International Journal of Engineering Inventions
e-ISSN: 2278-7461, p-ISSN: 2319-6491
Volume 15, Issue 11 [November 2026] PP: 169-177

Swarm-Optimized Fuzzy-PD Control with Neural Residual Compensation for Multi-DOF Robotic Manipulators

Trung-Kien Le[1], Minh-Tuan Phan[1], Anh-Nam Tran-Nguyen[1], Mai-Phuong Dao-Thi[2], N-K. Nguyen[1*]

[1]*Faculty of Automation, Artificial Intelligence and Intelligent Systems, Electric Power University, Hanoi, Vietnam*

[2]*Faculty of Automation, School of Electrical and Electronic Engineering, Hanoi University of Industry, Hanoi, Vietnam*

[*]*Corresponding Author*

ABSTRACT: *High-precision motion control of multi-degree-of-freedom (multi-DOF) robotic manipulators is still a challenging problem due to strong inter-joint dynamic coupling, configuration-dependent inertia, unmodeled dynamics and parametric uncertainties. However, it is well known that fixed-gain linear controllers cannot guaranty robust performance in general during fast transients or under changing operating conditions. In this paper, a multi-layer hybrid intelligent control framework for robotic manipulators is proposed to solve these restrictions. In this context, the baseline stabilizing action is achieved by a fuzzy proportional-derivative (fuzzy-PD) controller, while a feedforward neural network compensates online for the residual acceleration error. Moreover, two metaheuristics based on swarm intelligence, the Particle Swarm Optimization (PSO) and the Gray Wolf Optimizer (GWO), are used to tune the controller gains in a systematic way, avoiding subjective trial-and-error procedures. The proposed approach combines rule-based fuzzy inference, adaptive neural approximation and population-based swarm search in a unified scheme to improve trajectory-tracking accuracy, suppress transient overshoot and ensure smooth motion in multi-DOF manipulation tasks.*

Date of Submission: 05-11-2026 Date of acceptance: 16-11-2026

I. INTRODUCTION

Industrial robotic manipulators are playing an important role in modern manufacturing systems due to their flexibility, precision and ability to perform complex tasks. However, the accurate trajectory tracking of multi-degree of freedom (multi-DOF) robotic manipulators still remains a challenging task due to nonlinear dynamics, joint coupling, changing inertia and uncertainties of system parameters [1], [2]. These characteristics can heavily influence control performance, particularly during high-speed movements and varying operating conditions.

Conventional control methods such as proportional-integral-derivative (PID) and proportional-derivative (PD) controllers are widely used due to their simple structure and easy implementation [3], [4]. However, their performance might degrade in the case of nonlinear systems and parameter variations. Therefore, intelligent control techniques have been studied more and more to overcome these limitations. One of them is fuzzy logic control proposed by Zadeh [5], which is a flexible approach based on linguistic rules and human reasoning. The integration of fuzzy logic with PD control provides a practical solution to enhance the control adaptability without the necessity of a complete dynamic model [6].

Artificial neural networks have also shown great potential in the approximation of complex nonlinear relationships [7], [8]. In robotic control systems feedforward neural networks can be used for the estimation of uncertain dynamic effects, and for the generation of compensation signals. In particular, neural-network-based acceleration compensation can assist the main controller in reducing the modeling errors and improving the trajectory tracking. But the effectiveness of this approach is dependent on the architecture of the network and the training process.

Furthermore, the good control performance relies on the choice of proper controller parameters. Population-based optimization algorithms such as Particle Swarm Optimization (PSO) [9], [10] and Gray Wolf Optimizer (GWO) [11] offer systematic replacements to the traditional manual tuning methods. Such algorithms can tune controller parameters based on pre-defined performance criteria, which may improve tracking accuracy and dynamic response. In addition,

hybrid intelligent control can be realized by combining fuzzy logic, neural networks and optimization techniques, which can benefit from their complementary characteristics [12].

In this regard, this paper presents an analytical overview of a hybrid intelligent control framework that integrates PD-Fuzzy logic, feedforward neural-network-based compensation and PSO-GWO optimization for multi-DOF robotic manipulators. The study focuses on the operating principles, functional roles and potential benefits of these techniques in trajectory tracking, nonlinear dynamic compensation and controller parameter optimization, laying a foundation for their application in intelligent robotic control systems.

II. ROBOT MODELING AND CONTROL METHODOLOGY

2.1. Robot Structure and Kinematics

The robotic manipulator considered in this study consists of interconnected rigid links and revolute joints, as illustrated in Fig. 1. The motion of the robot is determined by the angular displacement of its joints, while the end-effector position depends on the geometric relationships between consecutive links.

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

The Denavit–Hartenberg (D–H) convention is employed to describe the kinematic relationships of the manipulator. Each joint is characterized by four parameters: link length (a_i), link twist (α_i), link offset (d_i), and joint angle (θ_i). Based on these parameters, the homogeneous transformation matrix between two consecutive coordinate frames can be established.

The overall transformation from the base frame to the end-effector is obtained by multiplying the individual transformation matrices:

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

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 between adjacent coordinate frames. This formulation provides the basis for determining the end-effector position and orientation.

2.2. Robot Dynamic Model

The dynamic behavior of a robotic manipulator is influenced by joint motion, link inertia, gravitational effects, and interactions between joints. To describe these characteristics, the Lagrange method is employed based on the kinetic and potential energies of the system.

The Lagrangian function is expressed as:

$$ L(\theta, \dot{\theta}) = K(\theta, \dot{\theta}) - U(\theta) \quad [2] $$

where (K) and (U) represent the total kinetic and potential energies, respectively.

Applying the Euler–Lagrange equations, the general dynamic model of the robotic manipulator can be written as:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau \quad [3]$$

where q , $\dot{q}$, and $\ddot{q}$ denote the joint position, velocity, and acceleration vectors, respectively. $M(q)$ is the inertia matrix, $C(q,\dot{q})$ represents Coriolis and centrifugal effects, $G(q)$ is the gravitational torque vector, and τ denotes the applied joint torque vector.

The nonlinear characteristics and interactions among these dynamic components introduce challenges for accurate trajectory tracking. Therefore, the dynamic model provides an important foundation for analyzing the application of intelligent control methods, particularly fuzzy logic control and neural-network-based compensation, to improve robotic motion performance.

2.3. PD-Fuzzy Control Structure

A PD-Fuzzy controller is the combination of the proportional-derivative (PD) control principle and the nonlinear reasoning ability of fuzzy logic. This combination facilitates a flexible control strategy for the robotic manipulators working under nonlinear dynamic conditions. The control action is obtained from the control rules based on the tracking error and the rate of change of tracking error, through a pre-defined fuzzy rule base.

The tracking error shown in Fig. 1 is obtained by comparing the desired joint position with the actual joint position. The error in tracking and its derivative are defined as (2) and (3) respectively.

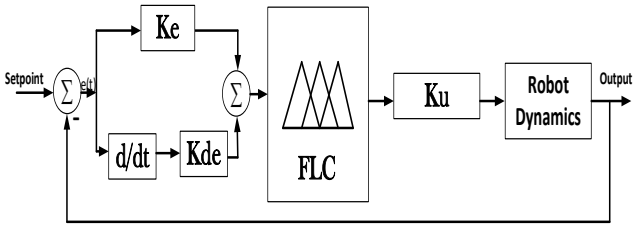


Fig. 1: The PD–Fuzzy controller is suitable for the model

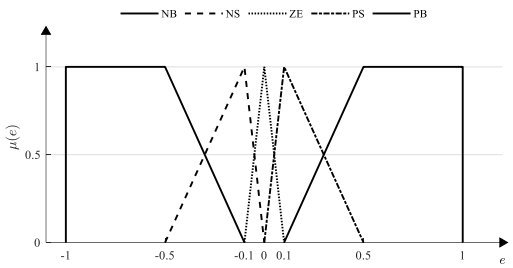


Fig. 2: Input variable *e*

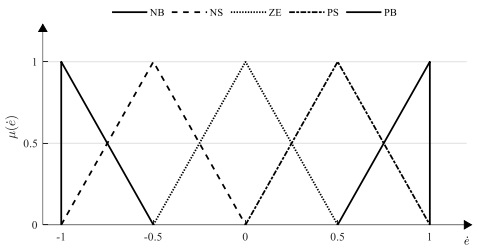


Fig. 3: Input variable *de*

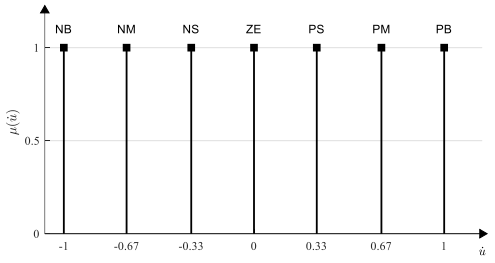


Fig. 4: Output torque applied to the robotic arm

The two input signals are multiplied by the scaling factors (Kp) and (Kd) before being processed by the fuzzy logic controller (FLC). The fuzzy inference system evaluates these inputs according to the established control rules and generates the corresponding output signal. The overall PD-Fuzzy control law is expressed in (4).

In this expression, $F(e,\dot{e})$ represents the fuzzy inference mechanism, while (Kp), (Kd), and (Ku) denote the input and output scaling factors. The resulting control signal is applied to the robotic manipulator to regulate joint motion and reduce trajectory tracking errors.

2.4. Fuzzy Membership Functions and Rule Base

The fuzzy inference system consists of three main stages: fuzzification, rule evaluation, and defuzzification. In the fuzzification stage, numerical input values are converted into linguistic variables using predefined membership functions. The fuzzy rule base then determines the appropriate control response, and the defuzzification stage converts the fuzzy output into a numerical control signal.

In the considered controller, the tracking error and its derivative are represented by five linguistic terms: Negative Big (NB), Negative Small (NS), Zero (ZE), Positive Small (PS), and Positive Big (PB). The output variable is described by seven linguistic terms, including NB, Negative Medium (NM), NS, ZE, PS, Positive Medium (PM), and PB. The membership functions are illustrated in Figs. 2–4.

The fuzzy control rules are established based on the relationship between the tracking error and its rate of change, as presented in Table 2. These rules determine the magnitude and direction of the control action under different operating conditions. By combining PD-based error processing with fuzzy inference, the controller provides a flexible mechanism for generating control signals and serves as the primary feedback component of the proposed hybrid intelligent control framework.

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
	PS	NS	ZE	PS	PM	PB
	PB	ZE	PS	PM	PB	PB

2.5.Offline metaheuristic parameter optimization framework

The performance of the PD-Fuzzy controller depends significantly on the selection of its scaling factors. Conventional manual tuning methods can be time-consuming and may not provide satisfactory performance under nonlinear operating conditions. Therefore, Particle Swarm Optimization (PSO) and Grey Wolf Optimizer (GWO) are considered as offline optimization methods to determine suitable controller parameters and improve trajectory tracking performance.

Particle Swarm Optimization (PSO): PSO is a population-based optimization algorithm in which particles search for optimal solutions by updating their velocities and positions according to individual and global best solutions. The updating process is described by (4) and (5), where (w) is the inertia weight, (c_1) and (c_2) are acceleration coefficients, and (r_1, r_2) are random values within [0, 1].

$$V_i^{k+1} = w.V_i^k + c_1 r_1\left(P_{best,i} - X_i^k\right) + c_2 r_2\left(G_{best} - X_i^k\right) \quad [4]$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad [5]$$

Grey Wolf Optimizer (GWO): GWO simulates the social hierarchy and hunting behavior of grey wolves. The three leading wolvesAlpha (α), Beta (β), Delta (δ) guide the search process, while the remaining wolves update their positions based on the leading solutions. The position updating mechanism is expressed by (6)–(16), allowing the algorithm to explore and exploit the search space.

$$D = \left|C.X_p(t) - X(t)\right| \quad [6]$$

$$X(t+1) = X_p(t) - A.D \quad [7]$$

$$\vec{X}_{(t+1)} = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad [8]$$

$$\vec{A} = 2.\vec{a}.\vec{r}_1 - \vec{a} \quad [9]$$

$$\vec{C} = 2.\vec{r}_2 \quad [10]$$

$$\vec{X}_1 = \left| \vec{X}_\alpha(t) - \vec{A}_1 . \vec{D}_\alpha \right| \quad 11] \qquad [\quad \vec{D}_\alpha = \left| \vec{C}_1 . \vec{X}_\alpha(t) - \vec{X}(t) \right| \quad [14]$$

$$\vec{X}_2 = \left| \vec{X}_\beta(t) - \vec{A}_2 . \vec{D}_\beta \right| \quad 12] \qquad [\quad \vec{D}_\beta = \left| \vec{C}_2 . \vec{X}_\beta(t) - \vec{X}(t) \right| \quad [15]$$

$$\vec{X}_3 = \left| \vec{X}_\delta(t) - \vec{A}_3 . \vec{D}_\delta \right| \quad 13] \qquad [\quad \vec{D}_\delta = \left| \vec{C}_3 . \vec{X}_\delta(t) - \vec{X}(t) \right| \quad [16]$$

In the proposed framework, both algorithms are considered for optimizing the PD-Fuzzy controller parameters by minimizing a predefined trajectory tracking error function. The optimized parameters are subsequently applied to the controller, providing a basis for comparison between PSO and GWO.

2.6. Objective Function Formulation

To evaluate and optimize the trajectory tracking performance of the 4-DOF robotic manipulator, the Integral of Time-weighted Absolute Error (ITAE) is selected as the objective function for both PSO and GWO algorithms, as expressed in (17).

$$J_{ITAE} = \int_0^{T_s} t . \sum_{i=1}^{4} |e_i(t)| dt \quad [17]$$

Here, $e_i(t)$ denotes the tracking error of the $i \in \{1,2,3,4\}$ joint, and T_s represents the total simulation time. By assigning greater weight to errors occurring at later times, the ITAE criterion encourages the reduction of persistent tracking errors and supports the optimization of PD-Fuzzy controller parameters.

2.7. Neural Network-Based Dynamic Compensation

A feedforward neural network (FNN) is employed to approximate the nonlinear dynamic characteristics of the robotic manipulator. Through its learning capability, the neural network can establish relationships between robot states and dynamic responses, providing a basis for acceleration compensation and improving trajectory tracking performance.

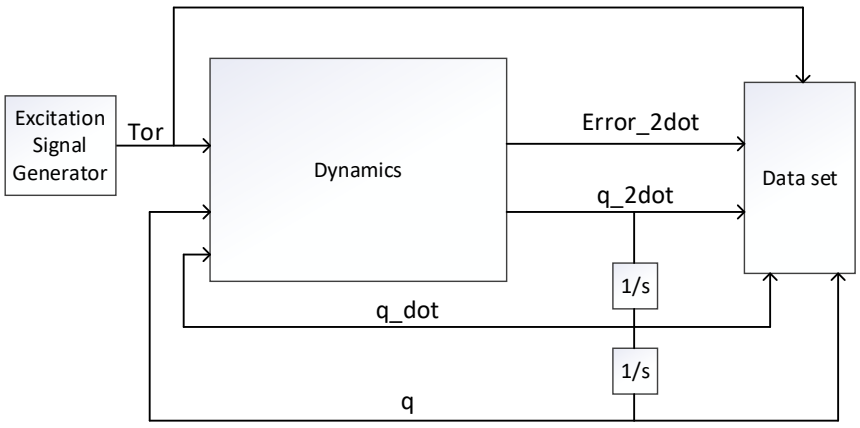


Fig. 11. Block diagram of the data generation process for neural network training.

As illustrated in Fig. 11, excitation signals are applied to the robot dynamic model to generate representative motion data. The resulting responses are processed through integration blocks to obtain joint velocities and positions, while the corresponding dynamic signals are collected to construct the training dataset.

The collected dataset is subsequently used for offline neural network training. During this process, the network weights and biases are adjusted to minimize the prediction error between the estimated and target outputs. After training, the neural network approximates the learned dynamic relationships and provides an estimated acceleration compensation component to support the PD-Fuzzy controller during robot operation.

2.8. Overall Hybrid Intelligent Control Framework

The overall structure of the proposed hybrid intelligent control system is illustrated in Fig. 12. The framework integrates a PD-Fuzzy controller, PSO/GWO-based parameter optimization, and a neural network for dynamic compensation in a 4-DOF robotic manipulator.

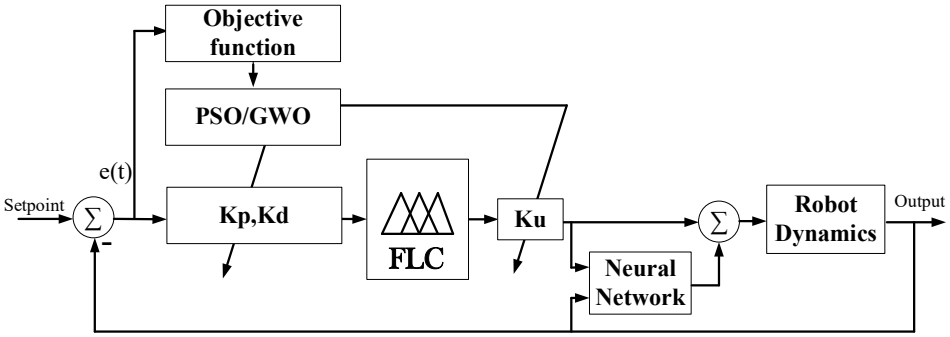


Fig. 12. Overall block diagram of the proposed PSO/GWO-optimized PD-Fuzzy control system with neural network-based dynamic compensation.

As shown in Fig. 12, the desired trajectory is compared with the actual joint position to generate the tracking error. This error is processed by the PD-Fuzzy controller to produce the primary control signal. During offline optimization, PSO and GWO are employed as alternative methods to adjust the controller parameters by minimizing the ITAE objective function. Meanwhile, the trained neural network estimates the residual dynamic effects and provides a compensation signal to support the feedback controller.

The resulting control action is applied to the robot dynamics, while the joint responses are fed back to continuously update the tracking error. By combining fuzzy feedback control, metaheuristic parameter optimization, and neural compensation, the proposed framework aims to improve trajectory tracking accuracy and reduce the influence of nonlinear dynamic uncertainties.

III. RESULTS AND DISCUSSION

To investigate the effectiveness of the proposed control framework, simulations were performed on a 4-DOF robotic manipulator using six control configurations: PID, Fuzzy, Fuzzy-PID, PD-Fuzzy with neural compensation, and its PSO- and GWO-optimized variants. The evaluation focuses on trajectory tracking accuracy, transient response characteristics, and the effectiveness of parameter optimization.

3.1. Optimization Performance

Table 3. Comparative Optimization Results of Scaling Factors and Tracking Performance for Joints 1–4

Joint	GWO				PSO			
	K_p	K_u	K_d	ITAE	K_p	K_u	K_d	ITAE
1	4.8745	72.6868	1.00001	0.0094	2.6367	43.6630	1.1402	0.0298
2	5.0001	44.7411	1.5243	0.0081	4.4973	90.6486	1.7665	0.0148
3	5.00009	57	1.3955	0.0062	3.6424	57.5097	1.5497	0.0085
4	5.00012	33.5342	1.607	0.0033	1.3134	36.2865	1.5225	0.0117

Table 3 presents the optimized scaling factors and corresponding ITAE values obtained using PSO and GWO. Both algorithms were configured with 100 search agents and a maximum of 100 iterations to optimize 12 controller parameters. The results indicate that GWO achieved lower ITAE values than PSO for all four joints, suggesting more effective parameter selection under the considered simulation conditions.

3.2. Trajectory Tracking Performance

Table 4. Step response performance indices for joint theta 1

Type	Overshoot	Rise time	Settling time
GWO Fuzzy & PD Neural	0.90359	2.0105	2.8039
PSO Fuzzy & PD Neural	1.9842	2.0932	2.9002
Fuzzy & PD Neural	2.1174	2.1346	2.905
Fuzzy & PID	6.9827	2.2849	2.9076
Fuzzy	29.2	2.2992	2.9083
PID	51.831	2.391	2.921

Table 5. Step response performance indices for joint theta 2

Type	Overshoot	Rise time	Settling time
GWO Fuzzy & PD Neural	0.25424	1.9496	2.5922
PSO Fuzzy & PD Neural	0.68053	2.0847	2.8412
Fuzzy & PD Neural	1.6772	2.0876	2.9051
Fuzzy & PID	3.4293	2.0991	2.9073
Fuzzy	4.2438	2.2393	2.9122
PID	14.363	2.7468	2.977

Table 6. Step response performance indices for joint theta 3

Type	Overshoot	Rise time	Settling time
GWO Fuzzy & PD Neural	0.19714	1.9787	2.6297
PSO Fuzzy & PD Neural	0.20969	2.0825	2.9005
Fuzzy & PD Neural	1.6772	2.0859	2.9022
Fuzzy & PID	3.0601	2.223	2.925
Fuzzy	4.6695	2.2349	2.923
PID	7.4218	2.476	2.9633

Table 7. Step response performance indices for joint theta 4

Type	Overshoot	Rise time	Settling time
GWO Fuzzy & PD Neural	0.2395	2.0773	2.8948
PSO Fuzzy & PD Neural	0.31664	2.0785	2.8955
Fuzzy & PD Neural	0.9427	2.0894	2.8958
Fuzzy & PID	3.0193	2.1123	2.8977
Fuzzy	5.939	2.1766	2.9537
PID	4.9839	2.3884	2.9992

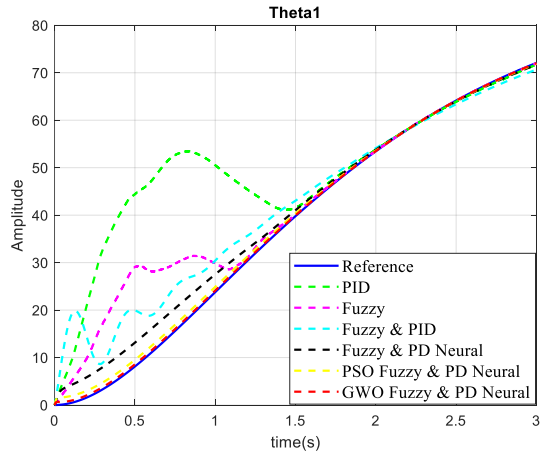


Fig 5: Joint 1 Trajectory Tracking Comparison.

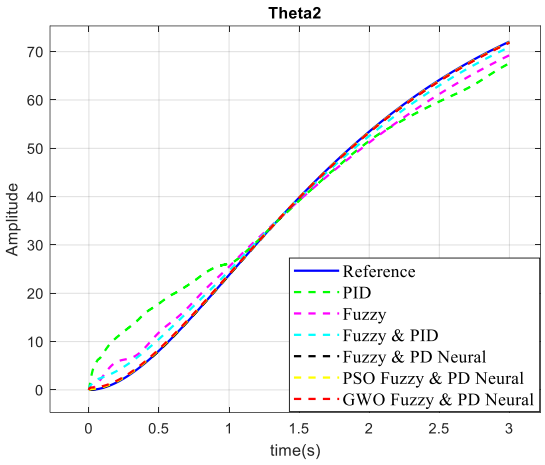


Fig 6: Joint 2 Trajectory Tracking Comparison.

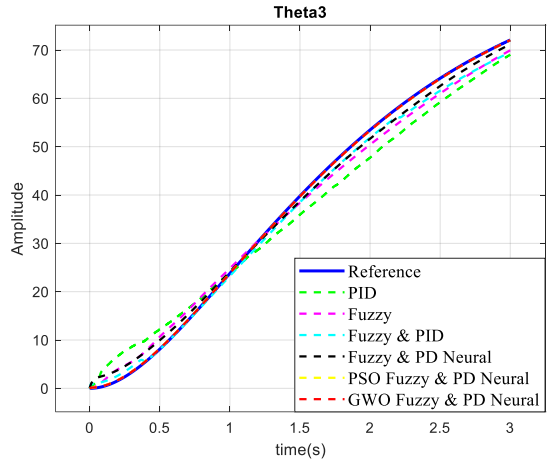


Fig 7: Joint 3 Trajectory Tracking Comparison.

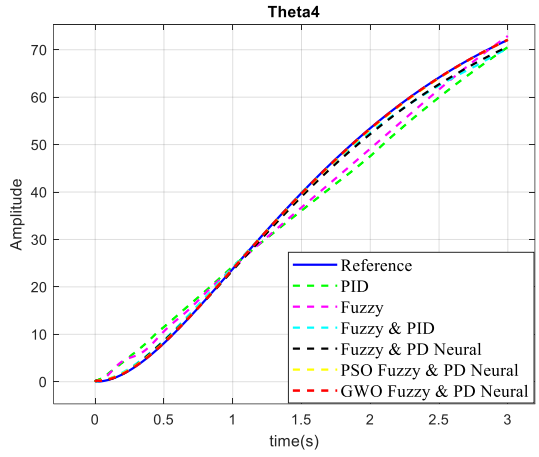


Fig 8: Joint 4 Trajectory Tracking Comparison.

The transient response characteristics of the six controllers are summarized in Tables 4–7, while the corresponding joint trajectories are illustrated in Figs. 5–8. The results show that incorporating PD-Fuzzy control and neural compensation reduces overshoot compared with the conventional PID and standalone Fuzzy controllers.

Among the evaluated configurations, the GWO-optimized controller provides the lowest overshoot and shortest rise and settling times across all four joints. In particular, the overshoot remains below 1%, with the largest improvement observed at Joint 1, where overshoot decreases from 51.831% for PID control to 0.90359% for the GWO-based approach.

The trajectory plots further illustrate that the optimized controllers follow the reference more closely, particularly during transient operation. Overall, the results indicate that combining PD-Fuzzy control, neural compensation, and metaheuristic parameter tuning offers improved tracking performance for the considered robotic manipulator.

IV. CONCLUSION AND FUTURE WORK

In this paper a hybrid intelligent control method was proposed for a 4-DOF robotic manipulator by integrating PD-Fuzzy control, neural network based dynamic compensation and meta-heuristic optimization techniques. Simulation results illustrated that the integration of these methods improves the performance of the trajectory tracking and reduces the transient errors compared to the conventional control strategies. The results indicated that the GWO gave lower ITAE values and better transient response characteristics for all the four joints by tuning the controller parameters using both PSO and GWO. In particular, the overshoot of the GWO-based controller was less than 1%, which demonstrates its capability of precisely and stably controlling the robot's motion under the conditions investigated.

Future work will be dedicated to the implementation of the proposed control framework to a real robotic manipulator in an effort to evaluate its practical feasibility. Further investigations will consider external disturbances, parameter variations and different reference trajectories to assess the robustness of the system. Moreover, online neural network adaptation and improved optimization strategies will be studied to enhance the controller's flexibility and performance in more complex operating environments.

ACKNOWLEDGMENTS

This research is funded by Electric Power University under research project code ĐTNH.31/2026.

REFERENCES

[1]. M. W. Spong, "Modeling and control of elastic joint robots," Journal of Dynamic Systems, Measurement, and Control, vol. 109, no. 4, pp. 310–319, 1987. https://doi.org/10.1115/1.3143860

[2]. J.-J. E. Slotine and W. Li, "On the adaptive control of robot manipulators," The International Journal of Robotics Research, vol. 6, no. 3, pp. 49–59, 1987. https://doi.org/10.1177/027836498700600303

[3]. J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," Transactions of the ASME, vol. 64, no. 8, pp. 759–768, 1942. https://doi.org/10.1115/1.2899060

[4]. L. A. Zadeh, "Fuzzy sets," Information and Control, vol. 8, no. 3, pp. 338–353, 1965. https://doi.org/10.1016/S0019-9958(65)90241-X

[5]. E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," International Journal of Man-Machine Studies, vol. 7, no. 1, pp. 1–13, 1975. https://doi.org/10.1016/S0020-7373(75)80002-2

[6]. J.-S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system," IEEE Transactions on Systems, Man, and Cybernetics, vol. 23, no. 3, pp. 665–685, 1993. https://doi.org/10.1109/21.256541

[7]. K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," Neural Networks, vol. 2, no. 5, pp. 359–366, 1989. https://doi.org/10.1016/0893-6080(89)90020-8

[8]. F. L. Lewis, A. Yesildirek, and K. Liu, "Multilayer neural-net robot controller with guaranteed tracking performance," IEEE Transactions on Neural Networks, vol. 7, no. 2, pp. 388–399, 1996. https://doi.org/10.1109/72.485674

[9]. J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proc. IEEE International Conference on Neural Networks, Perth, WA, Australia, vol. 4, pp. 1942–1948, 1995. https://doi.org/10.1109/ICNN.1995.488968

[10]. R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization: An overview," Swarm Intelligence, vol. 1, pp. 33–57, 2007. https://doi.org/10.1007/s11721-007-0002-0

[11]. S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," Advances in Engineering Software, vol. 69, pp. 46–61, 2014. https://doi.org/10.1016/j.advengsoft.2013.12.007

[12]. H. Faris, I. Aljarah, M. A. Al-Betar, and S. Mirjalili, "Grey wolf optimizer: A review of recent variants and applications," Neural Computing and Applications, vol. 30, pp. 413–435, 2018. https://doi.org/10.1007/s00521-017-3272-5