Analysis And Comparison of Different Tuning Method of PID Control in Robot Manipulator

Jiyan Li *

International College, Chongqing University of Posts and Telecommunication, Chongqing, 400065, China

* Corresponding Author Email: jl3597@nau.edu

Abstract. This paper provides a comprehensive examination of the background and theory of Proportional-Integral-Derivative (PID) control in robot manipulators. It not only compares the different tuning methods, but also offers analysis of the commonly used Metaheuristic algorithms tuning approaches employed by previous researchers. Furthermore, it discusses the challenges encountered and the potentials unlocked when applying PID control to robot manipulators, considering factors such as system complexity, robustness and stability. The insights gained from this research are of utmost significance for the continued development and industrial implementation of PID control in the field of robotics. By improving the understanding of PID control and its performance characteristics, this study contributes to enhancing the precision, stability, and adaptability of robot manipulators in various industries and applications, including manufacturing, automation, and autonomous systems. Overall, this paper sheds light on the future directions and potential advancements in PID control for robot manipulators.

Keywords: PID control, robot manipulator, Metaheuristic algorithms.

1. Introduction

Proportional, integral, and differential components collectively form the PID acronym, representing a comprehensive control method. The PID control algorithm, as its name suggests, integrates these three elements to deliver a robust control strategy. In continuous systems, it stands as the most well-established and commonly employed control algorithm. Particularly in scenarios where the model of the controlled system is not fully understood, the PID control algorithm, which emerged in the 1930s and 1940s, proves to be suitable. PID control is widely used in many areas such as manufacturing, chemical processing, and robotics. The reason behind its widespread adoption lies in its ability to maintain desired setpoints with precision and optimize the overall performance of systems [1]. In robot manipulators. By measuring feedback signals such as the position, speed or torque of the robot arm, the PID controller can calculate the appropriate control output to achieve accurate control and movement of the robot arm. PID control is simple, stable and easy to implement, so it is widely used in many robot arm applications. However, traditional PID control may have some limitations in some cases.

In order to, Explore the application of different PID tuning methods in the robot manipulator. First, this paper introduces the background of PID and some applications in the robot manipulator. In section 2, this paper introduces the basic principle of PID and some tuning methods. In section 3, The tuning method used by some predecessors is analyzed. In section 4, Some of the challenges and potential for applying PID to robotic manipulators are described. The research in this paper will contribute to the further development of PID control in the field of robotics and its further application in industrialization.

2. The Principle of PID control

2.1. The analysis of PID control principle

PID control, which stands for Proportional-Integral-Derivative control, is a widely used feedback control algorithm employed in various control systems. The PID control flow is shown in figure 1 [2].

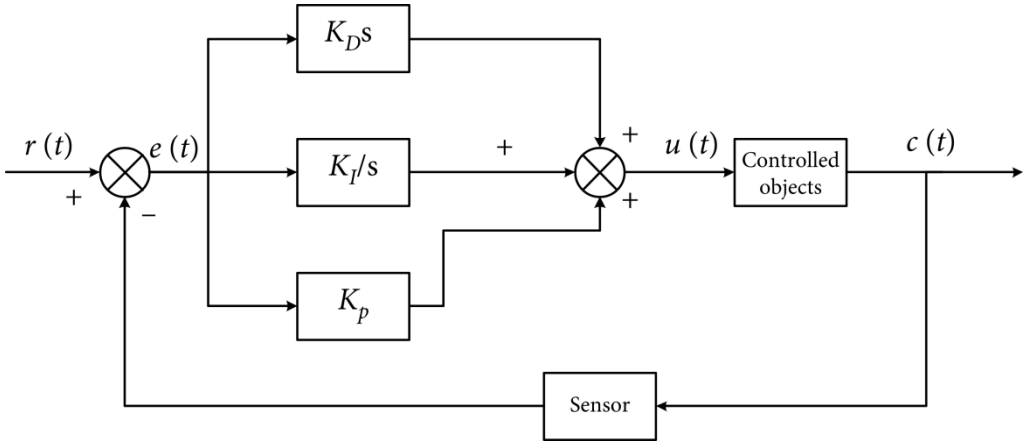


Fig 1. A block diagram of a PID controller in a feedback loop [2]

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \tag{1}$$

(1) Control that is proportionate to the current error is computed by the proportional term (P). The discrepancy between the setpoint and the actual process variable is the error.

By dividing the mistake by a proportional gain constant K_p, the adjustment is calculated. The size of the corrective action for a particular error increase with the proportional gain. A low gain, on the other hand, can cause a sluggish response, while a high gain could cause overshooting or instability.

Integral (I) Control: The integral term corrects the output by integrating the accumulated error over time. By using this method, steady-state mistakes that could happen with proportional control alone are reduced. The integral term is calculated by dividing the accumulated error by the integral gain constant K_i ($K_i = K_p T_i$, T_i is reset time). It accounts for both the size and its duration of the error. The control system can gradually reduce any prolonged departures from the setpoint by integrating the error over time.

3)Derivative (D) Control: Based on the rate of change of the error, the derivative term forecasts its future trend. In order to lessen overshooting and slow the response of system, it offers a corrective action proportional to the rate of error change. The derivative term is calculated by dividing the rate of change of the error by the derivative gain constant K_d ($K_d = K_p / T_d$, T_d is the derivative time). It enables the system to foresee the behavior of the error and respond appropriately [3].

The PID controller continuously adjusts the output by calculating and summing these three terms. The controller parameters (K_p K_i K_d) must be carefully tuned to achieve the desired response. Tuning involves adjusting the gains to balance the control system's stability, responsiveness, and accuracy.

2.2. Tuning method of PID control

2.2.1 Classical PID controller tuning method

1) Ziegler-Nichols Method

The most fundamentally used tuning method was first put forth by Ziegler and Nichols in 1942. The Ziegler-Nichols approach can be debugged by initially setting the proportional gain to zero, then progressively raising it until the limit gain K_u is reached, at which time the controller output value oscillates at a constant value. Table 1 displays how the K_u and oscillation period T_u was selected, along with the proportional, integral, and differential gains for the various types [4].

Table 1. Ziegler-Nichols method parameters [4]

Control type	K_p	K_i	K_d
P	$K_u / 2$		
PI	$K_u / 2.2$	$1.2K_p / T_u$	
PID	$0.6K_u$	$2K_p / T_u$	$K_p T_u / 8$

2) Cohen-Coon method

A method with PID controller settings generated by a First Order Plus Time Delay model was proposed by Cohen and Coon. The elimination of load turbulence is the main design requirement. Cohen presents the controller parameter configuration. The process of the Cohen-Coon turning method is as follows: Observe the process until it achieves a steady state. Change the input step by step.

Get a rough first-order process based on the output with a time constant τ (the time for the response to take place) that is τ_{dead} (dead time, also known as transportation lag, is the period of time between the introduction of a disturbance and the output signal's first indication of a change) units behind the introduction of the input step [5]. Cohen and Coon's technique results are not significantly better than Ziegler and Nichols approach The Cohen-Coon parameters is shown in Table 2 and its explanation is shown in table 3.

Table 2. Cohen-Coon method parameters [6]

Control type	K_p	T_d	T_d
P	$\frac{p}{NL}(1+\frac{R}{3})$		
PI	$\frac{p}{NL}(0.9+\frac{R}{12})$	$L\frac{30+3R}{9+20R}$	
PID	$\frac{p}{NL}(1.33+\frac{R}{4})$	$L\frac{30+3R}{9+20R}$	$\frac{4L}{11+2R}$

Table 3. Explanations of Cohen-Coon method parameters

P	Percent change of input
N	Percent change of output/ τ
L	τ_{dead}
R	$\frac{\tau_{dead}}{\tau}$

2.2.2 Optimization techniques

These approaches involve modifying PID parameters using cost function optimization methods and data modeling. Their strategies for controlling controller parameters are based on a cost function that they seek to minimize.

1)Ant Colony Optimization (ACO) is a metaheuristic optimization technique that draws inspiration from ant foraging behavior. It is predicated on the idea that ants use pheromone trails to explore and communicate while in search of food. Numerous optimization issues, including combinatorial optimization and routing issues, have been solved with success using ACO [7].

The main goal of ACO is to simulate ant colonies in order to identify optimal or nearly optimal solutions. The algorithm creates a set of solutions iteratively and modifies the pheromone trails according to the effectiveness of the solutions. The main idea behind ACO is to simulate the collective

behavior of ants to find optimal or near-optimal solutions. The algorithm works by iteratively constructing a set of solutions and updating pheromone trails based on the quality of the solutions.

2)Whale Optimization Algorithm (WOA) is a metaheuristic optimization method that draws its inspiration from whales' propensity for hunting. It is a population-based program that imitates humpback whales' social interactions and hunting techniques [8]. Numerous optimization issues, including PID controller tuning, have been solved with WOA.

3)Particle Swarm Optimization (PSO) is a population-based stochastic optimization technique that mimics the actions of a swarm of particles in search of the best course of action. Every particle updates its position and velocity based on both its own best-known position and the best position determined by the swarm. Each particle represents a potential solution. To converge to an ideal or nearly ideal solution, the PID parameters are modified iteratively [9].

4)Simulated Annealing (SA) is a stochastic optimization technique that draws inspiration from the metallurgical annealing process. To avoid being stuck in local optima, it starts with a solution and then probabilistically accepts "worse" solutions as it moves across the search space. The search is guided by the objective function, and the temperature parameter is gradually decreased to modify the PID settings [10].

3. Analysis different tuning method using in manipulator

3.1. Classical PID controller analysis in manipulator

Traditional PID control may have some limitations in some cases. For example, for robotic arm systems that are nonlinear, time-varying, or have complex dynamics, traditional PID control may not meet the requirements. Therefore, many researchers focus on improving PID control algorithms to improve the performance of manipulator systems [11].

3.2. Metaheuristic algorithms analysis for PID tuning in manipulator

According to recent research, heuristic algorithm-based optimization strategies have become effective tools for addressing a range of control engineering problems. Control of processes frequently employs metaheuristic algorithms for their quick responses, potent optimization capabilities, and simple architectural layout. When it comes to tackling higher-dimensional problems in optimization, metaheuristic techniques are more effective than conventional ones. Because of their robustness, they are highly adaptable to existing conventional controller design methodologies. Regardless of model order, metaheuristic algorithms can be used as a core mechanism to develop conventional and better organized controllers for a set of unstable operational models [5]. In the sections that follow, a robotic manipulator employs a few meta-heuristic methods.

1)Ant Colony Optimization

A controller for the robotic arm was created by Sandoval. This controller ought to be able to supply the motors with enough current to enable the movement of each Degrees of Freedom (DOF). The controller should also be able to recall the original controller and the locations of each DOF. Each DOF's movement speed can be controlled by the built-in controller. A PID algorithm is used to control each DOF in accordance with the required movement of each axis (degrees of freedom and mechanical torque). ACO is used to modify the PID tuning, enabling control of the robotic arm. ACO-PID outperformed ZN-tuned PID in terms of performance. Figure 2 below illustrates the idea of ACO [12].

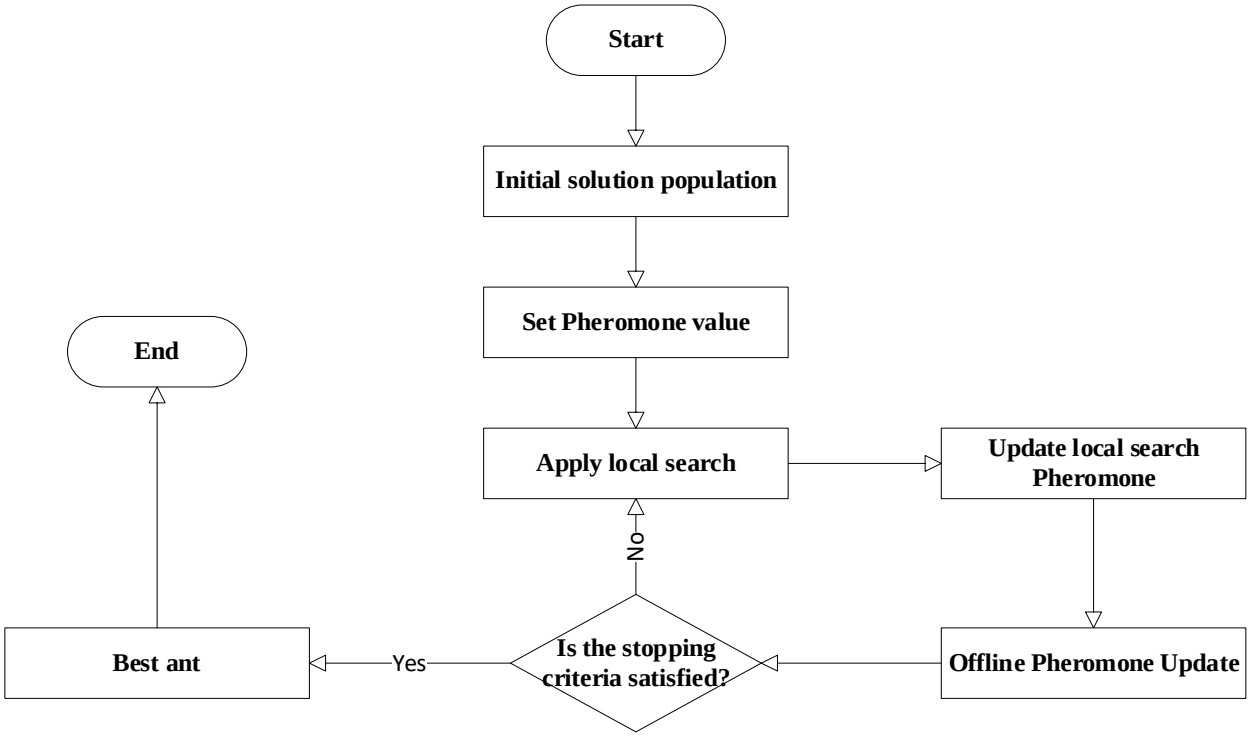


Fig 2. Ant colony optimization search algorithm flowchart (Photo/Picture credit: Original)

2)Whale Optimization Algorithm

On a simulation of a robotic manipulator and its controller, Loucif.F tested three metaheuristic algorithms, namely PSO-PID, GWO-PID, and WOA-PID, using Simulink/MATLAB. The WOA-PID controller performs better than the other controllers in terms of trajectory tracking, settling time, and ITAE error, according to the simulation findings. The outcomes show that the WOA-PID controller controls the robotic manipulator more effectively than it did before. The WOA-PID controller can more precisely follow the specified trajectory, according to the trajectory tracking metric [13]. The settling time suggests that the manipulator can be stabilized at the target position by the WOA-PID controller more quickly. Additionally, the WOA-PID controller may be able to more effectively reduce the integral error of the system given the benefit in terms of ITAE error. In similar robotic manipulator applications, the WOA-PID controller might be a superior option for obtaining improved control performance. To assure the stability and applicability of these results, taking into account additional parameters and practical application constraints, additional validation and testing are advised. The development of WOA is depicted in figure 3 below.

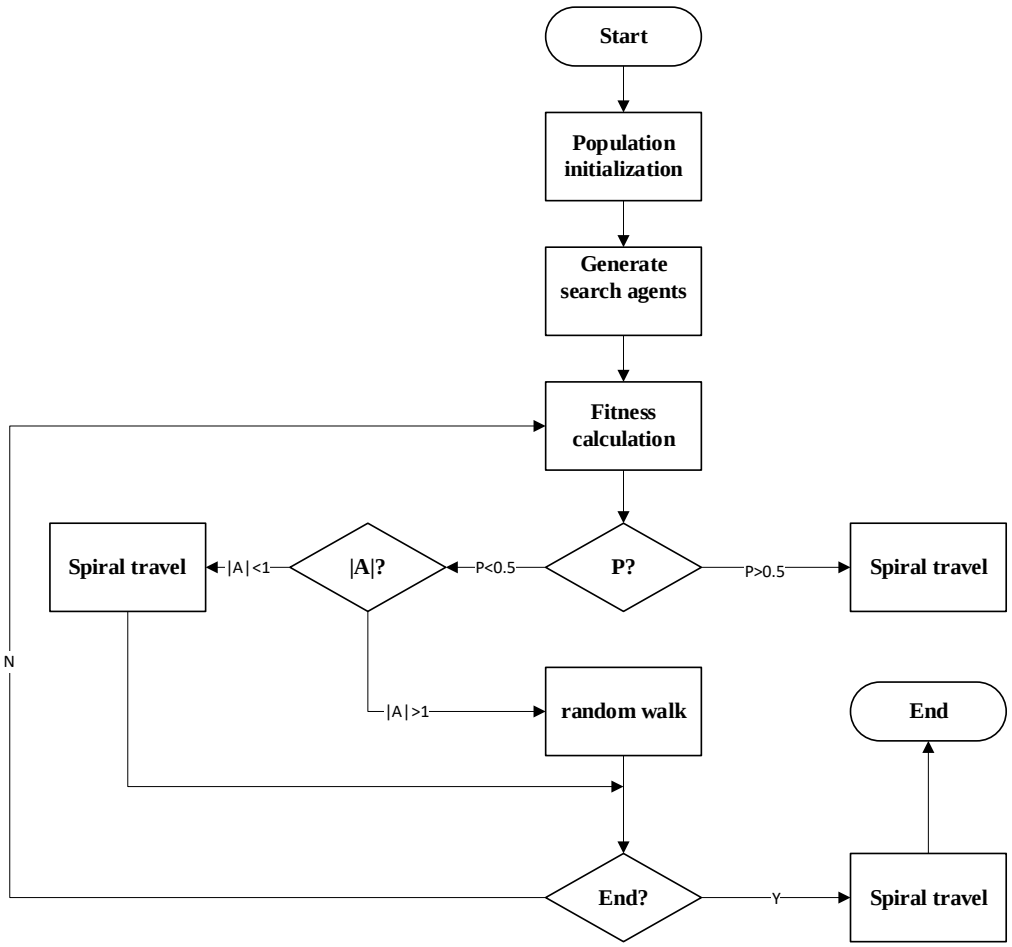


Fig 3. Flow chart of WOA (Photo/Picture credit: Original)

3)Quick Particle Swarm Optimization (QPSO)

Space Vector Modulation (SVPWM) was used by Souza, D.A., to control the joint velocity of a robot manipulator. A Quick Particle Swarm Optimization technique was suggested as a way to improve the PSO algorithm. The optimization procedure was verified by computer simulations, and the digital signal processor was used to implement the optimized parameters in an actual system. The effectiveness of the suggested strategy was demonstrated in experiments, and it was contrasted with existing tuning techniques for PI and PID controllers, such as Genetic Algorithm (GA), PSO, and Differential Evolution (DE). The simulations also showed that Quick Particle Swarm Optimization (QPSO) had the best performance, the most reliable robustness analysis, and the least expensive computing cost [14]. The design of the QPSO is depicted in Figure 4.

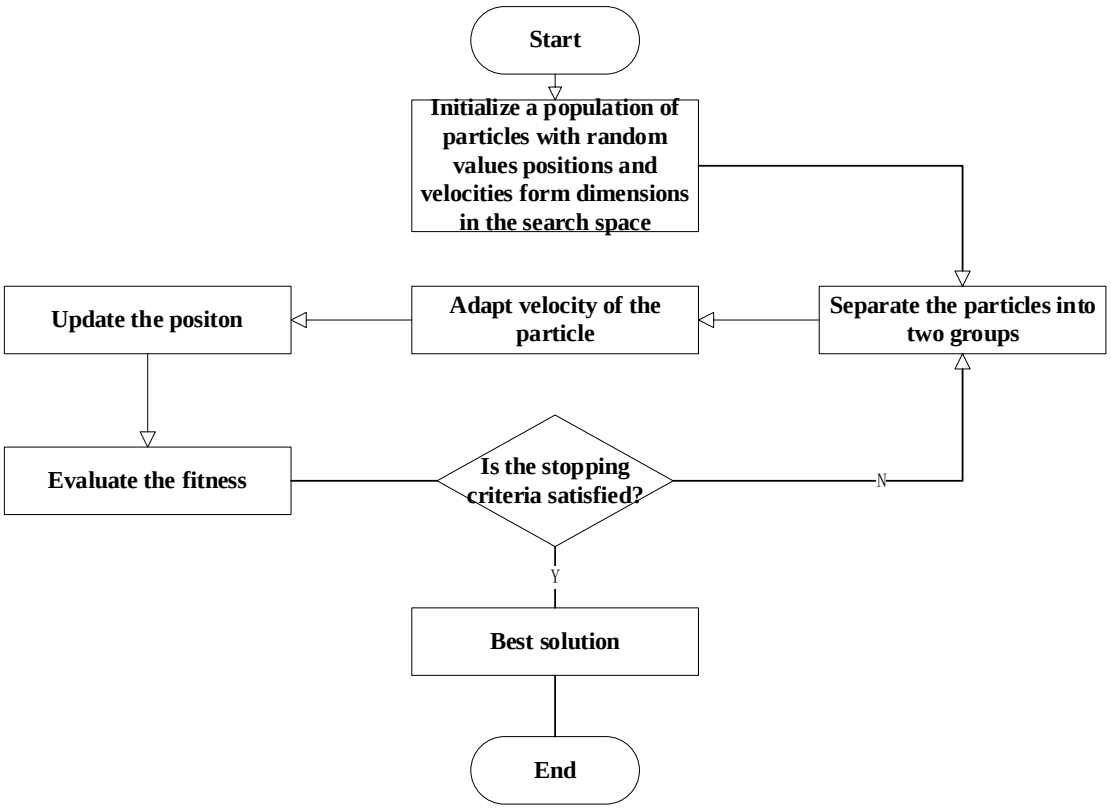


Fig 4. The flow chat of QPSO (Photo/Picture credit: Original)

4. The challenges and potential of PID in robotic arms

4.1. The challenges

Robotic arms often exhibit complex nonlinear dynamics due to factors such as friction, inertia, and varying payloads. Traditional PID control assumes linearity, which may lead to suboptimal performance.

Although PID optimization by metaheuristic algorithm has been able to better deal with the problem of robotic arm, the reasons behind their operation are only partially understood. Despite some interesting earlier attempts, it is unfortunately challenging to mathematically explain why these algorithms are so successful. Similar to this, there is no acknowledged standard for measuring how well different algorithms perform [5].

Metaheuristic algorithms are often developed to address specific optimization problems. Applying these algorithms to different PID control problems or varying system dynamics may require adaptation and customization. Generalizing the algorithm's performance across diverse control scenarios and ensuring adaptability to different control objectives can be challenging and may require careful algorithm design and parameter tuning.

4.2. The potentials

1)Optimal Parameter Tuning: Metaheuristic algorithms can automatically tune the PID controller's parameters to achieve optimal performance. Traditional methods of manual tuning or heuristic-based approaches may be time-consuming and inefficient. Metaheuristic algorithmscan efficiently search the parameter space and find near-optimal or optimal PID gains that result in improved control performance.

2)Robustness to System Variations: PID controllers are often sensitive to changes in the controlled system, such as parameter variations or disturbances. Metaheuristic algorithms can enhance the robustness of PID control by adapting the controller's parameters in real-time. By continuously

searching for the best parameter values, these algorithms can compensate for system variations and maintain stable and accurate control.

3)Nonlinear Control: Many real-world control problems involve nonlinear dynamics that cannot be adequately handled by traditional linear PID control. Metaheuristic algorithms can assist in designing nonlinear PID controllers that are capable of handling complex system dynamics. By incorporating nonlinear transformations or adaptive mechanisms into the PID control structure, metaheuristic algorithms enable effective control in nonlinear systems.

4)Intelligent Control Strategies: Metaheuristic algorithms can be utilized to develop intelligent control strategies beyond traditional PID control. For example, by combining PID control with fuzzy logic or neural networks, metaheuristic algorithms can optimize the structure and parameters of intelligent controllers. This allows for more adaptive and intelligent control actions, leading to improved system performance.

5)Real-Time Control Adaptation: Metaheuristic algorithms can enable real-time adaptation of PID control parameters based on changing system conditions. By continuously evaluating the system's performance and dynamically adjusting the PID gains, these algorithms can optimize control performance in dynamic environments.

5. Conclusion

This research focuses on exploring various tuning methods for the PID control system in robot manipulators. In the contemporary landscape of robotic arm control, the traditional approach of manually adjusting PID parameters has become less prevalent. Instead, researchers have turned to the utilization of metaheuristic algorithms to optimize the PID controller and address the manipulator's inherent issues such as ACO, WOA, QPSO. Despite there are some challenges, the utilization of metaheuristic algorithms in addressing manipulator problems offers notable advantages. Such advantages include the ability to search a vast solution to improve PID parameters avoiding local optima, and handle non-linear and complex systems effectively, and they generally have good robustness. Moreover, ongoing advancements in metaheuristic algorithms continue to pave the way for even more promising methods in the future. With further research and development, these algorithms have the potential to revolutionize robot manipulator control and significantly enhance performance and productivity.

References

[1] Yuval Zaidel, Albert Shalumov, Alex Volinski, Lazar Supic and Elishai Ezra Tsur1. Neuromorphic NEF-Based Inverse Kinematics and PID Control. Frontiers in Neurorobotics, 2021, 15.

[2] Zhou Yi. A Summary of PID Control Algorithms Based on AI-Enabled Embedded Systems. Security and Communication Networks, 2022.

[3] Wang Yong. Research on Nonlinear PID Control. Nanjing University of Science and Technology, 2000.

[4] Ziegler and Nichols. Optimum settings for automatic controllers. Transactions of the ASME, 1995, 42(6): 94-100.

[5] Stephen Bassi Joseph,Emmanuel Gbenga Dada Afeez Abidemi, David Opeoluwa Oyewola and Ban Mohammed Khammas. Metaheuristic algorithms for PID controller parameters tuning: review, approaches and open problems. HELIYON, 2022, 8(5).

[6] James Bennett, Ajay Bhasin, Jamila Grant and Wen Chung Lim. PID Tuning via Classical Methods. university of Michigan.

[7] Marco Dorigo, Vittorio Maniezzo, Alberto Colorni. Ant system: optimization by a colony of cooperating agents. in IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 1996, 26(1): 29-41.

[8] Gaganpreet Kaur and Sankalap Arora. Chaotic whale optimization algorithm. Journal of Computational Design and Engineering, 2018, 5.

[9] James Kennedy and Russell C. Eberhart. Particle swarm optimization. Proceedings of ICNN'95 - International Conference on Neural Networks, 1995, 4: 1942-1948.

[10] Fabian Viola. Simulated annealing simulated. Computers & Mathematics with Applications, 1997, 33(2): 81-94.

[11] Vcerklr Reis, Maris Santos, Mars Carmo and Flastab Ferreira. Control of level systems by arduino via PC platform. 20th International Conference on System Theory, Control and Computing (ICSTCC), 2016, 251-256.

[12] Diego Sandoval and Ismael Soto and Pablo Adasme. Control of direct current motor using Ant Colony optimization. 2015 Chilean Conference on Electrical, Electronics Engineering, Information and Communication Technologies, 2015, 79-82.

[13] Loucif Fatiha, Kechida Sihem and Sebbagh Abdennour. Whale optimizer algorithm to tune PID controller for the trajectory tracking control of robot manipulator, 2019, 42(1).

[14] Souza Darielson, Batista Josias, Nogueira Dos Reis, Laurinda Lucia, Junior Antonio Beira de Andrade, et al. PID controller with novel PSO applied to a joint of a robotic manipulator. Journal of the Brazilian society of mechanical sciences and engineering, 2021, 43(8).