The Aeronautical Journal (2025), **129**, pp. 1810–1832
doi:10.1017/aer.2025.12

RESEARCH ARTICLE

Tuning of cascade PID controller gains of quadcopter under bounded disturbances using metaheuristic based research algorithm

E.H. Çopur[1], E. Balta[2] and H.H. Bilgic[2]

[1]Department of Aerospace Engineering, Necmettin Erbakan University, Konya 42040, Türkiye
[2]Department of Aeronautical Engineering, Necmettin Erbakan University, Konya 42040, Türkiye
Corresponding author: E.H. Çopur; Email: ehcopur@erbakan.edu.tr

Received: 5 October 2023; **Revised:** 25 June 2024; **Accepted:** 11 February 2025

Keywords: Cascade PID controller; search algorithm; metaheuristic algorithm; quadcopter; optimization

Abstract

The proportional–integral–derivative (PID) controller remains widely used in industrial applications today due to its simplicity and ease of implementation. However, tuning the controller's gains is crucial for achieving desired performance. This study compares the performance of PID controllers within a cascade control architecture designed for both position and attitude control of a quadcopter. Particle swarm optimisation (PSO), grey wolf optimisation (GWO), artificial bee colony (ABC), and differential evaluation (DE) methods are employed to optimally tune the PID parameters. A set of PID gains is determined offline by minimising various modified multi-objective functions based on different fitness measures: IAE, ISE, ITAE and ITSE. These measures are adapted as fitness functions for position and attitude control. A simulation study is conducted to determine which fitness function yields the optimal PID gains, as evidenced by the lowest values of the objective functions. In these simulations, two different desired trajectories are designed, and the controllers are applied to ensure the quadcopter tracks these trajectories accurately. Additionally, to test the robustness of the flight control architecture and the finely tuned PID controllers against various environmental effects and parametric uncertainties, several case scenarios are also explored.

Nomenclature

b	thrust factor
d	drag factor
g	gravitational acceleration
m	quadcopter mass
$\boldsymbol{r}$	position vector in NED frame
u_F	total thrust force
$\boldsymbol{u}_\tau$	torque control vector
x, y, z	position in NED frame
CoG	centre of gravity
F_i	thrust force of individual rotor
L	distance from the propeller to CoG
I_r	motor moment of inertia
I_{xx}, I_{yy}, I_{zz}	inertias around x, y, z axes
$K_{p_*}, K_{i_*}, K_{d_*}$	proportional, integral and derivative gains
$\boldsymbol{R}$	rotation matrix

© The Author(s), 2025. Published by Cambridge University Press on behalf of Royal Aeronautical Society.

https://doi.org/10.1017/aer.2025.12 Published online by Cambridge University Press

Greek symbol

ϕ	roll angle
θ	pitch angle
$\boldsymbol{\xi}$	orientation vector
ψ	yaw angle
Ω_i	angular speed of each motor

1.0 Introduction

Unmanned air vehicles (UAVs) can be operated autonomously or remotely via a flight control system. Unlike manned aircraft, UAVs do not risk pilot lives during operation. Advanced technological developments have enhanced UAV manoeuverability, reduced manufacturing costs, and simplified assembly processes. Due to these advantages, UAVs have gained widespread popularity in the aviation industry and are increasingly used for hobby purposes.

Quadcopters, a type of UAV with rotary wings, have attracted significant attention from researchers over the past two decades. Often referred to as drones in military contexts, quadcopters have become a prominent research topic due to features such as vertical take-off and landing capabilities, light weight and extended endurance [1,2]. These advantages make quadcopters widely used in military operations, search-and-rescue missions, agriculture, mapping, surveillance and reconnaissance [3–5]. However, the attitude control of quadcopters presents ongoing challenges due to aerodynamic uncertainties, the nonlinear nature of their dynamics, parameter uncertainties and sensitivity to environmental factors [6–8]. Thus, an effective controller is essential for overcoming these challenges and achieving successful trajectory tracking missions [9].

In recent years, more complex controllers have been tested for altitude and attitude control of quadcopters. Advanced control methods often draw inspiration from traditional ones found in the literature. PID, LQR, and H-infinity are well-known examples of linear control methods [10–12]. Nonlinear control methods such as feedback linearisation, model predictive control, backstepping control, sliding mode control and adaptive control [13–17], are developed to address the stability and robustness of systems affected by external disturbances and uncertainties [18]. Additionally, fuzzy methods and artificial neural networks have become popular for model-free, learning-based controllers in recent years [19–21].

Although various approaches have been tested and implemented for UAV control under a wide range of flight conditions, the PID controller still holds a significant place in flight control systems. PID controllers are favoured in industrial applications for their easy implementation, straightforward design process, ability to reduce steady-state error through parameter tuning and low computational demand [22–24]. However, classical PID tuning methods may not always yield satisfactory results in precision control applications, prompting the use of artificial intelligence-based algorithms for fine-tuning PID gains in recent years [25].

Traditional approaches such as the Ziegler-Nichols [26] and Cohen-Coon [27] methods have been used to tune PID gains. Recent studies have evaluated various tuning techniques; for instance, proportional-derivative (PD) controllers tuned using adaptive particle swarm optimisation (APSO) and Ziegler-Nichols (ZN) methods were tested on a quadcopter, revealing that APSO-based PD controllers delivered superior tracking accuracy, shorter settling times, and lower overshoots compared to those tuned with ZN methods [28]. However, this study only focused on tuning parameters for position control and did not consider fine-tuning attitude controllers for safety reasons.

Another study developed a comprehensive model of a quadcopter, incorporating gyroscopic moments, external disturbances such as wind and the Earth's atmospheric rotation. This complex model was used to design individual PID controllers for each translational and rotational motion in all three axes [29]. Unlike the previous study, the ZN method was utilised to tune PID coefficients. Further research compared the Ziegler-Nichols-based PID and the self-tuning fuzzy PID controllers on a linear quadcopter model to achieve fine-tuning of the attitude and altitude controllers. It was concluded that

https://doi.org/10.1017/aer.2025.12 Published online by Cambridge University Press

the self-tuning fuzzy PID controller improved time response characteristics in terms of settling time and overshoot [30]. Another ZN-based PID controller was also introduced for attitude control of an X-configuration quadcopter without considering the position control. It was assumed that applying the ZN-based optimisation procedure only for tuning PID coefficients for pitch movement was sufficient due to the symmetric structure of the quadcopter. The results indicated that although the ZN-based PID controller produced lower rise times in the pitch axis, the PSO-based PID controller provided better overall control performance [31]. However, traditional PID tuning methods cannot produce satisfactory responses in precise control applications, which is why AI-based algorithms have been increasingly used in recent years to tune PID coefficients.

In a relevant study, a PD controller was implemented into two distinct control architectures for both altitude and attitude control of a quadcopter in hover flight mode [32]. The first architecture utilised a genetic algorithm (GA) and particle swarm optimisation (PSO) to design PD controllers for a nonlinear quadcopter model, while the second used an LQR-based PD controller for a linear model. In the latter case, PSO and GA first optimised the weighting matrices of the LQR controller, which was then used to tune PD gains. The simulation results indicated that the PSO-based PD controller provided the best performance for ISE and their own performance index, while the GA-based PD excelled in IAE values. In another study, a robust controller was developed to manage weight variations in quadcopters, employing both PSO-based PID and differential evaluation (DE)-based PID controllers [33]. These controllers were designed to enable the quadcopter to follow a straight trajectory along the x-axis and maintain its initial position in the y-z plane. Tested in the GAZEBO simulation environment across different quadcopter weights, the evolutionary optimisation algorithms outperformed classical tuning methods in terms of controller effectiveness.

The gains of four individual PID controllers were separately tuned using GA, evolution strategies (ES), DE, and cuckoo search (CS) algorithms for a linear quadcopter model controlling motions in the x-y plane, z-axis and roll-pitch-yaw axes. To assess these methods, two different flight trajectories were established. The results varied per trajectory and controller, with GA and ES generally outperforming the others [34]. The PSO method was also compared with the GA method for tuning classical PID and fractional order PID (FOPID) controllers using a linear quadcopter model. The findings showed that PSO-FOPID delivered superior performance in the x-y, z, roll-pitch axes, while the PSO-PID controller was more effective in the yaw axis [35]. These studies predominantly used a linear model for tuning PID coefficients, focusing mainly on attitude stabilisation, which can compromise control performance during tracking flight missions. Therefore, there is a need for a more efficient optimisation procedure to tune PID coefficients for successful tracking in autonomous flight missions. Since nature-inspired optimisation algorithms require a mathematical model of the controlled system, this study proposes an alternative optimisation procedure with a modified multi-objective function based on a nonlinear quadcopter model. Moreover, the PID coefficients used in the attitude and position controllers are simultaneously tuned during the optimisation process for a trajectory path designed in 3D space.

In this study, a cascade PID control algorithm is designed using a nonlinear model of a quadcopter. Three PID controllers are used in the inner loop and three in the outer loop. The outer loop controls the position of the quadcopter, aiming to move it to desired locations and regulate the attitude angles—roll, pitch and yaw—along with the required total thrust force. The desired attitude angles are then sent to the inner loop, where the attitude controller generates the necessary torques around the roll, pitch and yaw axes. The fine-tuning of controller gains has been adjusted with PSO, grey wolf optimisation (GWO), artificial bee colony (ABC), and (DE). Integral time absolute error (ITAE), integral time square error (ITSE), integral square error (ISE), and integral absolute error (IAE) criteria are selected as the fitness functions for the optimisation algorithms. A modified multi-objective function was designed by weighting the position and attitude errors in these fitness functions according to their effects on the system responses. The tracking performance of the designed controller has been tested in simulations. Square and spiral trajectories were generated, and the results of the simulation studies showed that the algorithms used to optimise the PID coefficients provide successful results. Additionally, the controller

https://doi.org/10.1017/aer.2025.12 Published online by Cambridge University Press

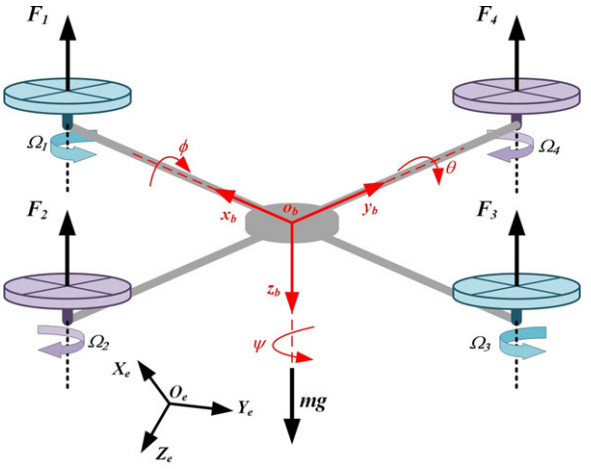


Figure 1. *The configuration of quadcopter UAV with respect to its frames.*

performances under bounded disturbances were analysed. The robustness analysis, considering important effects such as ground effect, wind effect and mass variation, was performed using the fine-tuned PID gains from previous simulations.

2.0 Modeling of quadcopter

A generic quadcopter consists of four separate rotors, each mounted on a symmetric framework, with each rotor connected by an identical link. Each rotor is operated by an electrical motor. Generally, two of them rotate in a clockwise direction, while the others rotate in a counter-clockwise direction. This arrangement guarantees zero net torque around the vertical axis at the CoG (centre of gravity) point, as shown in Fig. 1.

In the derivation of the quadcopter model, some generally accepted assumptions are made: i) the quadcopter possesses a rigid and symmetric structure, and ii) the origin O_b of the body-fixed frame $\mathcal{F}_B$ coincides with the CoG. To represent the dynamics of the translational and rotational motions of the quadcopter by using Newton-Euler formulation, a North-East-Down (NED) inertial frame $\mathcal{F}_E$, attached to the Earth, and a body-fixed frame $\mathcal{F}_B$ are defined. The position of the quadcopter is given by the vector $r = \left[x, y, z\right]^{\mathrm{T}}$ in the frame $\mathcal{F}_E$ and the orientation of the quadcopter is defined by the vector $\xi = [\phi, \theta, \psi]^{\mathrm{T}}$, which includes Euler angles (i.e. roll, pitch and yaw, respectively) in the frame $\mathcal{F}_B$ with respect to the frame $\mathcal{F}_E$. The translational dynamics of the quadcopter in three-dimensional space $\mathbb{R}^3$ are given as:

$$\ddot{r} = g\upsilon - \frac{1}{m} R u_F \tag{1}$$

where the unit vector $\upsilon = [0\ 0\ 1]^{\mathrm{T}}$ and the transformation matrix R from the body-fixed to the inertial reference frame system is given by

$$R = \begin{bmatrix} \mathrm{c}_\psi \mathrm{c}_\theta & \mathrm{c}_\psi \mathrm{s}_\theta \mathrm{s}_\phi - \mathrm{s}_\psi \mathrm{c}_\phi & \mathrm{c}_\psi \mathrm{s}_\theta \mathrm{c}_\phi + \mathrm{s}_\psi \mathrm{s}_\phi \\ \mathrm{s}_\psi \mathrm{c}_\theta & \mathrm{s}_\psi \mathrm{s}_\theta \mathrm{s}_\phi + c_\psi \mathrm{c}_\phi & \mathrm{s}_\psi \mathrm{s}_\theta \mathrm{c}_\phi - \mathrm{c}_\psi \mathrm{s}_\phi \\ -\mathrm{s}_\theta & \mathrm{c}_\theta \mathrm{s}_\phi & \mathrm{c}_\theta \mathrm{c}_\phi \end{bmatrix}$$

where $\mathrm{c}_* = \cos(*)$ and $\mathrm{s}_* = \sin(*)$. The total thrust force, u_F, is the sum of the individual thrusts, F_i, produced by each rotor driven by an electrical motor, expressed as $u_F = \sum_{i=1}^{4} F_i$ with $F_i = b\Omega_i$ where b is the thrust factor and Ω_i is the angular speed of each motor.

https://doi.org/10.1017/aer.2025.12 Published online by Cambridge University Press

Table 1. *Quadcopter parameters*

Parameters	Description	Value
I_{xx}	Moment of inertia about	0.007 kgm^2
I_{yy}	Moment of inertia about	0.007 kgm^2
I_{zz}	Moment of inertia about	0.012 kgm^2
I_r	Rotor moment of inertia	6.5×10^{-5} kgm^2
m	Mass	0.68 kg
L	Distance from propellers	0.17 m
d	Drag coefficient	8.5×10^{-7} Nms^2
b	Thrust coefficient	4.13×10^{-5} Ns^2
g	Gravitational constant	9.81 m/s^2

The rotational dynamics of the quadcopter are expressed as follows:

$$\boldsymbol{I}\ddot{\boldsymbol{\xi}} = -\dot{\boldsymbol{\xi}} \times \boldsymbol{I}\dot{\boldsymbol{\xi}} - \boldsymbol{I}_r\dot{\boldsymbol{\xi}} \times \boldsymbol{\Omega} + \boldsymbol{u}_{\boldsymbol{\tau}} \tag{2}$$

where $\boldsymbol{\Omega} = \begin{bmatrix} 0 & 0 & \Omega_r \end{bmatrix}$ with $\Omega_r = \Omega_1 - \Omega_2 + \Omega_3 - \Omega_4$, I_r represents the rotor moment of inertia, $I = \text{diag}\left(I_{xx}, I_{yy}, I_{zz}\right)$ denotes the moments of inertia with respect to its principal axes, and $\boldsymbol{u}_{\boldsymbol{\tau}} = \begin{bmatrix} u_\phi & u_\theta & u_\psi \end{bmatrix}^{\mathrm{T}}$ is the torque control vector. The complete dynamic model of the quadcopter can now be assembled as follows:

$$\begin{aligned}
I_{xx}\ddot{\phi} &= \left(I_{yy} - I_{zz}\right)\dot{\theta}\dot{\psi} - I_r\dot{\theta}\Omega_r + u_\phi \\
I_{yy}\ddot{\theta} &= \left(I_{zz} - I_{xx}\right)\dot{\phi}\dot{\psi} + I_r\dot{\phi}\Omega_r + u_\theta \\
I_{zz}\ddot{\psi} &= \left(I_{xx} - I_{yy}\right)\dot{\theta}\dot{\phi} + u_\psi \\
m\ddot{x} &= -\left(\mathrm{c}_\psi \mathrm{s}_\theta \mathrm{c}_\phi + \mathrm{s}_\psi \mathrm{s}_\phi\right) u_F \\
m\ddot{y} &= -\left(\mathrm{s}_\psi \mathrm{s}_\theta \mathrm{c}_\phi - \mathrm{c}_\psi \mathrm{s}_\phi\right) u_F \\
m\ddot{z} &= mg - \left(\mathrm{c}_\theta \mathrm{c}_\phi\right) u_F
\end{aligned}$$

where the control inputs are defined as

$$\begin{bmatrix} u_\phi \\ u_\theta \\ u_\psi \\ u_F \end{bmatrix} = \begin{bmatrix} 0 & Lb & 0 & -Lb \\ Lb & 0 & -Lb & 0 \\ d & -d & d & -d \\ b & b & b & b \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \tag{3}$$

where L is the distance from the propeller to the CoG and d is the drag factor. The model parameters of the quadcopter used as an example are given in Table 1 [36].

3.0 Design of the cascade pid controller

The classical control architecture typically includes only a single feedback loop. This structure often suffers from poor robustness performance when disturbances affect the plant under control, potentially resulting in slow settling times or high steady-state errors. To mitigate these disadvantages, a cascade control architecture can be employed, as depicted in Fig. 2.

This structure yields higher effectiveness in terms of control measures, particularly when the inner loop operates faster than the outer loop and when disturbances applied to the inner loop are effectively suppressed, thus mitigating the impact on the plant [37–39]. Generally, this control structure comprises two interlinked feedback loops: the inner and outer loops. The outer loop generates the command input for the inner loop, serving as the set-point.

https://doi.org/10.1017/aer.2025.12 Published online by Cambridge University Press

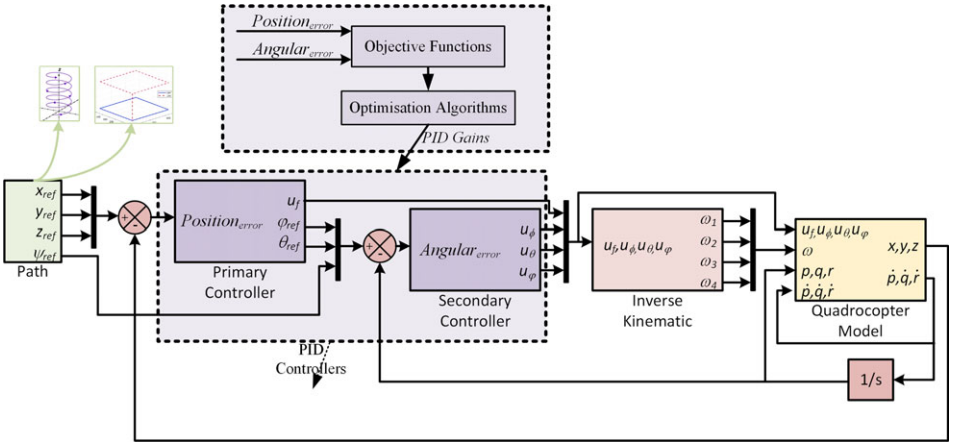


Figure 2. *A general cascade control architecture.*

In the control architecture developed here, the inner loop is tasked with performing attitude control, while the outer loop is designed to manage the altitude and the motion in the $x-y$ plane of the quadcopter. The outer loop generates the desired roll and pitch angles, denoted ϕ_d and θ_d, respectively. The yaw angle ψ is maintained at zero throughout the flight, except when another PID controller, tuned via an optimisation method, is implemented to track a desired non-zero yaw angle. As depicted in Fig. 2, $\boldsymbol{r_d} = \left[x_d, y_d, z_d\right]^{\mathrm{T}}$ represents the desired positions defined in the frame $\mathcal{F}_E$, and the position error vector for the outer loop is given by $[e_x, e_y, e_z]^{\mathrm{T}}$ calculated as $[x_d - x, y_d - y, z_d - z]^{\mathrm{T}}$. The position control of the quadcopter in the outer loop is executed using individual PID controllers for each axis, with the corresponding control inputs specified below:

$$\begin{aligned} u_x &= K_{p_x} e_x + K_{i_x} \int_0^t e_x dt + K_{d_x} \dot{e}_x \\ u_y &= K_{p_y} e_y + K_{i_y} \int_0^t e_y dt + K_{d_y} \dot{e}_y \\ u_z &= K_{p_z} e_z + K_{i_z} \int_0^t e_z dt + K_{d_z} \dot{e}_z \end{aligned} \tag{4}$$

where K_{p_*}, K_{i_*} and K_{d_*} are, respectively, the proportional, integral and derivative gains. Here, the subscript * indicates the corresponding axis (x, y or z axis). The control inputs (4) are utilised to determine the desired attitude angles and the total thrust force as follows:

$$u_F = m\sqrt{u_x^2 + u_y^2 + (u_z + g)^2} \tag{5}$$

$$\phi_d = \arcsin\left(-\frac{u_y}{\sqrt{u_x^2 + u_y^2 + (u_z + g)^2}}\right) \tag{6}$$

$$\theta_d = \arctan\left(\frac{u_x}{u_z + g}\right) \tag{7}$$

In order to maintain the desired attitudes, a PID controller is separately designed for yaw, roll and pitch axes, similar to the method used for controlling the position of the quadcopter. Therefore, the torque control inputs can be computed as follows:

https://doi.org/10.1017/aer.2025.12 Published online by Cambridge University Press

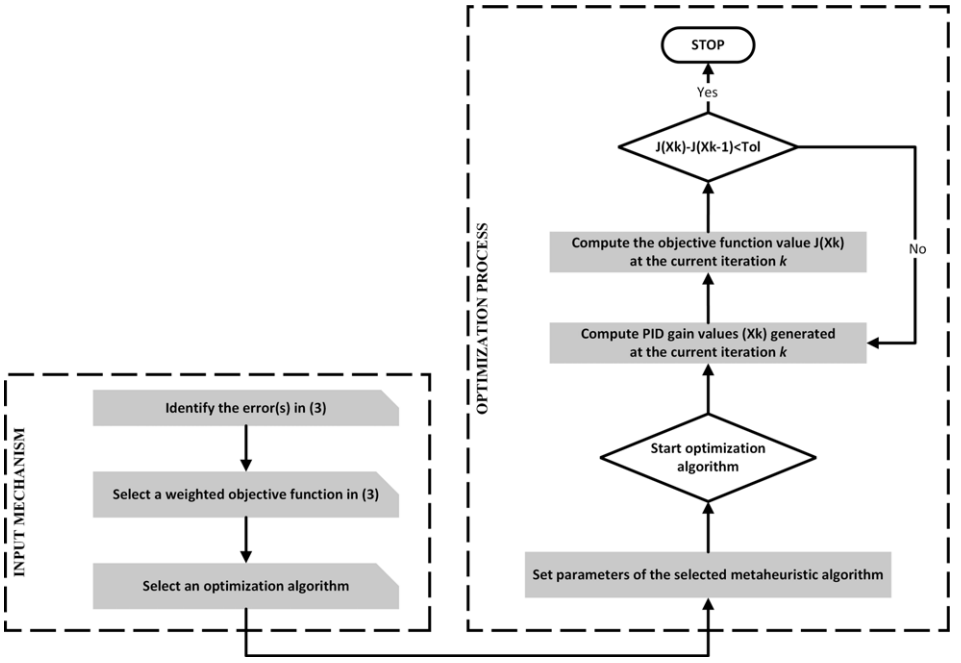


Figure 3. *The flowchart used in the optimisation mechanism.*

$$\begin{aligned} u_\phi &= K_{p_\phi} e_\phi + K_{i_\phi} \int_0^t e_\phi dt + K_{d_\phi} \dot{e}_\phi \\ u_\theta &= K_{p_\theta} e_\theta + K_{i_\theta} \int_0^t e_\theta dt + K_{d_\theta} \dot{e}_\theta \\ u_\psi &= K_{p_\psi} e_\psi + K_{i_\psi} \int_0^t e_\psi dt + K_{d_\psi} \dot{e}_\psi \end{aligned} \tag{8}$$

where e_ϕ, e_θ and e_ψ denote the attitude errors, i.e. $e_\phi = \phi_d - \phi$, $e_\theta = \theta_d - \theta$ and $e_\psi = \psi_d - \psi$. Here, K_{p_*}, K_{i_*} and K_{d_*} are the proportional, integral and derivative gains, respectively. The subscript $*$ indicates the corresponding axis (ϕ, θ or ψ). Now, the motor speeds required to track the desired positions and attitudes can be calculated by taking the inverse of (3), given by:

$$\begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{4b}u_F + \frac{1}{4d}u_\psi + \frac{1}{2Lb}u_\theta \\ \frac{1}{4b}u_F - \frac{1}{4d}u_\psi + \frac{1}{2Lb}u_\phi \\ \frac{1}{4b}u_F + \frac{1}{4d}u_\psi - \frac{1}{2Lb}u_\theta \\ \frac{1}{4b}u_F - \frac{1}{4d}u_\psi - \frac{1}{2Lb}u_\phi \end{bmatrix}$$

4.0 Optimisation algorithms

In this section, the optimisation algorithms employed to fine-tune the PID gains, as specified in (4) and (8), are briefly introduced. Figure 3 illustrates the generic optimisation procedure utilised in the simulations.

4.1 Particle swarm optimsation

Particle swarm optimisation (PSO) is a population-based optimisation method, first developed by Kennedy and Eberhart in 1995 [40]. They drew inspiration from the social behaviours observed in schools of fish and flocks of birds.

https://doi.org/10.1017/aer.2025.12 Published online by Cambridge University Press

PSO has gained popularity as an optimisation technique due to its minimal control parameter requirements and straightforward structure. In PSO, the current position of each individual in a swarm is recorded. This includes the position of the individual with the best value in the current iteration (P_{best}) and the position with the best global value (G_{best}). P_{best} converges towards G_{best} as each individual shares its location with others throughout the optimisation process [41]. The velocity of each individual is then computed and their positions are updated using:

$$v_{ij}^{k+1} = wv_{ij}^{k} + c_1 r_1 \left(\text{pbest}_{ij}^{k} - x_{ij}^{k}\right) + c_2 r_2 \left(\text{gbest}_{ij}^{k} - x_{ij}^{k}\right) \tag{9}$$

$$x_{ij}^{k+1} = x_{ij}^{k} + v_{ij}^{k+1} \tag{10}$$

where c_1 and c_2 are acceleration constants that regulate each individual's convergence rate to its best position. High acceleration constants can cause individuals to overshoot their best positions, while low values might prevent reaching these positions altogether. r_1 and r_2 are random numbers between 0 and 1. The inertia weight w is also chosen from a range of 0 to 1; higher values facilitate global search capabilities, while lower values enhance local search effectiveness. pbest_{ij}^{k} is the best solution reached by the individual i up to iteration k, and gbest_{ij}^{k} is the best solution found in the group up to iteration k. v_{ij}^{k} and x_{ij}^{k} denote the velocity and position of individual i at iteration k, respectively. After updates, the fitness values of all individuals in the new population are recomputed, and these computations are repeated until the maximum allowed number of iterations is reached.

4.2 Differential evolution (DE)

Differential evolution (DE), first introduced in 1995 by Rainer Storn and Kenneth Price, is based on a population-based optimisation algorithm. DE is a meta-heuristic algorithm that simulates natural evolution by combining different individuals to generate new ones with superior fitness values [42].

The DE algorithm employs three distinct operators: mutation, crossover and selection. To produce a new individual, DE uses a differential mutation operator that combines three randomly selected individuals from the current population. A new candidate solution is then generated by combining this mutation with the current individual using a crossover operator. The selection of the individual subject to mutation and the transition point is stochastic, enabling the algorithm to explore the search space randomly. The new population is evaluated using an objective function, and the best solutions are stored for the next iteration [42].

The initial population is generated by

$$X_{i,j} = X_{min} + \text{rand}\,[0, 1]\,(X_{max} - X_{min})$$

where $X_{i,j}$ represents the candidate solution, and X_{min} and X_{max} denote the lower limit and upper limit. Here, rand [0, 1] is a random number between 0 and 1. In the mutation process, new solutions are are formulated as:

$$Y_i = X_{r1} + F\,(X_{r2} - X_{r3})$$

where Y_i is the new solution and X_{r1}, X_{r2} and X_{r3} are the three randomly selected solutions. F is a scale factor, typically set between 0 and 2. The crossover process is then performed as follows:

$$Z_i = \begin{cases} Y_{i,j} & \text{rand}\,[0, 1] < CR \\ X_{i,j} & \text{otherwise} \end{cases}$$

where CR is the crossover ratio. Finally, in the selection stage, the choice is made based on the following criterion:

$$X_{i,j} = \begin{cases} Z_i & f\,(Z_i) \leq f\left(X_{i,j}\right) \\ X_{i,j} & \text{otherwise} \end{cases}$$

https://doi.org/10.1017/aer.2025.12 Published online by Cambridge University Press

4.3 Artificial bee colony (ABC)

First proposed in 2005 by Dervis Karaboga, inspired by the food search behaviour of honey bees [43], the ABC algorithm is a swarm intelligence-based optimisation method. It consists of three different bee populations: employed bees, onlooker bees and scout bees. Employed bees are tasked with searching for better solutions near the current solution. When a new solution is discovered, it is compared with the current solution; if superior, the current solution is updated; otherwise, it is retained. Subsequently, employed bees share their solutions with onlooker bees, who then search for the best solution among those provided by the employed bees. Scout bees are tasked with exploring new regions within the search space to discover superior solutions. When a food source is exhausted, the corresponding employed bee transitions into a scout bee.

The locations of the initial food sources are determined by:

$$x_{ij} = x_j^{min} + \text{rand}\,[0, 1]\left(x_j^{max} - x_j^{min}\right) \tag{11}$$

with $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, M$ where M and N denote the number of the food sources and the parameters to be optimised, respectively. A new source is then generated for the employed bee associated with the solution x_i using:

$$v_{ij} = x_{ij} + \Phi_{ij}\left(x_{ij} - x_{kj}\right) \tag{12}$$

subject to

$$v_{ij} = \begin{cases} x_j^{min} v_{ij} < x_j^{min} \\ v_{ij} x_j^{min} \leq v_{ij} \leq x_j^{max} \\ x_j^{max} v_{ij} > x_j^{max} \end{cases}$$

where v_{ij} is a candidate solution and $\Phi_{ij} \in [-1, 1]$ is a random number. The probability value P_i used by the onlooker bees in the selection process is calculated by

$$P_i = \frac{\text{fit}_i}{\sum_{i=1}^{N} \text{fit}_i}$$

In the roulette wheel selection mechanism, a random number between 0 and 1 is generated for each source. If the random number is less than P_i, then a new source is generated using (12). If the new source is superior to the current source, it replaces the current source and the dimension failure is reset. Otherwise, the dimension failure count is incremented by one. This process is repeated until all onlooker bees have settled into food source areas. Dimension failures are then reviewed, and a new source is generated using (11) for the exhausted sources. Subsequently, the best solution is recorded. Finally, all these steps are repeated until the stopping criterion is met.

4.4 Grey wolf optimisation (GWO)

The grey wolf optimiser (GWO) is a swarm intelligence-based metaheuristic algorithm first proposed by Mirjalili et al. in 2014 [44]. The GWO algorithm is inspired by the hunting behaviour, leadership and social hierarchy observed in grey wolf packs, which include four distinct ranks: alpha (α), beta (β), delta (δ) and omega (ω). This algorithm models the wolves' tactics of encircling, hunting and attacking their prey. In the wild, grey wolves encircle their prey and coordinate their attacks, a strategy that is emulated in GWO by dividing the search space into subspaces assigned to each wolf.

The encircling behaviour is mathematically modeled by:

$$\vec{D} = \left|\vec{C}\vec{X}_p(t) - \vec{X}(t)\right|$$

$$\vec{X}\,(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D}$$

https://doi.org/10.1017/aer.2025.12 Published online by Cambridge University Press

where t denotes the iteration step, $\vec{A}$ and $\vec{C}$ are the coefficient vectors, $\vec{X}_p$ denotes the position of the prey, and $\vec{X}$ is the position vector of a grey wolf. The coefficients $\vec{A}$ and $\vec{C}$ are calculated as follows:

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a}$$
$$\vec{C} = 2\vec{r}_2$$

where components of $\vec{a}$ linearly decrease from 2 to 0 across each iteration, and $\vec{r}_1$ and $\vec{r}_2$ are random vectors within the range of 0 to 1. The hunting behaviour is mathematically represented by:

$$\vec{D}_\alpha = \left|\vec{C}_1\vec{X}_\alpha - \vec{X}\right|$$
$$\vec{D}_\beta = \left|\vec{C}_2\vec{X}_\beta - \vec{X}\right|$$
$$\vec{D}_\delta = \left|\vec{C}_3\vec{X}_\delta - \vec{X}\right|$$

and

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha$$
$$\vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta$$
$$\vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot \vec{D}_\delta$$
$$\vec{X}\,(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}$$

where the subscripts α, β and δ denote the wolves in the wolf pack. The attacking phase is initiated when $\left|\vec{A}\right| \geqslant 1$, allowing the pack to search the area, and approaches the prey (or optimum solution) when $\left|\vec{A}\right| < 1$.

5.0 Fitness function

The PID controller aims to eliminate the error between the desired and current trajectories. However, completely eradicating this tracking error in real-time applications is not feasible. Instead, control practitioners strive to minimise the error as much as possible.

The control architecture proposed in this study incorporates six distinct PID controllers, as detailed in Section 3.0. The PSO algorithm is employed to minimise both the tracking errors and the optimisation time. Consequently, the most critical step involves designing an appropriate fitness function. The fitness functions used for the optimisation process are based on well-known performance measures, including IAE, ISE, ITSE, and ITAE. These fitness functions are defined as follows:

$$J_i = \begin{cases} \int e_i(t)^2 dt, & \text{for ISE} \\ \int |e_i(t)|\, dt, & \text{for IAE} \\ \int t\,|e_i(t)|\, dt, & \text{for ITAE} \\ \int t\left[e_i(t)^2\right] dt, & \text{for ITSE} \end{cases} \tag{13}$$

where $e_i(t)$ denotes the tracking error in the corresponding state. Utilising the fitness function (13), the objective function is expressed as:

$$J = \sum_{i=1}^{n} \tilde{J}_i$$

where n is the dimension of the state vector and $\tilde{J}_i$ is the weighted fitness function defined by $\tilde{J}_i = w_i J_i$ with a positive weighting coefficient w_i. The weighting coefficients are selected to ensure that the value of $\tilde{J}_i$ is approximately equal for each state. Moreover, the same weighting parameters are applied across all objective functions. The weighting coefficients ω_1, ω_2, ω_3, ω_4, ω_5 and ω_6 correspond to the states x, y, z, ϕ, θ and ψ, respectively. Table 2 provides the values of the weighting coefficients for all states.

https://doi.org/10.1017/aer.2025.12 Published online by Cambridge University Press

Table 2. *The weighting coefficients*

w_i	ISE	IAE	ITSE	ITAE
w_1	109.108	5.763	97.134	1.30
w_2	109.108	9.234	132.885	1.444
w_3	3547.986	51.080	7418.947	36.170
w_4	1199.587	39.635	1913.912	22.723
w_5	199.334	10.480	434.971	13.119
w_6	41.375	11.289	542.093	43.529

Table 3. *Parameters values of the optimisation algorithms*

Algorithm	Parameter	Value
PSO	Inertia weight limits	0.1–0.9
	c_1 (Personal acceleration constant)	1.5
	c_2 (Global acceleration constant)	1.5
GWO	Number of variables	18
ABC	Number of food sources	PN/2
	Number of variables	18
DE	Crossover ratio	0.9
	Scale factor	0.6

6.0 Results

The performance of the cascade flight control system, equipped with PID controllers tuned by the aforementioned optimisation algorithms, is evaluated in simulations designed for the trajectory tracking of a quadcopter. For these optimisation algorithms, the maximum number of iterations, the population size and the lower bound are respectively set to 100, 40 and 0. Conversely, the upper bound is selected within the range of 50 to 100. The values of the remaining parameters, which are set to search for the optimum gains of the PID controllers, are provided in Table 3. These parameters are selected arbitrarily, hence, this parameter setting yields good results rather than the best possible outcomes.

Simulations are conducted for two distinct flight missions. In the first mission, the quadcopter operates in autonomous flight mode, following a spiral-shaped reference trajectory while maintaining a constant heading. In the second mission, the spiral trajectory is replaced by a rectangular trajectory in the $x-y$ plane at a fixed altitude. Initially, the quadcopter ascends to an altitude of 1 m, then proceeds to follow a square trajectory with 1 m sides in the x-y plane. Consequently, the tracking performances of the designed controllers during sharp turns are compared.

6.1 Spiral trajectory tracking mission

The reference position of the quadcopter following the spiral trajectory is defined by the following time-dependent function:

$$x = \sin(t)$$
$$y = \cos(t) - 1$$
$$z = t$$

and the heading angle is consistently maintained at 30°.

Figure 4 illustrates the positions of the quadcopter controlled by the cascade flight control system, with PID coefficients independently tuned using PSO, ABC, GWO and DE optimisation algorithms to

https://doi.org/10.1017/aer.2025.12 Published online by Cambridge University Press

Table 4. *The best values of the fitness functions for spiral trajectory*

Algorithm	IAE	ISE	ITAE	ITSE
PSO	5.535	5.136	6.757	5.423
GWO	5.195	4.523	5.221	5.250
ABC	6.378	5.710	9.650	5.450
DE	5.640	4.504	6.645	4.985

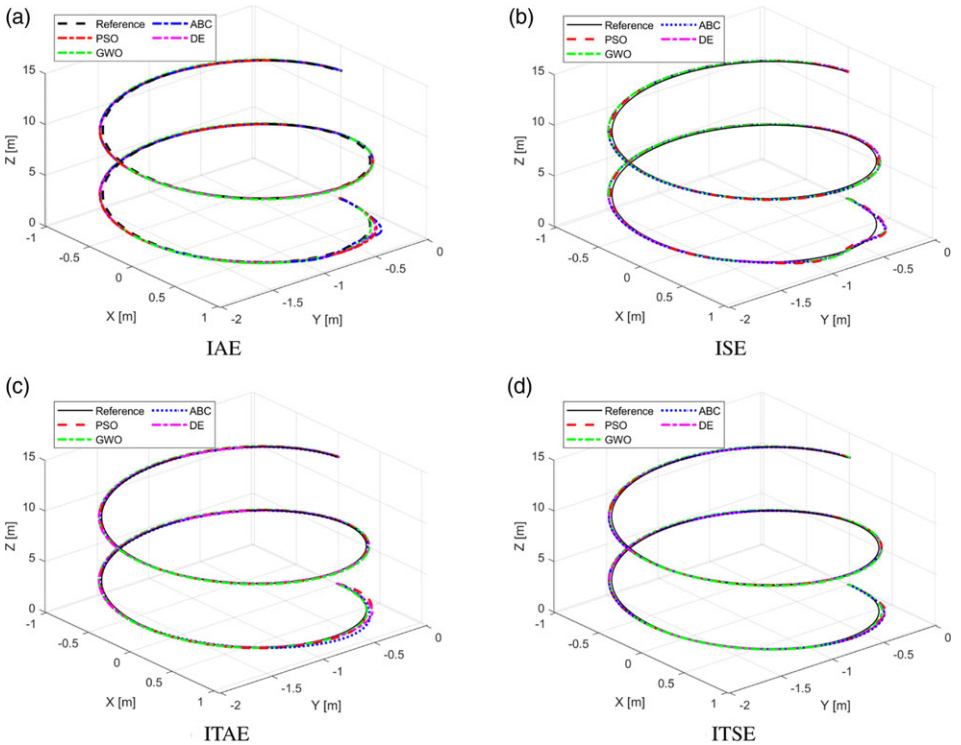


Figure 4. *Position response of the quadcopter in the spiral trajectory tracking mission with different optimisation algorithms.*

minimise the fitness functions. It is evident that the quadcopter successfully follows the reference trajectory in all cases. However, notable differences in the transient response phase are observed due to the effects of the optimisation algorithms. To further analyse these differences, a comprehensive examination of the error evolution is conducted. Figures 5 and 6 display the position and attitude error trajectories, respectively. As indicated, the GWO algorithm yields a less oscillatory response with faster convergence in the transient phase while minimising the IAE and ITAE fitness functions. Conversely, in scenarios employing the ISE and ITSE fitness functions, the DE algorithm provides satisfactory time responses. These outcomes are also reflected in Table 4, which presents the best values of the fitness functions for the spiral trajectory. According to the data in Table 1, the best values of IAE and ITAE are achieved using the GWO, while the DE algorithm produces the best values of ISE and ITSE. These findings are corroborated by the error analysis presented in Table 4.

https://doi.org/10.1017/aer.2025.12 Published online by Cambridge University Press

https://doi.org/10.1017/aer.2025.12 Published online by Cambridge University Press

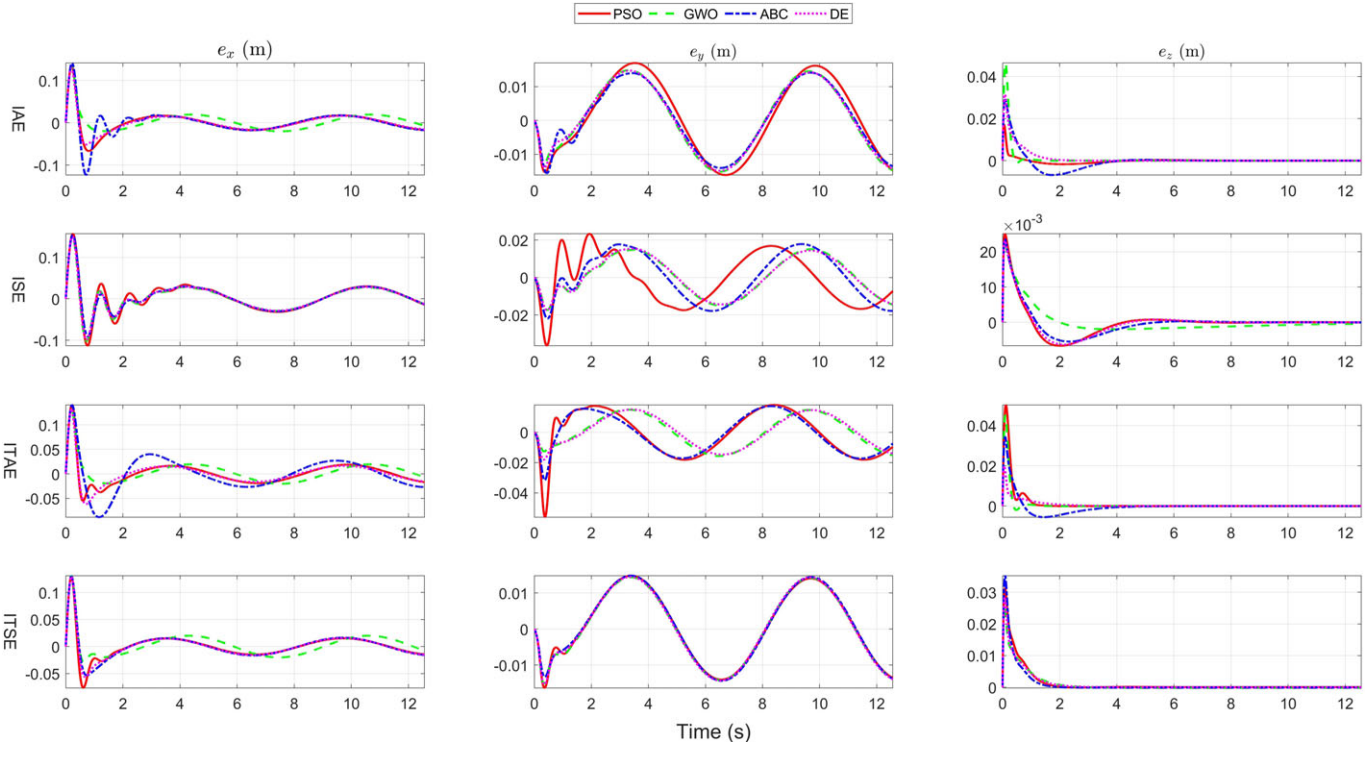


Figure 5. *Position errors of the quadcopter in the spiral trajectory tracking mission with different optimisation algorithms.*

https://doi.org/10.1017/aer.2025.12 Published online by Cambridge University Press

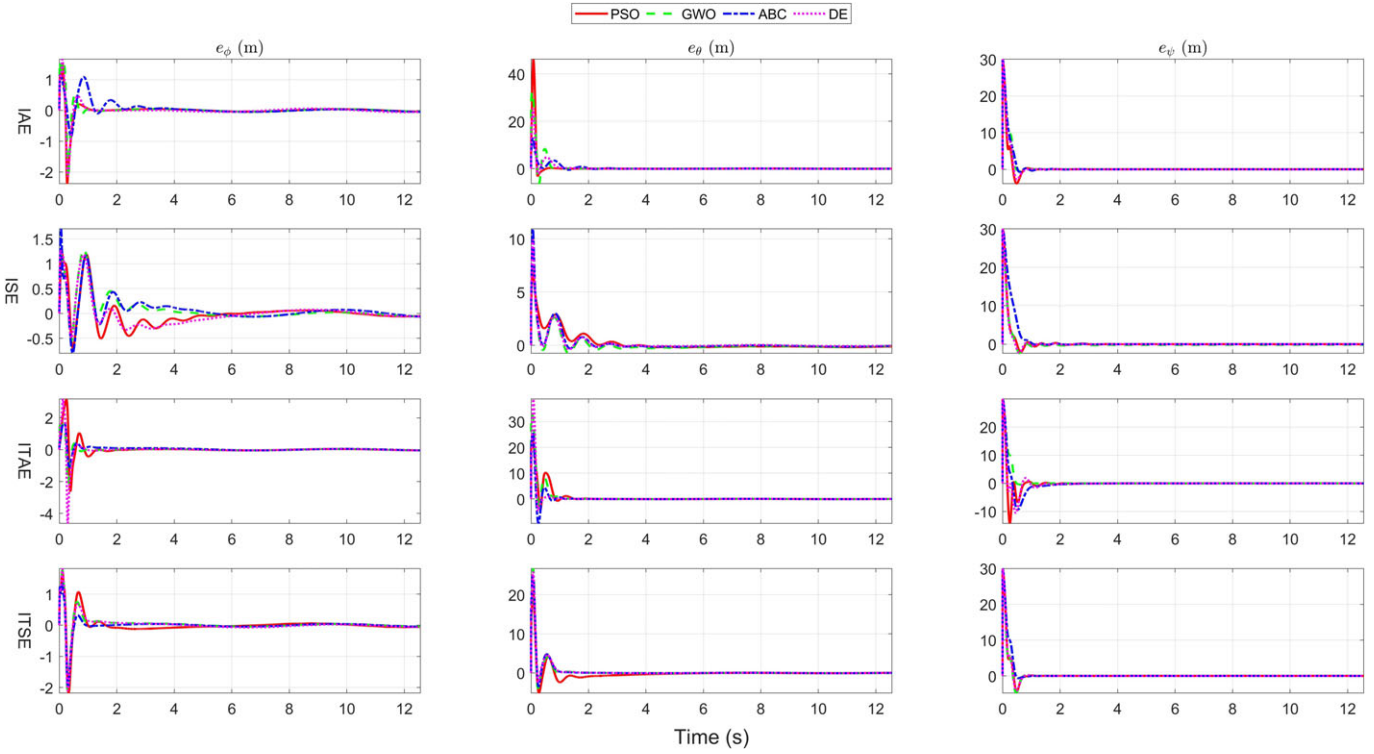


Figure 6. *Attitude errors of the quadcopter in the spiral trajectory tracking mission with different optimisation algorithms.*

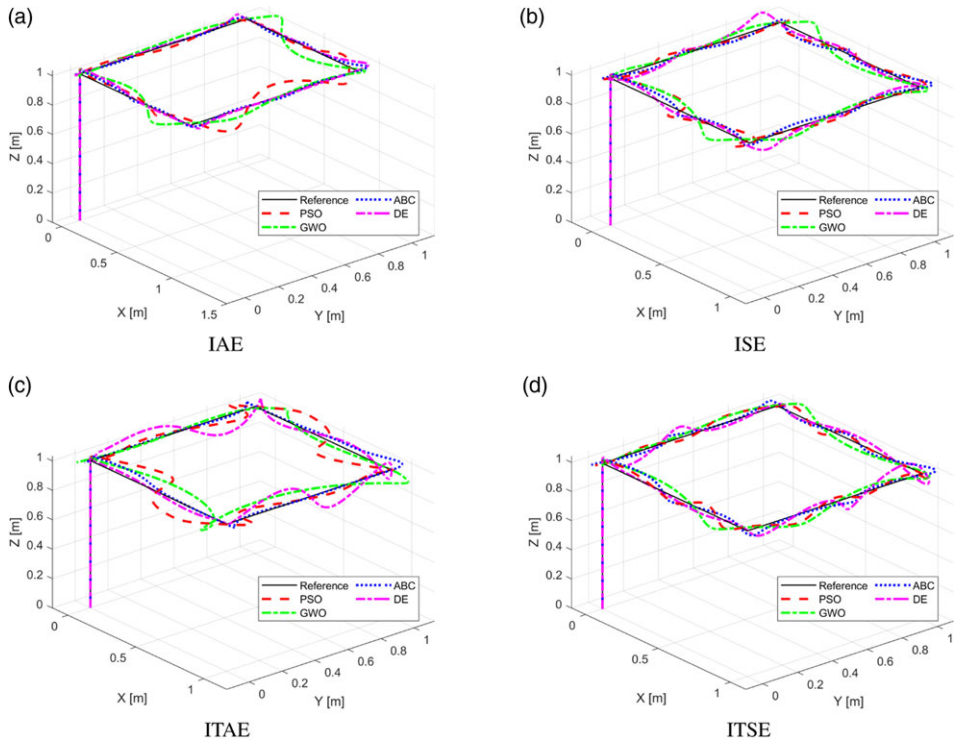


Figure 7. *Position response of the quadcopter in the square trajectory tracking mission with different optimisation algorithms.*

6.2 Square trajectory tracking mission

From the launch position (0,0,0), the quadcopter initially ascends to an altitude of 1m, reaching the coordinates (0,0,1). Subsequently, it moves 1 m along the positive x-axis to the coordinates (1,0,1). Following this, the quadcopter travels 1 m along the positive y-axis to the coordinate(1,1,1). It then proceeds to move 1 m in the negative x-axis direction, arriving at the point (0,1,1). Lastly, it travels 1 m in the negative y-axis direction, returning to (0,0,1), where it concludes the movement by maintaining a hover position. Similar to the spiral trajectory, the reference yaw angle (ψ) remains constant at 30°. For this mission, the PID optimisation process is reinvoked for all scenarios.

The performance of various PID controllers in tracking the reference square trajectory is assessed by comparing the quadcopter's position with the predefined trajectory. Figure 7 displays the positions of the quadcopter controlled by PID controllers, each tuned separately by the optimisation algorithms previously employed to minimise four distinct fitness functions. From Fig. 7, it is evident that some PID controllers, such as those tuned by the DE algorithm to minimise the ITAE function and by the PSO algorithm to minimise the IAE function, exhibit unreliable performance for this mission. These observations are further detailed in the error analysis, the results of which are presented in Figs. 8 and 9. While the ABC algorithm minimising the IAE fitness function achieves lower position and attitude errors compared to other algorithms, the GWO algorithm is found to be the most effective for tracking the square trajectory with other fitness functions. Table 5 corroborates these results.

6.3 Robustness against ground effect, mass variation and disturbance

In the previous section, a comprehensive analysis was conducted to evaluate the performance of the meta-heuristic algorithms used to fine-tune PID coefficients, focusing on trajectory-tracking control

https://doi.org/10.1017/aer.2025.12 Published online by Cambridge University Press

https://doi.org/10.1017/aer.2025.12 Published online by Cambridge University Press

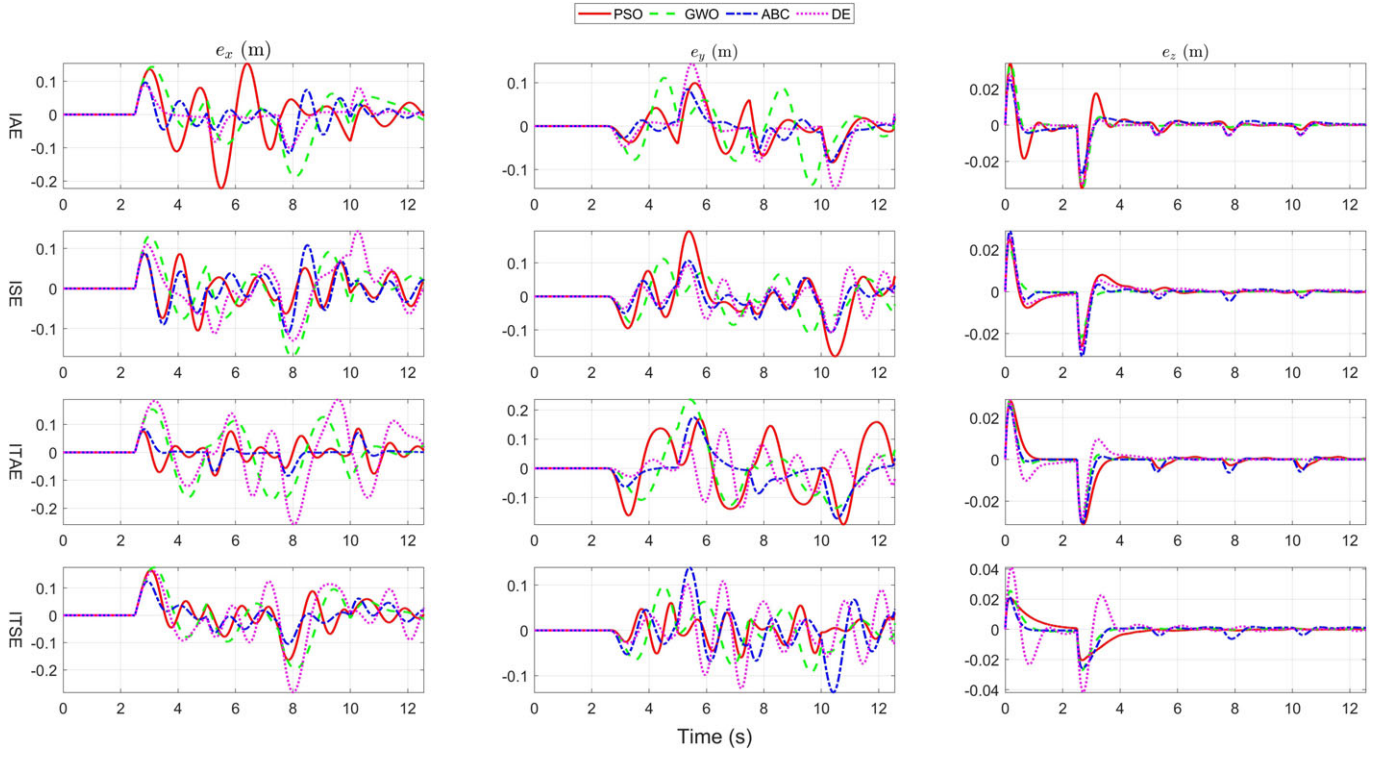


Figure 8. *Position errors of the quadcopter in the square trajectory tracking mission with different optimisation algorithms.*

https://doi.org/10.1017/aer.2025.12 Published online by Cambridge University Press

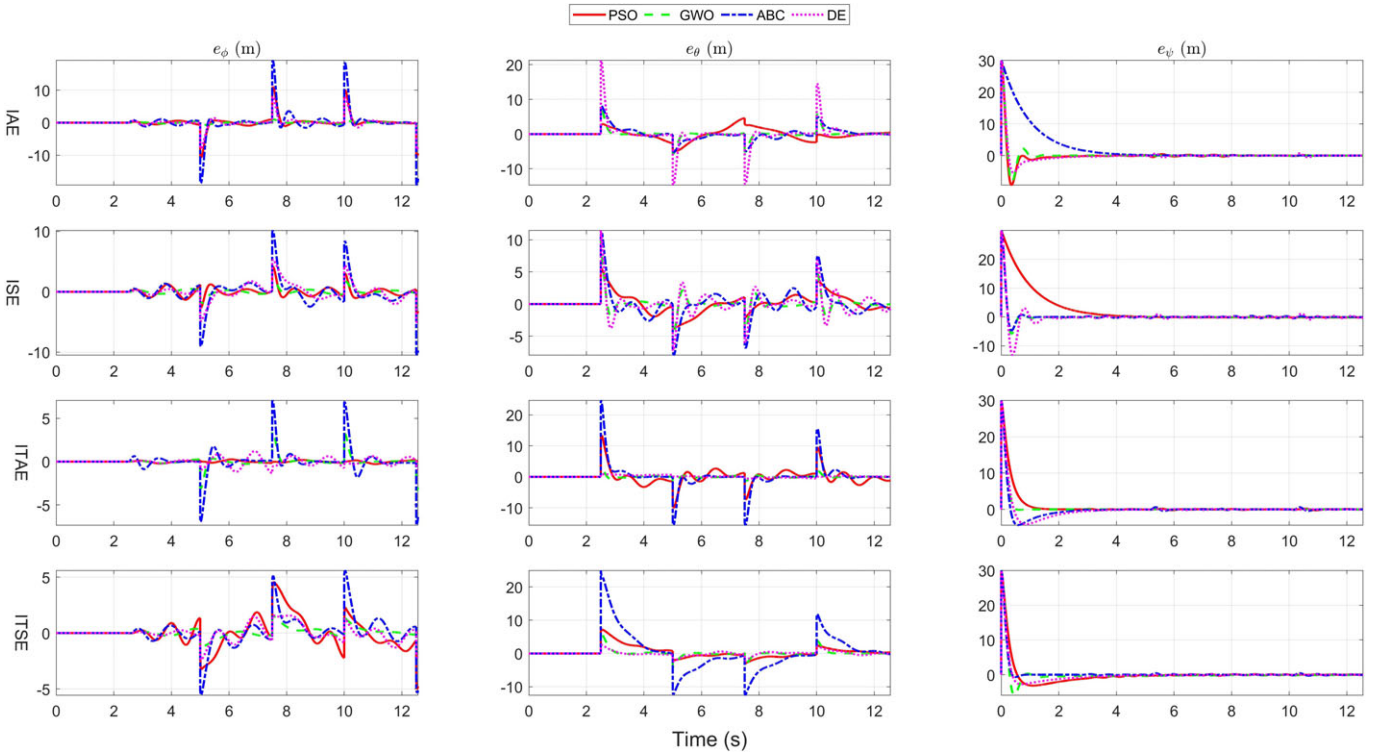


Figure 9. *Attitude errors of the quadcopter in the square trajectory tracking mission with different optimisation algorithms.*

Table 5. *The best values of the fitness functions for square trajectory*

Algorithm	IAE	ISE	ITAE	ITSE
PSO	18.616	18.778	52.353	163.452
GWO	27.922	11.894	28.947	85.516
ABC	14.500	19.537	42.202	145.728
DE	23.806	27.908	59.498	373.971

based on the nonlinear kinematics of the quadcopter. Currently, the robustness of the fine-tuned PID controllers is being tested in simulations. Given the low-altitude flight trajectories of the quadcopter, the ground effect is a significant factor and must be considered. Several approaches to modeling the ground effect have been documented [45–48]. Additionally, a linear model has been developed by Wei et al. [49], which is based on experimental results. This linear model is described as follows:

$$\frac{u_{F_{IGE}}}{u_F} = \begin{cases} -0.0373\frac{z}{R} + 1.1651 & \frac{z}{R} < 4.42 \\ 1 & \text{otherwise} \end{cases}$$

In the model, z represents the height from the ground, R denotes the propeller radius, and $u_{F_{IGE}}$ is the in-ground effect thrust force. Additionally, the flight controller confronts significant challenges from wind disturbances. To achieve accurate trajectory tracking in harsh environments, the flight controller must exhibit robustness against wind disturbances. While unsteady atmospheric conditions pose substantial control challenges at high altitudes, their effects are also non-negligible at very low altitudes, particularly during take-off or payload dropping phases. Consequently, for a more comprehensive analysis of robustness, wind disturbance is incorporated into the nonlinear quadcopter model as external moment disturbances applied to the roll and pitch axes. In the simulations, the following torques are employed [50]:

$$\tau_{d_\phi} = \tau_{d_\theta} = 7 + 2cos\,(2\pi t/3)$$

In the robustness study, two different flight scenarios are considered. In the first scenario, the PID controllers, previously fine-tuned for a spiral path, are tested in the presence of wind disturbances and ground effects. In the second scenario, the quadcopter undertakes a flight mission involving payload transportation. During this mission, the quadcopter, carrying three identical payloads each weighing 200 grams, follows the square route described in the previous section. The quadcopter is programmed to drop one of the payloads at each of the following locations: (1,0,1), (1,1,1), and (0,1,1). This mission also facilitates the investigation of the robustness of the fine-tuned controllers against parameter uncertainty.

6.3.1 Spiral trajectory tracking mission

The total length of the reference spiral trajectory is 42.4 m. The total distances traveled by the quadcopter and the corresponding RMSE values are provided in Table 6. The results reveal that the minimum RMSE and path length values are achieved by the PID controller fine-tuned by the ABC algorithm to minimise the IAE objective function. Figure 10 clearly demonstrates the quadcopter's capability, controlled by this specific controller, to accurately track the reference spiral trajectory.

6.3.2 Square trajectory tracking mission

This mission is depicted in Fig. 11, with the total path length being 5 m. The total path traveled by the quadcopter and the RMSE errors are summarised in Table 7. The minimum path length and RMSE value are achieved using PID coefficients optimised by the ABC algorithm, which minimised the IAE objective

https://doi.org/10.1017/aer.2025.12 Published online by Cambridge University Press

Table 6. *Performance of PID controllers for spiral path*

		PSO	ABC	GWO	DE
IAE	Path	55.07	43.90	47.57	5.50×10^{4}
	RMSE	0.0037	4.94×10^{-4}	0.0019	133.86
ISE	Path	5.30×10^{4}	5.51×10^{4}	47.41	5.36×10^{4}
	RMSE	133.45	140.23	0.0016	135.35
ITAE	Path	5.60×10^{4}	5.59×10^{4}	48.76	1.25×10^{4}
	RMSE	143.00	142.57	0.0021	21.42
ITSE	Path	44.15	5.57×10^{4}	46.56	46.14
	RMSE	7.31×10^{-4}	142.38	0.0016	0.0015

Table 7. *Performance of PID controllers for 2nd Scenario in Fig. 11*

		PSO	ABC	GWO	DE
IAE	Path	13.98	5.59	9.67	5.88
	RMSE	0.0013	2.84×10^{-4}	0.0013	5.86×10^{-4}
ISE	Path	7.14	5.81	9.43	6.37
	RMSE	3.36×10^{-4}	4.04×10^{-4}	8.73×10^{-4}	3.12×10^{-4}
ITAE	Path	19.84	5.99	13.00	50.93
	RMSE	0.0017	6.97×10^{-4}	0.0015	0.0075
ITSE	Path	14.98	5.62	7.36	8.90
	RMSE	0.0012	5.06×10^{-4}	9.33×10^{-4}	5.19×10^{-4}

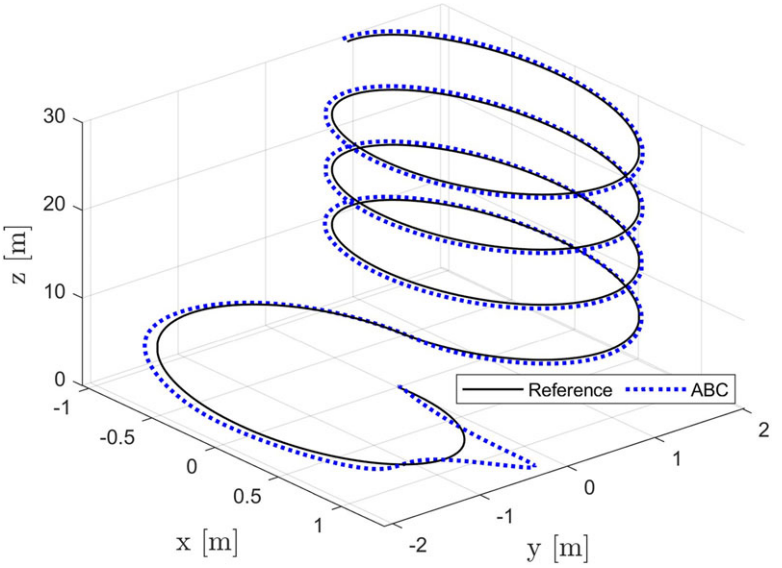


Figure 10. *Position response of the quadcopter in the spiral trajectory tracking mission with ABC-IAE algorithm.*

function. As shown in Fig. 12, the PID controller based on the ABC-IAE algorithm demonstrates considerable robustness in the presence of variations in total mass and external disturbances such as ground effect and wind.

https://doi.org/10.1017/aer.2025.12 Published online by Cambridge University Press

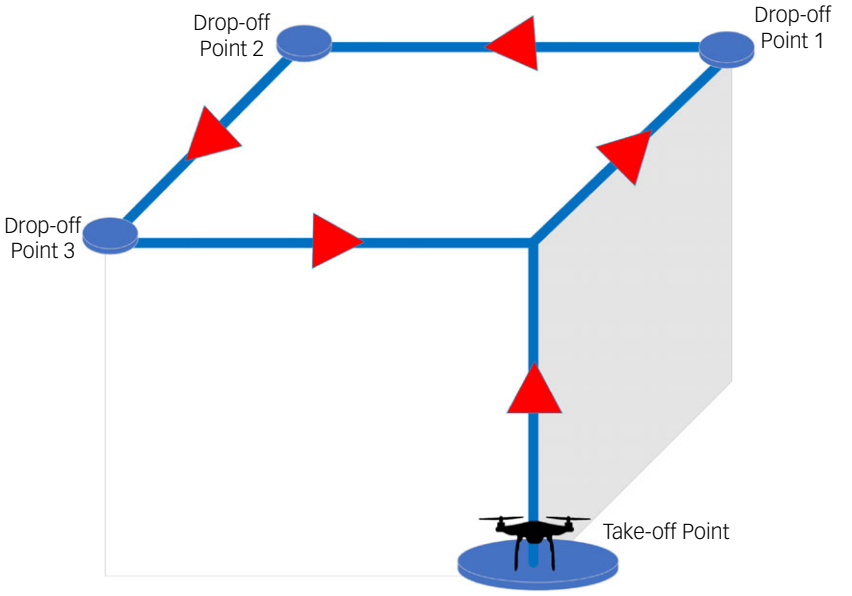


Figure 11. *Reference path for the second scenario.*

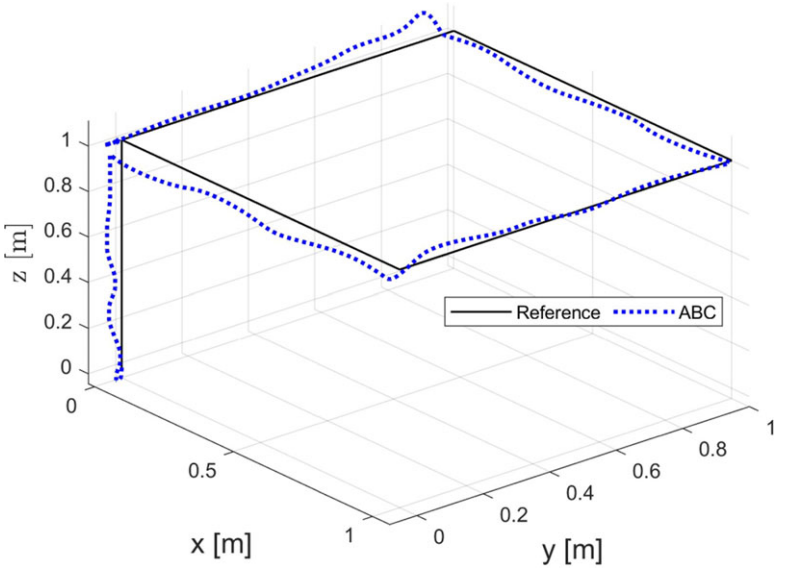


Figure 12. *Position response of the quadcopter with ABC-IAE algorithm for the* 2^{nd} *scenario.*

7.0 Conclusion

This study compares the tracking performances of different PID controllers within a flight control architecture for a quadcopter, which are tuned using various optimisation algorithms: PSO, ABC, GWO and DE. These controllers operate within a bounded search region of interest. The flight control system employs a cascade structure comprising a position controller and an attitude controller, resulting in six PID controllers being finely tuned. During the optimisation process, four distinct weighted fitness functions based on IAE, ISE, ITAE and ITSE performance criteria are utilised.

https://doi.org/10.1017/aer.2025.12 Published online by Cambridge University Press

In the simulations, two different reference trajectories are considered: a spiral trajectory and a square trajectory. The simulation results reveal that the GWO algorithm achieves the minimum values for IAE and ITAE, while the DE algorithm provides the minimum values for ISE and ITSE for the spiral trajectory. For the square trajectory mission, the ABC algorithm delivers the best IAE performance measure. Among the tested algorithms, GWO proves to be the most effective in minimising ISE, ITAE and ITSE performance measures.

References

[1] Mardan, M., Esfandiari, M. and Sepehri, N. Attitude and position controller design and implementation for a quadrotor, *Int. J. Adv. Robot. Syst.*, 2017, **14**, (3). doi: 10.1177/1729881417709242

[2] Saif, E. and Eminoğglu, İ. Modelling of quad-rotor dynamics and hardware-in-the-loop simulation, *J. Eng.*, 2022, **2022**, (10), pp 937–950. doi: 10.1049/tje2.12152

[3] Zainuddin, K, Ghazali, N. and Mat Arof, Z. The feasibility of using low-cost commercial unmanned aerial vehicle for small area topographic mapping, in *2015 IEEE International Conference on Aerospace Electronics and Remote Sensing Technology (ICARES)*, IEEE, Bali, Indonesia, 2015. doi: 10.1109/icares.2015.7429825

[4] Xuan-Mung, N. and Hong, S.-K. Improved altitude control algorithm for quadcopter unmanned aerial vehicles, *App. Sci.*, 2019, **9**, (10), p 2122. doi: 10.3390/app9102122

[5] Ju, C., Kim, J., Seol, J. and Il Son, H. A review on multirobot systems in agriculture, *Comput. Electron. Agric.*, 2022, **202**, p 107336. doi: 10.1016/j.compag.2022.107336

[6] Mian, A.A. and Daobo, W. Modeling and backstepping-based nonlinear control strategy for a 6 DOF quadrotor helicopter, *Chin. J. Aeronaut.*, 2008, **21**, (3), pp 261–268. doi: 10.1016/s1000-9361(08)60034-5

[7] Razmi, H. and Afshinfar, S. Neural network-based adaptive sliding mode control design for position and attitude control of a quadrotor UAV, *Aerosp. Sci. Technol.*, 2019, **91**, pp 12–27. doi: 10.1016/j.ast.2019.04.055

[8] Kamal Hosseini Sani, S. and Kakavand, E. Online sliding mode-based integrated guidance and control system for a quadcopter UAV in the presence of unknown obstacles, *Iran. J. Sci. Technol. Trans. Electric. Eng.*, 2022, (1), pp 193–201. doi: 10.1007/s40998-022-00537-x

[9] Zhu, X.-Z., Cabecinhas, D., Xie, W., Casau, P., Silvestre, C., Batista, P. and Oliveira, P. Kalman–Bucy filter-based tracking controller design and experimental validations for a quadcopter with parametric uncertainties and disturbances, *Int. J. Syst. Sci.*, 2022, pp 1–25. doi: 10.1080/00207721.2022.2096939

[10] Ataka, A., Tnunay, H., Inovan, R., Abdurrohman, M., Preastianto, H., Cahyadi, A.I. and Yamamoto, Y. Controllability and observability analysis of the gain scheduling based linearization for UAV quadrotor, in *2013 International Conference on Robotics, Biomimetics, Intelligent Computational Systems*, IEEE, Jogjakarta, Indonesia, 2013, pp 212–218. doi: 10.1109/robionetics.2013.6743606

[11] Ahmad, F., Kumar, P., Bhandari, A. and Patil, P.P. Simulation of the quadcopter dynamics with LQR based control, *Mater. Today: Proc.*, 2020, **24**, pp 326–332. doi: 10.1016/j.matpr.2020.04.282

[12] Alexandrov, V., Rezkov, I., Shatov, D. and Chestnov, V. Discrete-time ${\mathrm{H}}_\infty $H∞ optimization for quadcopter altitude control, in *2021 29th Mediterranean Conference on Control and Automation (MED)*, IEEE, Puglia, Italy, 2021, pp 126–131. doi: 10.1109/med51440.2021.9480354

[13] Bhatia, A.K., Jiang, J., Zhen, Z., Ahmed, N. and Rohra, A. Projection modification based robust adaptive backstepping control for multipurpose quadcopter UAV, *IEEE Access*, 2019, **7**, 154121–154130. doi: 10.1109/access.2019.2946416

[14] Martins, L., Cardeira, C. and Oliveira, P. Feedback linearization with zero dynamics stabilization for quadrotor control, *J. Intell. Robot. Syst.*, 2020, **101**, (1). doi: 10.1007/s10846-020-01265-2

[15] Ullah, S., Mehmood, A., Khan, Q., Rehman, S. and Iqbal, J. Robust integral sliding mode control design for stability enhancement of under-actuated quadcopter, *Int. J. Control, Automat. Syst.*, 2020, **18**, (7), pp 1671–1678. doi: 10.1007/s12555-019-0302-3

[16] Kimura, T., Hoshino, K., Asano, Y., Honda, A., Motooka, N. and Ohtsuka, T. Application of nonlinear model predictive control to quadcopter equipped with internal control system, in *2022 SICE International Symposium on Control Systems (SICE ISCS)*, IEEE, Yokohama, Japan, 2022, pp 51–57. doi: 10.23919/SICEISCS54350.2022.9754519

[17] Farber, B.E. and Richards, C.M. Adaptive control and parameter-dependent anti-windup compensation for inertia-varying quadcopters, *Int. J. Control*, 2022, pp 1–14. doi: 10.1080/00207179.2022.2138555

[18] Roy, R., Islam, M., Sadman, N., Mahmud, M.P., Gupta, K.D. and Ahsan, M. A review on comparative remarks, performance evaluation and improvement strategies of quadrotor controllers, *Technologies*, 2021, **9**, (2), p 37. doi: 10.3390/technologies9020037

[19] Jiang, F., Pourpanah, F. and Hao, Q. Design, implementation, and evaluation of a neural-network-based quadcopter UAV system, *IEEE Trans. Ind. Electron.*, 2020, **67**, (3), pp 2076–2085. doi: 10.1109/tie.2019.2905808

[20] Canbek, K.O. and Oniz, Y. Real-time implementation of an interval type-2 fuzzy logic controller for the trajectory tracking of an UAV, in *2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, IEEE, Ankara, Turkey, 2021, pp 418–423. doi: 10.1109/ISMSIT52890.2021.9604539

https://doi.org/10.1017/aer.2025.12 Published online by Cambridge University Press

[21] Guzay, C. and Kumbasar, T. Aggressive maneuvering of a quadcopter via differential flatness-based fuzzy controllers: from tuning to experiments, *Appl. Soft Comput.*, 2022, **126**, p 109223. doi: 10.1016/j.asoc.2022.109223

[22] Jiao, Q., Liu, J., Zhang, Y. and Lian, W. Analysis and design the controller for quadrotors based on pid control method, in *2018 33rd Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, IEEE, Nanjing, China, 2018, pp 88–92. doi: 10.1109/YAC.2018.8406352

[23] Behrooz, F., Mariun, N., Marhaban, M., Radzi, M.M., and Ramli, A. Review of control techniques for HVAC systems—nonlinearity approaches based on fuzzy cognitive maps, *Energies*, 2018, **11**, (3), p 495. doi: 10.3390/en11030495

[24] Idrissi, M, Salami, M. and Annaz, F. Modelling, simulation and control of a novel structure varying quadrotor, *Aerosp. Sci. Technol.*, 2021, **119**, p 107093. doi: 10.1016/j.ast.2021.107093

[25] Wang, Q.-G., Lee, T.-H., Fung, H.-W., Bi, Q. and Zhang, Y. PID tuning for improved performance, *IEEE Trans. Control Syst. Technol.*, 1999, **7**, (4), pp 457–465. doi: 10.1109/87.772161

[26] Ziegler, J.G. and Nichols, N.B. Optimum settings for automatic controllers, *J. Dyn. Syst. Meas. Control*, 1993, **115**, (2B), pp 220–222. doi: 10.1115/1.2899060

[27] Cohen, G.H. and Coon, G.A. Theoretical consideration of retarded control, *J Fluids Eng.*, 1953, **75**, (5), pp 827–834. doi: 10.1115/1.4015451

[28] Yazid, E., Garrat, M. and Santoso, F. Optimal PD tracking control of a quadcopter drone using adaptive PSO algorithm, in *2018 International Conference on Computer, Control, Informatics and Its Applications (IC3INA)*, IEEE, Tangerang, Indonesia, 2018, pp 146–151. doi: 10.1109/IC3INA.2018.8629504

[29] Parfiryev, A.V., Ischuk, I.N., Dushkin, A.V., and Smolyakova, S.D. Algorithm for controlling the trajectory of an unmanned aerial vehicle with the possibility of flying around obstacles, in *2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, IEEE, St. Petersburg and Moscow, Russia, 2020, pp 2395–2400. doi: 10.1109/EIConRus49466.2020.9039467

[30] Sarbass, H., Tatli, A. and Onrat, A. Control of unmanned aerial vehicles using self tuning fuzzy PID, in *2018 26th Signal Processing and Communications Applications Conference (SIU)*, IEEE, Izmir, Turkey, 2018, pp 1–4. doi: 10.1109/SIU.2018.8404778

[31] Khodja, M.A., Tadjine, M., Boucherit, M.S. and Benzaoui, M. Experimental dynamics identification and control of a quadcopter, in *2017 6th International Conference on Systems and Control (ICSC)*, IEEE, Batna, Algeria, 2017, pp 498–502. doi: 10.1109/ICoSC.2017.7958668

[32] Artale, V., Milazzo, C.L.R., Orlando, C. and Ricciardello, A. Comparison of GA and PSO approaches for the direct and LQR tuning of a multirotor PD controller, *J. Ind. Manag. Opt.*, 2017, **13**, (4), pp 2067–2091. doi: 10.3934/jimo.2017032

[33] Can, M.S. and Ercan, H. Real-time tuning of PID controller based on optimization algorithms for a quadrotor, *Aircr. Eng. Aerosp. Technol.*, 2021, **94**, (3), pp 418–430. doi: 10.1108/aeat-06-2021-0173

[34] Hasseni, S.-E.-I., Abdou, L. and Glida, H.-E. Parameters tuning of a quadrotor PID controllers by using nature-inspired algorithms, *Evol. Intell.*, 2019, **14**, (1), pp 61–73. doi: 10.1007/s12065-019-00312-8

[35] Saribas, H. and Kahvecioglu, S. PSO and GA tuned conventional and fractional order PID controllers for quadrotor control, *Aircr. Eng. Aerosp. Technol.*, 2021, **93**, (7), pp 1243–1253. doi: 10.1108/aeat-08-2020-0185

[36] Kayacan, E. and Maslim, R. Type-2 fuzzy logic trajectory tracking control of quadrotor VTOL aircraft with elliptic membership functions, *IEEE/ASME Trans. Mechatron.*, 2017, **22**, (1), 339–348. doi: 10.1109/tmech.2016.2614672

[37] Oh, S-K., Jang, H.-J. and Pedrycz, W. Optimized fuzzy PD cascade controller: a comparative analysis and design, *Simul. Model. Pract. Theory*, 2011, **19**, (1), pp 181–195. doi: 10.1016/j.simpat.2010.06.004

[38] Raja, G.L. and Ali, A. Series cascade control: an outline survey, in *2017 Indian Control Conference (ICC)*, 2017, IEEE, Guwahati, India, pp 409–414. doi: 10.1109/indiancc.2017.7846509

[39] Soares, D.M., Calil, H.A.M. and Stephan, R.M. Cascade control vs full-state feedback, in *2019 IEEE 15th Brazilian Power Electronics Conference and 5th IEEE Southern Power Electronics Conference (COBEP/SPEC)*, 2019, IEEE, Santos, Brazil. doi: 10.1109/cobep/spec44138.2019.9065696

[40] Eberhart, R. and Kennedy, J. A new optimizer using particle swarm theory, in *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, IEEE, Nagoya, Japan, 1995, pp 39–43. doi: 10.1109/MHS.1995.494215

[41] Shi, Y. and Eberhart, R. A modified particle swarm optimizer, in *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98TH8360)*, IEEE, Anchorage, AK, USA, 1998, pp 69–73. doi: 10.1109/ICEC.1998.699146

[42] Storn, R. and Price, K. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, *J. Global Opt.*, 1997, **11**, (4), pp 341–359. ISSN 1573-2916. doi: 10.1023/A:1008202821328

[43] Karaboga, D. An idea based on honey bee swarm for numerical optimization, Technical report, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[44] Mirjalili, S., Mirjalili, S.M. and Lewis, A. Grey wolf optimizer, *Adv. Eng. Softw.*, 2014, **69**, pp 46–61. doi: 10.1016/j.advengsoft.2013.12.007

[45] Cheeseman, I.C. and Bennett, W.E. The effect of ground on a helicopterrotor in forward flight, Techreport RM No. 3021, Aeronautical Research Council Reports and Memoranda, London, UK, 1955.

[46] Danjun, L., Yan, Z., Zongying, S. and Geng, L. Autonomous landing of quadrotor based on ground effect modelling, in *2015 34th Chinese Control Conference (CCC)*, IEEE, 2015. doi: 10.1109/chicc.2015.7260521

[47] Hayden, J.S. The effect of the ground on helicopter hovering power required, in *American Helicopter Society 32nd Annual National V/STOL Forum*, Washington, DC, USA, 1976.

https://doi.org/10.1017/aer.2025.12 Published online by Cambridge University Press

[48] Sanchez-Cuevas, P., Heredia, G. and Ollero, A. Characterization of the aerodynamic ground effect and its influence in multirotor control, *Int. J. Aerosp. Eng.*, 2017, **2017**, pp 1–17. ISSN 1687-5974. doi: 10.1155/2017/1823056

[49] Wei, P., Chan, S.N., Lee, S. and Kong, Z. Mitigating ground effect on mini quadcopters with model reference adaptive control, *Int. J. Intell. Robot. Appl.*, 2019, **3**, (3), pp 283–297. ISSN 2366-598X. doi: 10.1007/s41315-019-00098-z

[50] Thanh, H.L.N.N. and Hong, S.K. Quadcopter robust adaptive second order sliding mode control based on PID sliding surface, *IEEE Access*, 2018, **6**, pp 66850–66860. ISSN 2169-3536. doi: 10.1109/access.2018.2877795

Cite this article: Çopur E.H., Balta E. and Bilgic H.H. (2025). Tuning of cascade PID controller gains of quadcopter under bounded disturbances using metaheuristic based research algorithm. *The Aeronautical Journal*, **129**, 1810–1832. https://doi.org/10.1017/aer.2025.12

https://doi.org/10.1017/aer.2025.12 Published online by Cambridge University Press