

4-4-2026

Improving the Tracking Performance of Par-4 Robot Using Nature-Inspired Optimization Algorithms to Tune the PID Controller

Shaymaa M. Mahdi

College of Control and Systems Engineering, University of Technology-Iraq, Baghdad, Iraq,
shaymaa.m.mahdi@uotechnology.edu.iq

Ahmed I. Abdulkareem

College of Artificial Intelligence Engineering, University of Technology-Iraq, Baghdad, Iraq,
60162@uotechnology.edu.iq

Amjad J. Humaidi

College of Artificial Intelligence Engineering, University of Technology-Iraq, Baghdad, Iraq,
amjad.j.humaidi@uotechnology.edu.iq

Follow this and additional works at: <https://ijccce.researchcommons.org/journal>

How to Cite This Article

Mahdi, Shaymaa M.; Abdulkareem, Ahmed I.; and Humaidi, Amjad J. (2026) "Improving the Tracking Performance of Par-4 Robot Using Nature-Inspired Optimization Algorithms to Tune the PID Controller," *Iraqi Journal of Computers, Communications, Control and Systems Engineering*: Vol. 26: Iss. 1, Article 8. Available at: <https://ijccce.researchcommons.org/journal/vol26/iss1/8>

This Article is brought to you for free and open access by Iraqi Journal of Computers, Communications, Control and Systems Engineering. It has been accepted for inclusion in Iraqi Journal of Computers, Communications, Control and Systems Engineering by an authorized editor of Iraqi Journal of Computers, Communications, Control and Systems Engineering.



RESEARCH ARTICLE

Improving the Tracking Performance of Par-4 Robot Using Nature-Inspired Optimization Algorithms to Tune the PID Controller

Shaymaa M. Mahdi ^{a,*}, Ahmed I. Abdulkareem ^b, Amjad J. Humaidi ^b

^a College of Control and Systems Engineering, University of Technology-Iraq, Baghdad, Iraq

^b College of Artificial Intelligence Engineering, University of Technology-Iraq, Baghdad, Iraq

ABSTRACT

The Par-4 Delta parallel robot is a powerful candidate in most pick-and-place applications that require high speed, high acceleration, and high precision. Improving the tracking accuracy of parallel robots primarily relies on control design, which is essential for achieving enhanced performance and precision. The PID controller includes proportional, integral, and derivative values. On the other hand, the process of optimum tuning for a PID controller is complex and time-consuming until optimization approaches are employed. The Particle Swarm Optimization (PSO), Flower Pollination Algorithm (FPA), and Whale Optimization Algorithm (WOA) algorithms are the three Nature-Inspired Optimization algorithms proposed in this research for tuning the controller settings. To achieve the global minimum of tracking errors, the purpose of these approaches is to optimize the tuning of the PID controller's parameters toward the ideal level. The Root Mean Square Error (RMSE) indicates that the Whale Optimisation Algorithm (WOA) outperforms other methods in terms of tracking accuracy. WOA outperforms the classical PID controller by 58.5% in the x-axis and 85% in the z-axis. It also improves tracking by 33.9% (x-axis) and 69.3 % (z-axis) over the PSO-PID controller, and by 33.9% (x-axis) and 77.2% (z-axis) compared to the FPA-PID controller.

Keywords: Par-4 Delta Parallel Robot, PID controller, PSO algorithm, FPA algorithm, WOA algorithm

Highlights

1. This study proposed a PID control design for each channel of the Par-4 delta parallel robot.
2. The study developed three algorithms based on PSO, FPA and WOA to tune the terms of PID controllers.
3. A comparative study has been conducted among the optimized PID controllers for the Par-4 Delta parallel robot in terms of tracking accuracy.

Received 17 July 2025; revised 14 August 2025; accepted 16 September 2025.

Available online 4 April 2026

* Corresponding author.

E-mail addresses: shaymaa.m.mahdi@uotechnology.edu.iq (S. M. Mahdi), 60162@uotechnology.edu.iq (A. I. Abdulkareem), amjad.j.humaidi@uotechnology.edu.iq (A. J. Humaidi).

<https://doi.org/xx.xxxxx/2617-3352.1520>

1. Introduction

The parallel robot manipulators have greater advantages than series manipulators due to their stiffness, accuracy, high speed, high acceleration and they could carry more weight. However, the parallel robots are limited in works space and mobility [1, 2]. One of the parallel robot types is the delta robot, which was designed by Reymond Clavel and his scientific group in the beginning of the eighties. In the past few years, high-speed pick-and-place parallel robots, like the Delta parallel robot, have been used in many applications such as electronics, food, pharmaceuticals, packing, and other light industries [3].

Researchers have explored several control methods for the Delta robot's trajectory tracking, including adaptive control, sliding mode control, and model predictive control. Due to the complex dynamics and interactions of parallel robots, developing an effective strategy is challenging [3-5]. Many schemes aim to enhance tracking accuracy and reduce errors. Below, we provide a brief description of these methods.

In [6], D. Zhu et al. have proposed adaptive backstepping fractional-order non-singular terminal sliding mode control (ABF-NTSMC) to address the effects of external disturbances and inaccuracies in the robot's modelling, which are lumped and updated by the developed adaptive law. In [7], F. Serrano and M. Cardona present a robust H-infinity control design for trajectory control of the Delta robot in the presence of disturbances. The proposed control methodology has developed a control law based on an appropriate Lyapunov functional to enhance tracking accuracy. In [8], T. Yigit and H. Celik applied particle and moth swarm optimizations to tune the parameters of the FOPID controller optimally. The results obtained with the optimised POPID controller are compared to those obtained with the optimised integer PID controller, using the same optimisation methods. The study demonstrated that the optimal FOPID controller provides superior tracking performance compared to the integer version. In [1], M.A. Laribi et al. have applied the Genetic Algorithm approach to avoid workspace limitations in the DELTA robot. In [9], M. Rachedi et al. presented a centralised H_∞ multi-variable controller for trajectory tracking of a delta robot based on sensitivity and complementary function matrices. As compared to the traditional PID controller, the proposed H_∞ controller showed 60% improvement in terms of steady-state error.

In [10], X. Lu and M. Liu improved a PID fuzzy logic controller to control a Delta parallel robot. They addressed an optimal design problem using the Non-dominated Sorting Genetic Algorithm (NSGA-II). Their work demonstrated an effective method for designing and adjusting the PID controller to achieve better performance. In [11], X. G. Lu and colleagues used a type-1 fuzzy logic controller (T1FLC) to manage the Delta robot's trajectory. They tuned the membership functions and then used the best version to create fuzzy sets for another controller, which improved trajectory tracking despite uncertainties. In [12], L. Angel and J. Viola compared fractional and integer-order PID controllers using a simplified model of a delta robot. They found that the fractional PID controller performed better in accuracy, torque applied, and resistance to disturbances. In [13], A. J. Humaidi et al. developed two controllers for a Delta/Par4-like parallel robot: an augmented PD (APD) controller and an augmented nonlinear PD (ANPD) controller. In [14] A. J. Humaidi et al., they enhanced performance by optimizing controller settings using particle swarm optimization (PSO). In [15], J. B. Villamizar and L. A. Silva introduced an FOPID controller for tracking a parallel robotic manipulator. In [16], L. Wu and others proposed a control design that adapts robustly using a fuzzy model for the Delta robot while accounting for changes, disturbances, and nonlinear friction. In [17], Yong-Ju Pak et al. optimized a robust PID controller for a Delta parallel robot using a combined strategy of PSO and differential evolution (DE). In [18], T.-L. Bui et al. utilised Hedge Algebras (HA) for

trajectory tracking of the Delta robot, demonstrating that it outperformed Fuzzy Logic and PID controllers. In [19], A. Zhao et al. presented a control-affine neural network model with sliding mode control for the Delta robot using stepper motors.

Recently, PID controller has demonstrated enhancements in stability and system performance despite the presence of model uncertainties and external disturbances after tuning its parameters with one of the optimization methods. As the present research studies the feasibility of a PID controller for a Delta parallel robot Par-4 via three optimization methods such as Particle Swarm Optimization (PSO), Flower Pollination Algorithm (FPA), and Whale Optimization Algorithm (WOA), to tune the primary three PID controller parameters, K_P , K_D , and K_I [20–32, 39].

The contributions of this study can be summarized as follows:

- Proposing PID control for each channel to manage the dynamics of the Par-4 Delta parallel robot.
- Developing and designing algorithms to tune the parameters of the PID controller for the Delta robot based on five optimization techniques: Particle Swarm Optimization (PSO), Flower Pollination Algorithm (FPA), and Whale Optimization Algorithm (WOA).
- Conducting a comparative study on tracking performance among optimal PID controllers, which have been optimized using the proposed algorithms.

The present research contains the dynamic equation of delta parallel robot Par-4 in Section 2, then the PID controller simplified details in Section 3. Section 4 introduce the optimization methods with each type algorithm. Section 5 shows the simulated results and finally conclusions are in 6.

2. Parallel robot structure and dynamics

The Delta/Par4-like robot is a parallel manipulator known for its speed and agility. It has lightweight, strong parts and consists of four actuators that enable movement in three directions: up and down (z-axis), left and right (x-axis), and forward and backward (y-axis). Fig. 1 illustrates the three main components of the Par-4 robot: Eq. (1) the fixed base, which includes the reducer, Eq. (2) servo motor, Eq. (3) mounting frame, and Eq. (4) robot arm; Eq. (5) the movable base with one actuator; and Eqs. (6) and (7) are considered as the spherical hinge connecting the arms to four symmetrically arranged chains [20].

Delta robot's function in both joint and Cartesian spaces, using 3D coordinates (x, y, z) for the end effector and base arm angles for the joint space. This setup allows for parallel movement in tasks like picking, placing, and tracking paths.

There are two types of kinematics and dynamics: forward and inverse. Forward kinematics uses joint angles to find the end effector's Cartesian location, while inverse kinematics finds them. Delta robots require efficient and precise mapping between these regions for precision control, especially in high-speed applications. A challenge with Cartesian control in joint space is adjusting the end effector's orientation quickly. This can be addressed by using dynamic kinematics in Cartesian space and calculating posture through forward kinematics.

The most important disadvantage of Cartesian control with respect to joint space control is the difficulty in obtaining a fast and accurate observation of the end effector pose, which represents a restriction for its utilization. In order to solve this issue, it is possible to do the control in Cartesian space by using the forward kinematics to compute the end-effector pose. However, it is shown in that the utilization of the measured end-effector pose increases the tracking performance of the controller [13].

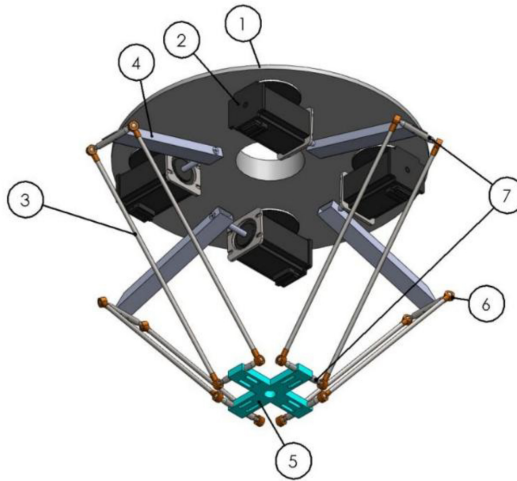


Fig. 1. The delta parallel robot par-4 schematic diagram.

First, we provide the Delta/Par4-like robot arm equilibrium analysis. We connect the acceleration vector q to the actuator torque vector τ_q using the following equation:

$$\tau_q - J_q^T F = I_{tot} \ddot{q} \quad (1)$$

where J_q^T denotes the joint Jacobian of the robot, F represents the forces vector of exerted on robot arm, and I_{tot} is a diagonal matrix.

Second, the moving plate's balance is looked at. We can write the motion equation for the moving plate as [20]:

$$J_x^T F = M_{tot} \ddot{X} \quad (2)$$

where J_x^T is Cartesian Jacobian of the robot, M_{tot} is a diagonal matrix, and the position parameter $X = [x \ y \ z]$. The dynamic model of Delta/Par4 like robot will be written in Cartesian space as [20]:

$$M_{tot} \ddot{X} = J_x^T J_q^{-T} (\tau_q - I_{tot} \ddot{q}) = J_m^T (\tau_q - I_{tot} \ddot{q}) \quad (3)$$

And it can be rewritten as the dynamic control schemes are based on the Lagrange formulation of the inverse dynamic model of a robot, which is given in the joint space by in the following equation [20, 30]:

$$M \ddot{X} + C \dot{X} = J_m^T \tau_q = F \quad (4)$$

Where the inertial matrix is $M = M_{tot} + J_m^T I_{tot} J_m$, and the centrifugal and Coriolis forces as $C = J_m^T I_{tot} \dot{J}_m$.

3. Controller design

The Proportional, Integral, and Derivative (PID) controller is widely used in industries but may struggle with nonlinear systems found in manufacturing. PID controllers often trade off between performance and robustness, making it hard to achieve both while

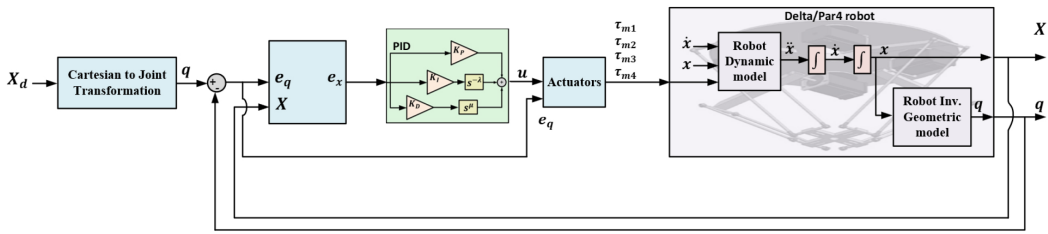


Fig. 2. Delta/par-4 parallel robot close-loop system.

maintaining stability. They can fail to manage unpredictable systems and may not provide effective disturbance rejection [25]. Despite these limitations, PID controllers are popular due to their simplicity and good performance, with many resources available on their operation and design. The PID controller is expressed as:

$$G_{IOPID}(s) = u(s) / e(s) = K_p + K_I s^{-1} + K_D s \tag{5}$$

where $u(s)$ symbolizes the control action, $e(s)$ symbolizes the error and K_p symbolizes the proportional gain, K_I symbolizes the integral gain, and K_D symbolizes the derivative gain.

According to control law of Eq. (5), the Par4-Delta parallel robot dynamic model can be described by Eq. (6):

$$M \ddot{X} + C \dot{X} = M \ddot{X}^d + C \dot{X}^d + K_p e + K_I s^{-1} e + K_D s e \tag{6}$$

Eq. (7) can be rewritten as:

$$M (\ddot{e}) + C (\dot{e}) + K_p e + K_I s^{-1} e + K_D s e = 0 \tag{7}$$

where \ddot{X}^d and \dot{X}^d are the desired travelling plate acceleration and velocity, correspondingly, and $e = [e_x \ e_y \ e_z]^T = X^d - X$ represents the position errors in the three channels, as shown in Fig. 2.

4. Optimizations with meta-heuristic

Metaheuristics are more complex processes or heuristics that try to locate, invent, or pick a partial search strategy that may correctly solve an optimization issue. Usually, we use them when we lack information or computational power. Meta algorithms may be employed for numerous optimization issues since they don't make many conclusions [9]. Due to their simplicity and ease of use, meta-heuristic optimization methods are becoming increasingly prominent in engineering. Additional reasons for their popularity in engineering include their ability to avoid local optima and their versatility across various domains [17].

The PID controller has to adjust its three parameters to work well. Instead of employing the time-consuming trial-and-error technique, this work formulates the tuning process as an optimization problem, as illustrated in Fig. 3. Then, three meta-heuristic optimizations are used to fix the multi-objective problems. These are the Particle Swarm Optimization (PSO), the Flower Pollination Algorithm (FPA), and the Whale Optimization Algorithm (WOA). The selection of these methods depended on their well-established and effective in solving nonlinear control and engineering optimization problems.

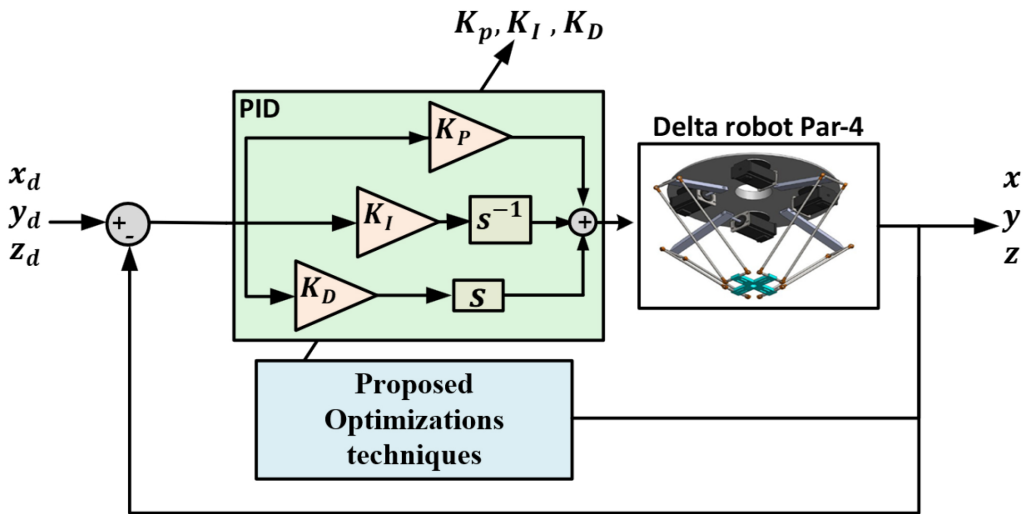


Fig. 3. Tuning pid controller with optimization metaheuristic.

4.1. Particle swarm optimization algorithm PSO

Use either the Microsoft Equation Editor or the *MathType* add-on for all math objects in your paper. When numbering equations, enclose numbers in parentheses and align with the right margin.

PSO population of particles explores space at a speed that changes based on its location without considering mass and volume [21]. Between iterations, swarm particles may update their virtual locations to enhance PSO search performance. In search of the ideal solution, each particle in the swarm goes to its prior best location (p_{best}) and the best position overall (g_{best}) [29–37].

The position of the global best particle is the optimal solution obtained by PSO, where the new velocity v_{t+1} is generated by applying the following equation:

$$v_i^{t+1} = v_i^t + \alpha \epsilon_1 [g^* - x_i^t] + \beta \epsilon_2 [x_i^{*(t)} - x_i^t] \quad (8)$$

Where ϵ_1 and ϵ_2 are two random vectors with values between 0 and 1. α and β are the learning parameters or acceleration constants, which are estimated as $\alpha \approx \beta \approx 2$. Also, determine the new position locations of the position

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (9)$$

The main stages of PSO may be delineated using the pseudocode presented in Algorithm below:

4.2. Flower pollination algorithm FPA

Four concepts may construct the flow pollination process that inspired FPA:

Rule 1: Levy flight patterns govern global pollination by biotic and cross-pollinators.

Rule 2: Self-pollination and abiotic causes cause localization.

Rule 3: Floral constancy links two flowers' relationships to reproduction.

Rule 4: Control pollination globally or locally by choosing an adjustment probability between 0 and 1.

Pseudocode of Particle Swarm Optimization

Objective function $f(x)$, $x = (x_1, \dots, x_n)^T$
 Set initial locations x_i and velocity v_i of n particles.
 Find g^* from $\min\{f(x_1) \dots, f(x_n)\}$ (at $t = 0$)
 while (criterion)
 for the cycle over all n particles and all D dimensions
 Generated new velocity v_{t+1} as in Eq. (8)
 Determine the new position locations of the position x_i^{t+1} as in equation Eq. (9)
 Measure objective functions in novel locations x^{t+1}
 Determining the current optimal value for each particle x_i^{*i}
 end for
 Determine the current global best value of g^*
 Update $t = t + 1$ (pseudo time or iteration counter)
 end while
 The final results x^{*i} and g^* should be output.

Pseudocode of Flower pollination algorithm

Objective \min or $\max f(x)$, $x = (x_1, x_2, \dots, x_d)$
 Initialize a population of n flowers/pollen gametes with random solutions
 Find the best solution g^* is the current best solution found among all solutions at the current generation/iteration.

 Define a switch probability $p \in [0, 1]$
 while ($t < \text{Max Generation}$) where t is the number of iteration
 for $i = 1 : n$ (all n flowers in the population)
 if $\text{rand} < p$, proximity probability p can be effectively used to switch between common global pollination to intensive local pollination.
 Draw a (d-dimensional) step vector L which obeys a L'evy distribution
 Global pollination via Eq. (10)
 Where x_i^t is the pollen i or solution vector x_i at iteration t , and γ is a scaling factor to control the step size

 Else
 Draw ϵ from a uniform distribution in $[0, 1]$
 Do local pollination via x_i^{t+1}
 end if Eq. (11)
 Evaluate new solutions
 If new solutions are better, update them in the population
 end For
 Find the current best solution g^*
 end while
 Output the best solution found

Table 1. Parameters for pid controller gains with three optimization types.

Channel of controller's gains	Controller's gains			
	Classical	PSO Algorithm	FPA Algorithm	WOA Algorithm
K_p	60×10^3	3.0297×10^4	2.8913×10^4	60×10^3
K_i	1×10^{-3}	0.0068	60.3737	44502.45
K_d	1×10^3	802.4862	817.4135	900

Pseudocode of Whale Optimization Algorithm

```

Initialize the whales population  $X_i$  ( $i = 1, 2, \dots, n$ )
Calculate the fitness of each search agent
 $\vec{X}_{best}(t)$  = the best search agent
while (t < maximum number of iterations)
    for each search agent
        Update a,  $\vec{A}$ ,  $\vec{C}$ , l, and p
    if1 ( $p < 0.5$ )
        if 2 ( $|A| < 0.5$ )
            Update the current position by Eq. (12)
        else if 2 ( $|A| \geq 1$ )
            Select a random search agent  $\vec{X}_{rand}$ 
            Update the position of the current search agent by applying Eq. (14)
        end if 2
    else if1 ( $p \geq 0.5$ )
        Update the position of the current search by the (Eq. (16))
    end if 1
end for
Check if any search agent goes beyond the search space and amend it
Calculate the fitness of each search agent
Update  $\vec{X}_{best}$  if there is a better solution
 $t = t + 1$ 
end while
return  $\vec{X}_{best}(t)$ 
    
```

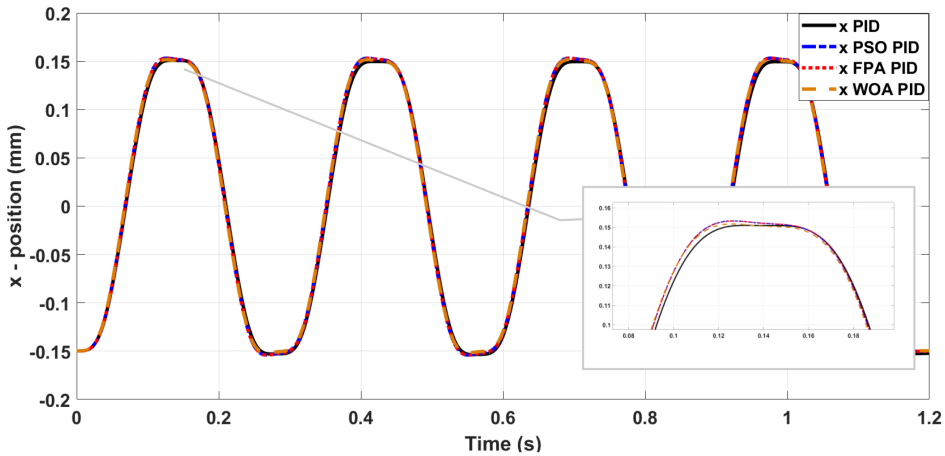


Fig. 4. X – axis position response of controlled 4DOF parallel robot.

Table 2. Evaluation of tracking errors based on both controllers.

Channel of controller's gains	Root Mean Square of Errors (RMSE) (mm)			
	Classical	PSO Algorithm	FPA Algorithm	WOA Algorithm
e_x	5.3×10^{-3}	3.5×10^{-3}	3.5×10^{-3}	2.2×10^{-3}
e_y	0	0	0	0
e_z	10.1×10^{-3}	3.1×10^{-3}	2.3×10^{-3}	1.5×10^{-3}

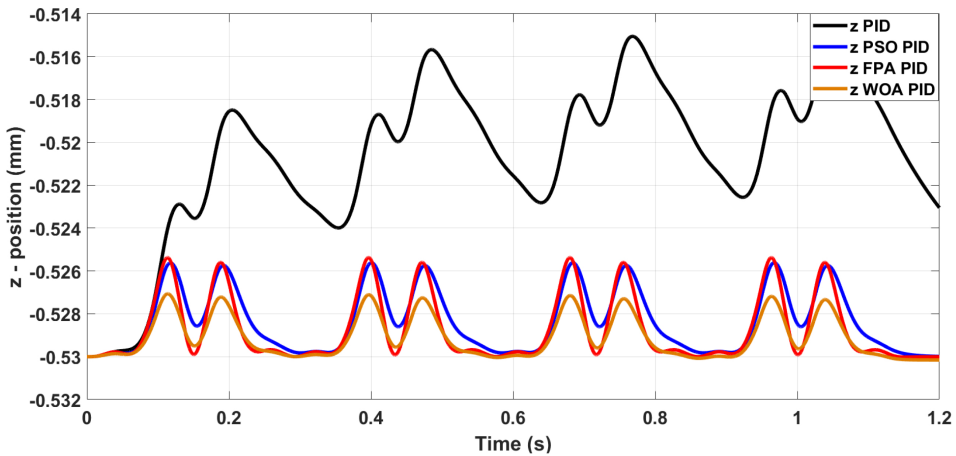


Fig. 5. Z – axis position response of controlled 4DOF parallel robot.

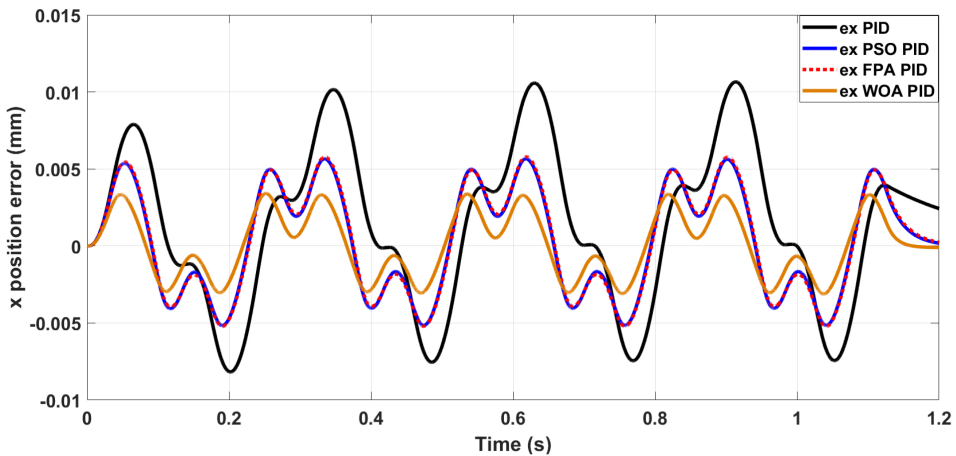


Fig. 6. X – axis position error response of controlled 4DOF parallel robot.

Table 3. Evaluation of actuating torques based on both controllers.

Channel of controller’s gains	Root Mean Square of Torque (RMST) (N.m)			
	Classical	PSO Algorithm	FPA Algorithm	WOA Algorithm
τ_{M1}	28.6199	34.3272	36.2545	35.1488
τ_{M2}	5.5626	7.1734	6.0577	6.1984
τ_{M3}	18.7908	21.4428	19.3486	21.1186
τ_{M4}	5.5626	7.1734	6.0577	6.1984

Flying insects carry flower pollen gametes in the worldwide pollination process [25, 37], where the above principles must be converted into updating equations to produce updating formulas as in the following, for both Rule 1 and Rule 3 associations to give global pollination as [15]:

$$X_i^{t+1} = X_i^t + L_\gamma (X_i^t - g^*) \tag{10}$$

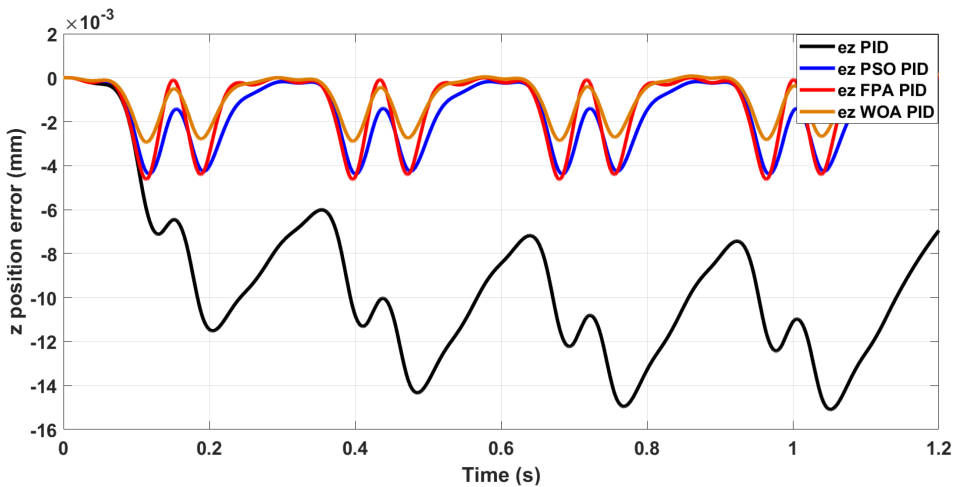


Fig. 7. Z – axis position error response of controlled 4DOF parallel robot.

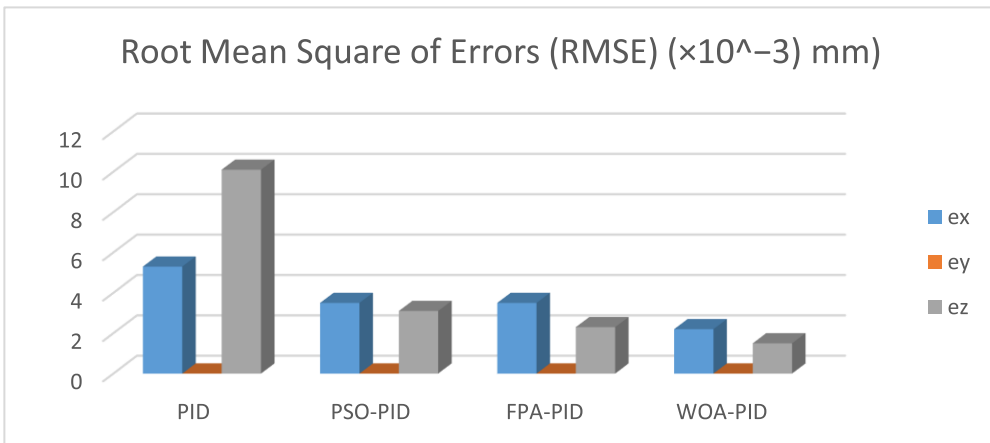


Fig. 8. The root-mean square error (RME) statistical analysis.

where X_i^{t+1} and X_i^t are $t - th$ iteration solutions, L is Levy flight step, is a scaling factor and global best is the present optimal solution, g^* is the global best solution. However, Rule 2 and Rule 3 combine to give local pollination as:

$$X_i^{t+1} = X_i^t + \varepsilon (X_j^t - X_k^t) \tag{11}$$

Where X_j^t and X_k^t are randomly generated numbers and ε is a random number between 0 and 1. Below is the pseudo-code for the FPA:

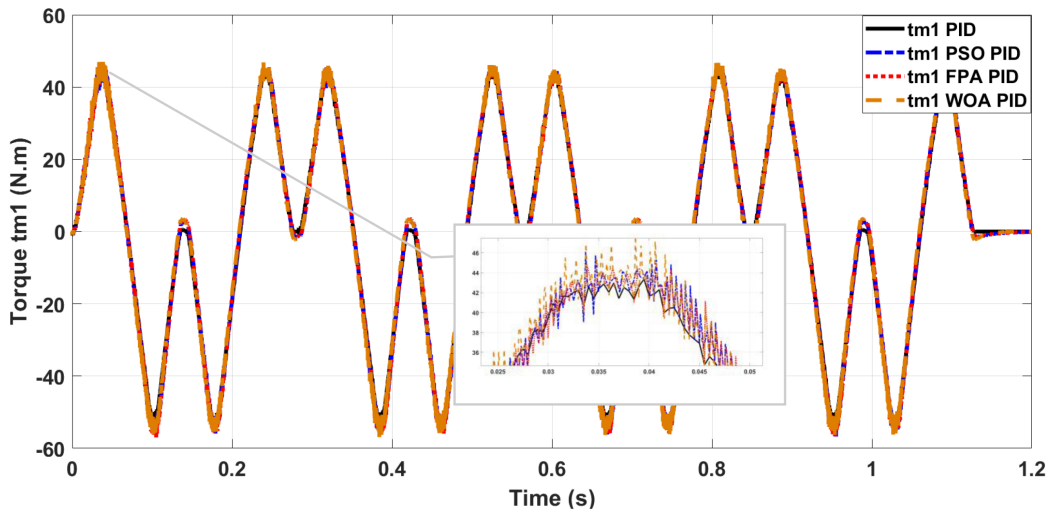
4.3. Whale optimization algorithm WOA

Define abbreviations and acronyms the first time they are used in the text, even after they have already been defined in the abstract.

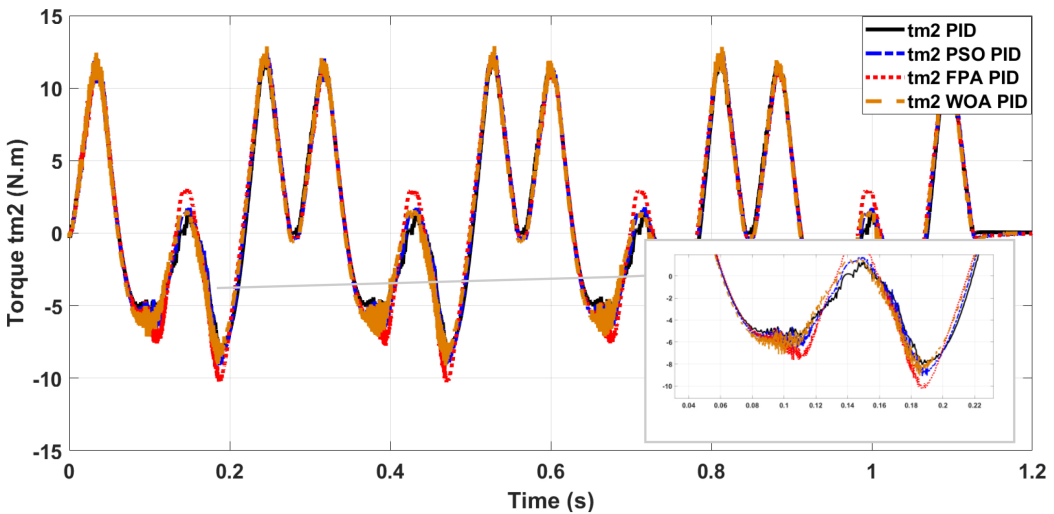
WOA is based on humpback whales’ bubble-net feeding society. Whales find and attack prey two ways. One tactic includes enclosing the prey, while the other uses bubble nets. Whales optimize by searching for prey and exploiting during the attack phase [37–39]. The algorithm simulates surrounding prey, spiral update, and exploration to improve an issue [24, 28]. Other search agents use the following equations to update the current search agent’s location toward the optimal answer [27, 28]:

$$\vec{D} = \left| \vec{C} \cdot \vec{X}^*(t) - \vec{X}(t) \right| \tag{12}$$

$$\vec{X}(t+1) = \vec{X}^*(t) - \vec{A} \cdot \vec{D} \tag{13}$$

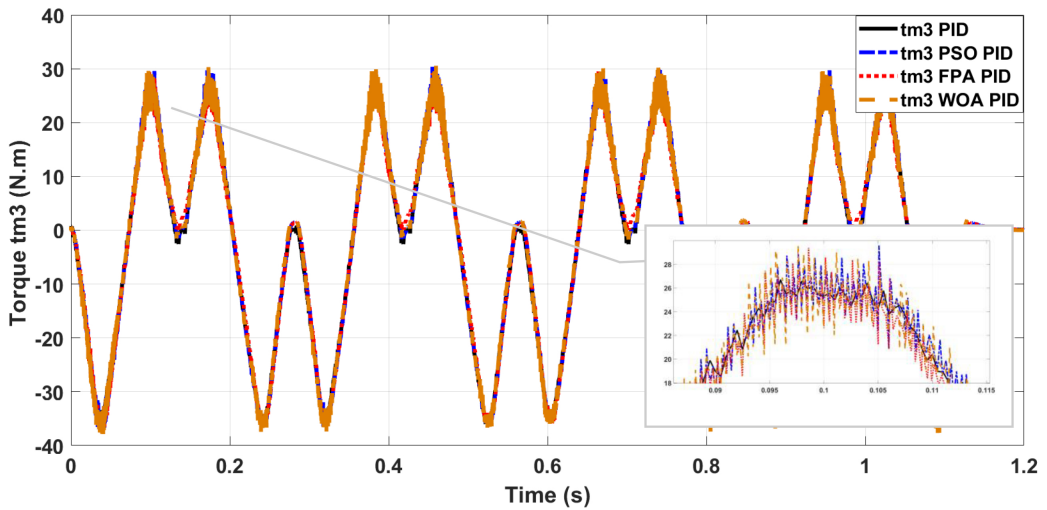


(a)

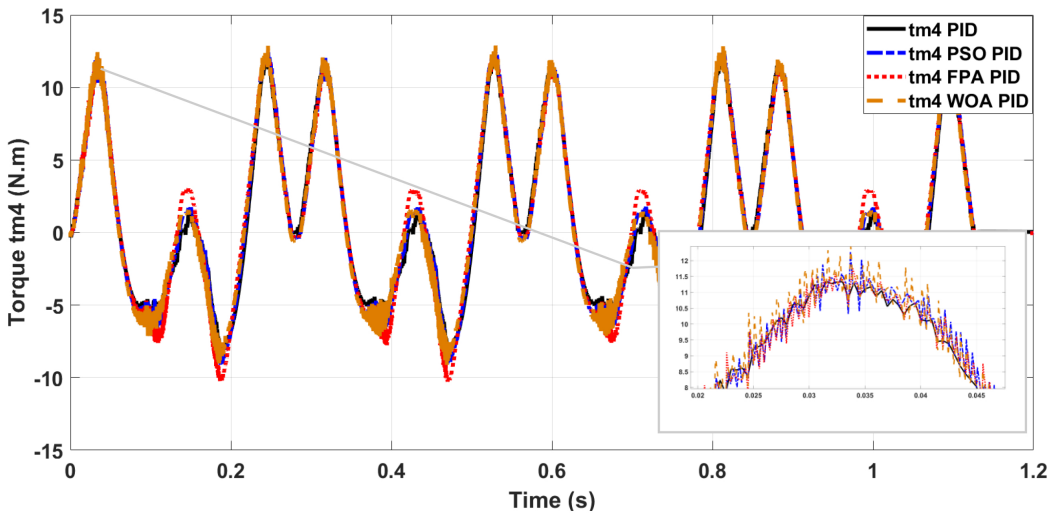


(b)

Fig. 9. Torque response of controlled delta parallel robot par-4: a)for motor 1, b) for motor 2, c) for motor 3, d) for motor 4.



(c)



(d)

Fig. 9. Continued.

Here \vec{A} and \vec{C} are the co-efficient vectors. \vec{X}^* is the position vector of the best solution obtained so far. \vec{X} is the current position vector. So, \vec{X}^* will be updated on each iteration if there exists a better solution.

The vectors \vec{A} and \vec{C} are calculated as follows: $\vec{A} = 2 \vec{a} \cdot \vec{r} - \vec{a}$, $\vec{C} = 2 \cdot \vec{r}$

Where \vec{a} is linearly decreased from 2 to 0 over the course of iterations (in both exploration and exploitation phases) and \vec{r} is a random vector in $[0, 1]$, p indicates a probability factor with random number in $[0, 1]$ based on whales that use two types of Bubble Net Attacking methods: Shrinking Encircling Mechanism (achieved by the decreasing value of \vec{a}) and Spiral Updating Position (the whale moves spirally towards its prey) to update the position. In Spiral Updating Method, distance 'D' between the position of the whale (X, Y)

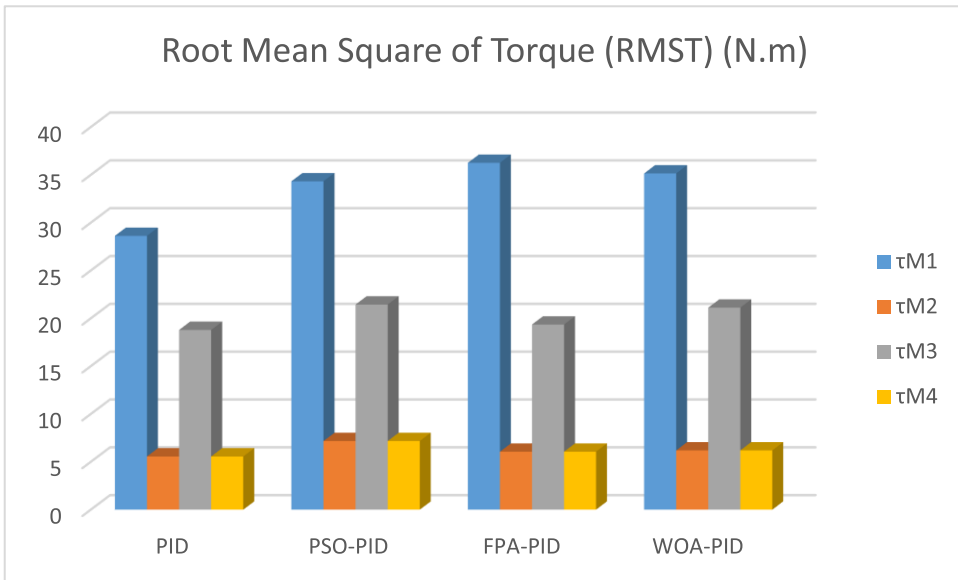


Fig. 10. The root mean square torque (N.M) statistical analysis.

and its prey (X^*, Y^*) is calculated as in [27]:

$$\ddot{\vec{X}}(t+1) = \vec{D} \cdot e^{bt} \cdot \cos(2\pi l) + \vec{X}^*(t) \quad (14)$$

Where b is a constant and l is a random number between $[-1, 1]$ and $D = |\vec{X}^* - \vec{X}|$

Then by tacking X_{rand} is a random search agent from current population, and the position of the current search agent is updated by:

$$D = \left| \vec{C} \vec{X}_{rand}(t) - \vec{X} \right| \quad (15)$$

$$\vec{X}(t+1) = \vec{X}_{rand}(t) - \vec{A} \cdot \vec{D} \quad (16)$$

Below is the pseudo-code for the Whale Optimization Algorithm (WOA):

5. Simulation results

The figures illustrate the position, position error, torque, and control input over time. The paper discusses the performance of a delta robot using a PID controller, with its parameters tuned by three methods: Particle Swarm Optimization (PSO), Flower Pollination Algorithm (FPA), and Whale Optimisation Algorithm (WOA). Each method was tested under the same conditions, using a population size of 30 and 100 iterations. WOA outperformed the others in convergence, accuracy, and computation time, making it the preferred method for final implementation. Table 1 lists the parameters for each optimization method.

In Figs. 4 and 5 reveal the time-varying position tracking for the delta parallel robot Par-4 in both the x-axis and z-axis, respectively, starting by the classical PID controller implementation then tuning the PID controller three parameters using PSO firstly, FPA, and finally WOA methods, where the results show significant improvement by applying WOA as compared to the other showed results, by producing a more stable response with

less overshoot and steady-state error. This was noticeable after calculating the root mean square (RMSE) error result as shown in [Table 2](#) and [Figs. 6 and 7](#), also a statistical analysis for the root mean square (RMSE) error shown in [Fig. 8](#).

The WOA-PID controller effectively minimises undesirable variations and shows less pronounced oscillations compared to other PID tuning methods, indicating greater stability in reaching the desired position. As shown in [Fig. 9](#), the torque over time for all four motors with the WOA-PID controller displays consistent behaviour with fewer peaks and troughs. A comparative study indicates that the WOA-PID controller delivers the best response, enhancing system reliability by reducing unwanted mechanical effects. [Table 3](#) presents the root mean square of the torque for different controller types, and [Fig. 10](#) illustrates this evaluation through histograms.

The implemented results and figures show that the WOA-PID controller system response improves the classical PID controller system response also the tuned PID controller by both PSO and FPA optimization methods, when it comes to decreasing position error and increasing the system's stability.

6. Conclusions

The delta parallel robot Par-4 is considered a nonlinear system, which makes its control challenging. Applying a PID controller requires manually tuning its three parameters as a starting point and implementing three optimization methods, such as the PSO, FPA, and WOA optimization methods. The robot's results demonstrate the efficacy of the WOA-PID controller in increasing system performance for precise trajectory tracking in Par-4 parallel robot control tasks. The PID controller has an improvement in the x-axis of 33.9 % after tuning its parameters with both the PSO and the FPA algorithms, of 58.5 % after tuning its parameters with the WOA algorithm. In comparison, the PID has an improvement in the z-axis of 69.3 % after tuning its parameters with the PSO algorithm, 77.2 % after tuning its parameters with the FPA algorithms, and of 85 % after tuning its parameters with the WOA algorithm, where the results showed oscillation reduction and more instantaneous error convergence, proposing that WOA-PID works excellently in high-accuracy applications where position error minimization is required. In order to extend this study for future work, one can compare the optimized controller for the suggested parallel robot to other control techniques used in the literature. Intelligent and nonlinear controllers may be used for such comparison to show the effectiveness of proposed controller compared to others.

Acknowledgment

There is no need for an acknowledgement in this steady.

Conflict of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that support the findings of this study are available on request from the corresponding author.

References

- [1.] M. A. Laribi, L. Romdhane, and S. Zeghloul, "Advanced Synthesis of the DELTA Parallel Robot for a Specified Workspace," *Parallel Manipulators, Towards New Applications*, Book edited by: Huapeng Wu, ISBN 978-3-902613-40-0, pp. 506, April 2008, I-Tech Education and Publishing, Vienna, Austria.
- [2.] D. Zhang, *Global Stiffness Optimization of Parallel Robots Using Kinetostatic Performance Indices*, chapter of book *Robot Manipulators Trends and Development* Edited by Agustin Jimenez and Basil M Al Hadithi, InTech Europe, 2010.
- [3.] Y. J. Pak, Y. S. Kong, and J. S. Ri, "Robust PID optimal tuning of a Delta parallel robot based on a hybrid optimization algorithm of particle swarm optimization and differential evolution," *Robotica*, 41, pp. 1159–1178, 2023, <https://doi.org/10.1017/S0263574722001606>.
- [4.] A. Al-Mufti, M. Z. Khalifa, and A. O. Hanfeesh, "Parallel Robot Manufacturing for Upper and Lower Limb Medical Rehabilitation," *Iraqi Journal of Computers, Communications, Control and Systems Engineering*, vol. 25, no. 1, pp. 1–15, April 2025, <https://doi.org/10.33103/uot.ijccce.25.1.1>.
- [5.] H. M. Alwan and H. S. Ahmed, "Singularity Analysis of Parallel Robot with Six Degrees-Of-Freedom of Six Legs," *Eng. & Tech. Journal*, vol. 32, Part (A), no. 4, pp. 960-972, 2014, <https://doi.org/10.30684/etj.32.4A.12>.
- [6.] D. Zhu, Y. He, and F. Li, "Trajectory Tracking of Delta Parallel Robot via Adaptive Backstepping Fractional-Order Non-Singular Sliding Mode Control," *Mathematics*, vol. 12, no. 14, pp. 2236, 2024.
- [7.] F. E. Serrano and M. Cardona, "Robust H-Infinity Control of Delta Parallel Robot with Disturbances," In book: *Advances in Automation and Robotics Research*, pp. 56–67, March 2024.
- [8.] T. Yigit and H. Celik, "Dimension Synthesis and Optimized FOPID Control of the Delta Robot with Moth Swarm Algorithm," *Arabian Journal for Science and Engineering*, vol. 48, pp. 6889–6902, 2023.
- [9.] M. Rachedi, H. boualem, and M. Bouri, "Design of an H_∞ controller for the Delta robot: Experimental results," *Advanced Robotics*, vol. 29, no. 18, April 2015.
- [10.] X.-G. Lu, and M. Liu, "Optimal Design and Tuning of PID-type Interval Type-2 Fuzzy Logic Controllers for Delta Parallel Robots," *International Journal of Advanced Robotic Systems*, vol. 13, no. 96, 2016, doi: [10.5772/63941](https://doi.org/10.5772/63941).
- [11.] X.-G. Lu, M. Liu, and J.-X. Liu, "Design and Optimization of Interval Type-2 Fuzzy Logic Controller for Delta Parallel Robot Trajectory Control," *International Journal of Fuzzy Systems*, Springer, vol. 19, pp. 190–206, 2016.
- [12.] L. Angel and J. Viola, "Fractional order PID for tracking control of a parallel robotic manipulator type delta," *ISA Transactions*, 2018.
- [13.] A. J. Humaidi and A. I. Abdulkareem, "Design of Augmented Nonlinear PD Controller of Delta/Par4-Like Robot," *Hindawi, Journal of Control Science and Engineering*, vol. 2019, doi: [10.1155/2019/7689673](https://doi.org/10.1155/2019/7689673).
- [14.] A. J. Humaidi, A. A. Oglah, S. J. Abbas, and I. K. Ibraheem, "Optimal Augmented Linear and Nonlinear PD Control Design for Parallel Robot Based on PSO Tuner," *International Review on Modelling and Simulations*, vol. 12, no. 5, October 2019.
- [15.] J. B. V. Villamizar and L. A. Silva, "Delta Parallel Robotic Manipulator Tracking Control using Fractional Order Controllers," *IEEE Latin America Transactions, Engineering, Computer Science*, 2019.
- [16.] L. Wu, R. Zhao, Y. Li, and Y.-H. Chen, "Optimal Design of Adaptive Robust Control for the Delta Robot with Uncertainty: Fuzzy Set-Based Approach," *Applied Sciences*, vol. 10, no. 10, 2020. <https://doi.org/10.3390/app10103472>
- [17.] Y.-J. Pak, Y.-S. Kong, and J.-S. Ri, "Robust PID optimal tuning of a Delta parallel robot based on a hybrid optimization algorithm of particle swarm optimization and differential evolution," *Robotica*, vol. 41, pp. 1159–1178, 2023.
- [18.] T.-L. Bui, T.-H. Nguyen, and X.-T. Nguyen, "A Controller for Delta Parallel Robot Based on Hedge Algebras Method," *Hindawi, Journal of Robotics*, vol. 2023, <http://doi.org/10.1155/2023/2271030>.
- [19.] A. Zhao, A. Toudeshki, R. Ehsani, J. H. Viers, and J.-Q. Sun, "Evaluation of Neural Network Effectiveness on Sliding Mode Control of Delta Robot for Trajectory Tracking," *Algorithms*, vol. 17, no. 113, 2024.
- [20.] X. Zhang and Z. Ming, "Trajectory Planning and Optimization for a Par4 Parallel Robot Based on Energy Consumption," *Applied Sciences*, vol. 9, no. 2770, 2019.
- [21.] Z. Qin and D. Pan, "Improved Dual-Center Particle Swarm Optimization Algorithm," *Mathematics*, vol. 12, no. 1698, 2024, <https://doi.org/10.3390/math12111698>.
- [22.] H. Al-Khazraji, "Optimal Design of a Proportional-Derivative State Feedback Controller Based on Meta Heuristic Optimization for a Quarter Car Suspension System," *Mathematical Modelling of Engineering Problems*, vol. 9, no. 2, pp. 437–442, April 2022, <http://iijeta.org/journals/mmep>.

- [23.] M. A. Al-Ali, O. F. Lutfy, and H. Al-Khazraj, "Investigation Of Optimal Controllers On Dynamics Performance Of Nonlinear Active Suspension Systems With Actuator Saturation," *Journal of Robotics and Control (JRC)*, vol. 5, no. 4, 2024, DOI: [10.18196/Jrc.V5i4.22139](https://doi.org/10.18196/Jrc.V5i4.22139).
- [24.] N. Rana, M. S. Abd Latiff, S. M. Abdulhamid, and H. Chiroma, "Whale optimization algorithm: a systematic review of contemporary applications, modifications and developments," *Neural Computing and Applications*, Springer Nature, 2020, <https://doi.org/10.1007/s00521-020-04849-z>.
- [25.] X. S. Yang, M. Karamanoglu, and X. S. He, "Flower Pollination Algorithm: A Novel Approach for Multi-objective Optimization, Engineering Optimization," *Mathematics, Optimization and Control*, vol. 46, no. 9, pp. 1222-1237, 2014, <https://doi.org/10.48550/arXiv.1408.5332>.
- [26.] M. Khursheed, M. F. Nadeem, A. Khalil, I. A. Sajjad, A. Raza, M. Q. Iqbal, R. Bo, and W. Rehman, "Review of Flower Pollination Algorithm: Applications and Variants," *2020 International Conference on Engineering and Emerging Technologies (ICEET)*.
- [27.] N. M. Laskar, K. Guha, I. Chatterjee, S. Chanda, K. L. Baishnab, and P. K. Paul, "HWPSO: A new hybrid whale-particle swarm optimization algorithm and its application in electronic design optimization problems," *Applied Intelligence, Springer Science Business Media*, LLC, part of Springer Nature, 2018. <https://doi.org/10.1007/s10489-018-1247-6>.
- [28.] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm. Advances in Engineering Software," *Advances in Engineering Software*, vol. 95, no. C, pp. 51-67, DOI: [10.1016/j.advengsoft.2016.01.008](https://doi.org/10.1016/j.advengsoft.2016.01.008).
- [29.] N. Rana, A. L. M. Shafie, S. M. Abdulhamid, and H. Chiroma, "Whale optimization algorithm: A systematic review of contemporary applications, modifications and developments," *Neural Computing and Applications*, vol. 32, Oct 2020.
- [30.] M. Zamani, M. Karimi-Ghartemani, and N. Sadati, "FOPID Controller Design for Robust Performance Using Particle Swarm Optimization," *An international journal of theory and application*, vol. 10, no. 2, 2007.
- [31.] G. Dhiman and V. Kumar, "Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications," *Advances in Engineering Software journal*, vol. 114, pp. 48-70, 2017. www.elsevier.com/locate/advengsoft Research paper.
- [32.] A. G. Gad, "Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review," *Archives of Computational Methods in Engineering*, vol. 29, pp. 2531-2561, 2022, <https://doi.org/10.1007/s11831-021-09694-4>.
- [33.] L. T. Rasheed, "Optimal Tuning of Linear Quadratic Regulator Controller Using Ant Colony Optimization Algorithm for Position Control of a Permanent Magnet DC Motor," *Iraqi Journal of Computers, Communications, Control and Systems Engineering*, vol. 20, no. 3, pp. 29-41, 2020, DOI: <https://doi.org/10.33103/uot.ijccce.20.3.3>.
- [34.] M. Zihin bin M. Zaina, J. Kanesana, J. H. Chuaha, S. Dhanapalb, and G. Kendall, "A multi-objective particle swarm optimization algorithm based on dynamic boundary search for constrained optimization," *Applied Soft Computing journal*, 2018, www.elsevier.com/locate/asoc.
- [35.] S. Charkoutsis, and M. Kara-Mohamed, "A Particle Swarm Optimization tuned nonlinear PID controller with improved performance and robustness for First Order Plus Time Delay systems," *Results in Control and Optimization*, vol. 12, 2023.
- [36.] Z. N. Mahmood, H. Al-Khazraji, S. M. Mahdi, "PID-Based Enhanced Flower Pollination Algorithm Controller for Drilling Process in a Composite Material," *Annales de Chimie - Science des Matériaux*, vol. 47, no. 2, pp. 91-96, April, 2023.
- [37.] F. Soleimani Gharehchogh, and Gholizadeh, "A Comprehensive Survey: Whale Optimization Algorithm and its Applications," *BASE DATA*, 2019, <https://doi.org/10.1016/j.swevo.2019.03.004>.
- [38.] Z. N. Mahmood, H. Al-Khazraji, and S. M. Mahdi, "Adaptive control and enhanced algorithm for efficient drilling in composite materials," *Journal Européen des Systèmes Automatisés*, vol. 56, no. 3, pp. 507-512, 2023, <https://doi.org/10.18280/jesa.560319>.
- [39.] A. J. Haleel and L. M. Dawood, "Binary Pareto Multi-Objective Whale Optimization Algorithm for Power Plant Maintenance Scheduling," *Iraqi Journal of Computers, Communications, Control and Systems Engineering*, vol. 24, no. 2, pp. 65-78, 2024, <https://doi.org/10.33103/uot.ijccce.24.2.6>.