Computational Engineering and Physical Modeling 8-3 (2025) 01-19

Contents lists available at CEPM

Computational Engineering and Physical Modeling

Journal homepage: www.jcepm.com

Meta-Heuristic Optimization of Pid Controllers for a 5-Dof Robotic Manipulator

Adeleke Olorunnisola [1,*]; Olurotimi Dahunsi [2]

1. Student, Department of Mechanical Engineering, Federal University of Technology, Akure, Ondo State. Nigeria
2. Professor, Department of Mechanical Engineering, Federal University of Technology, Akure, Ondo State. Nigeria

* Corresponding author: *olorunnisola01@gmail.com*

http://doi.org/10.22115/cepm.2025.544303.1383

ARTICLE INFO

Article history:
Received: 04 September 2025
Revised: 29 October 2025
Accepted: 30 November 2025

Keywords:
Industrial robotics;
Intelligent control;
Motion control;
Trajectory tracking;
Metaheuristic optimization;
Control system optimization.

ABSTRACT

Accurate motion control is a critical requirement for robotic manipulators in advanced industrial and research applications. Proportional–Integral–Derivative (PID) controllers remain the preferred choice due to their simplicity and reliability, although their performance is highly dependent on effective gain tuning. Conventional tuning techniques are often inefficient and fail to deliver consistent results across complex robotic systems. This study investigates the application of metaheuristic optimization methods, specifically Genetic Algorithm (GA) and Ant Colony Optimization (ACO), for tuning PID controllers in a five-degree-of-freedom robotic manipulator. The manipulator was modeled in SolidWorks, simulated in Simscape, and integrated with MATLAB-based control. A sinusoidal trajectory was employed as the reference input, and performance was evaluated using Integral Time Absolute Error (ITAE) and overshoot metrics across all joints. The results show that both GA and ACO outperform manual tuning. GA reduced the average overshoot by approximately 51% and ACO by 61% compared with manual tuning, while the GA and ACO algorithms achieved fitness (ITAE) improvements of 0.36% and 0.04%, respectively. GA demonstrated faster convergence (within 10 generations), whereas ACO achieved more stable fitness reduction and superior trajectory tracking, indicating enhanced robustness. These findings suggest that while GA offers computational efficiency, ACO provides improved stability and accuracy, making it a more effective strategy for PID tuning in this system.

How to cite this article: Adeleke O, Dahunsi O. Meta-Heuristic Optimization of Pid Controllers for a 5-Dof Robotic Manipulator. Computational Engineering and Physical Modeling. 2025;8(3): 1-19.. https://doi.org/10.22115/cepm.2025.544303.1383

1. Introduction

Robotic manipulators play a central role in modern industrial automation, supporting tasks such as welding, painting, assembly, and material handling. Their ability to operate with high precision and repeatability in hazardous or high-speed environments makes them indispensable in advanced manufacturing. Serial-chain manipulators are particularly valued for their load capacity and point-to-point accuracy, yet achieving reliable motion control in such systems remains a significant challenge.

Among the various control strategies developed for robotic manipulators, the Proportional–Integral–Derivative (PID) controller remains the most widely applied due to its simplicity, transparency, and proven effectiveness across diverse applications [1]. Despite this ubiquity, the performance of PID controllers is highly dependent on proper gain tuning [2]. Manual or classical tuning methods are often time-consuming and rarely deliver optimal results for complex multi-degree-of-freedom systems, where nonlinearities and dynamic interactions are prevalent [3].

To address these limitations, optimization-based approaches have gained increasing attention. Metaheuristic algorithms, inspired by natural processes, offer a powerful means of exploring the high-dimensional and nonlinear parameter space associated with PID tuning [4]. Genetic Algorithms (GA), representing evolutionary search methods [5], and Ant Colony Optimization (ACO), derived from swarm intelligence [6], have both been successfully applied to control problems and shown to outperform manual tuning techniques [4][7]. However, limited studies have provided a direct comparison of their effectiveness for robotic manipulator control, particularly for systems with higher degrees of freedom [8–10].

This study investigates the application of GA and ACO for tuning PID controllers in a five-degree-of-freedom (5-DOF) robotic manipulator. The manipulator is modeled in SolidWorks, simulated in Simscape, and integrated with MATLAB-based control. The objective is to evaluate the extent to which these algorithms can improve trajectory tracking, stability, and robustness compared to manual tuning.

The research specifically addresses the following questions: (i) How effectively can GA and ACO optimize PID parameters for a 5-DOF robotic manipulator? (ii) What performance improvements in terms of accuracy, speed, and stability can be achieved relative to conventional tuning methods?

2. Related works

Robotic manipulators play a vital role across a wide range of fields, including manufacturing, healthcare, agriculture, and service industries. In manufacturing, they support tasks such as assembly, welding, and material handling, offering high precision and consistency [11–13]. In the medical field, robotic arms enable minimally invasive surgery and assistive rehabilitation, improving patient outcomes [14,15]. Agricultural applications include automated harvesting and transplantation, where they reduce labor demand and enhance productivity [16]. Despite these diverse applications, the effectiveness of robotic manipulators fundamentally depends on their ability to achieve precise, reliable, and stable motion [17–19]. This reliance underscores the central challenge: developing control strategies that can deliver accurate trajectory tracking and adaptability under varying dynamic conditions.

These diverse applications emphasize the importance of precise control, yet few studies critically link application requirements to the limitations of existing control methods [20,21].

2.1. Control strategies for robotic manipulators

The performance of a robotic manipulator is inherently tied to the effectiveness of its control system. Over the years, a wide range of strategies have been developed to address the challenges of precision, robustness, and adaptability in dynamic environments. Classical approaches such as Proportional-Integral-Derivative (PID) controllers remain dominant in industry due to their simplicity, transparency, and ease of implementation. However, they often require extensive manual tuning, which limits their effectiveness in complex or nonlinear systems [21,22].

To overcome these limitations, advanced control methods such as adaptive control [23], robust control [24], sliding mode control [25], and model predictive control have been introduced [26,27]. Adaptive control adjusts parameters in real time to accommodate model uncertainties and external disturbances.

Robust control enhances system stability against parameter variations but can be conservative in design. Sliding mode control offers strong robustness but introduces chattering effects that may damage actuators. Model predictive control provides optimal performance under constraints, although it demands significant computational resources.

In addition to these methods, intelligent and nature-inspired algorithms have gained attention for their ability to optimize controller parameters and improve performance without requiring exact system models. Techniques such as fuzzy logic control, neural networks, evolutionary optimization (Genetic Algorithms) and Swarm Intelligence Optimization (Ant Colony Optimization) enable flexible and adaptive controller design. These methods have shown promising results in enhancing trajectory tracking, minimizing overshoot, and achieving faster convergence in robotic systems [28–32].

The diversity of these strategies highlights that no single method universally outperforms others across all applications. Instead, the choice of control strategy depends on the trade-offs between simplicity, robustness, computational efficiency, and adaptability required for the specific manipulator task.

Although advanced controllers such as MPC and robust control offer superior performance, their complexity has limited industrial adoption, leaving PID as the dominant strategy. This reinforces the need for improved PID tuning methods [1].

Table 1
Comparison of Major Control Strategies [1,32–34].

Category	Control Strategy	Key Features	Advantages	Limitations
Classical	PID Control	Linear feedback with proportional, integral, derivative terms	Simple, widely adopted in industry; easy to implement	Requires manual tuning; poor performance in nonlinear or uncertain systems
Robust / Adaptive	Adaptive Control	Online parameter adjustment for uncertainties	Handles system variations and disturbances	Complex design; stability proof required
	Robust Control	Designed to maintain stability under bounded uncertainties	Guarantees stability margins	Often conservative; may reduce performance
	Sliding Mode Control (SMC)	Discontinuous switching law	Strong robustness to matched disturbances	Chattering may damage actuators
Optimal	Model Predictive Control (MPC)	Optimization-based with constraints	High accuracy; handles constraints explicitly	Computationally intensive; requires accurate model
Intelligent / Optimization-based	Fuzzy Logic Control	Rule-based reasoning without exact model	Effective for nonlinear and uncertain systems	Requires expert-defined rules
	Neural Network Control	Learning-based nonlinear approximation	Captures unmodeled dynamics; adaptive	Training complexity; generalization issues
	Genetic Algorithm (GA)	Evolutionary optimization for controller gains	Finds near-optimal PID parameters; global search ability	Convergence speed depends on tuning parameters
	Ant Colony Optimization (ACO)	Swarm intelligence with probabilistic path construction	Efficient search in large parameter spaces; adaptable	Computationally heavy for high-dimensional problems

2.2. Challenges of PID tuning

Proportional-Integral-Derivative (PID) controllers remain the most widely adopted control strategy in industrial and robotic applications due to their simplicity, cost-effectiveness, and reliable performance across a range of systems. However, their effectiveness is highly dependent on the appropriate selection of gain parameters. The tuning of these parameters; proportional (Kp), integral (Ki), and derivative (Kd) directly influences system stability, transient response, and steady-state performance.

Classical tuning techniques, such as the Ziegler–Nichol's method, Cohen-Coon method, Tyreus-Luyben method, and trial-and-error approaches, have long been employed for PID parameter selection. These methods provide initial estimates of controller gains that are often sufficient for linear, time-invariant, and relatively simple systems. While they are straightforward to apply, their reliance on heuristic rules and empirical adjustments limits their applicability to more complex or nonlinear systems such as robotic manipulators. Furthermore, these methods typically involve offline tuning, which does not account for time-varying disturbances, model uncertainties, or dynamic operating conditions. As a result, systems tuned with classical methods may exhibit overshoot, oscillations, or degraded performance under unanticipated conditions [35–39].

Another major limitation is the lack of adaptability. Classical tuning assumes fixed plant dynamics, yet robotic manipulators often experience parameter variations due to payload changes, friction, joint flexibility, and external disturbances. A fixed set of PID gains tuned manually may not maintain performance across these variations. Additionally, iterative manual tuning is laborious and impractical for systems requiring real-time precision [38,40].

These challenges motivate the exploration of intelligent and metaheuristic optimization approaches for PID tuning. Unlike classical methods, metaheuristic algorithms such as Genetic Algorithms, Particle Swarm Optimization, and Ant Colony Optimization provide systematic search mechanisms within high-dimensional parameter spaces. They are capable of converging toward near-optimal gain values while balancing competing performance objectives such as fast settling time, minimal overshoot, and robustness. Moreover, these algorithms can adapt to nonlinearities and uncertainties, making them particularly suited for complex robotic manipulator systems [41–44].

Classical tuning techniques provide only approximate solutions and fail under nonlinear or time-varying conditions. This gap motivates the exploration of intelligent tuning methods.

2.3. Metaheuristic approaches for PID tuning

The limitations of classical PID tuning methods have led to increasing interest in metaheuristic algorithms, which provide a more systematic and adaptive framework for controller optimization. Metaheuristics, inspired by evolutionary processes and swarm intelligence, excel in exploring nonlinear, high-dimensional parameter spaces and avoiding premature convergence to local optima. Among these, Genetic Algorithms (GA) and Ant Colony Optimization (ACO) have emerged as prominent approaches for PID gain selection in robotic and other dynamic systems.

Genetic Algorithms employ mechanisms of selection, crossover, and mutation to iteratively evolve candidate solutions toward optimal PID parameters. Their ability to balance exploration and exploitation has made them effective in tuning controllers for mobile robots, underwater vehicles, and robotic manipulators. Prior studies have shown that GA-based tuning improves transient performance and robustness compared to manual or classical tuning approaches [9,30,45–48].

Ant Colony Optimization, on the other hand, is derived from the foraging behavior of ants, where pheromone trails guide the collective search process. In control applications, ACO has been applied to optimize PID parameters for systems such as robotic arms, unmanned aerial vehicles, and rehabilitation robots. Its probabilistic search mechanism enables efficient exploration of parameter spaces, particularly when discrete search spaces or nonlinear dynamics are involved [49–53].

While both GA and ACO have demonstrated strong potential in PID tuning, the majority of prior studies have applied them independently to different robotic systems. Comparative evaluations are limited, especially for multi-degree-of-freedom (multi-DOF) manipulators where the control task is inherently more complex due to coupling effects, nonlinear dynamics, and parameter uncertainties. A direct comparison of GA and ACO in such systems is therefore necessary to understand their relative strengths and trade-offs in achieving optimal performance.

This gap in the literature provides the motivation for the present study, which systematically investigates the effectiveness of GA and ACO in tuning PID controllers for a five-degree-of-freedom robotic manipulator. By evaluating both approaches within the same experimental framework, the study seeks to provide deeper insights into the suitability of these algorithms for advanced robotic control applications.

3. Methodology

This section describes the framework adopted for modeling, control design, and optimization of the 5-DOF robotic manipulator. The process is structured into four main stages. First, the PID control law and its integration within the feedback architecture are presented. Second, the manipulator is modeled in SolidWorks and simulated in MATLAB/Simscape to capture its kinematic and dynamic behavior. Third, metaheuristic optimization techniques, specifically Genetic Algorithm (GA) and Ant Colony Optimization (ACO), are implemented to tune the PID controller gains using the Integral Time Absolute Error (ITAE) as the objective function. Finally, the performance of the tuned controllers is evaluated against manual tuning through comparative simulations, focusing on convergence characteristics, overshoot behavior, and trajectory tracking accuracy across all joints. The simulations were executed on a system equipped with an Intel Core i7 (2.3 GHz) processor, 32 GB RAM, using MATLAB/Simulink R2023a.

For sinusoidal reference inputs, the term overshoot in this study refers to the maximum deviation of the output from the reference trajectory (i.e., peak tracking error).

3.1. PID control and system architecture

The control objective is to ensure accurate trajectory tracking for the 5-DOF robotic manipulator. A Proportional–Integral–Derivative (PID) controller was adopted due to its simplicity, proven industrial reliability, and effectiveness in handling multi-joint robotic systems. The PID control law [54] is expressed in equation 1 as

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{d}{dt} e(t) \tag{1}$$

Where $u(t)$ is the control output, $e(t)$is the tracking error, k_p, k_i, k_d represents the proportional, integral, and derivative gains, respectively.

The PID controller was embedded within a feedback architecture, as shown in Figure 1. The reference trajectory serves as the input, while the controller generates joint-level commands based on the instantaneous error between the desired and actual positions. The closed-loop control system comprises the input trajectory, the PID controller, the manipulator dynamics (plant), and feedback signals from joint sensors. This configuration facilitates continuous error correction, thereby enabling the manipulator to maintain stable operation and achieve precise motion tracking, as illustrated in Figure 2.

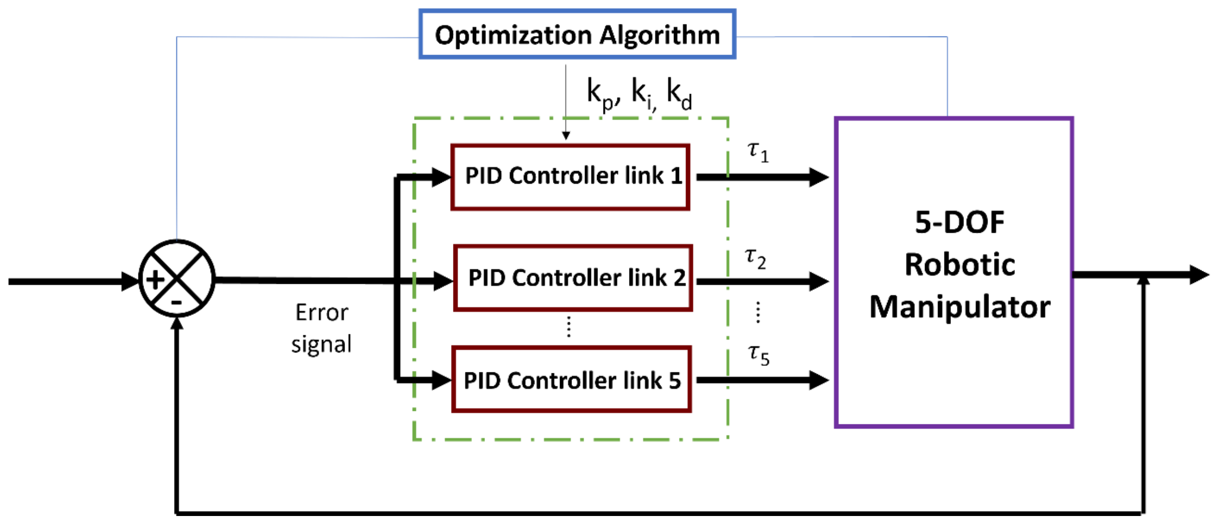

Fig. 1. Control structure architecture for the 5DOF robotic arm.

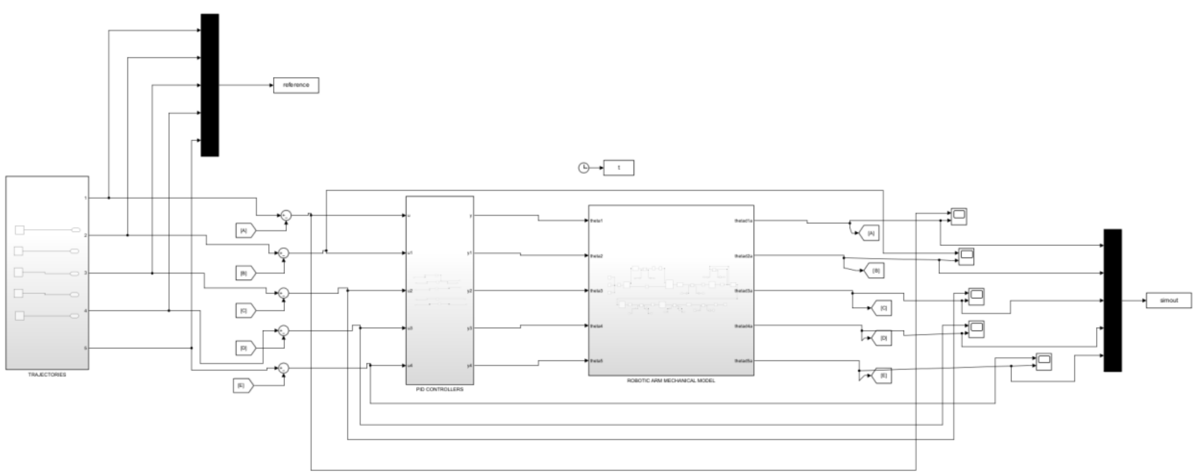

Fig. Simulink model of the closed-loop control architecture for the robotic manipulator.

3.2. System modeling and design

A CAD model of the robotic arm was developed using SolidWorks and incorporated into a MATLAB simulation for system validation. Figure 3 illustrates the CAD design of the robotic arm

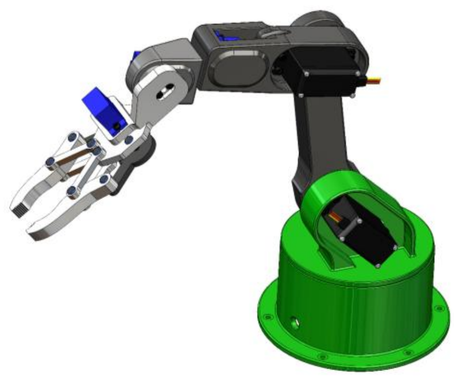

Fig. 3. Conceptual design of the robot arm.

3.3. Simscape model

Simscape is a MATLAB toolbox that allows modeling and simulating systems using a block-based approach. It was used to simulate the system. Figure 4 shows the Simscape model developed.

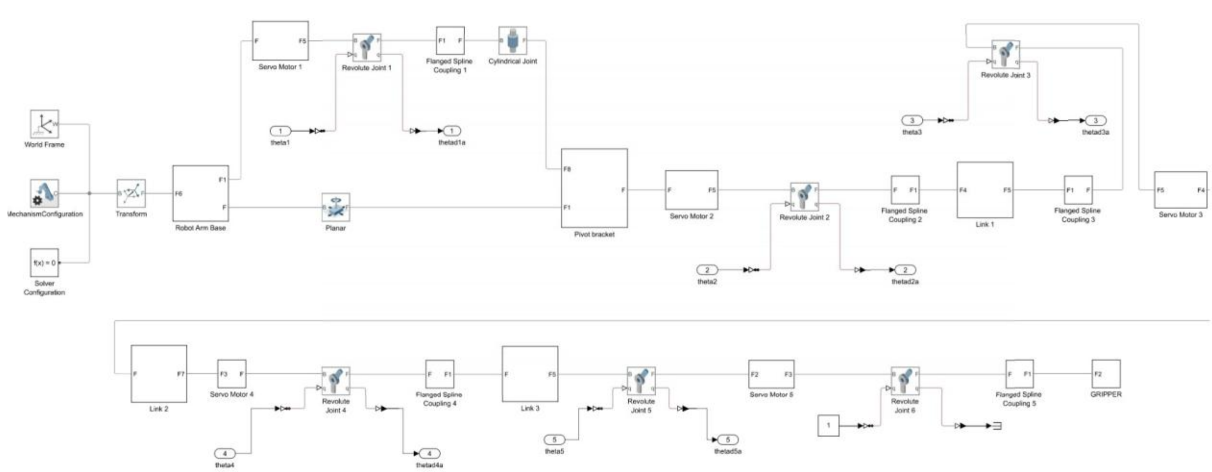

Fig. 4. Simscape model of the designed 5DOF robot arm.

3.4. Optimization problem formulation

In this study, Genetic Algorithm (GA) and Ant Colony Optimization (ACO) were employed to compute the optimal PID gains for the five-degree-of-freedom robotic manipulator. Both methods are stochastic global optimizers, capable of exploring diverse regions of the solution space. Their stochastic nature reduces the likelihood of premature convergence to local minima and enables the discovery of near-optimal gain configurations. The objective function selected for this work is the Integral Time Absolute Error (ITAE), defined as:

$$MinimizeITAE = \sum_{i=1}^{N} t_i . \vee e(t_i) \vee \tag{2}$$

Subject to:

$$k_p{}^{lower} \leq k_{p_j} \leq k_p{}^{upper}, \qquad j = 1, \dots, 5 \tag{3}$$

$$k_i^{lower} \leq k_{i_j} \leq k_i^{upper} \;, \qquad j = 1, \dots, 5 \tag{4}$$

$$k_d^{lower} \leq k_{d_j} \leq k_d^{upper} \;, \qquad j = 1, \dots, 5 \tag{5}$$

Where $t_i$is time at i-th sample, $e(t_i)$ is the tracking error at that instant, N is the total number of time steps. The tuning variables are the proportional $\left(k_{p_j}\right)$, integral $(k_{i_j}$, and derivative$\left(k_{d_j}\right)$ represent the proportional, integral, and derivative gains for the respective parts of the control system.

The ITAE criterion was selected over IAE and ISE because it penalizes sustained errors more heavily by weighting them with time. This makes it particularly suitable for robotic manipulators, where prolonged deviations from the desired trajectory degrade accuracy and dynamic performance.

The search space for each controller parameter was determined empirically from preliminary experiments and literature guidelines. The following ranges were applied:

$$k_p \in [0.1, 0.1, 0.1, 0.1, 0.1] \tag{6}$$

$$k_i \in [0.0001, 0.0001, 0.0001, 0.0001, 0.0001] \tag{7}$$

$$k_d \in [0.0001, 0.0001, 0.0001, 0.0001, 0.0001] \tag{8}$$

The optimization framework employed in this study is illustrated through the flowcharts of the Genetic Algorithm (GA) and Ant Colony Optimization (ACO), shown in Figures 4 and 5, respectively. These flowcharts outline the sequential steps of initialization, iterative search, and convergence, providing a systematic representation of how each algorithm was applied for PID parameter tuning.

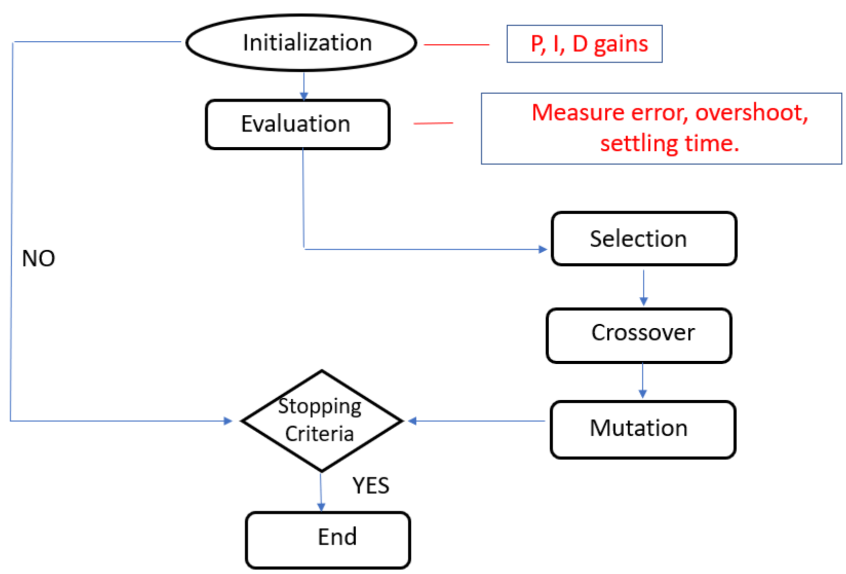

Fig. 5. Genetic Algorithm flowchart for PID controller tuning.

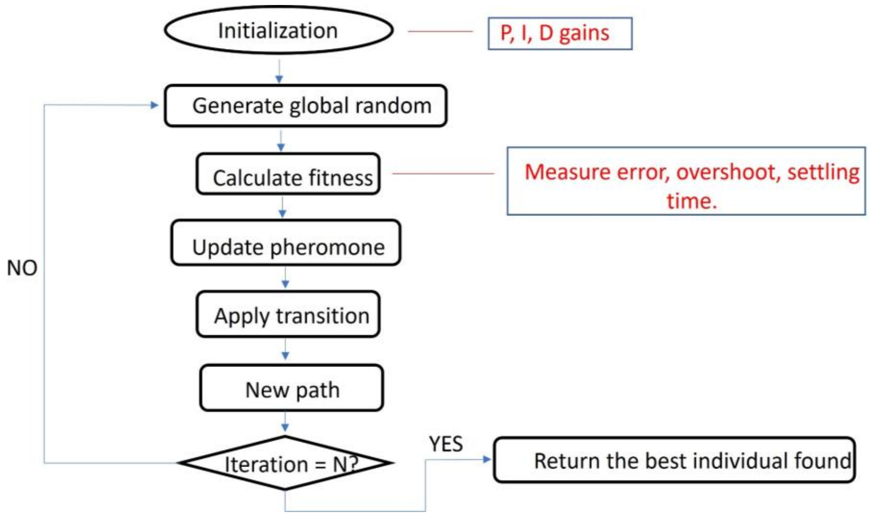

Fig. 6. Ant Colony Optimization algorithm flowchart for PID controller tuning.

To ensure reproducibility, the GA and ACO algorithms were configured with the parameter values listed in Tables 2.0 and 3.0 respectively. A maximum of 30 generations (iterations) was selected as the stopping criterion, or earlier if convergence tolerance was satisfied.

Table 2
Genetic Algorithm Parameters.

Genetic Algorithm parameters	Type/value
Maximum generation	30
Population size	5
Encoding	Binary
Selection	Uniform
Crossover	Single point crossover
Mutation	Uniform

Table 3
Ant Colony Optimization Parameters.

ACO parameters	Type/value
Maximum iteration/generation	30
Number of ants	5
Number of nodes	100
Evaporation rate	0.05
Alpha	0.5
Beta	0.2

4. Results and discussion

The effectiveness of the proposed optimization-based PID tuning strategies was assessed using the 5-DOF robotic manipulator model developed in Simscape. Genetic Algorithm (GA) and Ant

Colony Optimization (ACO) were applied to determine the controller gains for each joint, and their performance was compared with that of manually tuned parameters. The analysis focused on the optimized gain values, the convergence behavior of both algorithms, the resulting trajectory tracking and overshoot characteristics across the five joints.

Table 4.0 summarizes the tuned PID parameters for all five joints obtained through manual tuning, GA, and ACO. These values represent the final configurations after optimization and serve as the basis for subsequent performance analysis. It is noted that the negative derivative coefficients observed in the manually tuned controller reflect the unconstrained nature of the trial-and-error process rather than simulation inaccuracies. Such values indicate compensatory parameter interactions within the PID structure and are therefore interpreted as evidence of suboptimal manual tuning rather than as invalid results.

Table 4.0
Gains for the system.

Tuning method	Joint number	Proportional gain (k_p)	Integral gain (k_i)	Derivative gain (k_d)
Manual	i.	0.98	0.063	0.0179
	ii.	1.15	0.035	-8.35
	iii.	0.97	0.035	-8.35
	iv.	1.14	0.035	-8.35
	v.	1.0	0.004	0.00065
ACO	i.	0.77	0.0017	0.0021
	ii.	0.94	0.0029	0.0024
	iii.	0.78	0.0026	0.0023
	iv	0.85	0.001	0.0026
	v.	0.87	0.012	0.0025
GA	i.	0.98	0.0011	0.0015
	ii.	0.63	0.0041	0.0012
	iii.	0.6	0.0029	0.0020
	iv.	0.83	0.0016	0.0019
	v.	0.97	0.0041	0.0013

Figure 5 illustrates the convergence of the Genetic Algorithm (GA) in minimizing the ITAE cost function. GA rapidly reduced the objective value within the first 10 generations and stabilized thereafter, indicating fast convergence. In contrast, Figure 6 shows the convergence behavior of the Ant Colony Optimization (ACO) algorithm. ACO exhibited a slower rate of reduction in the early generations but continued to improve steadily, ultimately reaching a lower ITAE value after 20 generations. This demonstrates that GA is computationally more efficient in terms of convergence speed, whereas ACO provides superior final accuracy by avoiding premature stagnation. The trade-off highlights GA's advantage in applications requiring rapid online tuning, while ACO is better suited for offline or high-precision scenarios where minimizing residual error is critical.

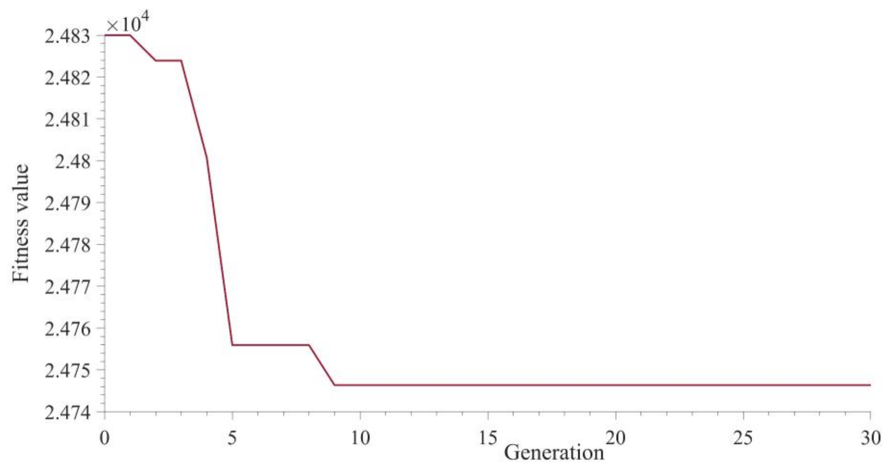

Fig. 7. Convergence plot for Genetic algorithm.

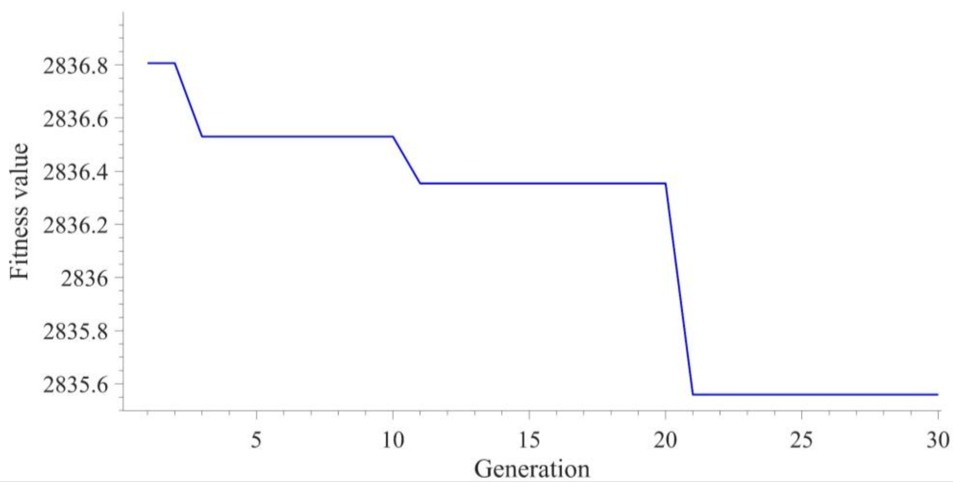

Fig. 8. Convergence plot for Ant Colony Optimization algorithm.

Figures 9 – 13 present the trajectory tracking performance of the five joints under manual tuning, GA-based tuning, and ACO-based tuning. The results indicate that both GA and ACO significantly improve tracking accuracy compared to manual tuning, with reduced overshoot and faster convergence. Notably, ACO achieves smoother responses for Joints 2–4, whereas GA provides closer tracking in Joints 1 and 5. These differences reflect the stronger global search capability of ACO at the expense of slower convergence, while GA favors rapid tuning but may sacrifice robustness across all joints.

In addition to ITAE evaluation, overshoot analysis was conducted across all five joints to assess transient performance and trajectory tracking accuracy. Overshoot is a critical performance index in robotic manipulators, as excessive deviations from the desired trajectory can induce mechanical stress, instability, and degraded precision in multi-joint coordination. Table 5.0 summarizes the overshoot values obtained for manual tuning, GA-optimized PID, and ACO-optimized PID. The results indicate that both GA and ACO significantly reduced overshoot compared to manual tuning, although their performance varied across different joints. GA achieved the lowest overshoot for Joints 1 and 5, demonstrating superior precision in these cases, whereas ACO consistently outperformed manual tuning in all joints and showed stronger suppression of overshoot in Joints

2, 3, and 4. These findings highlight the trade-off between the two algorithms: GA offers localized improvements in precision for certain joints, while ACO provides more uniform overshoot reduction across the manipulator.

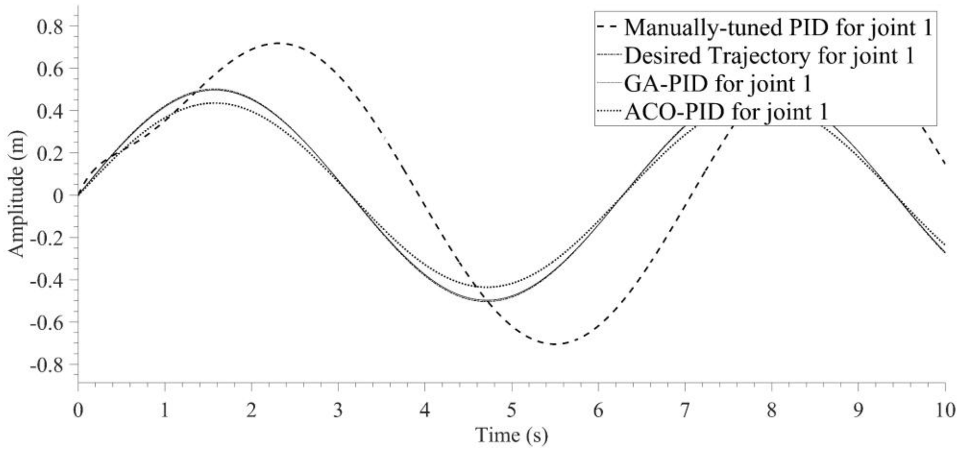

Fig. 9. Trajectory tracking graph for joint 1.

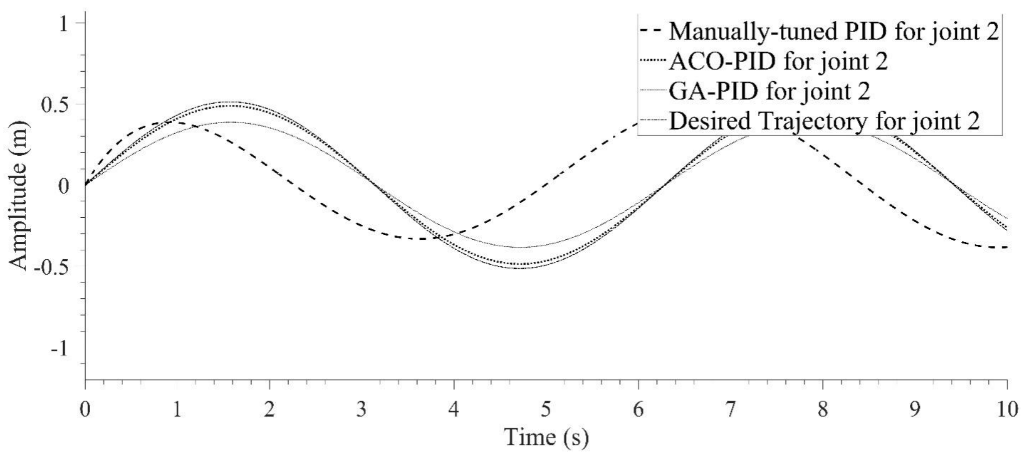

Fig. 10. Trajectory tracking graph for joint 2.

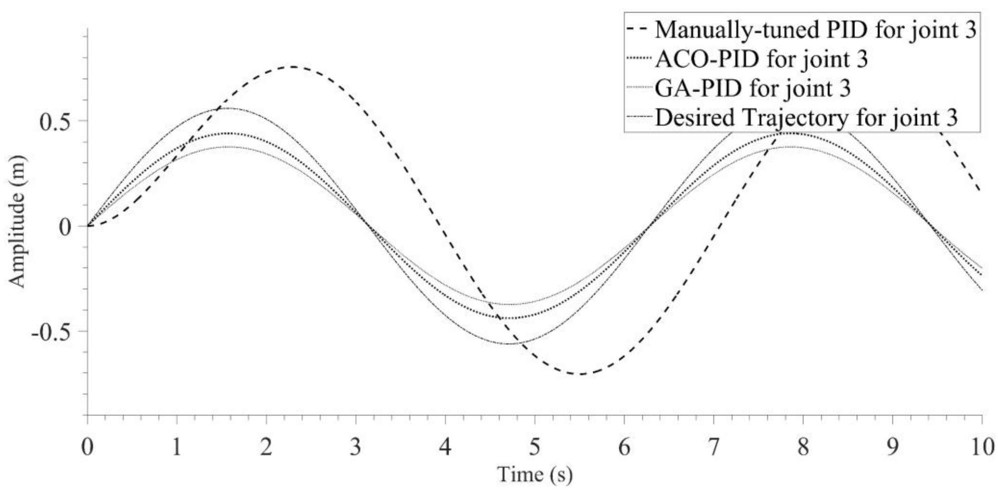

Fig. 11. Trajectory tracking graph for joint 3.

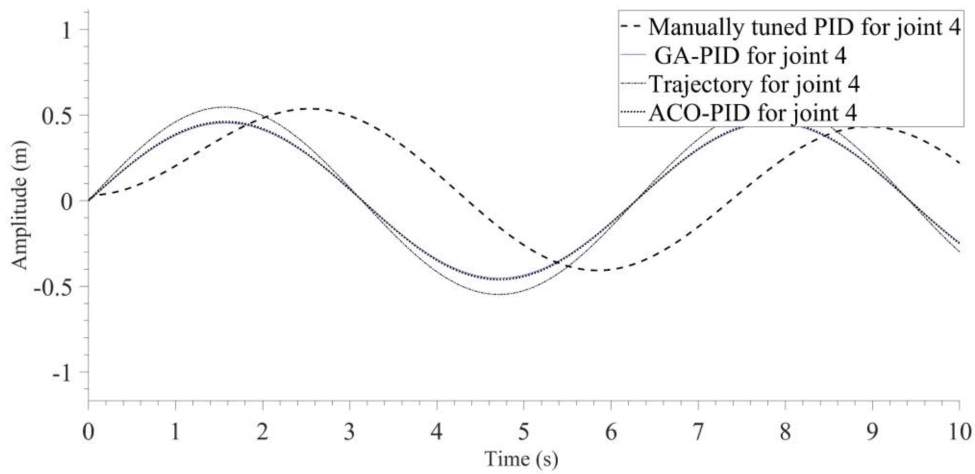

Fig. 12. Trajectory tracking graph for joint 4.

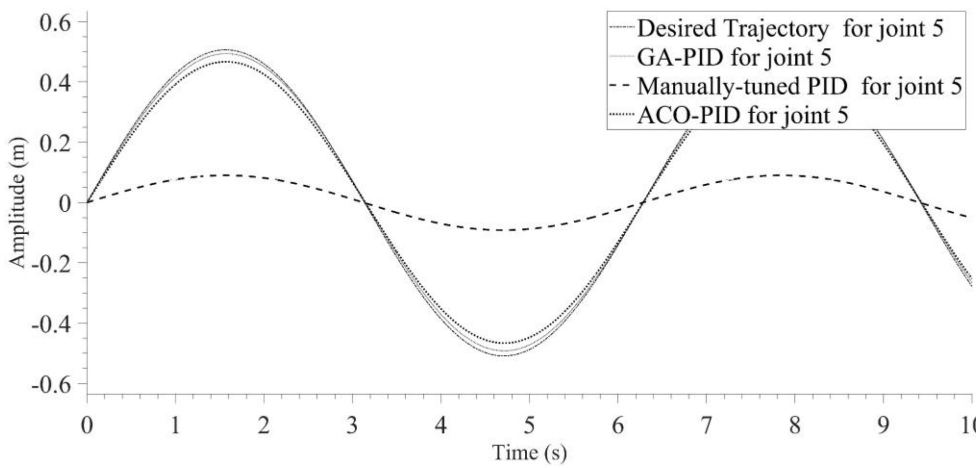

Fig. 13. Trajectory tracking graph for joint 5.

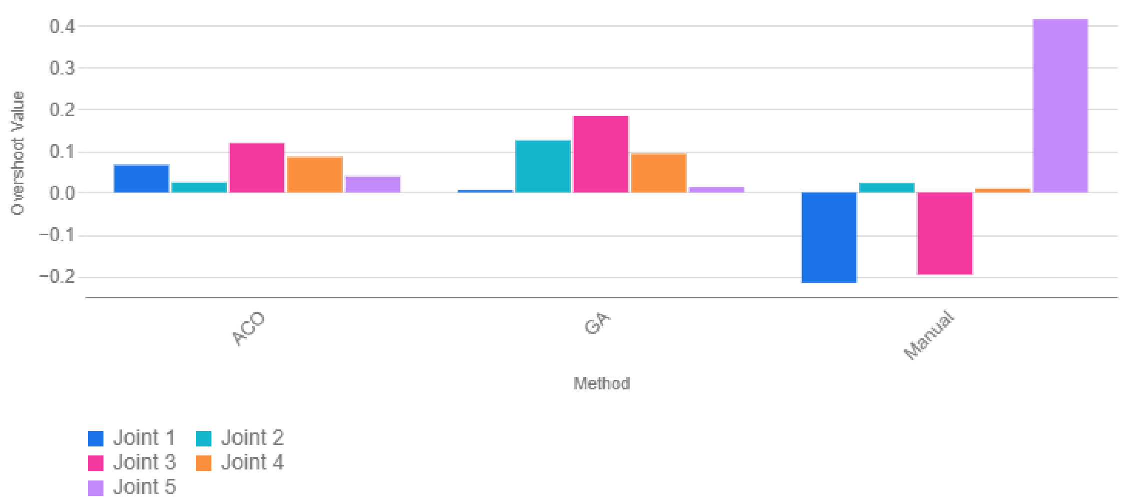

Fig. 14. Overshoot Comparison by Algorithm and Joint.

Table 5
Overshoot values for each joint for each tuning method applied.

Overshoot	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5
Manual	-0.2164	0.0234	-0.1965	0.0098	0.41613
ACO	0.0665	0.0248	0.1194	0.0852	0.0396
GA	0.0051	0.1258	0.1835	0.0928	0.0126

The chart below in Figure 14 shows a visual representation of the overshoot clearly denoting the lower overshoot that occurred in ACO as compared to other methods.

5. Conclusion and recommendation

This study compared Genetic Algorithm (GA) and Ant Colony Optimization (ACO) for PID controller tuning in a five-degree-of-freedom robotic manipulator. Both optimization methods significantly improved trajectory tracking and stability compared to manual tuning, confirming the effectiveness of metaheuristic techniques in robotic control.

Quantitatively, GA and ACO reduced the average overshoot by approximately 51% and 61%, respectively, relative to manual tuning. In terms of convergence, GA reached its optimal fitness value within 10 iterations, whereas ACO required about 30 iterations but achieved a lower final ITAE value, improving by approximately 0.04% compared to GA's 0.36%. These results demonstrate that both algorithms achieved substantial enhancements in tracking precision and stability across all joints. The observed performance differences stem from their underlying search mechanisms. GA's crossover and mutation operations enable rapid global exploration, resulting in faster convergence but slightly higher steady-state error. Conversely, ACO's pheromone-update process promotes gradual local refinement, yielding smoother convergence and more accurate joint tracking.

Overall, the findings reveal a distinct trade-off between computational efficiency and control accuracy: GA excels in speed, while ACO provides superior precision and robustness. By systematically comparing these two metaheuristic approaches on a multi-DOF manipulator, this work offers practical insights into the complementary strengths of evolutionary and swarm-based optimization in PID controller tuning. Future work will focus on experimental validation using a physical robotic platform to assess performance under variable payloads and disturbances. Additionally, hybrid or multi-objective optimization strategies may be explored to further balance accuracy, convergence speed, and energy efficiency in advanced robotic applications.

Funding

This research received no external funding.

Conflicts of Interest

The authors declare no conflict of interest.

Authors contribution statement

Adeleke Olorunnisola: Model development, Simulation, Data analysis, Software implementation, Visualization, original draft, review & editing.

Olurotimi Dahunsi: Model development, Simulation, Data analysis, Software implementation, Visualization, original draft, review & editing.

References

[1] Tinoco V, Silva MF, Santos FN, Morais R, Magalhães SA, Oliveira PM. A review of advanced controller methodologies for robotic manipulators. Int J Dyn Control 2025;13:36. https://doi.org/10.1007/s40435-024-01533-1.

[2] Sariyildiz E, Yu H, Ohnishi K. A Practical Tuning Method for the Robust PID Controller with Velocity Feed-Back. Machines 2015;3:208–22. https://doi.org/10.3390/machines3030208.

[3] Somefun OA, Akingbade K, Dahunsi F. The dilemma of PID tuning. Annu Rev Control 2021;52:65–74. https://doi.org/10.1016/j.arcontrol.2021.05.002.

[4] Joseph SB, Dada EG, Abidemi A, Oyewola DO, Khammas BM. Metaheuristic algorithms for PID controller parameters tuning: review, approaches and open problems. Heliyon 2022;8:e09399. https://doi.org/10.1016/j.heliyon.2022.e09399.

[5] Bartz-Beielstein T, Branke J, Mehnen J, Mersmann O. Evolutionary Algorithms. WIREs Data Min Knowl Discov 2014;4:178–95. https://doi.org/10.1002/widm.1124.

[6] Gonzalez TF, editor. Handbook of Approximation Algorithms and Metaheuristics. Chapman and Hall/CRC; 2007. https://doi.org/10.1201/9781420010749.

[7] Ünal M, Ak A, Topuz V, Erdal H. Optimization of PID controllers using ant colony and genetic algorithms. vol. 449. Springer; 2012.

[8] Sharma R, Rana KPS, Kumar V. Comparative Study of Controller Optimization Techniques for a Robotic Manipulator, 2014, p. 379–93. https://doi.org/10.1007/978-81-322-1771-8_33.

[9] Son VN, Van Cuong P, Minh ND, Nha PH. Optimize the parameters of the PID Controller using Genetic Algorithm for Robot Manipulators. ArXiv Prepr ArXiv250104759 2025.

[10] Dahmane SA, Azzedine A, Megueni A. Ant colony optimization algorithm based on optimal PID parameters for a robotic arm. Int J Control Syst Robot 2020;5.

[11] Cao Y, Zhou Q, Yuan W, Ye Q, Popa D, Zhang Y. Human-robot collaborative assembly and welding: A review and analysis of the state of the art. J Manuf Process 2024;131:1388–403. https://doi.org/10.1016/j.jmapro.2024.09.044.

[12] Thomas G, Akporhuarho J. The Basics and Significance of Industrial Robots in Manufacturing: A Review. Niger J Eng Sci Res 2021;4:38–54.

[13] Adebayo RA, Obiuto NC, Festus-Ikhuoria IC, Olajiga OK. Robotics in manufacturing: A review of advances in automation and workforce implications. Int J Adv Multidiscip Res Stud 2024;4:632–8.

[14] Ansari ZJ, Aher A, Thitame SN. Advancements in Robotics and AI Transforming Surgery and Rehabilitation. J Pharm Bioallied Sci 2025;17:S46–8. https://doi.org/10.4103/jpbs.jpbs_1937_24.

[15] Banyai AD, Brișan C. Robotics in Physical Rehabilitation: Systematic Review. Healthcare 2024;12:1720. https://doi.org/10.3390/healthcare12171720.

[16] Yang Q, Du X, Wang Z, Meng Z, Ma Z, Zhang Q. A review of core agricultural robot technologies for crop productions. Comput Electron Agric 2023;206:107701. https://doi.org/10.1016/j.compag.2023.107701.

[17] Mills JK. Stability and control aspects of flexible link robot manipulators during constrained motion tasks. J Robot Syst 1992;9:933–53. https://doi.org/10.1002/rob.4620090705.

[18] Xu S, Wu Z, Shen T. High-Precision Control of Industrial Robot Manipulator Based on Extended Flexible Joint Model. Actuators 2023;12:357. https://doi.org/10.3390/act12090357.

[19] Lyu G, Wang P, Li G, Lu F, Dai S. Precise torques and sliding mode compensation for trajectory tracking of manipulator with uncertainty. Int J Adv Robot Syst 2022;19:172988062211212. https://doi.org/10.1177/17298806221121212.

[20] Azeez MI, Atia KR. Modeling of PID controlled 3DOF robotic manipulator using Lyapunov function for enhancing trajectory tracking and robustness exploiting Golden Jackal algorithm. ISA Trans 2024;145:190–204. https://doi.org/10.1016/j.isatra.2023.11.033.

[21] Rakhmatillaev J, Bucinskas V, Kabulov N. An integrative review of control strategies in robotics. Robot Syst Appl 2025. https://doi.org/10.21595/rsa.2025.25014.

[22] Pezeshki S, Badalkhani S, Javadi A. Performance Analysis of a Neuro-PID Controller Applied to a Robot Manipulator. Int J Adv Robot Syst 2012;9. https://doi.org/10.5772/51280.

[23] Wei B. Adaptive Control Design and Stability Analysis of Robotic Manipulators. Actuators 2018;7:89. https://doi.org/10.3390/act7040089.

[24] Dawson D, Quf† Z, Duffie J. Robust tracking control for robot manipulators: theory, simulation, and implementation. Robotica 1993;11:201–8. https://doi.org/10.1017/S0263574700016064.

[25] Shoja Majidabad S, Toosian Shandiz H. Discrete-time based sliding-mode control of robot manipulators. Int J Intell Comput Cybern 2012;5:340–58. https://doi.org/10.1108/17563781211255880.

[26] Durmuş B, Temurtaş H, Yumuşak N, Temurtaş F. A study on industrial robotic manipulator model using model based predictive controls. J Intell Manuf 2009;20:233–41. https://doi.org/10.1007/s10845-008-0221-2.

[27] Dai L, Yu Y, Zhai D-H, Huang T, Xia Y. Robust Model Predictive Tracking Control for Robot Manipulators With Disturbances. IEEE Trans Ind Electron 2021;68:4288–97. https://doi.org/10.1109/TIE.2020.2984986.

[28] Kern J, Marrero D, Urrea C. Fuzzy Control Strategies Development for a 3-DoF Robotic Manipulator in Trajectory Tracking. Processes 2023;11:3267. https://doi.org/10.3390/pr11123267.

[29] Liu Y, Chen B, Ma L, Yang S, Li R, Guo Y. A novel trajectory tracking control approach for uncertain 6-DOF manipulators based on fuzzy sliding mode of radial basis function neural network. Proc Inst Mech Eng Part I J Syst Control Eng 2023;237:1032–44. https://doi.org/10.1177/09596518221144796.

[30] Sheng L, Li W. Optimization Design by Genetic Algorithm Controller for Trajectory Control of a 3-RRR Parallel Robot. Algorithms 2018;11:7. https://doi.org/10.3390/a11010007.

[31] Baghli FZ, Lakhal Y, Kadi YA El. The Efficiency of an Optimized PID Controller Based on Ant Colony Algorithm (ACO-PID) for the Position Control of a Multi-articulated System. J Robot Control 2023;4:289–98. https://doi.org/10.18196/jrc.v4i3.17709.

[32] Patidar AK, Sinha SS, Mukherjee S. Dimensional Synthesis of Delta Manipulator Using Genetic Algorithm-Based Multi-objective Optimization, 2021, p. 609–24. https://doi.org/10.1007/978-981-15-4477-4_44.

[33] Alvarez-Ramirez J, Cervantes I, Kelly R. PID regulation of robot manipulators: stability and performance. Syst Control Lett 2000;41:73–83. https://doi.org/10.1016/S0167-6911(00)00038-4.

[34] Schwenzer M, Ay M, Bergs T, Abel D. Review on model predictive control: an engineering perspective. Int J Adv Manuf Technol 2021;117:1327–49. https://doi.org/10.1007/s00170-021-07682-3.

[35] Ziegler JG, Nichols NB. Optimum Settings for Automatic Controllers. J Dyn Syst Meas Control 1993;115:220–2. https://doi.org/10.1115/1.2899060.

[36] Cohen GH, Coon GA. Theoretical Consideration of Retarded Control. J Fluids Eng 1953;75:827–34. https://doi.org/10.1115/1.4015451.

[37] Skogestad S. Simple analytic rules for model reduction and PID controller tuning. J Process Control 2003;13:291–309. https://doi.org/10.1016/S0959-1524(02)00062-8.

[38] Åström KJ, Hägglund T. The future of PID control. Control Eng Pract 2001;9:1163–75. https://doi.org/10.1016/S0967-0661(01)00062-4.

[39] Unbehauen H. Control systems, robotics and automation-volume III: system analysis and control: classical approaches-III. EOLSS Publications; 2009.

[40] Visual Servoing, 2010, p. 407–67. https://doi.org/10.1007/978-1-84628-642-1_10.

[41] Mezura-Montes E, Coello CAC. An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. Int J Gen Syst 2008;37:443–73. https://doi.org/10.1080/03081070701303470.

[42] D MR, B K, P SD, C G, S NK, K K. PID Controller Optimization for a Three-Link Robot Manipulator Using RGA. 2024 Int. Conf. Integr. Emerg. Technol. Digit. World, IEEE; 2024, p. 1–5. https://doi.org/10.1109/ICIETDW61607.2024.10940004.

[43] Chandrababu Naik B, Ravi Kumar C V., Jweeg MJ, Tolephih MH, Hari Prasad M, Mohammed MN, et al. Designing a PID Controller Using Ant Colony Optimization and Implementing It with FPGA, 2025, p. 965–76. https://doi.org/10.1007/978-3-031-84628-1_82.

[44] Eberhart R, Kennedy J. A new optimizer using particle swarm theory. MHS'95. Proc. Sixth Int. Symp. Micro Mach. Hum. Sci., IEEE; n.d., p. 39–43. https://doi.org/10.1109/MHS.1995.494215.

[45] Goud H, Chandra Sharma P, Nisar K, Reazul Haque M, Asri Ag. Ibrahim A, Singh Yadav N, et al. Metaheuristics Algorithm for Tuning of PID Controller of Mobile Robot System. Comput Mater Contin 2022;72:3481–92. https://doi.org/10.32604/cmc.2022.019764.

[46] Tanveer A, Ahmad SM. Genetic-Algorithm-Based Proportional Integral Controller (GAPI) for ROV Steering Control. INTERACT 2023, Basel Switzerland: MDPI; 2023, p. 4. https://doi.org/10.3390/engproc2023032004.

[47] Souza DA, Batista JG, dos Reis LLN, Júnior ABS. PID controller with novel PSO applied to a joint of a robotic manipulator. J Brazilian Soc Mech Sci Eng 2021;43:377. https://doi.org/10.1007/s40430-021-03092-4.

[48] Delavari H, Ghaderi R, Ranjbar NA, HosseinNia SH, Momani S. Adaptive fractional PID controller for robot manipulator. ArXiv Prepr ArXiv12062027 2012.

[49] Duan H, Wang D, Yu X. Novel approach to nonlinear PID parameter optimization using ant colony optimization algorithm. J Bionic Eng 2006;3:73–8. https://doi.org/10.1016/S1672-6529(06)60010-3.

[50] Herlambang T, Rahmalia D, Yulianto T. Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) for optimizing PID parameters on Autonomous Underwater Vehicle (AUV) control system. J Phys Conf Ser 2019;1211:012039. https://doi.org/10.1088/1742-6596/1211/1/012039.

[51] Kang Y, Li Z, Wang T. Application of PID Control and Improved Ant Colony Algorithm in Path Planning of Substation Inspection Robot. Math Probl Eng 2022;2022:1–10. https://doi.org/10.1155/2022/9453219.

[52] Suwoyo H, Tian Y, Adriansyah A, Ibnu Hajar MH, Zhou T. The ACA-based PID Controller for Enhancing a Wheeled-Mobile Robot. J FORTEI-JEERI 2020;1:19–23. https://doi.org/10.46962/forteijeeri.v1i2.15.

[53] Singh R, Bhushan B. Improved ant colony optimization for achieving self-balancing and position control for balancer systems. J Ambient Intell Humaniz Comput 2021;12:8339–56. https://doi.org/10.1007/s12652-020-02566-y.

[54] Astrom KJ, Hagglund T. PID Controllers: Theory, Design and Tuning. Research. Triangle Park NC Instrum Soc Amer 1995:241–3.