

Original Article

# Towards an Efficient Control Strategy for an Industrial Multi-DoF Robotic Arm

Ngoc-Khoat Nguyen<sup>1\*</sup>, Sy-Viet Ho<sup>1</sup>, Duy-Trung Nguyen<sup>1</sup>, Quoc-Hoan Tran<sup>1</sup>, Trung-Nguyen Tran<sup>1</sup>, Tien-Dat Nguyen<sup>1</sup>

<sup>1</sup>Faculty of Control and Automation, Electric Power University, Hanoi, Vietnam.

\*Corresponding Author : [khoatnn@epu.edu.vn](mailto:khoatnn@epu.edu.vn)

Received: 03 March 2025

Revised: 09 May 2025

Accepted: 14 May 2025

Published: 31 May 2025

**Abstract** - This study focuses on developing an optimized control scheme for a multi-Degree-of-Freedom (DoF) robotic manipulator, using a representative 4-DoF articulated arm as a case study. Various angular position control methodologies are implemented, encompassing both classical approaches, such as Proportional-Integral-Derivative (PID) control, and advanced intelligent techniques, including Fuzzy Logic Control (FLC) and Sliding Mode Control (SMC). Additionally, the Particle Swarm Optimization (PSO) method is used to find optimal tuning parameters for the controllers, significantly influencing the manipulator's control performance. Comprehensive simulations, comparative analyses, and performance evaluations conducted in the MATLAB/Simulink environment validate the advantages and limitations of each proposed control strategy. Based on the theoretical framework and empirical findings, the PID controller optimized via the PSO algorithm, along with the SMC, demonstrates superior performance, establishing them as viable and effective solutions for multi-DoF robotic manipulators in industrial applications.

**Keywords** - Multi-DoF robotic arm, PID, FLC, SMC, PSO, Control performance.

## 1. Introduction

Robots, especially industrial robots, can be defined as automated mechatronic systems that are designed and programmed to execute specific operational tasks. Within the industrial sector, these systems play a critical role, yielding significant advantages. It is found that their implementation is widespread across automated production lines, serving as a substitute for human labor in tasks characterized by repetitive motion, inherent hazards, or stringent precision requirements [1-5]. Robots have many useful applications; however, the design of control strategies has faced a lot of challenges [6-8]. A primary challenge in the design of robot control systems lies in accurately modeling the robotic system itself, given its inherent high non-linearity and susceptibility to uncertainties such as friction, variable payloads, and sensor inaccuracies. Moreover, the development of control algorithms requires meticulous attention to ensure system stability, precision, and robustness, while simultaneously satisfying real-time operational requirements across diverse applications. Additionally, integrating sensors and actuators into the control architecture poses significant challenges, particularly in mitigating sensor errors and actuator latency. Globally, established control methodologies, including Proportional-Integral-Derivative (PID) control and Fuzzy Logic Control (FLC), have been widely implemented in robotic systems. PID controllers are recognized for their inherent stability and high

reliability [2, 9-16]. However, they are susceptible to limitations such as overshoot, typically in the range of 15–20%, and exhibit significant sensitivity to torque disturbances. Conversely, FLC performance is strongly contingent upon the designer's expertise in defining membership functions, resulting in protracted parameter optimization procedures. For example, Nguyen et al. [1] documented that an average of 8–10 iterative tuning cycles was necessary to achieve a steady-state error of less than 5%. Both control paradigms traditionally rely on empirical knowledge and iterative refinement, leading to parametric uncertainty in robotic manipulator systems. To address these issues, contemporary research has investigated Particle Swarm Optimization (PSO) as a high-performing methodology for control parameter optimization [8]. Originally proposed by Kennedy and Eberhart in 1995, the PSO demonstrates efficacy in balancing the exploration of the parameter space while mitigating the risk of convergence to local minima. For example, Ahmed et al. (2021) implemented PSO in conjunction with PID control for a 6-DoF robot, achieving a 27% reduction in overshoot compared to conventional PID implementations. Analogously, Wang et al. (2022) utilized PSO to optimize the membership functions within a FLC, resulting in a 40% decrease in tuning time. However, most current investigations have focused on the independent optimization of singular controllers, either PID or FLC, neglecting the potential



benefits of a hybrid control strategy that integrates classical and intelligent control methodologies. Such an integrated approach holds the promise of developing a control system that effectively balances stability, accuracy, and response speed, thereby more adequately satisfying the stringent requirements of contemporary industrial applications.

The authors of the aforementioned studies have demonstrated that the PSO algorithm is among the most effective methods for optimizing controller parameters. Despite these notable improvements, a key limitation of this body of research is the isolated optimization of individual controller types—either PID or FLC—without exploring the potential benefits of integrating multiple control strategies within a single system.

Additionally, there is a paucity of studies evaluating such approaches on 4-DoF robotic platforms, limiting the generalizability and applicability of the findings to more complex robotic systems.

This investigation focuses on the dynamic analysis of a multi-DoF robotic system, using a representative 4-DoF configuration as a case study. The study develops and evaluates distinct control methodologies for the robotic model, including the PID control, FLC, and Sliding Mode Control (SMC).

Additionally, PSO is employed to optimize and determine the appropriate parameters for the PID and FLC controllers. Simulation results obtained using MATLAB/Simulink, along with detailed evaluations, analyses, and comparative assessments, illustrate the respective advantages and limitations of each control approach. Consequently, robotic system designers can select a suitable methodology for their specific applications. Although the findings presented herein are primarily theoretical, they offer significant prescriptive insights for real-world implementations.

## 2. Literature Review

Recent studies have demonstrated the effectiveness of the PSO algorithm in enhancing the performance of PID and FLC in robotic systems. For instance, Elsayed et al. (2024) applied PSO to optimize an FLC for a continuous soft robot, reporting a 29.9% reduction in the Integral of Time-weighted Absolute Error (ITAE) and a decrease in settling time to 0.7 seconds compared to a conventional FLC. However, this study focused exclusively on FLC optimization, without considering integration with other control strategies such as PID or SMC, and did not evaluate performance in multi-DoF robotic systems.

Similarly, Liu et al. (2021) utilized PSO to tune the parameters of a PID controller for a 7-DOF exoskeleton, resulting in reduced trajectory tracking error and improved stability. Nevertheless, their work was limited to PID control,

omitting consideration of alternative methods such as FLC or SMC, thereby restricting the diversity and adaptability of the control strategy. Barış Gökçe et al. (2021) applied PSO for PID optimization in an agricultural robot, achieving enhanced trajectory tracking and stability in outdoor environments. However, their study was confined to agricultural robots and did not extend to systems with higher DoF, such as 4-DOF manipulators.

A. K. Kashyap et al. (2021) implemented PSO to optimize PID controllers for the joints of a robotic manipulator, yielding improved transient response and reduced oscillations. Yet, the study focused solely on PID optimization, without comparative analysis against alternative control methodologies such as FLC or SMC. Likewise, H. H. Ammar and A. T. Azar (2020) demonstrated that PSO-enhanced PID control improved the stability and balance of a two-wheeled robot, although their work lacked generalizability to robots with higher degrees of freedom.

In summary, current research predominantly concentrates on optimizing a single type of controller—either PID or FLC—with limited investigation into the integration of multiple control strategies within a unified robotic framework. Moreover, there remains a notable gap in the literature regarding the combined application of PID, FLC, and SMC in the control of a 4-DoF robot, and the simultaneous optimization of their parameters using PSO.

Accordingly, this research aims to validate and assess the effectiveness of three control strategies—PID, FLC, and SMC—applied to a 4-DoF robotic manipulator. This includes the use of PSO for the optimization of PID and FLC parameters to enhance overall controller performance. The evaluation will be conducted through simulations in the MATLAB/Simulink environment, employing key performance metrics such as response time, overshoot, steady-state error, and robustness.

## 3. Research Methodology

### 3.1. Modeling of a 4-DoF Robot

#### 3.1.1. Forward Kinematics of the 4-DoF Robot

A planar four-degree-of-freedom (4-DoF) robotic manipulator, articulated by four rotary joints, serves as a fundamental platform for optimizing and stabilizing control strategies in practical applications such as object manipulation and additive manufacturing. The forward kinematics problem for this manipulator entails the determination of the end-effector's pose (position and orientation) as a function of the specified joint angles.

Denavit describes the dynamics of a 4-DoF robotic arm—the Hartenberg (DH) method, which helps determine the robot's initial parameters. Figure 1 illustrates a 4-DoF robotic arm model with four interconnected joints, all of which are rotary joints. The  $z$ -axis is oriented outward from the axis of rotation of the joints. Once the  $z$ -axis is determined, the  $x$ -axis can be identified accordingly.

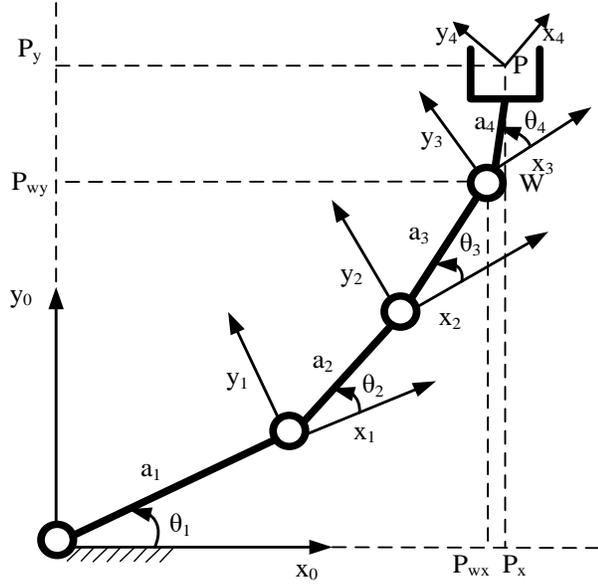


Fig. 1 Four-DoF robotic arm model with axes

Table 1. Denavit - Hartenberg (D-H) parameter table

$i$	$a_i$	$\alpha_i$	$d_i$	$\theta_i$
1	$a_1$	0	0	$\theta_1$
2	$a_2$	0	0	$\theta_2$
3	$a_3$	0	0	$\theta_3$
4	$a_4$	0	0	$\theta_4$

Where:

$a_i$  represents the offset distance between the  $z_{i-1}$  and  $z_i$  axes, measured along the direction of the axis  $x_i$ ;

$\alpha_i$  is the angle between the axis  $z_i$  and the axis  $z_{i-1}$  about the axis  $x_i$ ;

$d_i$  is the distance from the axis  $x_{i-1}$  to the axis  $x_i$  measured along the axis  $z_i$ ,

$\theta_i$  is the rotation angle from the axis  $x_{i-1}$  to the axis  $x_i$  about the axis  $z_i$ .

The general homogeneous transformation matrix for kinematics is given in (1):

$${}^{i-1}A_i = \begin{bmatrix} c(\theta_i) & -s(\theta_i)c(\alpha_i) & s(\theta_i)c(\alpha_i) & a_i c(\theta_i) \\ s(\theta_i) & c(\theta_i)c(\alpha_i) & -c(\theta_i)s(\alpha_i) & a_i s(\theta_i) \\ 0 & s(\alpha_i) & c(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Where:

$$s(\theta_i) = \sin(\theta_i); s(\alpha_i) = \sin(\alpha_i); c(\theta_i) = \cos(\theta_i); c(\alpha_i) = \cos(\alpha_i)$$

The homogeneous transformation matrix representing the spatial relationship between the base frame (frame 0) and the end-effector frame (frame 4) is given by:

$${}^0_4A = {}^0_1A \cdot {}^1_2A \cdot {}^2_3A \cdot {}^3_4A \quad (2)$$

With  ${}^0_1A, {}^1_2A, {}^2_3A, {}^3_4A$  should be calculated as follows:

$${}^0_1A = \begin{bmatrix} c_1 & -s_1 & 0 & a_1 c_1 \\ s_1 & c_1 & 0 & a_1 s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^1_2A = \begin{bmatrix} c_2 & -s_2 & 0 & a_2 c_2 \\ s_2 & c_2 & 0 & a_2 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2_3A = \begin{bmatrix} c_3 & -s_3 & 0 & a_3 c_3 \\ s_3 & c_3 & 0 & a_3 s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^3_4A = \begin{bmatrix} c_4 & -s_4 & 0 & a_4 c_4 \\ s_4 & c_4 & 0 & a_4 s_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where:  $s_i = \sin \theta_i, c_i = \cos \theta_i$  with  $(i = 1,2,3,4)$

The matrix  ${}^0_4A$  can be deduced as:

$${}^0_4A = \begin{bmatrix} c_{1234} & -s_{1234} & 0 & p_x \\ s_{1234} & c_{1234} & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where:

$$s_{12} = \sin(\theta_1 + \theta_2), c_{12} = \cos(\theta_1 + \theta_2)$$

$$s_{123} = \sin(\theta_1 + \theta_2 + \theta_3), c_{123} = \cos(\theta_1 + \theta_2 + \theta_3)$$

$$s_{1234} = \sin(\theta_1 + \theta_2 + \theta_3 + \theta_4), c_{1234} = \cos(\theta_1 + \theta_2 + \theta_3 + \theta_4)$$

$p_x, p_y, p_z$  define the end-effector's spatial position within the global coordinate frame, where the value of each joint variable is incorporated into the subsequent kinematic equations.

$$\begin{aligned} p_x &= a_1 c_1 + a_2 c_{12} + a_3 c_{123} + a_4 c_{1234} \\ p_y &= a_1 s_1 + a_2 s_{12} + a_3 s_{123} + a_4 s_{1234} \\ p_z &= 0 \end{aligned} \quad (3)$$

The orientation of the robot is determined using the following formula:

$$\gamma = \theta_1 + \theta_2 + \theta_3 + \theta_4 \quad (4)$$

Where  $\gamma$  is the orientation of the robot;  $\theta_1, \theta_2, \theta_3, \theta_4$  are the rotation angles of each joint of the 4-DoF robot.

### 3.1.2. Inverse Kinematics of a 4-DoF Robot

The inverse kinematics problem of a robot is a problem that assumes the position of the end-effector of the robotic arm and requires determining the initial joint angle parameters [2].

Based on the 4-DoF robot model given in Figure 1, it is possible to calculate  $\theta_2$  as follows:

$$\begin{cases} x^2 + y^2 = (a_2s_2 + (a_1 + a_2c_2))^2 \\ \quad = a_2^2s_2^2 + a_1^2 + a_2^2c_2^2 + 2a_1a_2c_2 \\ c_2 = \frac{x^2+y^2-a_1^2-a_2^2}{2a_1a_2} \end{cases}$$

Thus,

$$\theta_2 = a \tan 2 (c_2, \sqrt{1 - c_2^2}) \quad (5)$$

Calculating  $\theta_1$  can be Implemented Below:

$$\begin{cases} x_2 = a_1c_1 + a_2c_{12} = c_1(a_1 + a_2c_2) - s_1(a_2s_2) \\ y_2 = a_1s_1 + a_2s_{12} = s_1(a_1 + a_2c_2) + c_1(a_2s_2) \\ c_1 = \frac{(a_1+a_2c_2)x_2+a_2s_2y_2}{x_2^2+y_2^2} \\ s_1 = \frac{(a_1+a_2c_2)y_2-a_2s_2x_2}{x_2^2+y_2^2} \end{cases} \quad (6)$$

Thus,

$$\theta_1 = a \tan 2 (s_1, c_1). \quad (7)$$

Calculate  $\theta_3$ :

$$\begin{cases} x_3 = c_3(a_1 + a_2c_2 + a_3c_3) \\ y_3 = s_3(a_1 + a_2s_2 + a_3s_3) \\ x_3^2 + y_3^2 = [c_3(a_1 + a_2c_2 + a_3c_3)]^2 + [s_3(l_1 + l_2s_2 + l_3s_3)]^2 \\ x_3^2 + y_3^2 = [(a_1 + a_2c_2)^2 + (a_2s_2)^2 + l_3^2] \end{cases}$$

From that, it can be deduced the following:

$$c_3 = \frac{x_3}{\sqrt{x_3^2 + y_3^2}}, s_3 = \frac{y_3}{\sqrt{x_3^2 + y_3^2}} \quad (8)$$

Derive  $\theta_3 = a \tan 2 (s_3, c_3)$

Calculate

$$\theta_4 = \gamma - (\theta_1 + \theta_2 + \theta_3) \quad (9)$$

### 3.2. Design of the PID Controller

The Proportional-Integral-Derivative (PID) controller [3] is a linear feedback control system comprising three constituent elements: the proportional, integral, and derivative terms. This controller is implemented for the regulation of a 4-DoF robotic manipulator. In this specific context, the PID controller [4] processes input signals, generates output signals, and utilizes digital signal processing. To ensure that the end point of the robotic arm follows the desired reference signal, the initial output value must converge to the setpoint value. The relationship between input and output of a PID is described in the continuous-time domain as follows:

$$\theta(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (10)$$

Where:  $K_p$  is the proportional component,  $K_i$  is the integral component, and  $K_d$  is the derivative component. The working principle of the PID controller applied for controlling

a robot arm is illustrated in Figure 2. It is also a closed-loop control where the error between the setpoint and the output is employed as the input of the PID controller.

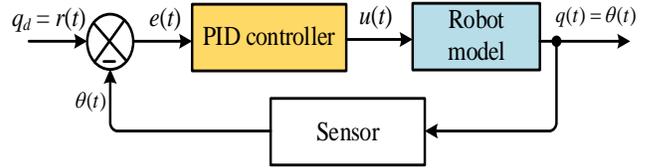


Fig. 2 PID controller to control a multi-DoF robot

Where  $q_d = r(t) \in R^{4 \times 1}$  is the setpoint signal at the robot joints,  $q(t) = \theta(t) \in R^{4 \times 1}$  is the output control signal representing the position of the motor, and  $e(t) = r(t) - \theta(t) \in R^{4 \times 1}$  is the error between the setpoint value and the desired control signal.

### 3.3. Particle Swarm Optimization (PSO) Algorithm for PID Controller

The PSO algorithm is a population-based stochastic optimization technique rooted in the principles of swarm intelligence, designed to identify optimal solutions within a defined search space [5]. Inspired by the social behavior of bird flocking in foraging activities, the PSO algorithm [6] is categorized as a swarm intelligence-based approach. Introduced in 1995 by James Kennedy and Russell C. Eberhart at an IEEE conference, this algorithm has demonstrated significant applicability across diverse domains necessitating the resolution of optimization challenges [7], [8]. The foraging process of an avian flock can be conceptualized as follows: the search space corresponds to the three-dimensional Euclidean space. Initially, the flock exhibits movement with potentially stochastic directional components. Over time, certain individuals within the swarm discover regions of resource concentration. Based on the perceived resource density, these individuals communicate information to conspecifics within proximity. This information propagates throughout the entire population. Consequently, each individual adjusts its velocity (both magnitude and direction) to vector towards the area exhibiting the highest resource availability, informed by the collective intelligence of the swarm. The purpose of the PSO algorithm [9] is to find the optimal values of three scaling factors of a PID controller  $K_p, K_i, K_d$  for a 4-DoF robot system [10] (see Figure 3).

The process of updating the particles is based on the following formula:

$$\begin{cases} v_{i,m}^k = xv_{i,m}^{k-1} + \varphi_1 r_1 (pbest - x_{i,m}^{k-1}) + \varphi_2 r_2 (gbest - x_{i,m}^{k-1}) \\ v_{i,m}^{(k+1)} = wv_{i,m}^{(k)} + c_1 * rand() * (pbest_{i,m} - x_{i,m}^{(k)}) \\ \quad + c_2 * rand() * (gbest_{i,m} - x_{i,m}^{(k)}) \\ x_{i,m}^{(k+1)} = v_{i,m}^{(k)} + x_{i,m}^{(k+1)} \end{cases} \quad (11)$$

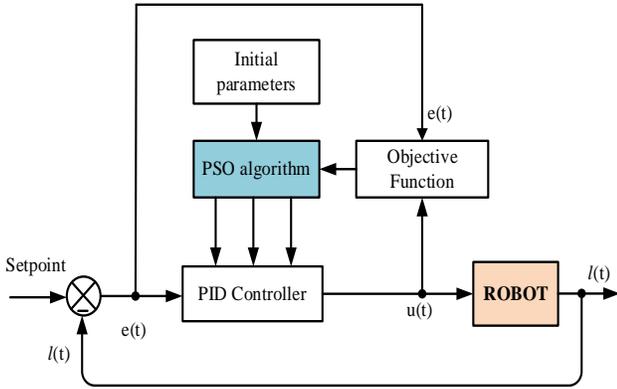


Fig. 3 Block diagram of the PID controller based on PSO

The algorithm for finding the parameters of the PID controller is divided into the following steps:

- Step 1.** Initialize PSO parameters.
- Step 2.** Randomly select the parameters  $k_p, k_i, k_d$  within the upper and lower bounds.
- Step 3.** Compute the objective function based on the ITAE standard of the PID controller.
- Step 4.** Update the position and velocity of each particle in the swarm.
- Step 5.** Select the optimal parameters in each iteration.
- Step 6.** Terminate the loop based on predefined stopping criteria.
- Step 7.** Use local optimization to select the best parameters  $k_p, k_i, k_d$  through robot simulation.

The flowchart of the parameter search process is illustrated in Figure 4. In it, the swarm intelligence algorithm finds the optimal PID controller parameters [9] based on the objective function. The stopping condition for the number of iterations is chosen based on experience in the search process.

**3.4. Design of Fuzzy Logic Controllers**

The fundamental principle of a fuzzy logic controller is predicated on the concept of fuzzy sets, which enable the representation of a parameter’s degree of membership within a set. This contrasts with classical set theory, which employs binary logic and discrete set boundaries. Fuzzy logic control utilizes linguistic variables, fuzzy inference rules, and approximate reasoning to emulate human-like decision-making in complex, nonlinear systems.

This methodology facilitates a decision-making process that approximates human cognitive reasoning. Fuzzy control has gained significant attention in robotic control systems due to its ability to handle nonlinearity, uncertainty, and changing operating conditions. Its successful applications have been observed in various fields, especially in robotic control. Compared to traditional control methods, fuzzy control offers superior performance, greater adaptability, and higher

reliability. The main components of the basic controller consist of three functional blocks: fuzzification, rule base, and defuzzification.

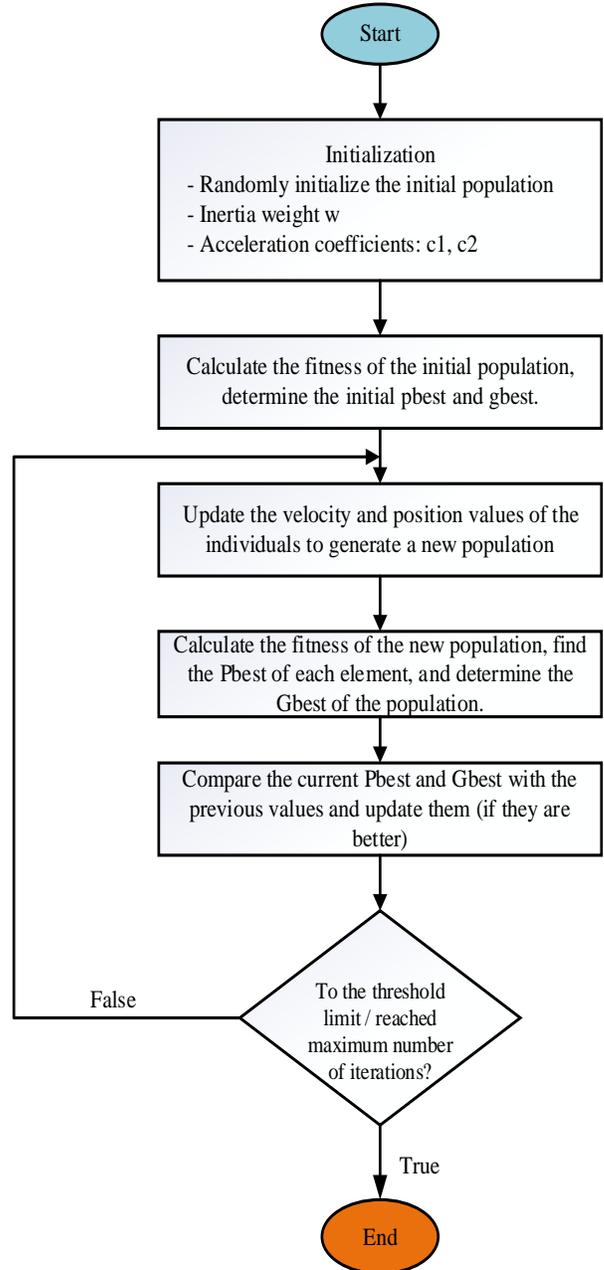


Fig. 4 Flowchart of PID controller parameter search using PSO

**3.5. Particle Swarm Optimization (PSO) Algorithm for Fuzzy Logic Controllers (FLCs)**

The process presented above for the FLC controller shows that its design is largely based on trial and error, as selecting the number, shape, and values of fuzzy sets, as well as choosing the rule base, depends on the designer’s experience. The desired outcome of the FLC controller is only an acceptable result rather than the optimal one.

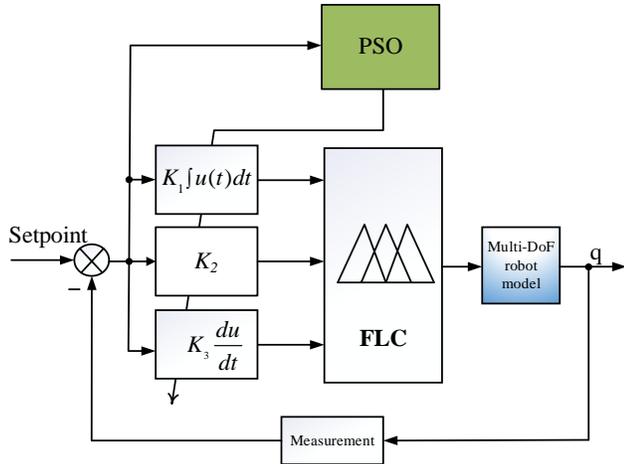


Fig. 5 Fuzzy control diagram for robot position tracking

To attain superior operational characteristics, this study will leverage the PSO algorithm for the automated tuning of the control system’s parameters. The PSO methodology is widely adopted for the automated optimization of FLC’s parameters, specifically encompassing the configuration of membership functions and the structure of fuzzy inference rule sets (see Figure 5). Drawing inspiration from the emergent collective behaviors observed in avian flocks and fish schools, PSO facilitates an efficacious exploration of optimal parameter sets within complex, non-linear search spaces. The incorporation of PSO aims to enhance the performance metrics, adaptability to dynamic conditions, and overall stability of the controller within operational contexts exhibiting multiple sources of uncertainty. The principal objective of implementing the Particle Swarm Optimization algorithm in conjunction with Fuzzy Logic Controllers [14] is to optimize the signal conditioning stages, encompassing both pre-processing and post-processing, for a robotic manipulator possessing four degrees of freedom (4-DoF).

3.5.1. Definition of Fuzzy Sets

- Input signal: Position error  $\Delta q$
- Output signal: control signal  $u$

3.5.2. Number of Fuzzy Sets

With respect to design requirements, the cardinality of fuzzy sets for each linguistic variable necessitates careful selection. Insufficient cardinality compromises the controller’s efficacy. An initial configuration of three triangular membership functions per variable is adopted, with subsequent partitioning as required. It is noteworthy that an excessive number of linguistic variables substantially increases computational complexity. In this analysis, the research team has determined the following linguistic variables:

- Position error: (NB, NS, ZE, PS, PB)
- Derivative of position error: (NB, NS, ZE, PS, PB)
- Output: (NB, NM, NS, ZE, PS, PM, PB)

Membership Functions (Tables 2-5)

Table 2. Position parameters (rad)

Position Parameters (rad)		
Name	Abbreviation	Value Range
Negative Big	NB	[-1 -0.5 -0.1]
Negative Small	NS	[-0.5 -0.1 0]
Zero	ZE	[-0.1 0 0.1]
Positive Small	PS	[0 0.1 0.5]
Positive Big	PB	[0.1 0.5 1]

Table 3. Differential position error parameters (m)

Position Parameters (rad)		
Name	Abbreviation	Value Range
Negative Big	NB	[-1 -0.5 ]
Negative Small	NS	[-1 -0.5 0]
Zero	ZE	[-0.5 0 0.5]
Positive Small	PS	[0 0.5 1]
Positive Big	PB	[0.5 1]

Table 4. Differential control signal parameters

Control Signal Differential Parameter		
Name	Abbreviation	Value Range
Negative Big	NB	-1
Negative Medium	NM	-0.67
Negative Small	NS	-0.33
Zero	ZE	0
Positive Small	PS	0.33
Positive Medium	PM	0.67
Positive Big	PB	1

Table 5. Fuzzy control rules for the robotic arm

DU	E					
	NB	NE	ZE	PO	PB	
DE	NB	NB	NB	NM	NS	ZE
	NE	NB	NM	NS	ZE	PS
	ZE	NM	NS	ZE	PS	PM
	PO	NS	ZE	PS	PM	PB
	PB	ZE	PS	PM	PB	PB

Constructing the Composition Rules

Based on practical experience, the control rules are as presented in Table 5.

Selection of Composition Method

For position control of the robotic arm, the MAX-MIN method is chosen.

Selection of Defuzzification Principle

Defuzzification is the process of determining the precise output value of the controller. The choice of defuzzification method also affects the system response. In this study, the authors use the centroid method.

3.6. Design of Sliding Mode Controller (SMC)

Sliding Mode Control (SMC) is a nonlinear control method with strong adaptability to disturbances and uncertainties, helping the system achieve high stability even

under environmental variations. As shown in Figure 6, this method is based on the feedback control principle, where the system is controlled to converge to a sliding surface and maintain its state on that surface. Once the system reaches the desired signal, it becomes stable and resistant to external disturbances or parameter variations of the model.

Sliding mode control is divided into three phases:

Phase 1: Reaching Mode – The trajectory converges to the sliding surface within a finite time.

Phase 2: Sliding Mode – The trajectory slides along the surface towards the setpoint.

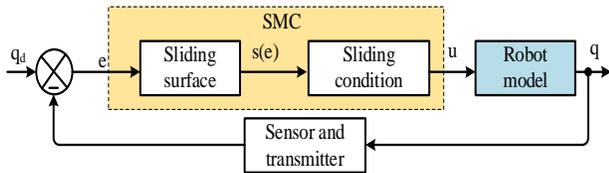


Fig. 6 Sliding mode control diagram for the control system of robots

Phase 3: Equilibrium Point – The intersection of the axis  $e$  and  $\dot{e}$  is a stable system state.

Sliding mode control uses a sliding surface to determine the desired system dynamics, generally expressed as:

$$s = \dot{e} + \lambda e \tag{12}$$

Where:  $e = q_d - q$  the error between the desired value and the actual value,  $\lambda$  is a constant that adjusts the convergence speed.

$$\tau = M(q)(\ddot{q}_d - \lambda \dot{e}) + C(q, \dot{q})\dot{q} + G(q) - K \operatorname{sgn}(s) \tag{13}$$

The system’s stability is ensured using the Lyapunov method by selecting a positive definite function.

$$V = \frac{1}{2} s^2(t) \tag{14}$$

So,  $\dot{V} = s(t)\dot{s}(t) < 0$ .

Thus, if  $\dot{V} < 0$  then with  $V \rightarrow 0$  leads to  $S \rightarrow 0$  and  $e \rightarrow 0$ . Therefore, the sufficient condition for sliding mode control is:

$$s(t)\dot{s}(t) < 0$$

The purpose of the controller is to achieve  $e \rightarrow 0$  where  $e$  is the error signal between the reference input and the system output. To achieve the error signal convergence  $e \rightarrow 0$ , sliding mode control utilizes the sliding function  $s = \dot{e} + \lambda e$ . Specifically, the system must satisfy the Hurwitz polynomial condition, ensuring that all positive coefficients and the roots of the polynomial lie on the left half of the complex plane.

The sliding surface must be designed to ensure  $e \rightarrow 0$  and satisfy the sliding condition  $s(t)\dot{s}(t) < 0$ . This is a crucial aspect of sliding mode control.

### 4. Simulation Results

After applying the PSO algorithm to optimize the PID and FLC controllers for the four-DoF robot system, with a maximum number of iterations of 100, we observe that the controller has been optimized to its best possible performance. Figure 7 describes the convergence of the PSO algorithm for the PID controller.

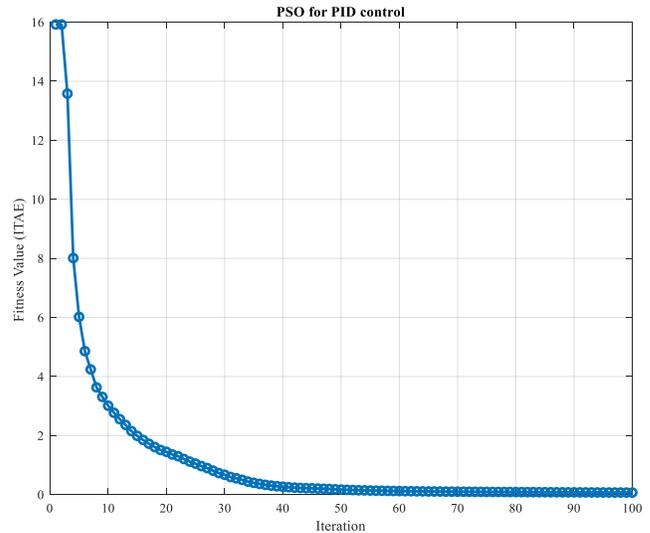


Fig. 7 Convergence of PID parameter search using PSO

Table 6. PSO search parameters in PID controller

PSO Algorithm Parameters	Value
Number of individuals in the population	12
Population size	50
Number of iterations	100
Cognitive coefficient	1.45
Social coefficient	1.45
Inertia weight	2

Table 7. PSO parameters for PID coefficient search

Search Coefficient	Value
$k_{p1}$	2.044915187537121e+02
$k_{i1}$	3.349424655637698e+03
$k_{d1}$	33.069418459651130
$k_{p2}$	3.551874643975046e+03
$k_{i2}$	3.700955730448828e+03
$k_{d2}$	0.994880511416388
$k_{p3}$	1.958522117477832e+03
$k_{i3}$	6.991074847046266e+05
$k_{d3}$	4.339488775468977
$k_{p4}$	28.768002950191168
$k_{i4}$	5.885635377612176e+02
$k_{d4}$	1.466750167666233

The results of the FLC controller, optimized using the PSO algorithm for a 4-DOF robot with 100 iterations, indicate that the fitness has a technical convergence (see Figure 8).

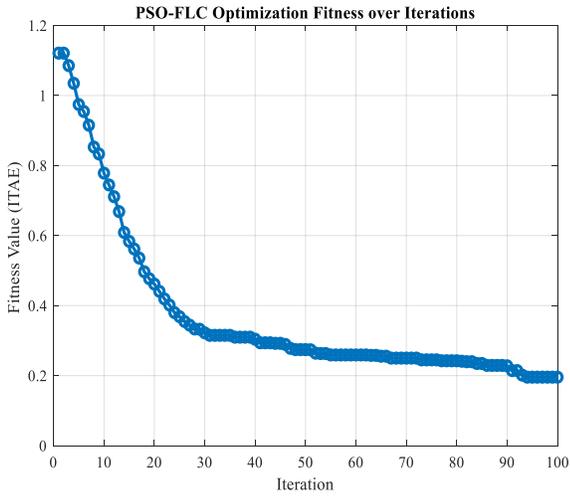


Fig. 8 Convergence of FLC coefficient search using the PSO algorithm

Table 8. PSO tuning parameters for the FLC controller

PSO Algorithm Parameters	Value
Number of individuals in the population	12
Population size	50
Number of iterations	100
Cognitive coefficient	1.45
Social coefficient	1.45
Inertia weight	2

Table 9. PSO parameters for searching FLC coefficients

Search Coefficient	Value
K1	0.8143
K2	11.8478
K3	100.3350
K4	0.1788
K5	86.9640
K6	26.7065
K7	0.2560
K8	0.8661
K9	72.0785
K10	0.3449
K11	1.8686
K12	85.8374

To evaluate the results of the different proposed controllers -PID, SMC, FLC, and PSO - simulations were conducted using MATLAB. The simulation results are presented in Figures 9-12.

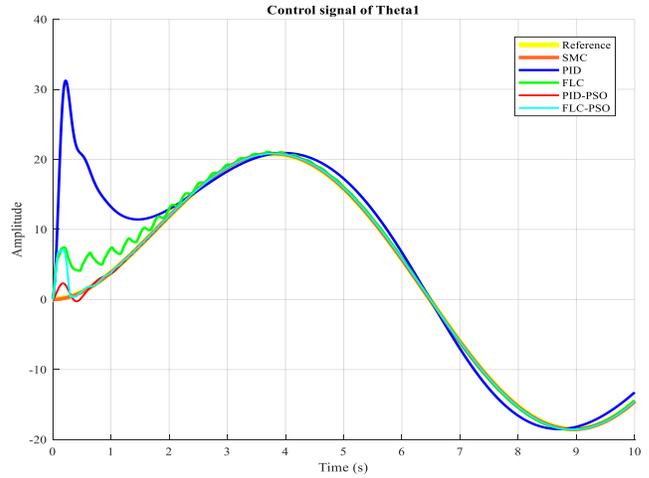


Fig. 9 Output response of Theta1

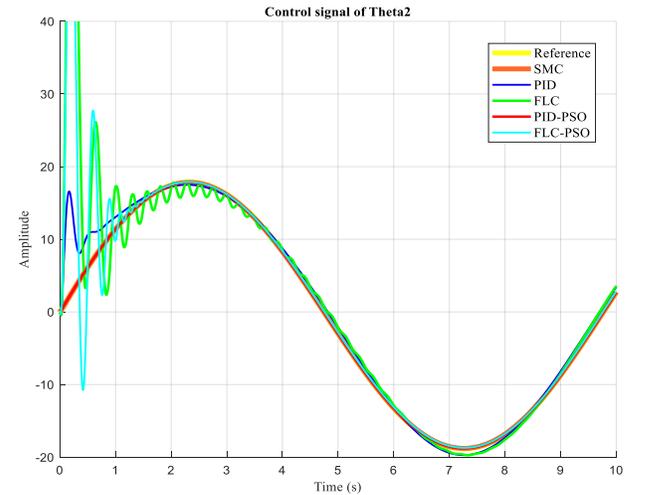


Fig. 10 Output response of Theta2

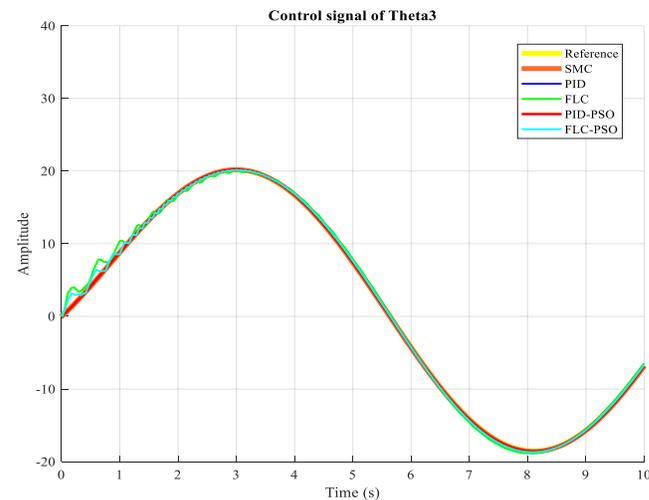


Fig. 11 Output response of Theta3

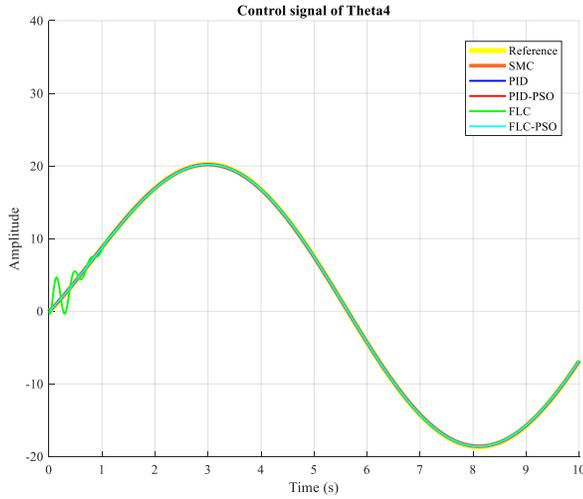


Fig. 12 Output response of Theta4

Table 10. Evaluation of controller quality criteria

Controller	Quality evaluation criteria
PID	- Overshoot: 182.096%, Risetime: 0.05 - ITAE1: 0.03896
PSO-PID	- Overshoot: 35.154%, Risetime: 0.598 - ITAE1: 0.028367303409015
SMC	- Overshoot: 3.644%, Risetime: 2.150 - ITAE1: 0.014721860484709
PSO-FLC	- Overshoot: 0.5982%, Risetime: 2.117 - ITAE1: 0.020829479731508
FLC	- Overshoot: 2.885%, Risetime: 1.751 - ITAE1: 0.189246412720922

Based on the data table, SMC has the best performance with a very low overshoot (3.644%) and the smallest integral error, but it responds slowly. The FLC and PSO-FLC balance between speed and stability, with a low overshoot (~2-3%) and relatively fast response time. PSO-PID improves compared to PID, but still has a high overshoot (35.154%). Traditional PID has the highest overshoot (182.096%) and strong oscillations, making it less effective. If accuracy is the top priority, choose SMC; if a balance between speed and stability is needed, choose FLC or PSO-FLC.

Table 11. Evaluation of controller quality for Theta2

Controller	Quality Evaluation Criteria
PID	- Overshoot: 50%, Risetime: 0.052091 - ITAE2: 0.1888
PSO-PID	- Overshoot: 1.21%, Risetime: 0.994923 - ITAE2: 0.012981523346529
SMC	- Overshoot: 0.962%, Risetime: 1.474 - ITAE2: 0.011142989966548
PSO-FLC	- Overshoot: 117.997%, Risetime: 0.047 - ITAE2: 0.166441259385680
FLC	- Overshoot: 163.116%, Risetime: 0.039 - ITAE2: 0.664113858529447

Based on the data table, SMC provides the best control quality with low overshoot (0.962%) and the smallest integral error (ITAE2 = 0.0111), but the response time is slightly slow. PSO-PID also achieves high performance with low overshoot (1.21%) and a small error, whereas PID has a high overshoot (50%) and a larger error. In contrast, PSO-FLC and FLC exhibit extremely high overshoot (117.997% to 163.116%), causing strong oscillations and instability. Overall, SMC is the best choice for accuracy, PSO-PID balances speed and stability, while PSO-FLC and FLC are not suitable due to excessive oscillations.

Table 12. Evaluation of controller quality for Theta3

Controller	Quality evaluation criteria
PID	- Overshoot: 82.60 %, Risetime: 0.98 - ITAE3: 4.57e-07
PSO-PID	- Overshoot: 27.725%, Risetime: 0.91 - ITAE3: 8.964856184645077e-05
SMC	- Overshoot: 0.50%, Risetime: 1.994 - ITAE3: 0.041309667194915
PSO-FLC	- Overshoot: 159.57%, Risetime: 0.41 - ITAE3: 0.105614332891668
FLC	- Overshoot: 32.52%, Risetime: 0.110 - ITAE3: 0.251414415608863

Based on the data table, SMC provides the best performance with an extremely low overshoot (0.50%) and a small integral error (ITAE3 = 0.0413), but the response time is slow. PSO-PID is a balanced choice with significantly lower overshoot compared to PID (27.725%) and a small error. PID has a high overshoot (82.606%) and a slow response time, causing strong oscillations. The PSO-FLC and FLC have high overshoots (159.573% and 32.524%), making the system less stable. Overall, SMC is the optimal method for accuracy, while PSO-PID is suitable when balancing speed and stability.

Table 13. Evaluation of controller quality for Theta4

Controller	Quality evaluation criteria
PID	-Overshoot:82.617 %, Risetime:0.986 - ITAE4: 3.392e-05
PSO-PID	-Overshoot:27.705%, Risetime: 0.9163 - ITAE4: 0.059082325879194
SMC	-Overshoot: 1.109%, Risetime: 1.915 - ITAE4: 0.044545211418203
PSO-FLC	-Overshoot:187.409%, Risetime: 0.47 - ITAE4: 0.003213812939863
FLC	-Overshoot: 27.705%, Risetime: 0.91 - ITAE4: 0.00399557879510223

Based on the data table presented above, SMC provides the best control quality with a very low overshoot (1.109%) and a small integral error (ITAE4 = 0.0445), but with a slow response time. PSO-PID and FLC have higher overshoot (27.705%), yet they remain more stable than PID. PID has a large overshoot (82.617%), causing the system to oscillate.

PSO-FLC has an extremely large overshoot (187.409%), making it the least stable system. Overall, SMC is the optimal choice for accuracy, while PSO-PID and FLC can be considered if a faster response is needed.

## 5. Conclusion and Future Work

Based on the extensive simulation data generated throughout this investigation, several salient conclusions can be derived:

- (1) The SMC demonstrates superior performance characteristics in terms of both accuracy and stability. In contrast, the PSO-PID control offers a pragmatic trade-off between response speed and system stability, presenting a balanced solution for applications necessitating both rapid dynamics and robust control.
- (2) Conventional PID control and the PSO-FLC exhibit significant limitations, primarily due to their substantial overshoot and inherent susceptibility to instability. Consequently, these control methodologies warrant thorough reevaluation and potential refinement before deployment in systems requiring stringent performance criteria.

- (3) The PSO method has been demonstrated to be one of the best optimization mechanisms for enhancing the control performance of a diverse range of control architectures, including, but not limited to, PID, FLC, and SMC. This algorithm's ability to navigate complex parameter spaces and converge to optimal solutions renders it applicable in control system design.

Future research efforts will be directed towards the design and fabrication of physical multi-degree-of-freedom robotic platforms for the experimental validation of the proposed control strategies. Implementing these controllers on real-world robotic systems will enable a more comprehensive assessment of their performance under realistic operating conditions. Furthermore, this practical implementation will serve to validate and reinforce the theoretical framework established in this study, thereby enhancing its applicability and significance in industrial robotics.

## Funding Statement

This research is funded by Electric Power University under research 2024, Project Granted number ĐTNH.100/2024.

## References

- [1] Mihai Crenganis et al., "Dynamic Analysis of a Five Degree of Freedom Robotic Arm Using Matlab-simulink Simscape," *MATEC Web of Conferences: 10<sup>th</sup> International Conference on Manufacturing Science and Education – MSE 2021*, vol. 343, pp. 1-11, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [2] Palvi Kulkarni, Ovi Kulkarni, and Javed K. Sayyad, "Tuning of a Robotic Arm using PID Controller for Robotics and Automation Industry," *2024 6<sup>th</sup> International Conference on Energy, Power and Environment (ICEPE)*, Shillong, India, pp. 1-6, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [3] Iqbal M. Batiha et al., "Design Fractional-order PID Controllers for Single-Joint Robot Arm Model," *International Journal of Advances in Soft Computing and its Applications*, vol. 14, no. 2, pp. 96-114, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [4] K. Renuka, N. Bhuvanesh, and J. Reena Catherine, "Kinematic and Dynamic Modelling and PID Control of Three Degree-of-Freedom Robotic Arm," *Select Conference Proceedings on Advances in Materials Research*, pp.867-882, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [5] Awatef Aouf, Lotfi Boussaid, and Anis Sakly, "A PSO Algorithm Applied to a PID Controller for Motion Mobile Robot in a Complex Dynamic Environment," *2017 International Conference on Engineering & MIS (ICEMIS)*, Monastir, Tunisia, pp. 1-7, 2017. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [6] Ammar Al-Jodah et al., "PSO-Based Optimized Neural Network PID Control Approach for a Four Wheeled Omnidirectional Mobile Robot," *International Review of Applied Sciences and Engineering*, vol. 14, no. 1, pp. 58-67, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [7] Shoubao Su et al., "Improved Fractional-Order PSO for PID Tuning," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 35, no. 6, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [8] Li-Chun Lai et al., "A PSO Method for Optimal Design of PID Controller in Motion Planning of a Mobile Robot," *2013 International Conference on Fuzzy Theory and Its Applications (iFUZZY)*, Taipei, Taiwan, pp. 134-139, 2013. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [9] Abhishek Kumar Kashyap, and Dayal R. Parhi, "Particle Swarm Optimization Aided PID Gait Controller Design for a Humanoid Robot," *ISA Transactions*, vol. 114, pp. 306-330, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [10] Hossam Hassan Ammar, and Ahmad Taher Azar, "Robust Path Tracking of Mobile Robot Using Fractional Order PID Controller," *International Conference on Advanced Machine Learning Technologies and Applications*, Cairo, Egypt, pp. 370-381, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]

- [11] Fatiha Loucif, Sihem Kechida, and Abdennour Sebbagh, "Whale Optimizer Algorithm to Tune PID Controller for the Trajectory Tracking Control of Robot Manipulator," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 42, pp. 1-11, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [12] M. Khairudin et al., "The Mobile Robot Control in Obstacle Avoidance Using Fuzzy Logic Controller," *Indonesian Journal of Science and Technology*, vol. 5, no. 3, pp. 334-351, 2020. [[Google Scholar](#)] [[Publisher Link](#)]
- [13] Nurul Nafisah Kamis et al., "Error Driven Fuzzy Logic Controller (FLC) for Spherical Mobile Robot: Simulation & Experimental Performance Analysis," *IOP Conference Series: Materials Science and Engineering: 5<sup>th</sup> International Conference on Mechanical, Automotive and Aerospace Engineering*, vol. 1244, pp. 1-13, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [14] Andres O. Pizarro-Lerma et al., "Experimental Evaluation of a Sectorial Fuzzy Controller Plus Adaptive Neural Network Compensation Applied to a 2-DOF Robot Manipulator," *IFAC-PapersOnLine*, vol. 52, no. 29, pp. 233-238, 2019. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [15] Zafer Bingül, and Oğuzhan Karahan, "A Fuzzy Logic Controller Tuned with PSO for 2 DOF Robot Trajectory Control," *Elsevier*, vol. 38, no. 1, pp. 1017-1031, 2025. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [16] Mohamed Fawzy El-Khatib, and Shady A. Maged, "Low Level Position Control for 4-DOF Arm Robot Using Fuzzy Logic Controller and 2-DOF PID Controller," *2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, Cairo, Egypt, pp. 258-262, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [17] Sinan İlgen et al., "The Bees Algorithm Approach to Determining SMC Controller Parameters for the Position Control of a SCARA Robot Manipulator," *European Journal of Science and Technology*, no. 33, pp. 267-273, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [18] Minghuang Qin et al., "Fractional-Order SMC Controller for Mobile Robot Trajectory Tracking Under Actuator Fault," *Systems Science & Control Engineering*, vol. 10, no. 1, pp. 312-324, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]