SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V13I3P103

Volume 13 Issue 3, 28-40, March 2026

Original Article

A Self-Tuning Hybrid Control Architecture Integrating Fuzzy Logic and Evolutionary Computation for Uncertain Robotic Dynamics

Thi-Mai-Phuong Dao[1], Minh-Tuan Phan[2], Trung-Kien Le[2], Anh-Nam Tran Nguyen[2], Tien-Loc Le[3], Tien-Dung Nguyen[2], Duy-Thuan Vu[2], Ngoc-Khoat Nguyen[2*]

[1]*School of Electrical and Electronic Engineering (SEEE), Hanoi University of Industry, Hanoi, Vietnam.*
[2]*Faculty of Control and Automation, Electric Power University, Hanoi, Vietnam.*
[3]*Faculty of Postgraduate Studies, Lac Hong University, Dongnai, Vietnam.*

[*]*Corresponding Author : nkn.research@gmail.com*

Received: 05 December 2025 Revised: 05 January 2026 Accepted: 04 February 2026 Published: 23 March 2026

Abstract *- A hybrid control approach applying fuzzy logic and evolutionary computation techniques, which can be considered an efficient methodology in an industrial robot manipulation system, is studied in this paper. The proposed approach combines the flexibility of fuzzy logic techniques with the optimization capabilities of evolutionary computation to improve motion accuracy, sensor-based decision-making, and operational robustness in nonlinear and uncertain environments. A hybrid control model is proposed, where key parameters of the fuzzy controller, such as membership functions and fuzzy rules from a swarm-based optimization algorithm with an auto-tuning organization for biological systems to work on input-output mappings, are automatically adjusted. This adaptive approach accelerates the convergence rate, minimizes steady state errors, and enhances the ability of a robot to handle external loads and environmental disturbances. Through numerical simulations, the results show that this newly developed method exceeds in trajectory accuracy and is more stable as well as adaptable to different loads than classic PID and independent fuzzy logic controllers. These results also validate the efficacy of the hybrid fuzzy-evolutionary control for the realization of intelligent, autonomous, and highly adaptive industrial robotic systems.*

Keywords *- Industrial robot, Fuzzy Logic Control, Evolutionary algorithm, Parameter optimization, Hybrid control system.*

1. Introduction

Modern industrial robot arms are programmable electromechanical systems and are now widely used in automated production lines. These robots are mainly used to perform repetitive tasks at high speed and with relatively stable accuracy. According to Ji and Wang [1], the increasing application of robots in industry has helped improve work efficiency, increase flexibility in operation, and contribute to increasing production productivity. A typical industrial robot is composed of many rotary and translational joints, which are arranged in a multi-degree-of-freedom configuration. Thanks to this arrangement, the robot can work in three-dimensional space and perform the necessary movements for production. Compared to humans, robots have better repetitive work capabilities, can withstand larger loads, and have higher movement accuracy. A robot arm usually consists of main parts such as the mechanical part, the drive unit, and the central controller. The drive system can be a servo motor or a hydraulic or pneumatic system, depending on the torque and load requirements of each specific application. The central controller is responsible for controlling and synchronizing the movement of the axes during the robot's operation. In addition, the robot is equipped with sensors such as position sensors, force sensors, or vision systems to collect feedback signals. These sensors work in conjunction with appropriate end parts for each job, for example, a welding torch or a gripper. Based on their kinematic structure, industrial robots are often divided into Cartesian, cylindrical, SCARA, and articulated types. Choosing the appropriate type of robot is important because it directly affects the workspace and the system's operational efficiency, as Dzedzickis et al. [2] stated.

Although industrial robotic arms are widely used in tasks such as assembly, welding, and material processing, precise control of these systems still faces many challenges. The main reason stems from the nonlinear characteristics in the dynamics of the robot, and the system parameters are often difficult to determine fully and accurately. Furthermore, the close connection between robot joints makes the construction of kinematic and dynamic models very challenging. As a result, traditional control methods often struggle to fully meet the requirements for noise immunity and model error compensation, which can easily lead to reduced stability or increased trajectory tracking errors [3]. In actual operation conditions, robots are also affected by many adverse factors such as changes in friction, structural resonance, and disturbances from the surrounding environment. In addition, temperature fluctuations or instability of lighting conditions can also significantly affect the performance of the system. When robots are not equipped with adaptive control strategies or a sufficiently high level of robustness, adverse impacts from the environment as well as from the system itself can quickly degrade tracking accuracy and affect the overall stability of the robot [5]. In addition, in practical industrial applications, robots must also meet stringent requirements for safety and reliability, including the ability to detect, diagnose, and handle errors in a timely manner, while ensuring effective coordination between humans and robots. According to [6], the lack of suitable fault tolerance mechanisms is considered one of the important causes of reduced continuous operation time and affects the overall reliability of the system.

Given the above limitations, many recent studies show a trend of shifting from rigid model-based linear controllers to more intelligent control architectures capable of effectively handling nonlinear characteristics and system uncertainties. Review works have classified advanced control methods for robots into six main groups, including adaptive control, sliding mode control, model-based predictive control, robust control, fuzzy logic, and artificial neural network-based methods [7]. In addition, studies on compliance force control also show a clear shift towards physical interaction problems, in which machine learning techniques are increasingly applied to real-time parameter tuning [8]. Recently, reinforcement learning, along with simulation-to-reality transfer methods, has been considered a potential approach to enhance the flexibility and adaptability of robots in multi-degree-of-freedom workspaces [4].

For multi-joint robotic systems, the tight dynamic linkage between joints, along with the nonlinear characteristics that change with load, makes system control very complex. Under such conditions, traditional controllers such as PID or LQR often reveal many limitations, especially when it is necessary to simultaneously ensure the stability and noise immunity of the system. Given this reality, Fuzzy Logic Control (FLC) is considered a suitable solution, because this method allows the construction of control laws based on human knowledge and experience in the form of language, instead of relying entirely on the exact mathematical model of the system [9]. Many experimental studies show that fuzzy logic controllers can significantly improve the accuracy of trajectory tracking, while reducing the influence of model error and uncertainty of system parameters [10]. In robotic applications, fuzzy logic controllers often use position error and its derivative as input signals, then represent them through linguistic variables such as “Big Negative”, “No”, or “Big Positive”. Therefore, the controller design process requires the appropriate selection of member functions as well as a rule-based structure to ensure a balance between control efficiency and computational cost [11].

Based on the above analysis, this study proposes the construction and implementation of a fuzzy logic control framework for a robotic arm, in which input signals are formed from control errors, and member functions are determined based on the knowledge and experience of experts.

The selection of an FLC aims to exploit the ability to maintain the stability and robustness of the controller against external disturbances, thereby meeting the requirements of modern industrial automation systems.

2. Literature Review

Many recent efforts in the literature are about how to improve suitable robots by dealing with nonlinearity, the image combination of enhancement between images, and unknown parameters. The fuzzy logic technique is currently used widely in the industry, as it is a good substitute for classical control techniques. While PID controllers remain widely used and are the subject of a vast research effort in the literature, they have shown limitations in the presence of load variation, friction, or interaction between joints as well.

This move to intelligent control is backed by observational data from several classic studies. It is then evident that fuzzy logic can be a very powerful alternative to many problems (and particularly in the field of control). In [12], a fuzzy inference system of 9 rules was developed in controlling a 3-DOF robotic arm. This work used two inputs: the tracking error and its derivative. The results demonstrated that the dominant feature of the present control method was superior to many traditional methods. In [13], a new framework is proposed to combine a time-lock estimation and an appropriate fuzzy system for the friction compensation. The integration of the hybrid approaches enriched tracking accuracy to more than 90%. Of course, as soon as one moves to such areas as soft or multi-joint robotics, which are extremely complex in terms of modeling, modeless fuzzy controllers show their force. We did not reach the very low RMS ethical care error (0.28-0.54 mm), but we demonstrated the controllability resistance class to environmental disturbances [14].

In recent years, a lot of work has been dedicated to the hybrid control design in robotic systems, with more intricate dynamical models. It has been proven in some references [15] that the super-torsion fuzzy logic controller based on hard-soft interaction control can significantly improve the tracking stability of the system and disturbances in the boundary-element case. Furthermore, fuzzy neural networks combined with fuzzy logic controllers perform better than traditional PID in cases that inertia load and friction at the robot joints are included with systems [16]. Accompanied by this trend, optimization approaches are receiving more attention, and they play an important role in controller design. One example is that the PSO algorithm has been used to optimize the PD-Fuzzy controller, in which the method worked by applying evolutionary algorithms for the purpose of optimizing membership functions [17]. Besides industrial robots, fuzzy logic has also been employed in mobile robotic tractors with great success for the system stability during harvesting operations when there are many disturbances resulting from unpredictable ground circumstances and variable load conditions [18].

3. Forward Dynamics of a 4-DOF Robot

Consider a 4-DOF robot model as shown in Figure 1. Using the Denavit–Hartenberg method to determine the several parameters, i.e., a_i; α_i; d_i and θ_i.

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

i_i	a_i	α_i	d_i	θ_i
1	a_1	0	0	θ_1
2	a_2	0	0	θ_2
3	a_3	0	0	θ_3
4	a_4	0	0	θ_4

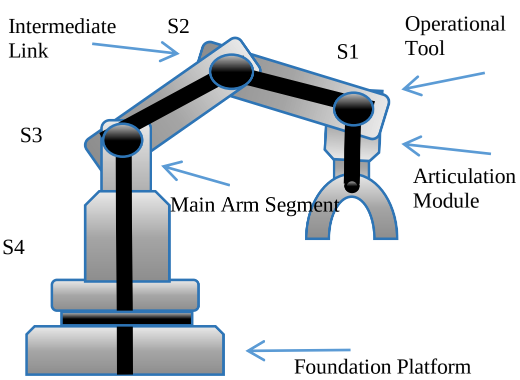

Fig. 1 An illustration of typical robotic arm models

Where:
a_i: the distance along the x_i axis between the z_{i-1} and z_i
α_i: the angle of rotation between the z_{i-1} and z_i axes about the x_i axis;
d_i: the displacement along the z_i axis from the x_{i-1} axis to the z_i axis;
θ_i: the rotation angle from the x_{i-1} axis to the x_i axis about the z_i axis.

A generalized homogeneous transformation matrix for kinematics is given below:

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

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 represents the spatial relationship between the base frame (frame 0) and the end-effector frame (frame 4)

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

3.1. Robot Dynamics

To establish the robot dynamics, the following steps need to be executed:
Step 1: Compute the total kinetic energy:

$$K(\theta,\dot{\theta}) = \sum_{i=1}^{n}\left(\frac{1}{2}m_i v_{Ci}^T v_{Ci} + \frac{1}{2}\omega_i^T I_i \omega_i\right) \quad (3)$$

Step 2: Calculate the total potential energy:

$$U(\theta) = \sum_{i=1}^{n}\left(-m_i g^T P_{Ci} + U_{ref\ i}\right) \quad (4)$$

Step 3: Applying the Lagrange equations:

$$L(\theta,\dot{\theta}) = K(\theta,\dot{\theta}) - U(\theta) \quad (5)$$

- Using the Lagrange differential equations:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = \tau \quad (6)$$

Step 4: Using the general dynamic equations to compute the three matrices M, C, and G:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau \quad (7)$$

Where:
q : Joint position vector
$\dot{q}$: Joint velocity vector
$\ddot{q}$: Joint acceleration vector
τ: Torque applied to the robot
M: Inertia matrix
C: Coriolis and centrifugal forces

4. Control Methods

4.1. Classical PID Controller

4.1.1. Concept

In operation, the algorithm operates by continuously calculating an error that represents a delta between a measured process variable and a desired setpoint. To correct or compensate for this error, the input of the controller is adjusted to drive the system in one direction toward a desired state. The key advantage of a PID controller is that it works, assuming the mathematical model of the system is known or even defined, or an accurate approximation. Universality of PID does not mean a "one size fits all" and to realize best performance, the PID parameters should be carefully adjusted with respect to the system dynamics (see Figure 2).

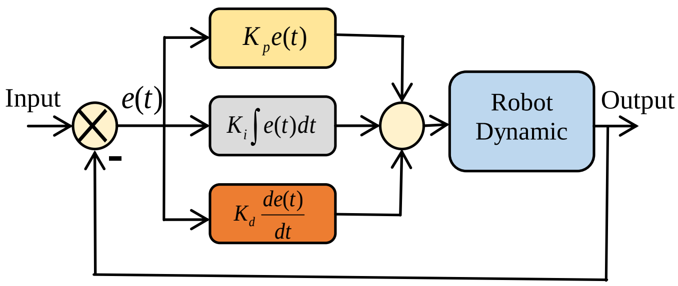

Fig. 2 A block diagram of the PID controller

4.1.2. Particle Swarm Optimization (PSO) Algorithm

A PSO is an evolutionary optimization technique based on a random population approach, initially proposed by Kennedy and Eberhart in 1995. The PSO is a swarm intelligence algorithm designed to imitate the searching behavior of group particles moving in search space and adjusting their direction according to their own best position value as well as the global one. In this vein, it is conducive to convergence to a global optimal solution in various cumbersome optimization tasks. Figure 3 depicts the convergence, one of the most important processes of a PSO algorithm.

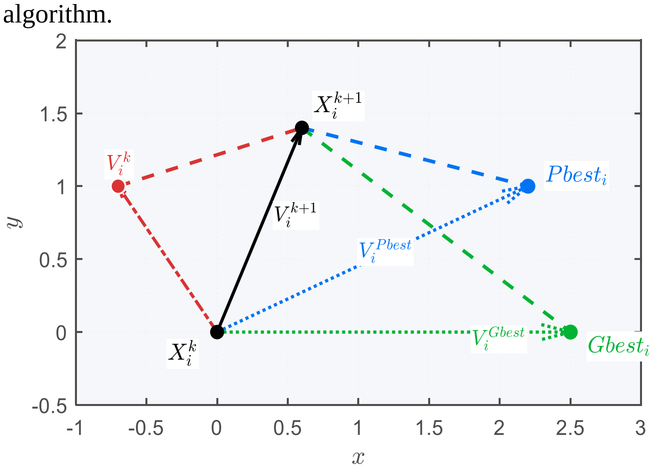

Fig. 3 Convergence description of the PSO algorithm

The velocity updating equation can be calculated as:

$$v_i = wv_i + c_1 r_1 (pbest_i - x_i) + c_2 r_2 (gbest - x_i) \quad (8)$$

The position updating equation should be expressed in (9).

$$x_i = x_i + v_i \quad (9)$$

Where:

$v(t)$: the current velocity of the particle

w: the inertia weight

$c_1$and c_2: the acceleration coefficients

r_1 and r_2: random numbers uniformly distributed between 0 and 1.

$pbest$: the personal best position of the particle

$gbest$: the global best position found by the entire swarm

A typical flowchart of the PSO algorithm is drawn in Figure 4.

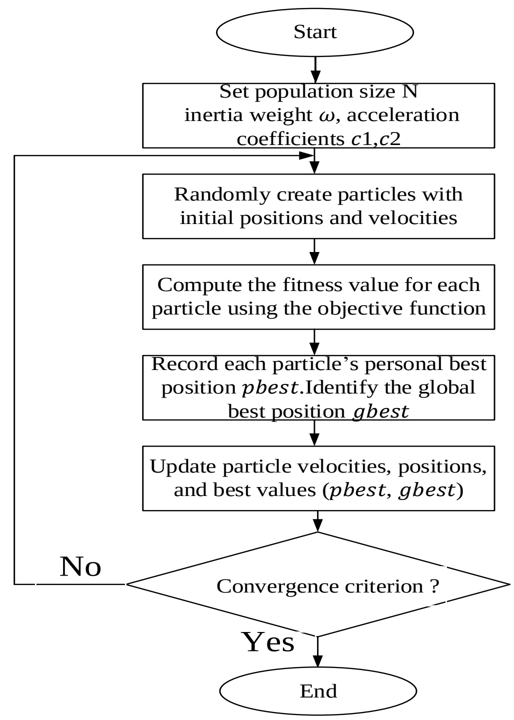

Fig. 4 A typical flowchart of the PSO algorithm

The PSO is motivated by the social learning behavior of living organisms, especially when exploring optimal solutions within a multi-dimensional search space. The PSO optimisation works as follows: Let the first step randomly initialise a population of particles, each particle being an individual that represents a solution to a problem. Particles' movement and speed are modified by the iterations under the guidance of two main vectors: one related to the best position of an individual particle (Pbest), and another that indicates the best position found so far by any particle in a swarm (Gbest). The swarm then begins looking through space using control parameters, celerity coefficients c1 and c2, as well as the current velocity.

In particular, the PSO algorithm is realised by a set of critical stages. Initially, the size and some similar parameters and acceleration coefficients are determined. Then the particles are randomly placed and moved throughout the search space. In every iteration, each particle is evaluated using the objective function based on which its fitness value is calculated. With this outcome, the optimum position of each particle (pbest) is updated, and so is the population best (gbest). Finally, the particle positions and velocities are updated to the new iteration. This step is iterated indefinitely until a termination criterion is satisfied or reaches a certain number of iterations.

4.1.3. Tuning PID Parameters using PSO

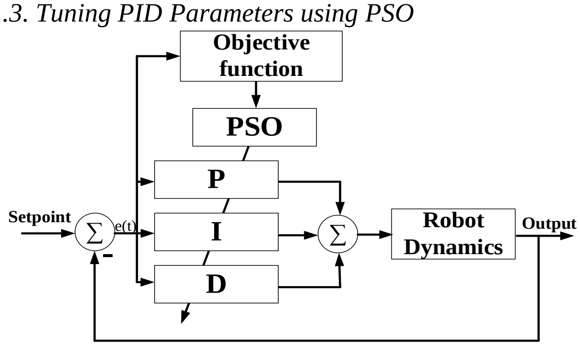

Fig. 5 A block diagram of the PID–PSO controller

In the development of automatic control systems, optimization in terms of Proportional-Integral-Derivative (PID) controllers by employing robust optimization algorithms, including PSO, has posed a new paradigm to achieve the maximum performance among intricate architectures. The principle of this controller is illustrated in Figure 5. Even though PID controllers demonstrate experimentally that they make the system stable along with better tracking accuracy, finding desired gain parameters is still an essential issue, in particular for nonlinear and/or time-variant systems. Therefore, the application of PSO to automatic PID controller tuning is far superior to conventional methods in terms of performance. Because it simulates particle swarm behavior, PSO leads to ideal P, I, and D values and involves no or minimal human work. In complex and dynamic systems, it ensures an efficient, stable, and accurate solution. Finally, the PSO and PID strategies merge to maximize the flexibility and adaptability of sophisticated control architectures in current-day deeply uncertain industrial surroundings.

4.2. Fuzzy Controller

Fuzzy controllers have made a significant contribution to the development of human-like reasoning and decision-making by adopting linguistic rules, rather than accurate mathematical models. Figure 6 shows the principle of a control system applying a fuzzy logic controller. Also, Figure 7 describes a PD-type fuzzy logic controller, which can be considered a typical fuzzy logic architecture.

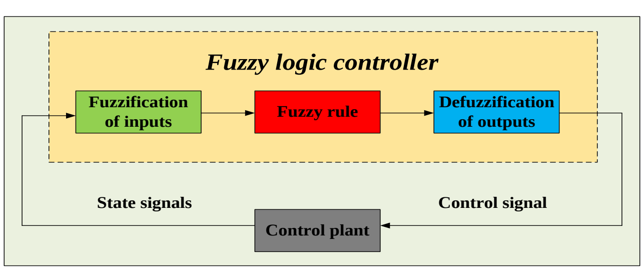

Fig. 6 A basic block diagram of a fuzzy logic controller

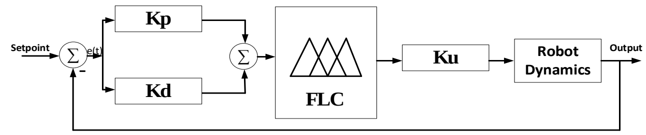

Fig. 7 The PD–Fuzzy controller is suitable for the model

- Fuzzy rule construction in a fuzzy logic controller. A set of fuzzy rules can be built as given in Table 2.

Table 2. Designing fuzzy rules for the model

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

- Values for the fuzzy rules are illustrated as membership functions shown in Figures 8-10.

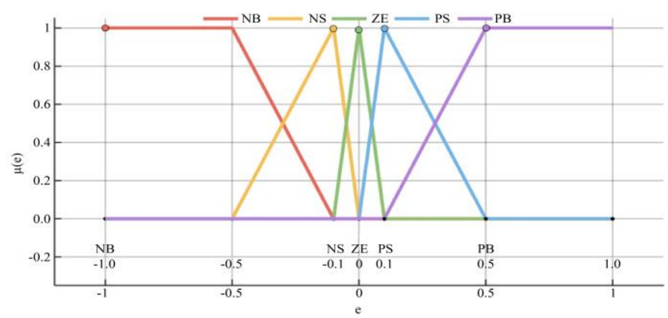

Fig. 8 Input variable "e"

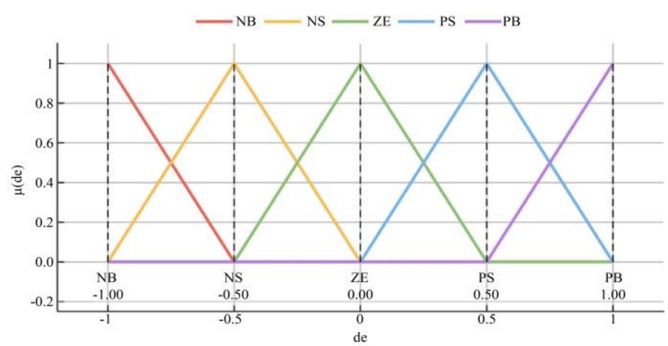

Fig. 9 Input variable "de"

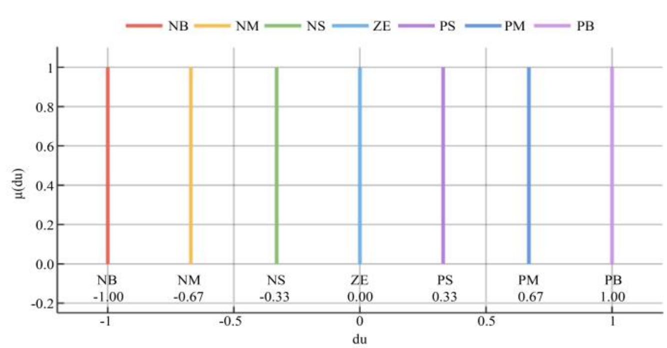

Fig. 10 Output torque applied to the robotic arm

4.2.1. Tuning PD Parameters using PSO in Combination with a Fuzzy Controller

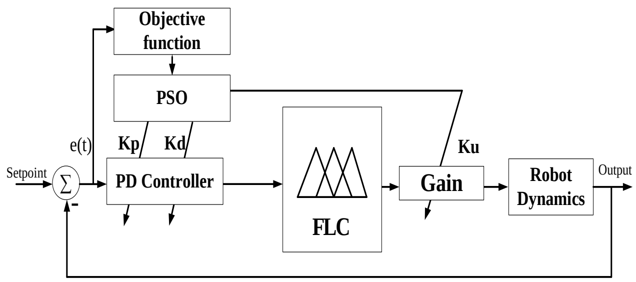

Fig. 11 The PD–Fuzzy controller is suitable for the model

In this study, a novel hybrid control strategy is introduced, which combines the strengths of PEMO and synthetic controllers such as PD and FLC. The control signal (the error signal) of the controller is considered the difference between a reference input and the output fed to a PD controller in this study. The Kp and Kd coefficients of the PD controller are selected by the PSO algorithm. Then, the output of the PD controller is fed to FLC, and after passing through the FLC, this signal is multiplied by Ku and sent to the dynamics of the robot (see Figure 11).

Stability time, percent overshoot, steady-state error, and tracking performance with respect to changes in system parameters are quantified for the proposed controller.

5. Simulation Results

On the basis of the results in the derivation of the dynamic model and synthesis, this section presents the results and analysis to better understand the presented control structure. To ensure a fair comparison, we test all methods on the cloud robotic 4-DOF robot arm system with uniform set-ups and the same initial conditions and reference trajectories.

The performance limitation of the dual-mode weaver base filters is also evaluated quantitatively and compared by using two strict performance indices: Integral Time Absolute Error (ITAE) and Integral Time Square Error (ITSE).

In the comparative analysis, various control methods such as classical PID and PSO-optimized PID (called PID-PSO), PD-Fuzzy controller, proposed in this study, as well as its PSO-optimized version called PD-fuzzy-PSO, were taken into consideration. It has been demonstrated that there are large differences in the regulatory ability across those methods based on experimental observations. It is shown that sharp differences in trajectory tracking accuracy, transient response factors such as settling time and peak overshoot, and steady-state residual amplitude, along with offset magnitude between separate controllers, are revealed, yielding a global view of each controller's level of robustness guarantees and final end-point precision.

Table 3. PSO algorithm parameter settings

Algorithm Parameters PSO	PID Controller	PD-Fuzzy Controller
Number of variables	12	12
Swarm size	100	100
Max iterations	100	100
Personal coefficient	1.5	1.5
Social coefficient	1.5	1.5
Inertia weight	0.09	0.09

Table 4. Optimized values for the PID controller

PSO for PID Controller	ITAE	ITSE
K_{p1}	202.9870	209.98
K_{i1}	63.4039	62.988
K_{d1}	47.9111	54.415
K_{p2}	164.8549	164.4
K_{i2}	43.4333	42.142
K_{d2}	26.0514	26.043
K_{p3}	74.8497	74.964
K_{i3}	57.8722	63.33
K_{d3}	8.3091	7.5319
K_{p4}	168.2827	154.09
K_{i4}	45.2190	26.432
K_{d4}	22.7250	91.838

Table 5. Optimized values for the PD–Fuzzy controller

PSO for PD-Fuzzy Controller	ITAE	ITSE
K_{p1}	3.4461	3.2052
K_{u1}	99.6908	40.0018
K_{d1}	1.0844	58.5211
K_{p2}	0.1361	0.9519
K_{u2}	52.1446	39.4544
K_{d2}	1.9448	1.7081
K_{p3}	0.2417	40.1557
K_{u3}	57.3532	40.8583
K_{d3}	1.5470	1.1610
K_{p4}	0.0811	39.6175
K_{u4}	85.4102	42.1043
K_{d4}	1.5502	1.9018

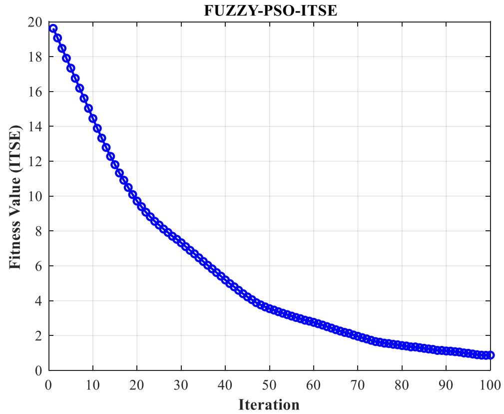

Fig. 12 Convergence of fuzzy–ITSE

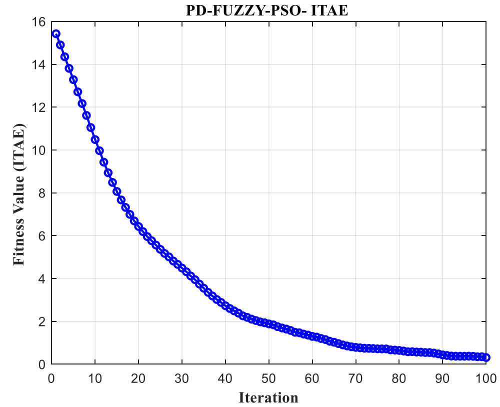

Fig. 13 Convergence of fuzzy–ITAE

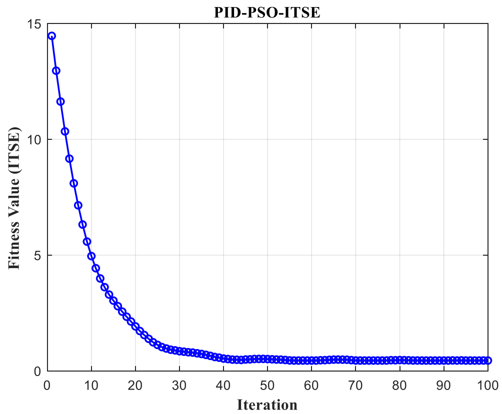

Fig. 14 Convergence of PID–ITSE

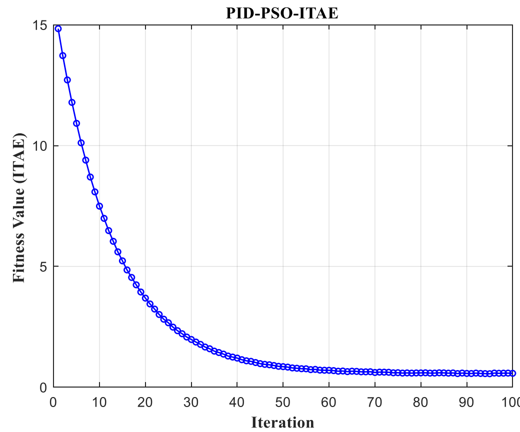

Fig. 15 Convergence of fuzzy–ITSE

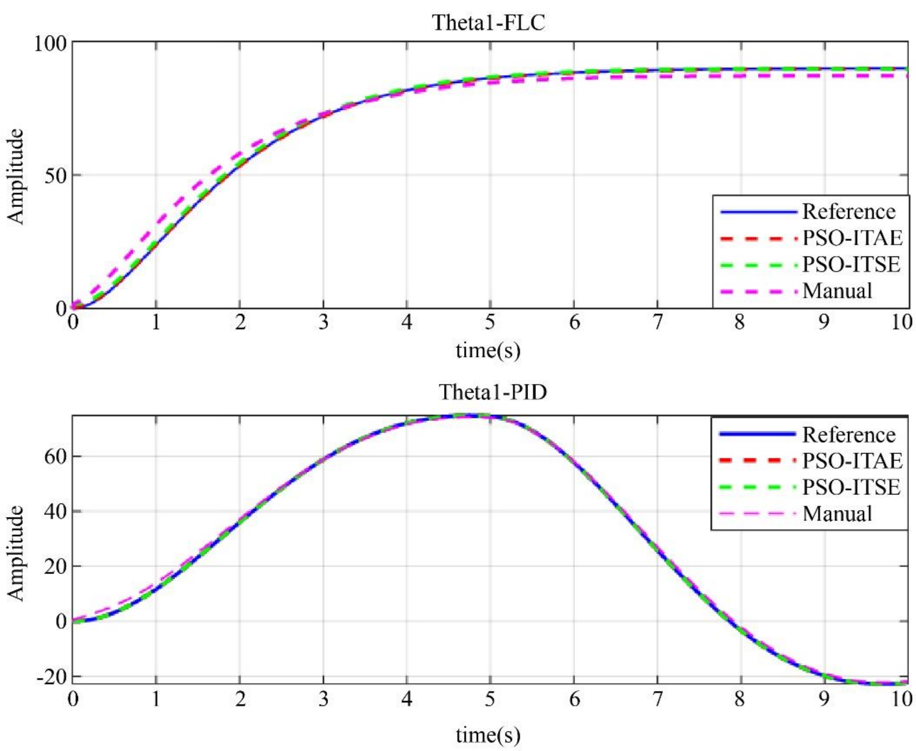

Fig. 16 Simulation results of Theta1

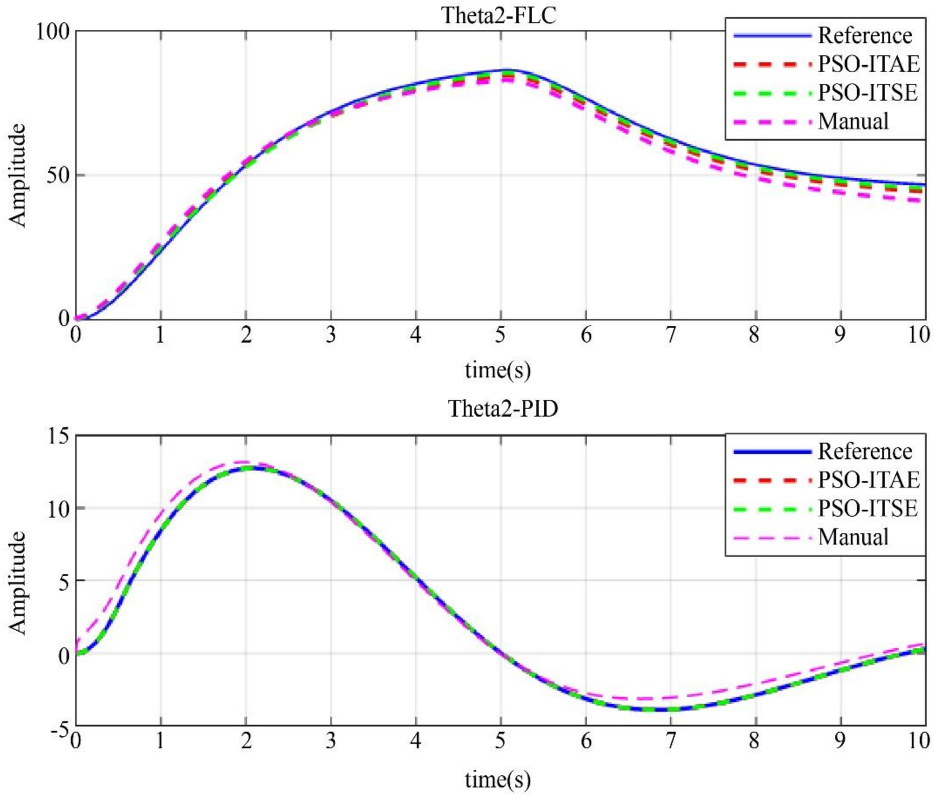

Fig. 17 Simulation results of Theta2

Table 6. Control performance parameters of THETA1

Controller	Performance Evaluation using the Objective Function ITAE1				Performance Evaluation using the Objective Function ITSE1			
	Overshoot (±)	Rise time (S)	Settling time (S)	ITAE1	Overshoot (±)	Rise time (S)	Settling time (S)	ITSE1
PID	7.5	2.8	6.8	0.6813	7.5	2.9	6.9	0.01011
PID-PSO	5.5	2.5	6.2	0.6407	5.6	2.5	6.1	0.01060
PD-fuzzy	3.0	2.2	5.4	0.1042	3.3	2.1	5.2	0.02419
PD-fuzzy-PSO	1.5	2.0	4.8	0.0973	1.9	2.0	4.7	0.002878

It can be seen from Table 6 that the surveyed controllers show considerably different control performance. The standard PID controllers, when the system is stable, show an overshoot (≈7.5%) and a large transient time (stabilization = 6.8 s approximately), also showing much steady state error. The performance of fine-tuning the PID parameters with the PSO algorithm was greatly improved: overshoot reached 5.6% and settling time decreased from 9.4 s to 6.2 s, suggesting better adaptability to transient responses.

In the case of a PD-fuzzy controller, fuzzy logic has largely reduced oscillations and transient errors. It is shown that the overshoot was reduced to 3.3% and the system stabilization time was reduced to 5.4 s with a smoother response and a shorter transient process than conventional PID controllers used. However, ITSE is still high due to the absence of global optimization. There is a better superiority index for the other three when compared with the PD-fuzzy-PS controller.

The system has relatively good trajectory tracking properties, the overshoot value is almost zero (1.9%), and the settling time of 4.8 s is the smallest while ITAE and ITSE values are also the lowest at 0.0973 and 0.002878, respectively; it keeps small response oscillation at zero with a much smaller steady-state error close to zero and with a zero steady-state error. It indicates that the fuzzy control and PSO swarm algorithm is combined more superiorly: not only stability but also reference signal tracking convergence rate are drastically increased, and the relative long-term deviation of the system is decreased.

Table 7. Control performance parameters of THETA2

Controller	Performance Evaluation using the Objective Function ITAE2				Performance Evaluation using the Objective Function ITSE2			
	Overshoot (±)	Rise time (S)	Settling time (S)	ITAE2	Overshoot (±)	Rise time (S)	Settling time (S)	ITSE2
PID	10.2	1.85	6.40	0.4226	9.8	1.80	6.35	0.004463
PID-PSO	7.5	1.60	5.80	0.4032	7.2	1.55	5.65	0.0044
PD-fuzzy	13.8	1.45	7.10	3.5659	12.9	1.40	6.90	0.06163
PD-fuzzy-PSO	5.4	1.30	5.10	3.5109	5.0	1.25	5.00	0.01814

The data summarized in Table 7 show a rather significant difference between the control methods considered. Despite using a traditional PID controller, the system operated stably and did not experience any loss of control throughout the operation. However, upon closer examination of the transient response, the inherent limitations of the PID structure began to become apparent.

Specifically, the system exhibited a relatively large overshoot, approximately 10%, and the time to reach a steady state extended to about 6.4 seconds. As a result, the trajectory tracking process at the second joint was significantly slower, while the steady-state error persisted and could not be ignored. From the above results, when used independently, the classical PID controller does not really meet the requirements of the nonlinear control problem involving Theta 2 joints.

The application of the PSO algorithm to adjust the parameters of the PID controller resulted in a significant improvement in system performance. The PID-PSO controller provided a faster response, demonstrated by a shortened stabilization time of approximately 5.8 seconds, while the overshoot was significantly reduced to about 7.5% Compared to the original PID, both the ITAE and ITSE values are lower.

As shown, the system collects signals better and has fewer errors after a period of operation. In general, optimizing PID with PSO helps the controller achieve a good balance between response speed and stability. This is an undeniable advantage of such a PID controller.

On the other hand, the PD-Fuzzy controller shows good results by using fuzzy logic. The system response is smoother, oscillations disappear faster, and the convergence speed increases. However, since there is no global optimization, the ITSE value remains high at 0.06163. Therefore, the cumulative error remains quite large, especially when the reference trajectory changes abruptly. Nevertheless, the results obtained still show that fuzzy control is very flexible and adapts well to nonlinear systems.

Of the methods tested, the PD-fuzzy-PSO system can be considered the best solution. Combining fuzzy logic with the optimal PSO algorithm truly yielded remarkable results. This system demonstrated excellent response quality, with minimal overshoot of approximately 5.4%, and the shortest stabilization time of approximately 5.1 seconds.

The ITAE and ITSE values reached their lowest levels, at 3.5109 and 0.01814, respectively. As a result, the system operated stably, closely followed the reference signal, and showed virtually no steady-state errors.

It is noted that though the conventional PID controller ensures the stability of the system, it also introduces a sluggish response with a relatively high percentage of overshoot, approximately 8.6, and a high settling time of about 5.9s. The ITAE3 and ITSE3 values also indicate a high amount of accumulated error, and hence, it is noted that the trajectory tracking potential of this controller is still limited.

It is observed that the quality of response performance of the system when the PID controller's parameters are tuned using PSO (PID-PSO) has improved significantly. The rise time is 2.1s, and the %OS becomes 6.3%, while the settling time is approximately 5.3s. Therefore, with respect to the performance index, the ITAE3 (ITSE) values of 0.2201 (0.0014) are considerably low compared to the PID conventional controller design.

On the part of system response, it was found that the performance due to nonlinear characteristics in fuzzy logic of the PD-fuzzy controller had increased speed, and there were fewer oscillations as compared with PID. But the ITAE3 value of 0.3560 was higher than those for the upper methods, which suggested that a certain error had accumulated in the system response.

This was primarily due to the non-global optimal parameters of the PD-fuzzy controller, and it is operationally dependent on the structure of the controller.

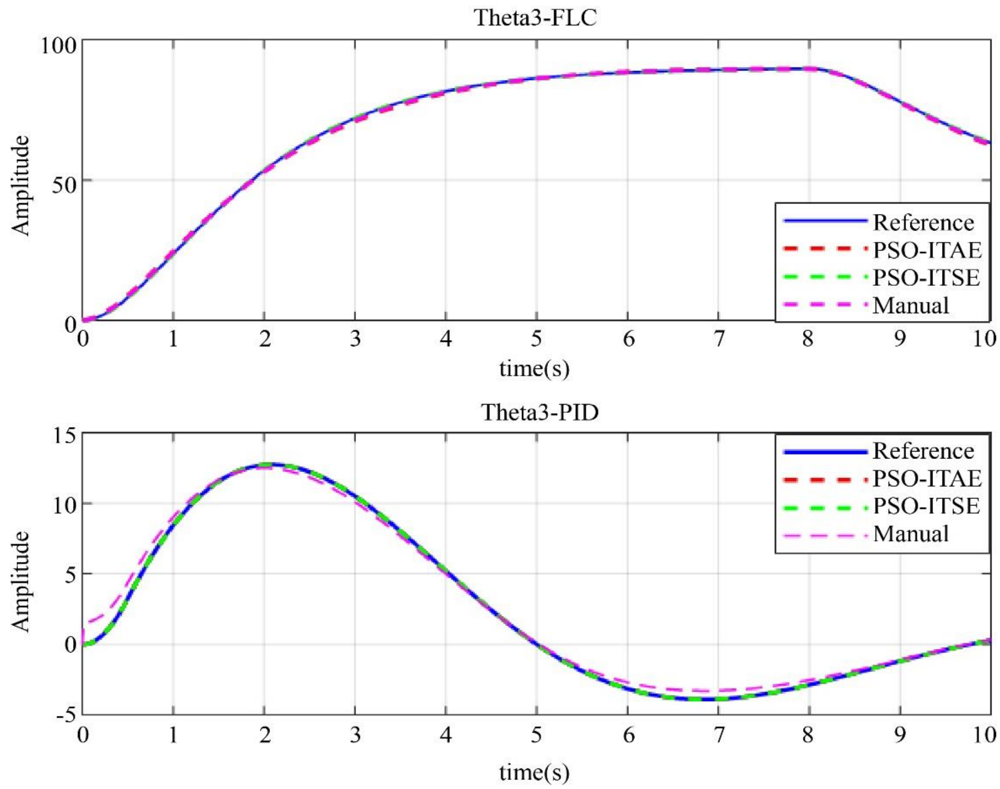

Fig. 18 Simulation results of Theta3

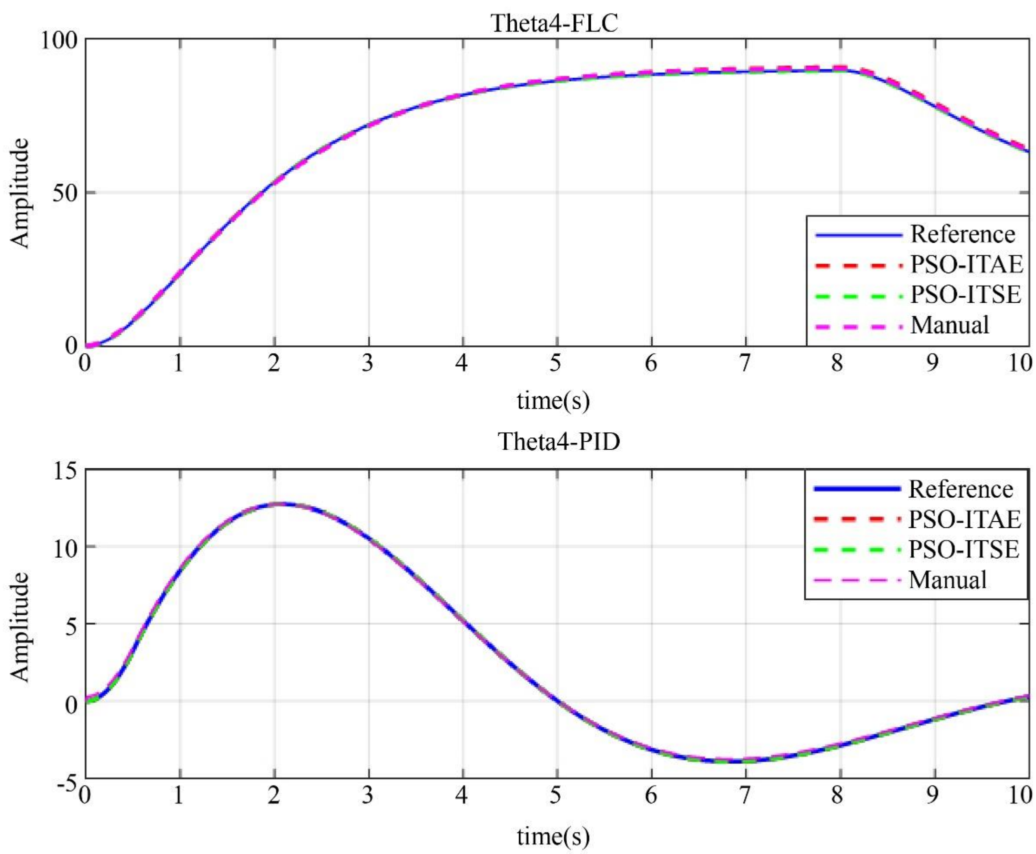

Fig. 19 Simulation results of Theta4

Table 8. Control performance parameters of THETA3

Controller	Performance Evaluation Using the Objective Function ITAE3				Performance Evaluation Using the Objective Function ITSE3			
	Overshoot (±)	Rise time (S)	Settling time (S)	ITAE3	Overshoot (±)	Rise time (S)	Settling time (S)	ITSE3
PID	8.6	1.75	5.90	0.2454	8.1	1.70	5.80	0.001389
PID-PSO	6.3	1.55	5.35	0.2201	6.0	1.50	5.20	0.001400
PD-FUZZY	9.8	1.40	5.10	0.3560	9.5	1.35	5.00	0.001339
PD-FUZZY-PSO	4.2	1.25	4.60	0.3194	4.0	1.20	4.55	7.395e-07

In addition, the hybrid PD-fuzzy-PSO controller performed better than all other methods , which illustrates the power and flexibility of both fuzzy logic and PSO. The percentage overshoot of the system response was approximately 4.2%, and the settling time was 4.6 seconds at the lowest level. The value of ITSE3 at $7.395*10^{-7}$ was less than that of the other methodologies.

Table 9. Control performance parameters of THETA4

Controller	Performance Evaluation using the Objective Function ITAE4				Performance Evaluation using the Objective Function ITSE4			
	Overshoot (±)	Rise time (S)	Settling time (S)	ITAE4	Overshoot (±)	Rise time(S)	Settling time (S)	ITSE4
PID	9.1	1.70	6.00	0.07721	8.8	1.65	5.90	0.0002251
PID-PSO	6.7	1.55	5.50	0.07127	6.4	1.50	5.35	2.2535e-04
PD-Fuzzy	11.5	1.40	5.20	0.67900	11.0	1.35	5.10	0.0018
PD-Fuzzy-PSO	4.8	1.25	4.70	0.65160	4.5	1.20	4.60	4.803e-07

The simulation results for Theta4 also validated the significant differences among the considered control techniques.

For the conventional PID regulator, it was found that the system is stabilized, but with a long settling time and a large overshoot (≈9.1%). The settling time of the system was 6.0 seconds, and ITAE4 = 0.07721, ITSE4 = 0.0002251 were still high in this case because it had not entirely eliminated the accumulated error of the system. Hence, it can be seen that the trajectory tracking performance of the conventional PID controller is not so good, particularly in the transient response region.

According to the comparison, we can see that by tuning the control parameters using the PSO algorithm (PID-PSO), system performance has been greatly enhanced. The overshoot is now 6.7% and it takes around 5.5s for the systems to be settled down. The ITAE4 = 0.07127, ITSE4 = 2.2535×10^{-4} values have also decreased now that the control process is stable and faster; the positive effect of the PSO algorithm for optimization of parameters of the PID controller is clear.

For PD-Fuzzy, the use of a combination of fuzzy logic itself results in a softening in the response, and for this reason, to reduce oscillations, with great adaptive capacity to the nonlinear characteristics of the system. Furthermore, there is no way to tune the common parameters, so effectively the overshoot is still very high (≈11.5%). In addition to that, the ITAE4 value (0.6790) is higher than all of the controllers. This indicates that PD-Fuzzy is conditioning qualitative stability only and does not have the quantitative error tuned. On the other hand, PD-fuzzy-PSOdemonstrated speed and accuracy in improvements. The overshoot decreased significantly to 4.8%, the settling time was lowest (≈4.7 s), and the ITAE4 and ITSE4 criteria were also minimal (0.6516 and 4.803×10^{-7}, respectively). This controller also has a quick and smooth response with almost no vibration and close to zero steady state error. It showed that the combination of fuzzy control with PSO optimization can have better effects on tracking accuracy and stability of trajectory under complex dynamics.

6. Conclusion and Future Directions

The simulation results for Theta1, Theta2, Theta3, and Theta4 angles demonstrate a big difference between the controllers we are analyzing. The results indicated that the Fuzzy controller-based PD-fuzzy-PSOswarm optimization is a more efficient method than either PID or PID-PSO controllers. This was demonstrated particularly in reduced oscillations, shorter transients, and increased trajectory tracking accuracy. At any rate, the quality of control was already remarkable, especially when compared head-to-head. Traditional PID controllers can only stabilize the system simply, but have a huge overshoot and a large steady state error; the PSO-based PID approach greatly improves the response quality, including fewer oscillations and shorter times to stabilize. However, this PID model is still not very good in the nonlinear and turbulent conditions. The flexibility of the PD-Fuzzy controller is improved by the fuzzy inference, which markedly reduces the transient and oscillation errors in comparison to the systematic PID control approach. With the aid of PSO to automatically adjust the fuzzy inference effect of the system, the PD-fuzzy-PSO system gets a better performance, such as faster response speed, smaller overshoot, and a steady state error almost zero. Possibly, the combination of Fuzzy intelligent control with the global searching ability of PSO is a very effective hybrid of control as a result of nonlinearity, but will it be promoted widely? While the conventional PID controllers still have limitations.

In addition, the control performances are further evaluated utilizing the ITAE and ITSE criterion functions. It is found that each criterion is excellent in a certain way. For instance, the ITAE criterion can make the system respond faster, having a smaller steady-state error and closer to the reference signal by designing the control in such a direction that the ITAE penalty function tends to decrease the most; whereas, ITSE criterion mainly enhance the system stability and reduce the magnitude of oscillations due to ITSE penalty function tends to increase with larger penalizing. ITAE and ITSE work well together; the former mainly enhances the response speed and output accuracy, while the latter mainly increases the system stability and reduces the magnitude of oscillations in the output. In the meantime, one can construct hybrid objective functions by combining the ITAE and ITSE penalty functions together for better performance in implementation, as illustrated below:

$$J = \alpha.ITSE + (1-\alpha).ITAE \tag{10}$$

The weighting factor can be a function of time or of the system error in the process. At the early part of the response, increasing weight is given more to ITSE in order to dampen oscillations and keep the system stable. As the response proceeds, one can slowly increase the weight of ITAE to bring down the settling time and steady-state error. And, more advanced Hybrid Fuzzy–PSO techniques such as the Adaptive Fuzzy–PSO method, Neuro–Fuzzy–PSO algorithm, or deep reinforcement fuzzy control are capable of further improving the control performance. These methodologies can enable the controller to automatically adjust others in terms of the real working conditions, which makes it possible for the system to perform adaptively and effectively under different operation environments.

In summary, the findings show that the scenario of the suggested hybrid fuzzy-PSO controller will promote system performance, as experimented with ITAE and ITSE. These methods can be considered as a good direction for the next research insofar as the NNES are needed with high precision for nonlinear control systems, such as robotics, drive systems, and orientation stability.

References

[1] Wei Ji, and Lihui Wang, "Industrial Robotic Machining: A Review," *International Journal of Advanced Manufacturing Technology*, vol. 103, pp. 1239-1255, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[2] Andrius Dzedzickis et al., "Advanced Applications of Industrial Robotics: New Trends, and Possibilities," *Applied Sciences*, vol. 12, no. 1, pp. 1-25, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[3] Janis Arents, and Modris Greitans, "Smart Industrial Robot Control Trends, Challenges and Opportunities within Manufacturing," *Applied Sciences*, vol. 12, no. 2, pp. 1-20, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[4] Pietro Bilancia et al., "An Overview of Industrial Robots Control and Programming Approaches," *Applied Sciences*, vol. 13, no. 4, pp. 1-14, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[5] Mukul Kumar Gupta, "Common Challenges in Control of Industrial Manipulators: A Review," *Online Journal of Robotics & Automation Technology*, vol. 1, no. 5, pp. 1-3, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[6] Zeashan Khan, Ali Nasir, and Samir Mekid, "Fault-Tolerant Control Strategies for Industrial Robots: State of the Art and Future Perspective on AI-based Fault Management," *Artificial Intelligence Review*, vol. 28, pp. 1-33, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[7] Javlonbek Rakhmatillaev, Vytautas Bucinskas, and Nozimjon Kabulov, “An Integrative Review of Control Strategies in Robotics,” *Robotic Systems and Applications*, vol. 5, no. 2, pp. 50-74, 2025. [CrossRef] [Google Scholar] [Publisher Link]
[8] Minglei Zhu et al., “Compliant Force Control for Robots: A Survey,” *Mathematics*, vol. 13, no. 13, pp. 1-22, 2025. [CrossRef] [Google Scholar] [Publisher Link]
[9] Phan Bui Khoi, “Application of Fuzzy Logic in the Robot Control for Mechanical Processing,” *Vietnam Journal of Science & Technology*, vol. 61, no. 4, pp. 531-569, 2023. [CrossRef] [Google Scholar] [Publisher Link]
[10] Claudio Urrea, John Kern, and Johanna Alvarado, “Design and Evaluation of a New Fuzzy Control Algorithm Applied to a Manipulator Robot,” *Applied Sciences*, vol. 10, no. 21, pp. 1-21, 2020. [CrossRef] [Google Scholar] [Publisher Link]
[11] Vítor Tinoco et al., “A Review of Advanced Controller Methodologies for Robotic Manipulators,” *International Journal of Dynamics and Control*, vol. 13, pp. 1-17, 2025. [CrossRef] [Google Scholar] [Publisher Link]
[12] John Kern, Dailin Marrero, and Claudio Urrea, “Fuzzy Control Strategies Development for a 3-DoF Robotic Manipulator in Trajectory Tracking,” *Processes*, vol. 11, no. 12, pp. 1-30, 2023. [CrossRef] [Google Scholar] [Publisher Link]
[13] Yao Sun, Xichang Liang, and Yi Wan, “Tracking Control of Robot Manipulator with Friction Compensation Using Time-Delay Control and an Adaptive Fuzzy Logic System,” *Actuators*, vol. 12, no. 5, pp. 1-18, 2023. [CrossRef] [Google Scholar] [Publisher Link]
[14] Jing Liu et al., “Design and Validation of a Fuzzy Logic Controller for Multi-Section Continuum Robots,” *IEEE Transactions on Industrial Electronics*, 2025. [CrossRef] [Google Scholar] [Publisher Link]
[15] Xiaoshan Qian, Lisha Xu, and Xinmei Yuan, “Fuzzy Super Twisting Mode Control of a Rigid-Flexible Robotic Arm based on Approximate Inertial Manifold Dimensionality Reduction,” *Frontiers in Neurorobotics*, vol. 17, pp. 1-16, 2023. [CrossRef] [Google Scholar] [Publisher Link]
[16] Bo-Ru Tseng, Jun-Yi Jiang, and Ching-Hung Lee, “Adaptive Position/Force Controller Design Using Fuzzy Neural Network and Stiffness Estimation for Robot Manipulator,” *International Journal of Fuzzy Systems*, vol. 27, pp. 2658, 2672, 2025. [CrossRef] [Google Scholar] [Publisher Link]
[17] Weiwei Sun et al., “Flexible Manipulator Trajectory Tracking based on an Improved Adaptive Particle Swarm Optimization Algorithm with Fuzzy PD Control,” *Mechanical Sciences*, vol. 16, pp. 125-141, 2025. [CrossRef] [Google Scholar] [Publisher Link]
[18] Putri Repina Kesuma et al., “Fuzzy Logic Controller Design for Mobile Manipulator Movement Control,” *Journal of Applied Smart Electrical Network and Systems*, vol. 1, no. 1, pp. 12-18, 2020. [CrossRef] [Google Scholar] [Publisher Link]