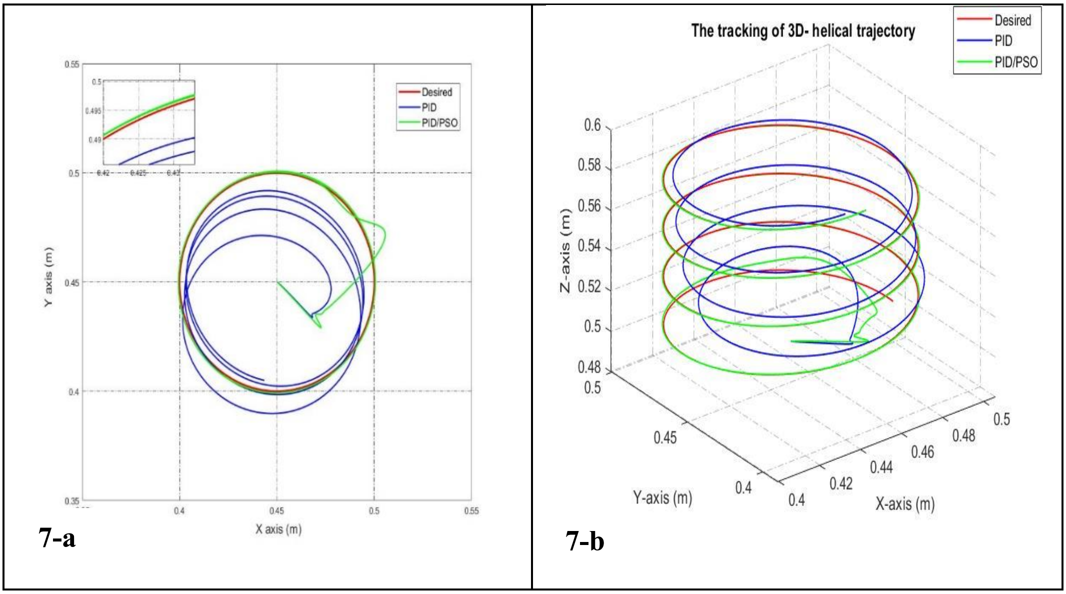


Figure 7-a: Circular trajectory tracking with PID and PID/PSO in (X-Y) plane
Figure 7-b: Trajectory tracking performance for the 3-DOF robot of 3D- helical loop trajectory using PID, and PID/PSO algorithm

Figure (7-a) illustrates the trajectory tracking efficacy of the 3-DOF robot manipulator utilizing a non-optimal PID controller, in comparison to the PID/PSO algorithm for circular trajectory tracking. Figure (7-b) illustrates the trajectory tracking efficacy of the 3-DOF robot manipulator executing a three-dimensional helical loop trajectory utilizing both PID and PID/PSO algorithms. Equation no. 41, show the procedure to calculate the Z-dimension in the trajectory.

$$P_Z = 0.5 + 0.003t \qquad \ldots (41)$$

Table (8) presents a comparison of the non-optimal PID techniques with the optimization algorithm PID-PSO, highlighting the performance criteria of RMSE across the three Cartesian spaces.

Table (8): Performance index for non-optimal PID, and PID/PSO with *RMSE* method in X-Position, Y-Position, and Z-Position

	Non- optimal PID	Optimal PID/PSO
X-axis	0.0110	0.0059
Y-axis	0.0122	0.0014
Z-axis	0.0113	0.0007436
Total *RMSE*	0.0345	0.0080436

Case 2: Finding the Optimal PID using the SSO Algorithm in Simulation

To assess the efficacy of the suggested PID controllers, derived from the SSO algorithm, the fundamental parameters of the SSO algorithm were modified as depicted in Table 9:

Table (9): Basic parameters used in the SSO algorithm

Initial parameters for social spider algorithm SSO	Selected parameters
Spider No.	20
Iteration	100
random number $[\alpha, \beta, \gamma]$	1
PF parameter	0.7
Upper female per cent	0.9
Lower female per cent	0.65

Table (10) illustrate the optimal parameters for PID with SSO algorithm respectively, as follows:

Table (10): Optimal parameters for PID controller with SSO algorithm

	Link number		
Parameters	L_1	L_2	L_3
K_p	105.2872	19.0765	201.6369
K_i	140.4973	136.4403	231.3952
K_d	3.9759	0.2470	49.4850

Figure (8) illustrates the iteration of SSO algorithms, displaying the Root Mean Square Error and objective function, indicating that SSO iteratively progresses until it attains the optimal parameter values, with the best fitness = 1.7270 e^{-4}.

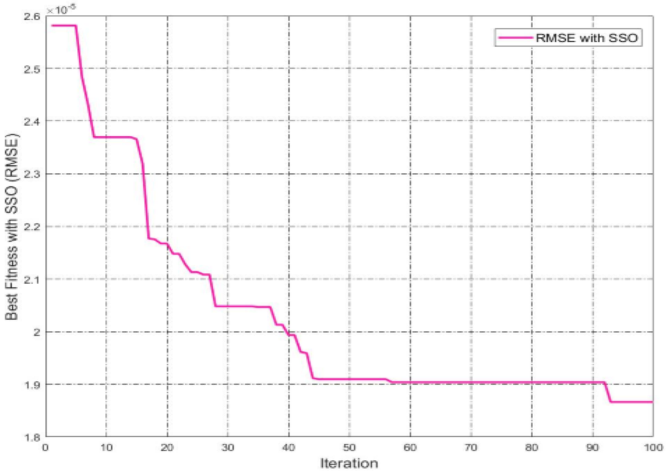
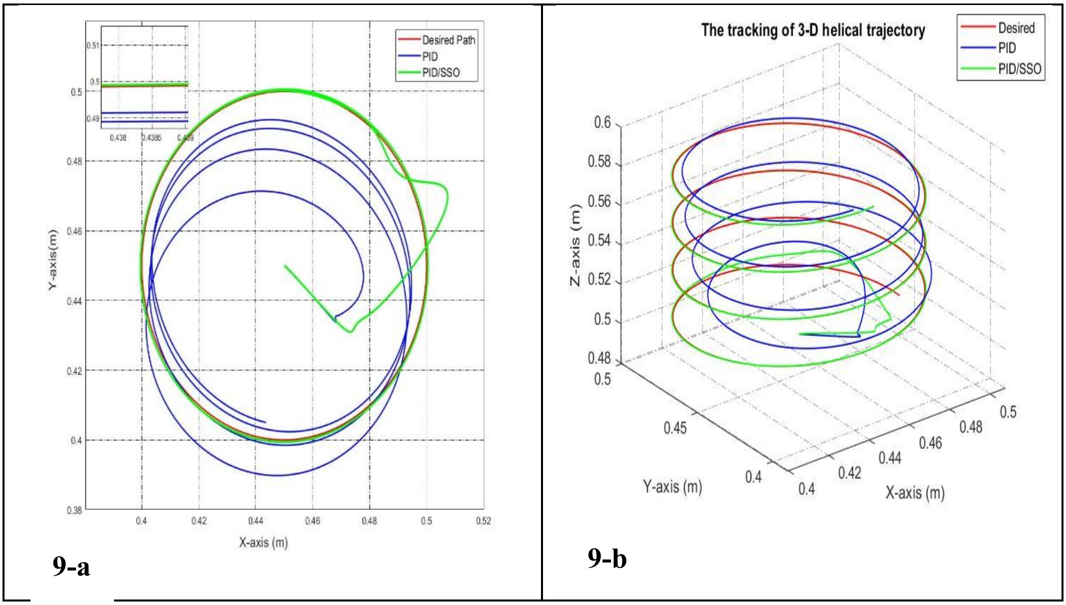


Figure (8): Reiteration and objective purpose for SSO algorithm

Figure (9-a) illustrates the rate of actual circular trajectory tracking error in Cartesian space for the three degrees of freedom (3-DOF) robot across joints one, two, and three. Figure (9-b) illustrates the trajectory tracking efficacy of the 3-DOF robot manipulator following a 3D helical loop trajectory, employing both PID and PID/SSO algorithms



9-a

9-b

Figure 9-a: Circular trajectory tracking with PID, and PID/SSO in planFigure 9-b: Trajectory tracking performance for the 3-DOF robot with 3D- helical loop trajectory using PID, and PID/SSO algorithm

Case 3: Simulation results of PID/PSO and PID/SSO algorithms

A comparative analysis will be performed between the PID/PSO and PID/SSO algorithms utilized in this work to optimize the settings of the PID controller. The results of each algorithm are shown in Figure (10-a), which uses circular trajectories with PID/PSO and PID/SSO. Figure (10-b) depicts the trajectory tracking efficacy of the 3-DOF robot in three dimensions for the helical loop trajectory employing PID/PSO and PID/SSO algorithms. PID/SSO exhibits superior stability and convergence relative to PID/PSO.

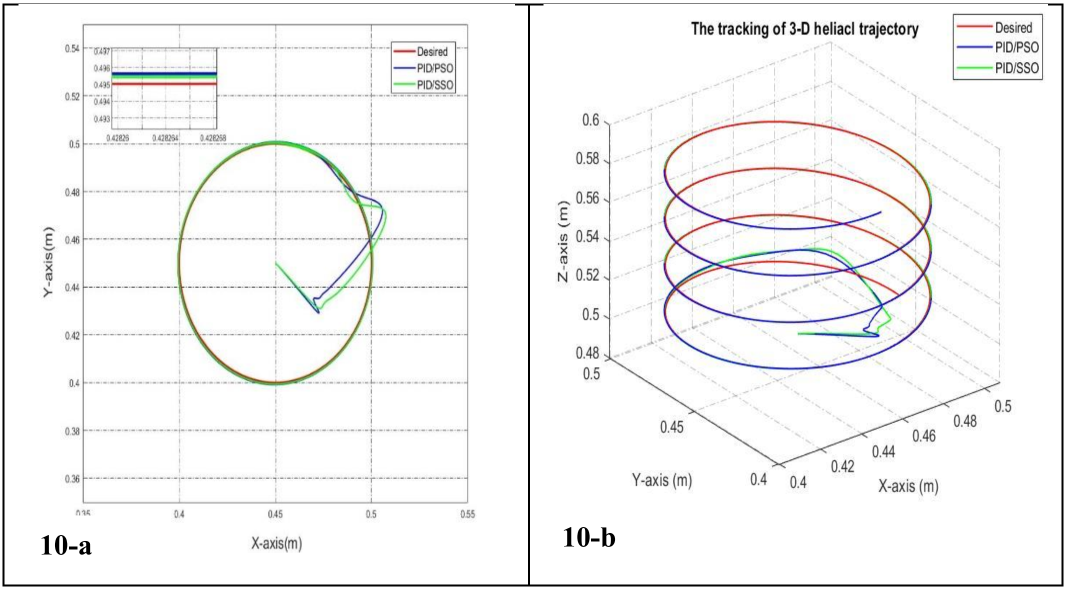


Figure 10-a: Circular trajectory tracking with PID/PSO, and PID/SSO in two planes. Figure 10-b: Trajectory tracking performance for the 3-DOF robot with 3D- of the helical trajectory using PID/PSO, and PID/SSO algorithm

The performance index calculates the quality and fitness function of the proposed controllers. Table (11) presents a comparison of PID/PSO techniques with the optimization algorithm and the PID/SSO algorithm, illustrating the performance criteria of RMSE across the three Cartesian spaces.

Table 11. presents the Performance Index for non-optimal PID and PID/SSO utilizing the RMSE method across X-Position, Y-Position, and Z-Position.

	$RMSE$ in PID/PSO (m)	**$RMSE$ in PID/SSO (m)**
X-axis	0.0059	0.0055
Y-axis	0.0014	0.0012
Z-axis	0.00074362	0.0006979
Total $RMSE$	0.00804362	0.0073979

8. Discussion the result:

The comparative analysis of the two control strategies, **PID–PSO** and **PID–SSO**, indicates that both approaches significantly enhance the performance of the conventional PID controller by reducing steady-state error and improving transient response. Nevertheless, the **PID–SSO** consistently outperformed the **PID–PSO** in terms of overall system stability, faster settling time, and reduced oscillatory behavior. Accordingly, the **PID–SSO** can be regarded as a more robust and reliable optimization technique, making it a more suitable choice for practical implementations that demand high accuracy and long-term stability, as shown in Table 11. Where the total RMSE is 0.0073979, with the best fitness = 1.7270 e^{-4}.

9. Conclusions

This study presents a 3-DOF manipulator robot, which enhances safety and efficiency across several applications. The primary problem for a robotics researcher is achieving optimal control of the robot manipulator with sufficient overall efficiency. This work allows for several conclusions on the control of the 3-DOF robot, as follows:

1. This study presents an ideal methodology and a resilient control strategy utilizing Proportional Integral Derived PID, along with two innovative Intelligent Swarm Algorithms: Social Spider Optimization (SSO) and Particle Swarm Optimization (PSO).
2. The evolutionary algorithms of SSO and PSO have been employed to ascertain optimal parameter values for the proposed controller, ensuring trajectory stability under diverse conditions affecting the Cartesian coordinates of the robot manipulator's end effector, with nine parameters optimized for the PID.
3. The simulation results illustrate the effectiveness of the proposed tuning method, exhibiting a high degree of stability and superior implementation for the 3-DOF robot. Notably, the PID/SSO approach yielded the most favorable outcomes in the circular and square trajectory responses, particularly in steady-state conditions, with a high RMSE of 0.0073979.

10. References:

[1] Åström, Karl J., and Tore Hägglund. "PID control." *IEEE Control Systems Magazine* 1066 (2006): 30-31.

[2] Karaboga, Dervis, and Bahriye Akay. "A comparative study of artificial bee colony algorithm." *Applied mathematics and computation* 214.1 (2009): 108-132.

[3] Mirjalili, Seyedali. "Evolutionary algorithms and neural networks." *Studies in computational intelligence* 780.1 (2019): 43-53.

[4] Cuevas, Erik, et al. "A swarm optimization algorithm inspired in the behavior of the social-spider." *Expert Systems with Applications* 40.16 (2013): 6374-6384.

[5] Zheng, Shufa, et al. "Viral load dynamics and disease severity in patients infected with SARS-CoV-2 in Zhejiang province, China, January-March 2020: retrospective cohort study." *bmj* 369 (2020).

[6] Izci, Davut, Serdar Ekinci, and Abdelazim G. Hussien. "Effective PID controller design using a novel hybrid algorithm for high order systems." *PLoS One* 18.5 (2023): e0286060.

[7] Kabir, Usman, et al. "Performance analysis of PID, PD and fuzzy controllers for position control of 3-DOF robot manipulator." *arXiv preprint arXiv:1910.12076* (2019).

[8] Hadi, Muhamad Sukri, et al. "Intelligent PID Controller for Vibration Suppression of Horizontal Flexible Plate Based on Social Spider Optimization." *Climbing and Walking Robots Conference*. Cham: Springer Nature Switzerland, 2024.

[9] Khan, Huma, et al. "Speed control of wheeled mobile robot by nature-inspired social spider algorithm-based PID controller." *Processes* 11.4 (2023): 1202.

[10] Souza, Darielson A., et al. "PID controller with novel PSO applied to a joint of a robotic manipulator." *Journal of the Brazilian Society of Mechanical Sciences and Engineering* 43.8 (2021): 377.

[11] Korayem, Moharam Habibnejad, et al. "Determining maximum load-carrying capacity of robots using adaptive robust neural controller." *Robotica* 28.7 (2010): 1083-1093.

[12] Yadegar, Sanaz. "Design stable robust intelligent nonlinear controller for 6-DOF serial links robot manipulator." *International Journal of Intelligent Systems and Applications* 6.8 (2014): 19.

[13] Ashagrie, Aderajew, Ayodeji Olalekan Salau, and Tilahun Weldcherkos. "Modeling and control of a 3-DOF articulated robotic manipulator using self-tuning fuzzy sliding mode controller." Cogent Engineering 8.1 (2021): 1950105.

[14] J. J. Craig “Introduction to robotics: mechanics and control,” 3^rd. Pearson Education India, 2009.

[15] Ashagrie, Aderajew, Ayodeji Olalekan Salau, and Tilahun Weldcherkos. "Modeling and control of a 3-DOF articulated robotic manipulator using self-tuning fuzzy sliding mode controller." *Cogent Engineering* 8.1 (2021): 1950105.

[16] Abbas, Zaheer. *Motion control of robotic arm manipulator using pid and sliding mode technique*. Diss. Department of Electrical Engineering, Capital University of Science and Technology, Islamabad, 2018.

[17] Singh, Tarun Pratap, P. Suresh, and Swet Chandan. "Forward and inverse kinematic analysis of robotic manipulators." *International Research Journal of Engineering and Technology (IRJET)* 4.2 (2017): 1459-1468.

[18] Spong, Mark W., and Mathukumalli Vidyasagar. *Robot dynamics and control*. John Wiley & Sons, 2008.

[19] Şahin, Hakan. *Design of a secondary packaging robotic system*. MS thesis. Middle East Technical University (Turkey), 2005.

[20] Akkar, Hanan AR, and Suhad Qasim G. Haddad. "Design Stable Controller for PUMA 560 Robot with PID and Sliding Mode Controller Based on PSO Algorithm." *International Journal of Intelligent Engineering & Systems* 13.6 (2020).

[21] Haddad, Suhad Qasim G., and Hanan AR Akkar. "Intelligent swarm algorithms for optimizing nonlinear sliding mode controller for robot manipulator." *International Journal of Electrical and Computer Engineering (IJECE)* 11.5 (2021): 3943-3955.

[22] Karaboga, Dervis, and Bahriye Akay. "A comparative study of artificial bee colony algorithm." *Applied mathematics and computation* 214.1 (2009): 108-132.

[23] Y. Wang, P. Bai, et al, “Reconnaissance Mission Conducted by UAV Swarms Based on Distributed PSO Path Planning Algorithms,” IEEE Access, Vol. 7, pp. 105086–105099, 2019.

[24] F. Hasan, L. Rashad, and A. Humod, “Integrating Particle Swarm Optimization and Routh-Hurwitz’s Theory for Controlling Grid-Connected LCL-Filter Converter,” Int. J. Intell. Eng. Syst., Vol. 13, No. 4, pp. 102–113, 2020.

[25] L. C. Kien, T. T. Nguyen, et al, “A Novel Social Spider Optimization Algorithm for Large-Scale Economic Load Dispatch Problem,” Energies, Vol. 12, No. 6, pp. 1–26, 2019.

[26] A. Luque-Chang, and E. Cuevas, “Social Spider Optimization Algorithm: Modifications, Applications, and Perspectives,” Math. Probl. Eng., Hindawi, Vol. 2018, 2018.

تصميم وحدة تحكم PID مثالية لذراع روبوت ثلاثي درجات الحرية

Dr. Suhad Qasim G. Haddad[1]
suhad.qasim@muc.edu.iq

Ass.Prof. Enass H. Flaieh[2]
Enass.h.flaieh@uotechnology.edu.iq.iq

Prof.Dr. Hanan A.R. Akkar[3]
Hanan.a. akkar@ uotechnology.edu.iq.iq

الخلاصة :

إن أصعب مهمة يواجهها باحث الروبوتات هي التحكم في مُشغل الروبوت بكفاءة إجمالية مناسبة. يقدم هذا العمل تقنيات دقيقة وسريعة لتحسين معاملات مُشغل الروبوت ثلاثي الدرجات. صُمم هذا النوع من الروبوتات لأغراض أكاديمية وتعليمية، ويُستخدم أيضًا في المجال الطبي. جميع وصلات مُشغل الروبوت ثلاثي الدرجات عبارة عن مفاصل دوارة أو دوارة ذات اتصالات تسلسلية. تركز تقنية التحسين على خوارزميات السرب الذكي المستخدمة لتحسين وضبط كسب وحدة التحكم المقترحة، وهي المشتق التكاملي النسبي التقليديPID) .

استُخدمت تقنيتا تحسين سرب الجسيمات الذكي PSO وتحسين العنكبوت الاجتماعي SSO للحصول على أفضل قيم لمعاملات وحدة التحكم، لتحقيق الاتساق والاستقرار والمتانة. يُستخدم MATLAB/Simulink لتسريع حساب FK وIK وDM لمُشغل الروبوت ثلاثي الدرجات. تظهر النتائج أن تعديل SSO / PID يعطي أفضل النتائج من حيث استجابة الطور المستقرة، وخطأ الجذر التربيعي المتوسط الأدنى مع أفضل دالة هدف، وإشارة تحكم مستقرة، وحالة مستقرة معقولة مع تتبع المسار المثالي للمسار الدائري.

الكلمات المفتاحية: وحدة التحكم PID، خوارزمية PSO، خوارزمية SSO، ذراع روبوت ثلاثي درجات الحرية، حركيات وديناميكيات ذراع الروبوت.

[1] مدرس دكتور سهاد قاسم حداد ، مدير دائرة المختبرات ، كلية المنصور الجامعة

[2] أستاذ مساعد إيناس فليح ، كلية الذكاء الاصطناعي ، الجامعة التكنولوجية

[3] أستاذ دكتور حنان عكار، كلية الهندسة الكهربائية ، الجامعة التكنلوجية