Arabian Journal for Science and Engineering (2023) 48:15241–15255
https://doi.org/10.1007/s13369-023-08088-x

RESEARCH ARTICLE-ELECTRICAL ENGINEERING

PSO–PID Controller for Quadcopter UAV: Index Performance Comparison

Nur Hayati Sahrir[1] · Mohd Ariffanan Mohd Basri[1]

Received: 17 November 2022 / Accepted: 18 June 2023 / Published online: 6 July 2023

Abstract

A quadcopter is underactuated where there are 6° of motion with only four rotors to control all six motions. Varying the speed of the four rotors can produce thrust, roll, pitch and yaw torque which results in specific movements of the quadcopter. This paper presents the dynamic modeling of a quadcopter, which derived using Newton–Euler formalism and Proportional–integral–derivative (PID) controller for a quadcopter unmanned aerial vehicle. The PID controller is employed in this study due to its simplicity and easy to design. However, it is relatively difficult to determine the optimal tuning gains for PID controller which requires an ample of time with consideration of quadcopter dynamics and nonlinearities. There are several traditional methods for PID gains tuning such as manual tuning and Ziegler–Nichols (ZN-PID) methods, but both are time-wasting with unreliable results specifically for quadcopter system. This work proposes PID gains optimization using Particle Swarm Optimization (PSO) algorithm (PSO–PID) for quadcopter to reduce tuning effort with optimal results. This meta-heuristic algorithm is implemented to provide the optimal PID gains for altitude and attitude stabilization through setup iterations and populations with fixed boundaries. PSO performance is evaluated using several index performances which are IAE, ISE, ITAE and ITSE. The results obtained confirm that the PSO meta-heuristic algorithm works acceptably good with all index performances, especially ITSE, in identifying optimal PID gains for stabilization and trajectory tracking of a quadcopter. It is also proven that PSO–PID controller is better than ZN-PID controller in escape maneuvering of roll motion during wind disturbance occurrence.

Keywords Index performance · Particle Swarm Optimization (PSO) · Proportional–integral–derivative (PID) controller · Quadcopter UAV

1 Introduction

Dual-system vertical take-off and landing (VTOL) UAV flight control is a rapidly developing topic with numerous current studies. Creating flight control systems that can efficiently manage the change from vertical to horizontal flight while providing stable flying in both modes is one of the main difficulties in this field of study. The use of different control systems, including neural networks and fuzzy logic controllers, as well as the integration of sensors and algorithms that can adapt to shifting flight conditions in real time have all been examined by researchers as potential solutions to this problem. Research has also been done to find ways to make VTOL UAVs more energy efficient, particularly by using hybrid propulsion systems, which mix gas and electric motors. This study has concentrated on minimizing energy consumption while preserving stability and safety by optimizing the flight control algorithms. The topic of dual-system VTOL UAV flight control is still developing as researchers investigate new methods and tools to raise the effectiveness, efficiency, and security of these aircraft. Generally, there are four types of UAV flight platforms which are single-rotor, multi-rotor, fixed-wing and hybrid UAV. Both single-rotor and multi-rotor can take-off and land vertically (VTOL) while fixed-wing can horizontally take-off and landing (HTOL). Multi-rotor is the focus in this research. There are also several types of multi-rotor such as tri-copter, quadcopter, hexa-copter and octocopter. Different names mean different number of propellers with different payload and endurance.

✉ Mohd Ariffanan Mohd Basri
ariffanan@utm.my

[1] Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 Johor Bahru, Johor, Malaysia

The quadcopter, being the most common, less complex, better stabilizing mechanism and cost-effective, is the one being researched. There is also a vast application of quadcopter in different fields nowadays such as aerial photography, sports, entertainment, geographic mapping, search and rescue, law enforcement, agriculture, disaster management, weather forecast, wildlife monitoring, inspection, mining, shipping and delivery and for defense system. These applications show that the quadcopter is so important that more improvements are expected in the future. However, a quadcopter is an underactuated system where it has 6° of freedom (DOF) but with only four actuator inputs which are the four rotors. Quadcopter is a nonlinear system and is easily exposed to system uncertainties and disturbances especially if being implemented in the real system [1]. This leads to another challenge which quadcopter has high coupling degree [2] between translational and rotational motions [3]. This brought a challenge to control the quadcopter movement from one place to another with accuracy.

Due to the downsides, there exists several control techniques such as Proportional–integral–derivative (PID) control [4–7], backstepping control [1, 8], adaptive control [9, 10], sliding mode control [11, 12], linear quadratic regulator (LQR) [13, 14], Linear Quadratic Gaussian (LQG) [15], fuzzy logic (FL) control [16, 17], model predictive control (MPC) [18] and other hybrid methods to produce a more robust controllers [19–22]. Despite, PID control technique has been utilized widely in various system fields and there are numerous studies on the use of PID controller for quadcopter UAV. A cascaded PID controller is designed in a study [23] to satisfy the 6° of motion of a quadcopter and to realize trajectory tracking. However, this study only utilized PID controller without exploring other control algorithms or advanced techniques for quadcopter trajectory tracking survival. A PID controller is adapted in another study [24] using a combination of nonlinear proportional–integral (NPI) controller and conventional PID controller strategy. This novel approach for quadcopter posture regulation has improved the stability and robustness of quadrotor. Similarly, the same author in another paper [25] utilized the combination of NPI and PID controller in quadcopter for the simulation of trajectory tracking, with disturbances included. The simulation results demonstrate that the presented controller performs better than the PID controller in terms of tracking accuracy, robustness toward disturbances, and trajectory tracking. Moreover, a study [26] was conducted using PID controller as a medium to test the optimized trajectory obtained from quadcopter's parameter identification.

There are several ways of PID tuning and for most beginners, conventional methods such as manual tuning and Ziegler–Nichols (ZN) seems to be the easiest way to tune PID gains for a system. Due to the time-wasting of PID manual tuning, ZN method is the better choice to tune the PID gains but it is also unreliable to be adapted in quadcopter system since this technique may result in system instability, as well as significant losses and damages. In recent years, there has been a growing trend of researchers using and exploring more meta-heuristic algorithms. Understanding the social swarm behavior of fish, birds, ants, and other animals in search of food and shelter, has aided researchers to ideate optimization search methods that can be effectively applied to a range of engineering issues [27]. Various studies have been conducted to find optimal PID gains using these meta-heuristic approaches such as Genetic Algorithms (GA) [3], Particle Swarm Optimization (PSO) [28], Capuchin Search Algorithm (CapSA) [27] and so much more. Those algorithms have been proven to be much more efficient in previous studies compared to conventional methods.

In this work, PID gains optimization was done using Particle Swarm Optimization (PSO) algorithm to provide optimal gains of PID controllers, based on the formulated fitness function, to achieve optimal maneuvering results of quadcopter. The main advantages of the PSO algorithm are it is the most researched, easily programmable, faster in convergence and mostly provides better solutions. PSO–PID controllers have been the subject of several studies looking into quadcopter control systems. A PSO–PID controller was utilized in a study [29]. There is a PID controller for each Euler angle and the parameters were optimized using the PSO technique, which considerably boosted the quadcopter's performance without the need to rely on more complex control algorithms. The controller performance was evaluated through sum absolute error fitness function. Similar to this, PSO–PID controller was utilized in [30] for trajectory tracking of a quadcopter. The PSO technique was utilized to increase the stability and tracking capabilities of the quadcopter by optimizing the PID settings of the attitude controller and position controller. The work proved to be more efficient than the traditional tuning method through graphical analysis and Mean Squared Error (MSE) evaluation. A PSO–PID controller was also utilized in a different study [31] for path following control tuning. The feed-forward path planning contains a set of intermediary waypoints, and the PD controller parameters were optimally tuned using the PSO algorithm, which increased the performance of quadcopter in path following. In addition [32], suggested a multi-objective function PSO algorithm to improve PID controller performance, which focuses only on the roll (ϕ) axes. The simulation was done using Parrot Mambo virtual model. The proposed work was found to provide better PID tuning compared to using MSE or MAE function evaluation.

The optimization algorithm was formulated based on [1, 28] and the objective function comprises of the output error and overshoot of altitude and attitude motions of quadcopter to assess the optimization performance. The output error in the fitness function was varied to integral of squared error

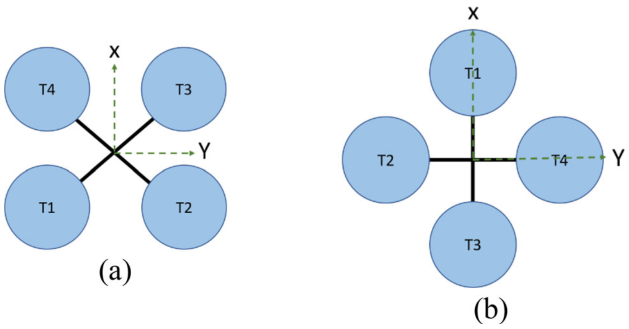


Fig. 1 **a** 'Cross' configuration **b** 'Plus' configuration

(ISE), integral of absolute error (IAE), integral of time multiplied by absolute error (ITAE) and integral of time multiplied by squared error (ITSE). The objective of this work is to compare the performance between IAE, ISE, ITAE and ITSE in producing the best PSO–PID controller performance for quadcopter. Based on the system verification done by PSO algorithm, the contribution of this paper is threefold: (1) discussion on the effect of each index performance on the quadcopter's altitude and attitude motions [33]; (2) simulation on trajectory tracking of quadcopter using circular and lemniscate pattern for different index performances with the identification of the best index error to be used [31]; (3) simulation of quadcopter altitude motion with wind disturbance and the performance comparison between PID gains obtained by ZN [34] and PSO [35].

Section 2 will briefly explain the mathematical model of quadcopter UAV based on several references. Section 3 contains the PID control mathematical model and description. Section 4 has some brief explanation on meta-heuristic algorithms and details on PSO–PID working algorithm. Section 5 presents the simulation results of PSO–PID algorithm with all index performances and Sect. 6 concludes this work briefly.

2 Mathematical Model of Quadcopter

2.1 Quadcopter Description

There are two types of quadcopter configurations, the 'cross' configuration and the 'plus' configuration as shown in Fig. 1.

Most studies use a 'plus' configuration by neglecting the nonlinear effect of quadcopter [29], and it is considerably more agile. However, a 'cross' configuration is considered as more stable because it uses two rotors to act during any movement [9]. For example, during pitch movement, speed of rotors 1 and 2 (3 and 4) will increase (decrease) simultaneously, for 'cross' configuration. However, in 'plus' configuration, only the speed of rotor 1(rotor 3) will decrease (increase) simultaneously. Figures 2 and 3 show the movements that a quadcopter can perform by changing the speeds of each rotor [1].

Figure 2a shows that all rotors have the same speed, can lift the quadcopter vertically, to certain altitude and hover in the air. Figure 2b shows that speed of rotor 1 and 4 increases, while rotor 2 and 3 decreases. This causes the quadcopter to roll along x-axis and change the quadcopter position along y-axis. Figure 2c explains the pitching of quadcopter along the y-axis where speed of rotor 1 and 2 increases, while rotor 3 and 4 decreases, thus change the quadcopter position along x-axis. Figure 2d shows the yawing of a quadcopter where two diagonal rotors with same direction of rotation increases (decreases) in speed, which causes the quadcopter to change its heading in the air [36]. When creating a dynamic model of a quadcopter, the following presumptions were considered:

1. All four rotors of the quadcopter are expected to be rigid and symmetrical throughout.
2. The origin and center of mass of the quadcopter body frame are identical.
3. At low speeds, aerodynamic effects and ambient wind disturbances are minimal.
4. Blade flapping is ignored in a reasonably fast rotor dynamic because rotors are thought to be rigid.

2.2 Quadcopter Dynamics

The translational dynamic equation of this quadcopter is obtained from Newton's Second Law and Euler's Rotational Equation of Motion and is defined as follows [1]:

$$\mathrm{m}\ddot{\Gamma} = u_{\mathrm{T}} R e_z - m g e_z \tag{1}$$

where m is the quadcopter mass, R is the rotation matrix, g is the gravitational acceleration, $e_z = (0\,0\,1)^T$ is the unit vector and $\underline{u}_{\mathrm{T}}$ is the total thrust of all four rotors. The rotation matrix of the body frame with respect to the earth frame is given by [37]:

$$R_b^e = \begin{bmatrix} C\theta C\psi & S\phi S\theta C\psi - S\psi C\phi & S\theta C\phi C\psi + S\phi S\psi \\ S\psi C\theta & S\phi S\theta S\psi + C\phi C\psi & S\theta S\psi C\phi - S\phi C\psi \\ -S\theta & S\phi C\theta & C\phi C\theta \end{bmatrix} \tag{2}$$

where C(angle) represents, cosine while S(angle) represents sine. Three Euler angles, namely roll angle (ϕ), pitch angle (θ) and yaw angle (ψ) form the orientation of the quadcopter. The rotational dynamic equation of this quadcopter is given

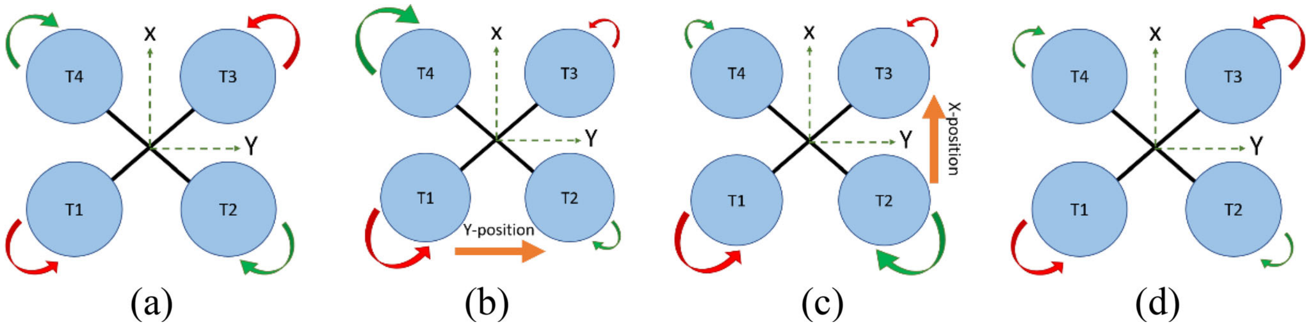


Fig. 2 ‘Cross’ configuration quadcopter movements **a** hovering **b** rolling **c** pitching **d** yawing

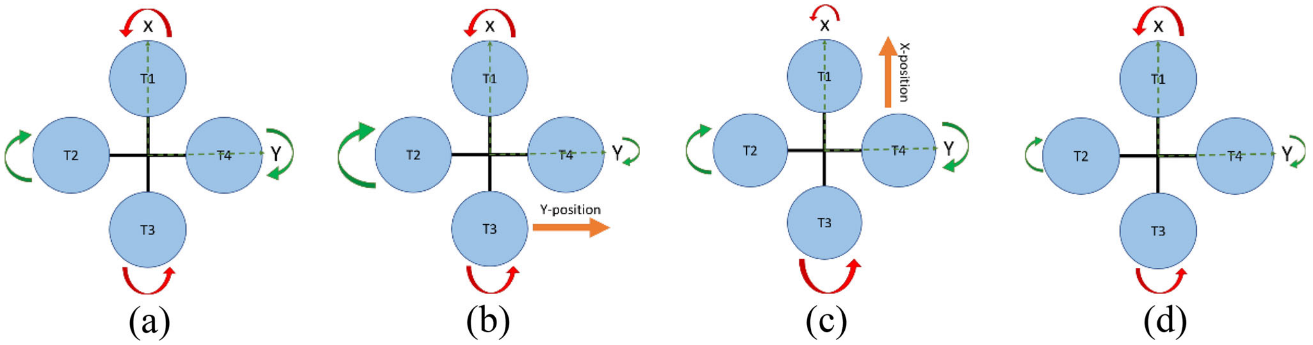


Fig. 3 ’Plus’ configuration quadcopter movements **a** hovering **b** rolling **c** pitching **d** yawing

by [1]:

$$I\dot{\omega} = -\omega \times I\omega - J_r(\omega \times e_z)\Omega_r + \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} \tag{3}$$

where it represents all airframe body torque, gyroscopic torque and Coriolis torque. I is the quadcopter inertia matrix, $\dot{\omega}$ is the angular acceleration vector, ω is the angular velocity vector, J_r is the rotor’s inertia, Ω_r is the relative speed of all rotors and τ_ϕ, τ_θ, τ_ψ represent roll, pitch and yaw torque. Based on Eqs. (1) and (3), the produced translational and rotational equation are as shown as follows:

$$\ddot{x} = \frac{1}{m}(S\theta C\phi C\psi + S\phi S\psi)U_1$$

$$\ddot{y} = \frac{1}{m}(S\theta C\psi C\phi + S\phi S\psi)U_1$$

$$\ddot{z} = \frac{1}{m}(C\phi C\theta)U_1 - g$$

$$\ddot{\phi} = \frac{1}{I_{xx}}\left[\left(I_{yy} - I_{zz}\right)\dot{\theta}\dot{\psi} - J_r\dot{\theta}\Omega_r + lU_2\right]$$

$$\ddot{\theta} = \frac{1}{I_{yy}}\left[(I_{zz} - I_{xx})\dot{\phi}\dot{\psi} + J_r\dot{\phi}\Omega_r + lU_3\right]$$

$$\ddot{\psi} = \frac{1}{I_{zz}}\left[\left(I_{xx} - I_{yy}\right)\dot{\theta}\dot{\phi} + U_4\right] \tag{4}$$

Note: $\Omega_r = (\Omega_1 - \Omega_2 + \Omega_3 - \Omega_4)$

The speed of each rotor is obtained from Eq. (5) [38], the speeds produced are used to generate a relative speed, Ω_r, which is required for quadcopter dynamics as shown in Eq. (4). Equation (6) shows the control inputs based on ‘cross’ configuration rotor speeds. Figure 4 shows the visualization of a quadcopter model based on these equations.

$$\begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} = \begin{bmatrix} K_T & K_T & K_T & K_T \\ 0 & -\ell K_T & 0 & \ell K_T \\ \ell K_T & 0 & -\ell K_T & 0 \\ K_d & -K_d & K_d & -K_d \end{bmatrix}^{-1} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} \tag{5}$$

$$U_1 = b({\Omega_1}^2 + {\Omega_2}^2 + {\Omega_3}^2 + {\Omega_4}^2)$$

$$U_2 = b\left({\Omega_1}^2 - {\Omega_2}^2 - {\Omega_3}^2 + {\Omega_4}^2\right)$$

$$U_3 = b({\Omega_1}^2 + {\Omega_2}^2 - {\Omega_3}^2 - {\Omega_4}^2)$$

$$U_4 = d\left({\Omega_1}^2 - {\Omega_2}^2 + {\Omega_3}^2 - {\Omega_4}^2\right) \tag{6}$$

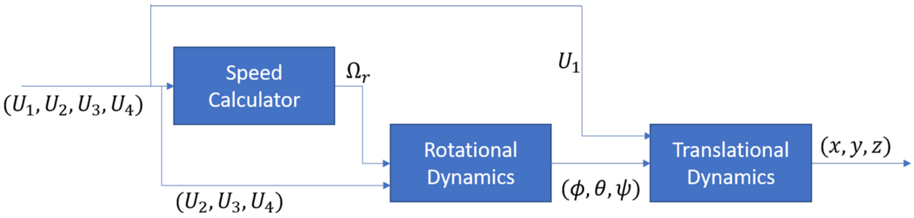


Fig. 4 Block diagram of quadcopter model

Table 1 Parameters of a quadcopter model

Parameter	Value
g	9.81 m s^2
m	0.5 kg
ℓ	0.2 m
$J_x = J_y$	4.85×10^{-3} kg m^2
J_z	8.81×10^{-3} kg m^2
J_r	3.36×10^{-5} kg m^2
K_T	2.92×10^{-6} kg m
K_d	1.12×10^{-7} kg m^2

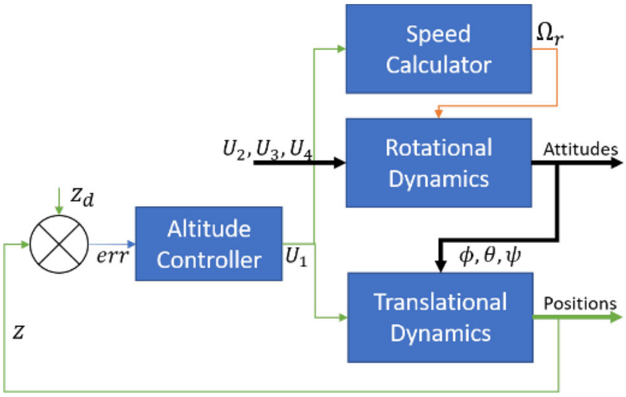


Fig. 6 Block diagram for altitude control

2.3 Quadcopter Parameters

The quadcopter model in this work has its parameters appropriately established by reference to [39], which are listed in Table 1.

3 Controller for Quadcopter UAV

In this part, block models are used to illustrate the structure of each controller in a quadcopter system. This section also includes the proposed PSO–PID controller design and methodology. Fig. 5 shows the whole visualization of quadcopter system including the inner and outer loop PID controllers for 6° of freedom (DOF) of quadcopter.

3.1 Control of Altitude

The error (difference between desired input and actual output) is taken into a controller for error attenuation in order to regulate the altitude of a quadcopter. The controller will then output an adjusted control input, U1, in order for the quadcopter to attain the desired height. The general altitude control structure for quadcopter system is shown in Fig. 6.

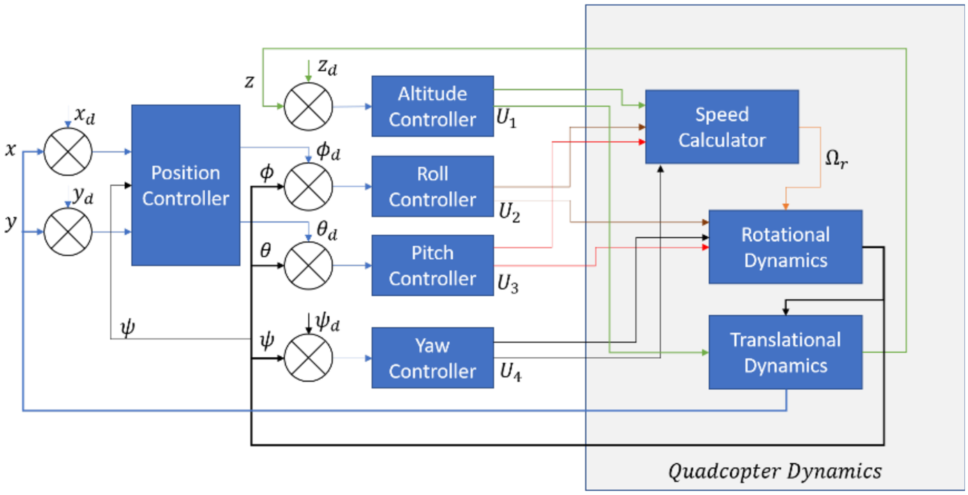


Fig. 5 Block diagram of a whole quadcopter system

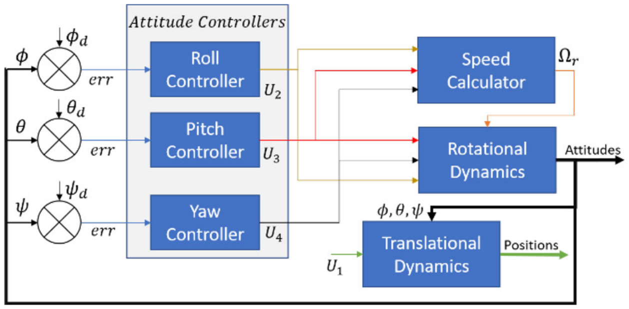


Fig. 7 Block diagram for attitudes control

3.2 Control of Attitudes

In a quadcopter, attitude refers to the three different angle movements that are known as roll, pitch, and yaw. In order to manage the three attitudes and get the quadcopter to comply with the specified requirements, the error resulting from the difference between the reference input and the actual output is used to produce the three control inputs, U2, U3, and U4. The quadcopter system's attitude control architecture is shown in Fig. 7.

3.2.1 Control of *x–y Position*

A quadcopter's underactuated and coupling dynamic features prevent direct control of the position x and y. According to certain studies, x and y positioning are the underactuated parts, while its attitudes and altitude are referred to as fully actuated parts. A quadcopter can only achieve a translational movement in the x–y axes through a change in roll and pitch angle movement, which technically represents the coupling dynamics in a quadcopter.

The position controller in Fig. 8 included conversion blocks from world frame to body frame as well as controls for a quadcopter's x and y translational movement. In a quadcopter's cascaded loop of controllers (see Fig. 8), the position controller is referred to as the outer loop. The observed yaw angle provides information to the position controller as well. The reason is that, unlike x–y position inaccuracy, which is relative to the ground, or the global reference frame, roll and pitch are relative to the quadcopter's body. Because of this, the quadcopter does not always move in the x-world direction in response (to pitch) or the y-world direction in response to roll.

$$x^G\cos\psi + y^G\sin\psi = x^B$$

$$x^G\sin\psi - y^G\cos\psi = y^B \tag{7}$$

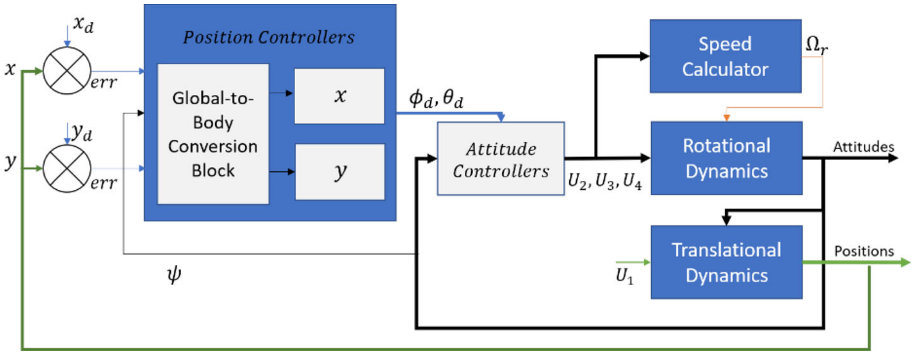
According to Eq. (7), the quadcopter will need to know its yaw angle, how it is rotated, and if roll, pitch, or a combination of the two would be required to move it to a very particular place in the room. In order to achieve the desired roll and pitch angle for the inner loop controllers, the outputs from Eq. (7) are then input into respective position controllers as shown in Eq. (8).

$$\begin{aligned}\phi_d &= K_p\left(y_d^B - y^B\right) + K_i\int_0^t\left(y_d^B - y^B\right)\mathrm{d}t + K_d\frac{\mathrm{d}\left(y_d^B - y^B\right)}{\mathrm{d}t}\\ \theta_d &= K_p\left(x_d^B - x^B\right) + K_i\int_0^t\left(x_d^B - x^B\right)\mathrm{d}t + K_d\frac{\mathrm{d}\left(x_d^B - x^B\right)}{\mathrm{d}t}\end{aligned} \tag{8}$$

3.3 PID Controller

In this paper, we look at the PID controller, which is one of the most well-known controllers in industry. The PID controller has several advantages, including minimizing downtime, rising time, and peak error, as well as smoothing output movement and reducing system overshoot. Meanwhile, altering the parameters of this type of controller is complicated. Poor performance, delayed control, and, in rare situations, system instability may come from its limited ability to alter the parameters. As a result, many types of PID controller tuning procedures necessitate close attention from the operator to determine the best set of values for those parameters in order to achieve a decent gain. The gains of the four PID controllers were tuned based on the derived mathematical model for the

Fig. 8 Block diagram for position control



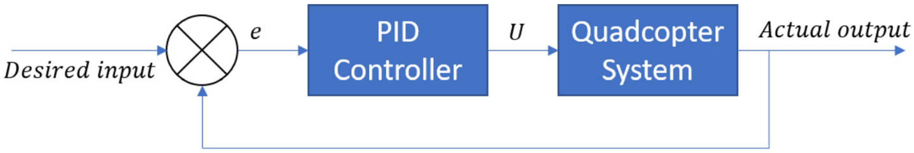


Fig. 9 General visualization of PID controllers for quadcopter UAV

quadcopter to control the altitude and attitudes and thus, the position of the quadcopter.

Equation (9) shows the equations of PID Controller for each of altitude and attitude controls. Only four PID controllers take place in the first stage of implementing conventional controller into the quadcopter system. The four controllers are for independent motions of quadcopter which are the altitude, roll, pitch and yaw. Figure 9 shows the block visualization of PID controllers.

$$\begin{aligned}
U_1 &= K_P e_z + K_I \int_0^t e_z \mathrm{d}t + K_D \frac{\mathrm{d}e_z}{\mathrm{d}t} \\
U_2 &= K_P e_\phi + K_I \int_0^t e_\phi \mathrm{d}t + K_D \frac{\mathrm{d}e_\phi}{\mathrm{d}t} \\
U_3 &= K_P e_\theta + K_I \int_0^t e_\theta \mathrm{d}t + K_D \frac{\mathrm{d}e_\theta}{\mathrm{d}t} \\
U_4 &= K_P e_\psi + K_I \int_0^t e_\psi \mathrm{d}t + K_D \frac{\mathrm{d}e_\psi}{\mathrm{d}t}
\end{aligned} \tag{9}$$

3.4 PSO–*PID Controller*

A metaheuristic is a higher-level process or heuristic used in computer science and mathematical optimization to find, generate, or select a heuristic (partial search algorithm) that may offer a good enough solution to an optimization problem, especially when there is incomplete or imperfect information or limited computing capabilities. Meta-heuristic algorithm, specifically the Swarm Intelligence (SI), is inspired by the behavior of animals' instinct to work in a group to search for food or shelter has been efficiently implemented by researchers over the years for optimization method. The primary goal of employing meta-heuristic algorithms in this controlling and optimization problem is to obtain the optimal or nearly optimal solutions through dependable optimization using a local search strategy and their random evolutionary processes [40]. Particle Swarm Optimization (PSO) algorithm is among the most highly rated in the literature on optimization has found extensive application in several fields in both the sciences and industry. It imitates the foraging and movement of a school of fishes or a flock of birds. As a nature-inspired algorithm with intelligent collective behavior of a grouped individuals with the aim to optimize a given problem, both exploration and exploitation take place in balance. The purpose of the PSO algorithm is to converge the search into all sub-optimal solutions discovered during the exploitation process, as it attempts to assess different sections of the search domain during the exploration process.

This paper intends to investigate the performance of PSO–PID algorithm by implementing different error performances in the fitness function equation. This is accordance with the previous study [40] where ITAE is implemented in the fitness equation to drive the PID control system design to an optimal state in terms of faster settling time. Step-by-step process of PSO–PID algorithm is detailed as follows:

1. Initialize the number of populations = 50, position = [0–1] and velocity = 0 of each particle, and the dimension of test cases (K_P, K_I and K_D for roll, pitch, yaw and altitude), which is 12 in total.

2. Run the quadcopter model and calculate fitness function (10) using initialized parameters, hold the *gbest* value.

$$\begin{aligned}
\text{Fitness} = {} & (\phi_e \times \alpha) + (\phi_{os} \times \alpha) + (\theta_e \times \alpha) + (\theta_{os} \times \alpha) \\
& + (\psi_e \times \alpha) + (\psi_{os} \times \alpha) + (z_e \times \alpha) + (z_{os} \times \alpha)(10)
\end{aligned}$$

where α is the weight assigned to each term. The fitness function consists of the output error performance (11) of roll, pitch, yaw and altitude (z) along with their overshoot value.

$$\begin{aligned}
&\text{IAE} = \int_0^{t_{ss}} |e(t)|\mathrm{d}t \quad \text{ISE} = \int_0^{t_{ss}} e^2(t)\mathrm{d}t \\
&\text{ITAE} = \int_0^{t_{ss}} t|e(t)|\mathrm{d}t \quad \text{ITSE} = \int_0^{t_{ss}} te^2(t)\mathrm{d}t
\end{aligned} \tag{11}$$

For each iteration,

3. Update the PID values and run the quadcopter system.

4. Calculate the fitness function (10) and update the new *gbest* and *pbest.*

5. Calculate the new position (13) and velocity (12).

$$\overrightarrow{V_i}^{t+1} = w\overrightarrow{V_i}^t + C_1 r_1\left(\overrightarrow{P_i}^t - \overrightarrow{X_i}^t\right) + C_2 r_2\left(\overrightarrow{G}^t - \overrightarrow{X_i}^t\right) \tag{12}$$

$$\overrightarrow{X_i}^{t+1} = \overrightarrow{X_i}^t + \overrightarrow{V_i}^{t+1} \tag{13}$$

6. Determine if the maximum iteration is reached. Otherwise, return to Step 3.

Equations (11) show the four performance criteria evaluated from the error signal obtained from the difference of the input reference and the system output where t_{ss} is the time when the system reached a steady state condition. IAE and

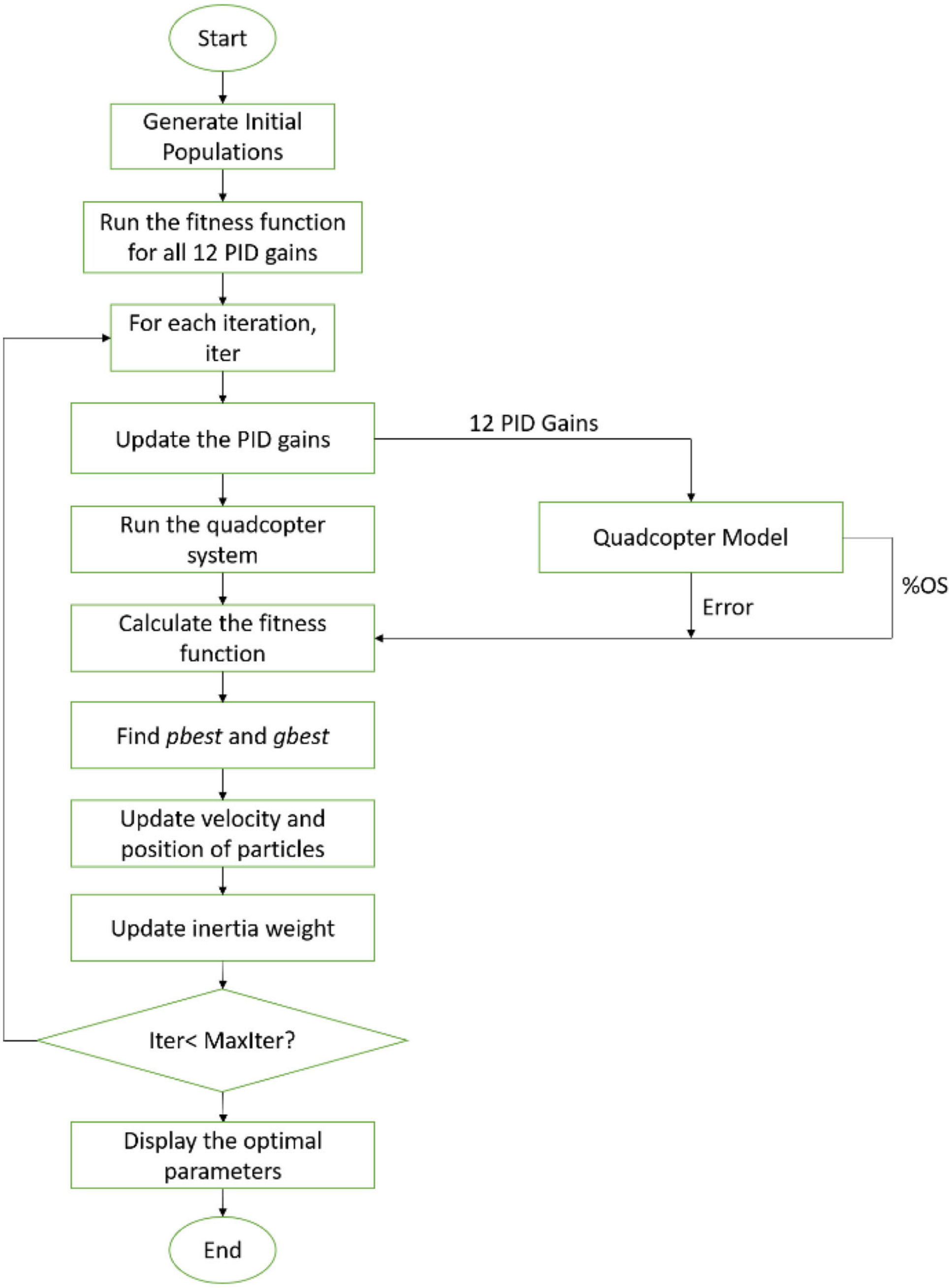

Start
Generate Initial Populations
Run the fitness function for all 12 PID gains
For each iteration, iter
Update the PID gains
12 PID Gains
Quadcopter Model
%OS
Error
Run the quadcopter system
Calculate the fitness function
Find *pbest* and *gbest*
Update velocity and position of particles
Update inertia weight
Iter< MaxIter?
Display the optimal parameters
End

Fig. 10 Flowchart for PSO algorithm

ISE are independent of time but hypothetically will result in small overshoot with longer settling time [41]. ITAE and ITSE come into action by including the late error values into account which is expected to be better than IAE and ISE [41]. Figures 10 and 11 depict the PSO flowchart and PSO–PID block diagram, respectively.

4 Simulation Results

The quadcopter system verification simulation is run on a personal computer with 8 GB RAM, which is sufficient for running the MATLAB/Simulink. The results of quadcopter simulation using PID controller optimized by PSO algorithm are being presented in this section. Altitude and attitude stabilization and trajectory tracking using circular and lemniscate pattern are tested and compared between the four performance criteria based on the system error.

4.1 Parameter Setting

The parameters setting for PSO algorithm are stated in Table 2. The number of population and iteration were changed

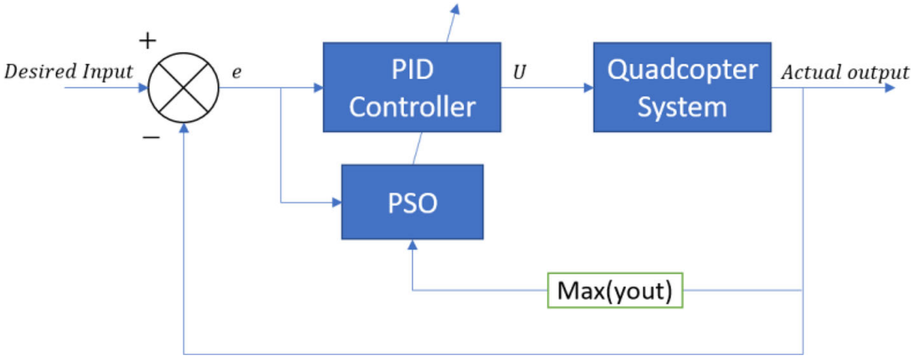


Fig. 11 Block diagram of PSO–PID

Table 2 Parameter setting for PSO algorithm

Parameter	Value
Number of populations	50
Number of dimensions	12
Inertia weight	1
Cognitive parameter	2.0
Social parameter	2.0
Lower bound	0
Upper bound	50
Number of iterations	50

Table 3 ZN-PID gains

Method	ZN		
Parameter	K_p	K_i	K_d
Roll	6.3	0.01031	0.71397
Pitch	1.1	1.25655	0.27163
Yaw	1.2	1.25655	0.27163
Thrust	8.32	8.79943	2.86412

Table 4 The PID gains obtained from each error performance with respective fitness cost

Method	IAE			
Parameter	K_p	K_i	K_d	Fitness cost
Roll	50	5.4441	2.8125	0.0631
Pitch	50	26.5817	10.5201	0.06347
Yaw	50	27.2595	1.4565	0.05619
Thrust	50	36.0432	9.7771	0.2514
ISE				
Roll	35.4736	6.7187	1.6262	0.01102
Pitch	12.9577	2.4879	4.4352	0.002691
Yaw	50	3.7975	1.3780	0.01704
Thrust	50	32.7011	8.5893	0.1244
ITAE				
Roll	50	50	2.3316	0.03659
Pitch	6.0914	5.6761	2.5822	0.02963
Yaw	50	50	1.2126	0.02541
Thrust	50	30.1158	9.1505	0.03926
ITSE				
Roll	50	4.1531	3.1645	0.0007913
Pitch	17.4897	3.9749	3.9290	0.0002231
Yaw	33.7736	2.4542	0.9508	0.0006625
Thrust	50	30.3777	8.8697	0.01136

several times, while the others were fixed until satisfactory results were obtained.

4.2 Optimal PID Gains

The optimal PID gains of K_p, K_i and K_d for roll, pitch, yaw and altitude (z) that are obtained from PSO–PID simulations are tabulated based on error performance in Table 4. PID gains obtained from ZN-PID calculation are recorded in Table 3.

The fitness cost values obtained in Table 4 shows the reliability of using PSO–PID in controlling a quadcopter. IAE produces the highest fitness cost values for roll, pitch, yaw and thrust, compared to others. For every index performance, fitness cost value for thrust is the highest compared to roll, pitch and yaw. For ITAE, there is an insignificant difference between the fitness cost value of pitch and fitness cost value of yaw. The fitness cost for ISE is comparable to ITAE with not so much difference between those two performances. ITSE is reported to produce the lowest fitness values among other error performances for roll, pitch, yaw and thrust. It is considered reliable to be used for controlling a quadcopter more efficiently.

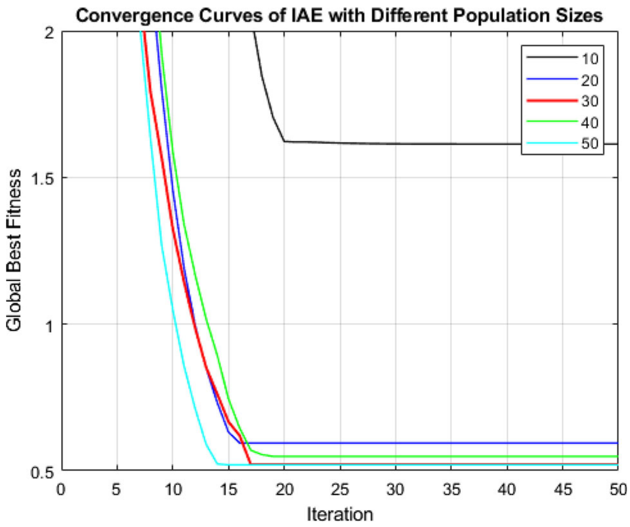


Fig. 12 Convergence curves of IAE with different population sizes

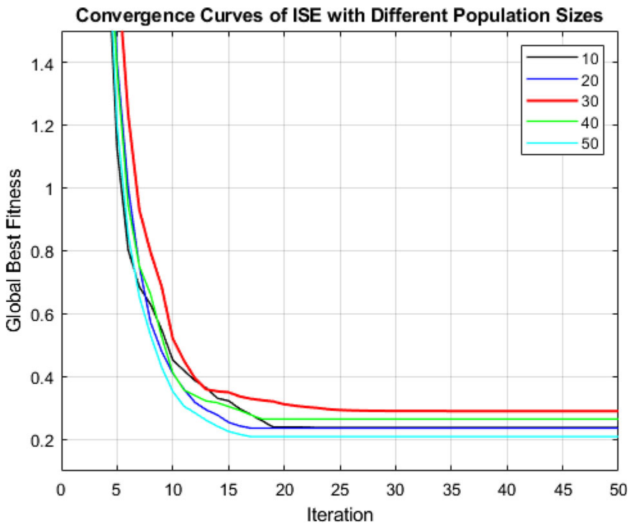


Fig. 13 Convergence curves of ISE with different population sizes

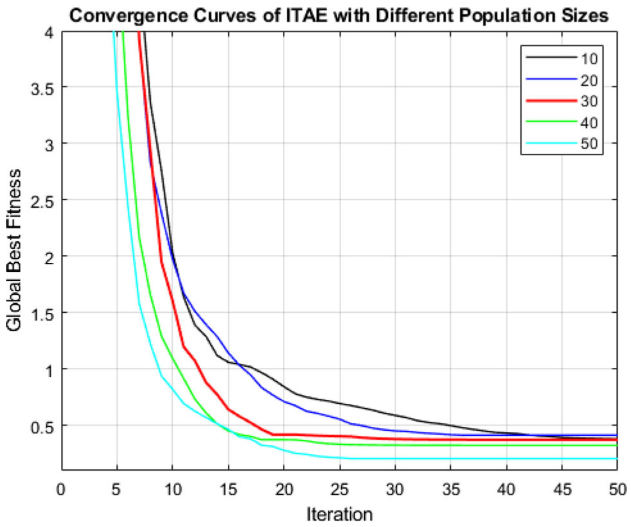


Fig. 14 Convergence curves of ITAE with different population sizes

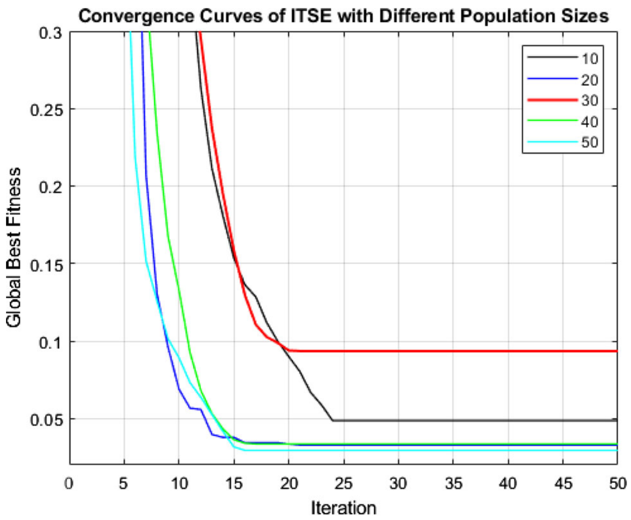


Fig. 15 Convergence curves of ITSE with different population sizes

4.3 Convergence Results

In this subsection, global best fitness convergence results are presented for each of error performances IAE, ISE, ITAE and ITSE. The convergence curves are presented using different population sizes to evaluate the best number of populations to be used to compare between all performances. Hypothetically, the lowest fitness cost approaching zero is the best outcome with the lowest error produced from quadcopter system. Figures 12, 13, 14, 15 show the convergence characteristic curves of the fitness cost of the PSO–PID quadcopter system using IAE, ISE, ITAE and ITSE, respectively.

In Fig. 12, there is a huge difference between population size of 10 and the other population sizes which clearly shows that it is not optimal to use IAE with population size of 10. Each figure proves that the highest population size of 50 gives the best fitness cost value converging toward zero and the fastest to converge, compared to smaller population sizes. However, this does not guarantee that further increase in population sizes will improve the fitness cost value and might cause a slight turndown in terms of fitness function and system performance. Among all the four index performances, ITSE gives the best fitness cost value with the nearest toward zero. Hypothetically, a more optimal fitness cost value will be achieved by increasing the number of populations. However, it does not comply to all situations such as in Fig. 15, population size of 30 produce a higher fitness cost value than population size of 10 and same goes to Fig. 13. These curves show the fitness cost values determined by using fitness function from equation. The significant convergence using PSO

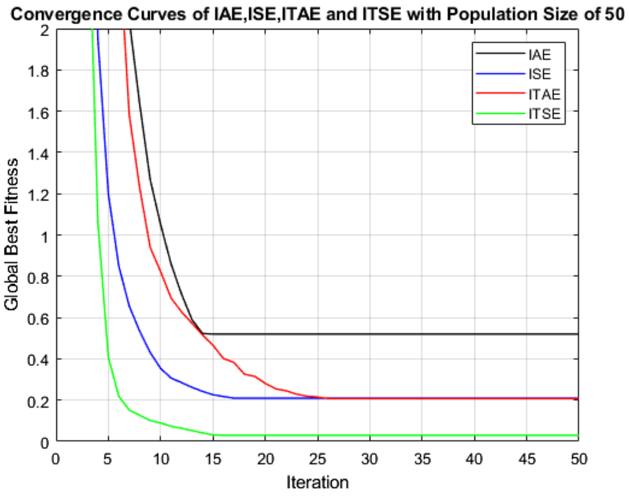


Fig. 16 Convergence curves of IAE, ISE, ITAE and ITSE with population size of 50

shown by each index performance is important to determine the best PSO parameter for optimizing the PID controller of a quadcopter system.

Fig. 16 shows the comparison of convergence curves of each index performance for population size of 50. The convergence curves prove that acceptable results can be achieved using PSO–PID for quadcopter system. By referring to Fig. 16, ITSE index performance helps PSO algorithm to converge the fastest toward the smallest fitness value compared to IAE, ISE and ITAE.

4.4 Altitude and Attitude Tracking

Figure 17 shows the simulation results of quadcopter system using different index performances for roll, pitch, yaw and altitude tracking. PSO–PID is implemented specifically for altitude and attitude stabilization and thus, helps to produce a better result for position tracking. Altitude and attitude tracking are using step signals for evaluations.

Figure 17a–d represents the simulated results of altitude and attitude stabilization for IAE, ISE, ITAE and ITSE. It shows the quality of each index performance in controlling the roll, pitch, yaw and altitude of a quadcopter. All index performances lead the quadcopter movement toward a zero steady-state error with different settling time, rise time and overshoot. As shown in Fig. 17, ITSE could provide the best performance if compared to IAE, ISE and ITAE. Overall, all index performances have measure-proved that PSO–PID controller can do escape maneuver of the quadcopter with each has their own priority measurement.

Table 5 presents the overshoot, rise time and settling time obtained after running PSO–PID controllers for quadcopter using IAE, ISE, ITAE and ITSE error performances. The overshoot obtained from all performances are acceptable and satisfying and this is because overshoot obtained from the quadcopter system is included in the fitness function in equation which is targeted to be reduced toward zero. The overshoot for roll, pitch and thrust from ISE is nearly the same but yaw has a lower overshoot. ITSE produces the lowest overshoot for roll, yaw and thrust compared to other performances but the overshoot value for pitch is the second highest among those. ITSE produces the overall longest settling time while ITAE produces the overall shortest settling time. The overall rise time from IAE is comparable to the overall rise time from ISE with insignificant difference. The overall rise time for all performances is acceptable and reliable to be implemented in a quadcopter system. By using the obtained optimal PSO–PID gains, a comparison of quadcopter's performance is being made with PID controller of ZN-PID [34] gains. It was mentioned before in this paper that ZN-PID may cause significant losses and damage to the system when facing uncertainties. The performance of roll motion is evaluated in this paper with inclusion of wind gust model as disturbance (as shown in Fig. 18) for the quadcopter system and the results are as shown in Fig. 19. ZN-PID [6.3, 0.01031149302, 0.713965] has some difficulties adapting with the disturbance while PSO–PID [50, 4.1531, 3.1645] that was obtained from ITSE evaluation shows reliability in maneuvering the quadcopter very well through the wind disturbance. Compared to the ZN-PID [34], the proposed PSO–PID can handle the disturbance better. (Fig. 19).

4.5 Position Tracking (Circular Trajectory)

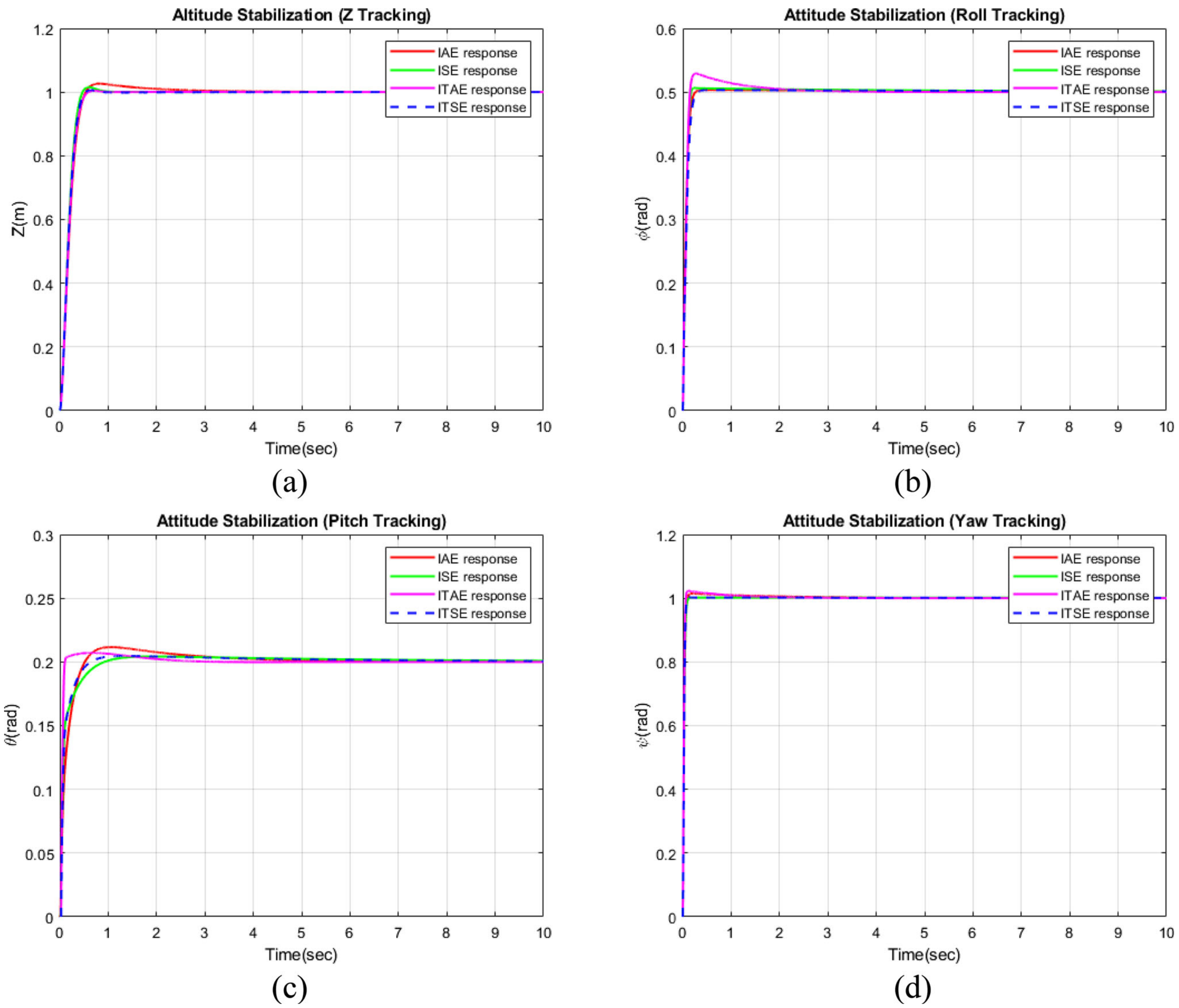
Figure 20 presents the circular trajectory performed by different PID gains obtained from IAE, ISE, ITAE and ITSE error performances. Starting from the ground at (0,0,0), the circular trajectory is desired to be performed at an altitude of 1 and with radius of 1.5.

Each IAE, ISE, ITAE and ITSE performed well by stabilizing its position based on the desired trajectory. ISE produces the best result in circular trajectory tracking with undershoot at the starting point. ITAE and ITSE require quite some moment to settle down on the desired track with high overshoot and zero undershoot. Table 6 represents the comparison of error performances for circular trajectory in terms of x–y position and roll–pitch.

4.6 Position Tracking (Lemniscate Trajectory)

A lemniscate trajectory tracking was also performed using different PID gains obtained using IAE, ISE, ITAE and ITSE error performances and is presented in Fig. 21. The starting point is at (0,0,0) going up to the altitude of 1 with the scale of $x = (-\ 1.5\ 1.5)$ and $y = (-\ 1.5\ 1.5)$.

Simulation shows that all error performances succeeded in following the desired lemniscate trajectory despite the time



(a)

(b)

(c)

(d)

Fig. 17 Simulated step responses for each index performances **a** *Z* tracking **b** roll tracking **c** pitch tracking **d** yaw tracking

Table 5 The overshoot, rise time and settling time produced using IAE, ISE, ITAE and ITSE

Method	IAE			ISE		
Evaluation	%OS	Rise Time (s)	Settling time (s)	%OS	Rise time (s)	Settling time (s)
Roll	0.505	0.122	29.534	1.531	0.107	25.461
Pitch	6.989	0.369	10.233	1.511	0.339	23.909
Yaw	1.531	0.050	6.353	0.504	0.047	18.962
Thrust	3.646	0.331	5.965	1.531	0.278	1.794

Method	ITAE			ITSE		
Evaluation	%OS	Rise time (s)	Settling time (s)	%OS	Rise time (s)	Settling time (s)
Roll	5.851	0.104	6.353	0.505	0.145	50.194
Pitch	0.887	0.076	6.450	2.553	0.243	20.514
Yaw	2.577	0.038	3.928	0.505	0.045	19.447
Thrust	0.505	0.306	4.316	0.505	0.295	5.480

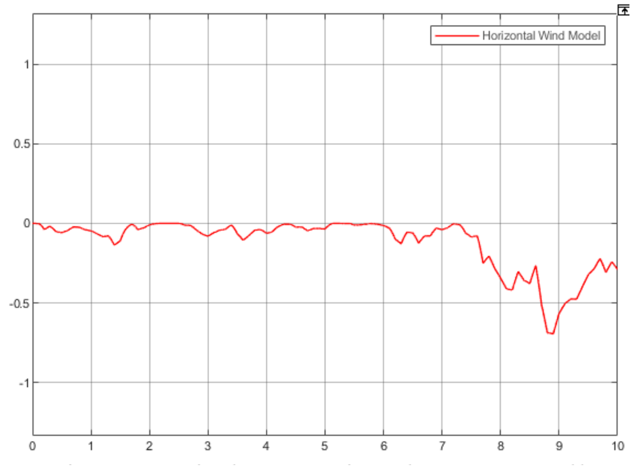


Fig. 18 Wind gust disturbance at roll motion

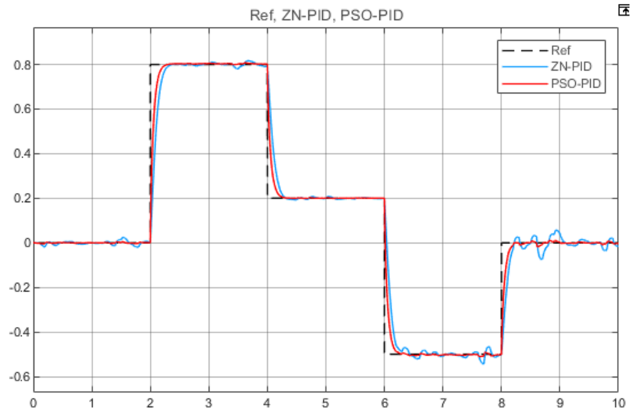


Fig. 19 Roll motion of quadcopter with disturbances

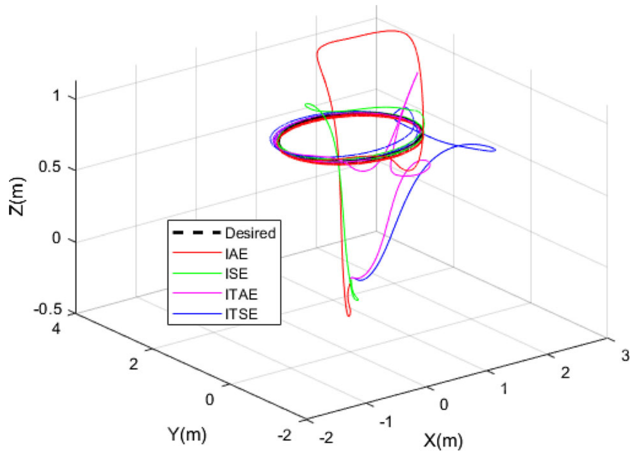


Fig. 20 Circular trajectory responses by IAE, ISE, ITAE and ITSE

taken to settle on the desired track. In this simulation, ITSE produces the best result with faster settling time, lowest overshoot and zero undershoot, compared to IAE, ISE and ITAE.

Table 6 Comparison of error performances for x–y position and roll–pitch in Circular Trajectory

Circular	ϕ	θ	X	Y
IAE	0.4824	0.4221	6.546	0.745
ISE	0.3416	0.02599	0.5784	0.4142
ITAE	0.5063	6.193	10.66	6.175
ITSE	0.01602	0.09017	2.423	2.135

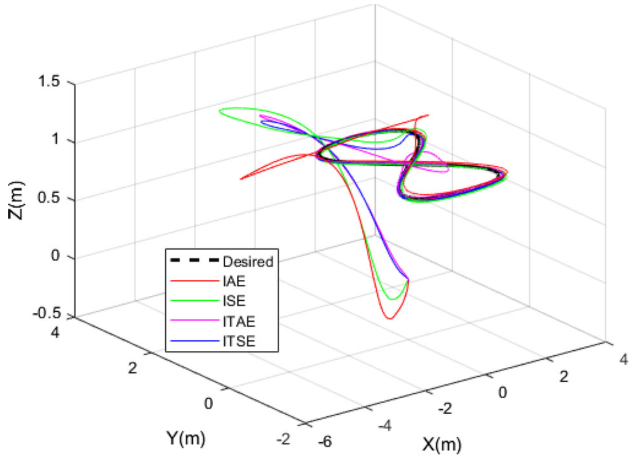


Fig. 21 Lemniscate trajectory responses by IAE, ISE, ITAE and ITSE

Table 7 Comparison of error performances for x–y position and roll–pitch in Lemniscate Trajectory

Lemniscate	ϕ	θ	X	y
IAE	0.4195	0.7208	4.014	4.598
ISE	0.2136	0.2216	3.137	1.388
ITAE	2.68	17.34	11.36	11.87
ITSE	0.1169	0.6537	3.171	0.9385

Table 7 represents the comparison of error performances for lemniscate trajectory in terms of x–y position and roll–pitch.

5 Conclusions

This paper has clearly presented the dynamic equations of quadcopter model, the PID controller equations, the PSO algorithm with fitness function equation and the equations for error performances. The results obtained from each error performance for PSO–PID controller in quadcopter system are also presented and evaluated. It is found that the results of overall error performances are acceptable and can escape maneuvering. ITAE produces the best overall result of overshoot, rise time and settling time for roll, pitch, yaw and thrust, compared to other performances. However, ITSE provides the lowest fitness cost values for roll, pitch, yaw and thrust if compared with other performances. It is also found that by increasing the number of populations, the lower the fitness values obtained but for some populations sizes, it does not conform to the idea. It is also does not guarantee that by further increasing the size of population, the fitness values will decrease any further. For circular trajectory tracking, ISE shows the best circular tracking compared to IAE, ITAE and ITSE with small undershoot. Both ITAE and ITSE show zero undershoot for both circular and lemniscate trajectory tracking. Wind disturbance is included in the roll motion to compare the performance of PSO–PID and ZN-PID, and it was seen that by using PID gains obtained from PSO–PID, the escape maneuver of roll motion of a quadcopter is much better than ZN-PID. However, the PSO–PID algorithm took so much time to finish running for 50 iteration and 50 population (optimal chosen condition). Moreover, the PID gains obtained from PSO–PID are offline and fixed, which is unfavorable for a real-time flight of quadcopter. In future research, neural network will be considered to further improve the optimal tuning of PID controller for an improved real-time escape maneuver of a quadcopter UAV.

Acknowledgements The authors would like to thank the Ministry of Higher Education (MOHE) through Fundamental Research Grant Scheme (FRGS/1/2021/TK0/UTM/02/56) and Universiti Teknologi Malaysia through UTMFR (Q.J130000.3823.22H67) for supporting this research.

References

1. MohdBasri, M.A.; Husain, A.R.; Danapalasingam, K.A.: Enhanced backstepping controller design with application to autonomous quadrotor unmanned aerial vehicle. J. Intell. Robot. Syst. Theory Appl. **79**(2), 295–321 (2015). https://doi.org/10.1007/s10846-014-0072-3
2. He, Z.; Zhao, L.: A simple attitude control of quadrotor helicopter based on Ziegler-Nichols rules for tuning PD parameters. Sci. World J. (2014). https://doi.org/10.1155/2014/280180
3. Siti, I.; Mjahed, M.; Ayad, H.; El Kari, A.: New trajectory tracking approach for a quadcopter using genetic algorithm and reference model methods. Appl. Sci. (2019). https://doi.org/10.3390/app9091780
4. Wang, P.; Man, Z.; Cao, Z.; Zheng, J.; Zhao, Y.: Dynamics modelling and linear control of quadcopter. Int. Conf. Adv. Mechatron. Syst. ICAMechS **0**, 498–503 (2016). https://doi.org/10.1109/ICAMechS.2016.7813499
5. Salih, A.L.; Moghavvemi, M.; Mohamed, H.A.F.; Gaeid, K.S.: Modelling and PID controller design for a quadrotor unmanned air vehicle. In: 2010 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR) 2010—Proc., vol. 1, pp. 74–78 (2010). https://doi.org/10.1109/AQTR.2010.5520914
6. Jiao, Q.; Liu, J.; Zhang, Y.; Lian, W.: Analysis and design the controller for quadrotors based on PID control method. In: Proceedings—2018 33rd Youth Academic Annual Conference of Chinese Association of Automation, YAC 2018, pp. 88–92. IEEE (2018). https://doi.org/10.1109/YAC.2018.8406352
7. Tang, Y.R.; Li, Y.; Ma, T.: Design, implementation and control of a small-scale UAV quadrotor. In: Proceeding of the 11th World Congress on Intelligent Control and Automation, vol. 2015-March, no. March, pp. 2364–2369 (2015). https://doi.org/10.1109/WCICA.2014.7053091
8. Basri, M.A.M.; Husain, A.; Danapalasingam, K.: Stabilization and trajectory tracking control for underactuated quadrotor helicopter subject to wind-gust disturbance. Sadhana Acad. Proc. Eng. Sci. **40**(5), 1531–1553 (2015). https://doi.org/10.1007/s12046-015-0384-4
9. Thu, K.M.; Gavrilov, A.I.: Designing and modeling of quadcopter control system using L1 adaptive control. Procedia Comput. Sci. **103**(October 2016), 528–535 (2017). https://doi.org/10.1016/j.procs.2017.01.046
10. Palunko, I.; Fierro, R.: Adaptive control of a quadrotor with dynamic changes in the center of gravity. In: IFAC Proceedings Volumes (IFAC-PapersOnline), IFAC, pp. 2626–2631 (2011). https://doi.org/10.3182/20110828-6-IT-1002.02564
11. Xu, R.; Özgüner, Ü.: Sliding mode control of a quadrotor helicopter. In: Proceedings of the 45th IEEE Conference on Decision and Control, pp. 4957–4962 (2006). https://doi.org/10.1109/cdc.2006.377588
12. Bouadi, M.T.H.; Bouchoucha, M.: Modelling and stabilizing control laws design based on sliding mode for an UAV type-quadrotor. Eng. Lett. **15**(2), 15–24 (2007)
13. Jafar, H.; Zareh, M.; Roshanian, J.; Nikkhah, A.: An optimal guidance law applied to quadrotor using LQR method. Trans. Jpn. Soc. Aeronaut. Space Sci. **53**(179), 32–39 (2010). https://doi.org/10.2322/tjsass.53.32
14. Okyere, E.; Bousbaine, A.; Poyi, G.T.; Joseph, A.K.; Andrade, J.M.: LQR controller design for quad-rotor helicopters. J. Eng. **2019**(17), 4003–4007 (2019). https://doi.org/10.1049/joe.2018.8126
15. Fessi, R.; Bouallègue, S.: Modeling and optimal LQG controller design for a quadrotor UAV. In: 3rd International Conference on Automation, Control, Engineering and Computer Science, pp. 264–270 (2016)
16. Fakurian, F.; Menhaj, M.B.; Mohammadi, A.: Design of a fuzzy controller by minimum controlling inputs for a quadrotor. In: 2014 2nd RSI/ISM International Conference on Robotics and Mechatronics, ICRoM 2014, pp. 619–624 (2014). https://doi.org/10.1109/ICRoM.2014.6990971
17. Pedro, J.O.; Kala, P.J.**:** Nonlinear control of quadrotor UAV using Takagi-Sugeno fuzzy logic technique. In: 2015 10th Asian Control Conference: Emerging Control Techniques for a Sustainable World, ASCC 2015. IEEE (2015). https://doi.org/10.1109/ASCC.2015.7244739
18. Andrien, A.; Kremers, D.; Kooijman, D.; Antunes, D.: Model predictive tracking controller for quadcopters with setpoint convergence guarantees. In: Proceedings of the American Control Conference, pp. 3205–3210 (2020). https://doi.org/10.23919/ACC45564.2020.9147947
19. Lyu, Y.; Lai, G.; Chen, C.; Zhang, Y.: Vision-based adaptive neural positioning control of quadrotor aerial robot. IEEE Access **7**, 75018–75031 (2019). https://doi.org/10.1109/ACCESS.2019.2920716
20. Rios, H.; Falcon, R.; Gonzalez, O.A.; Dzul, A.: Continuous sliding-mode control strategies for quadrotor robust tracking: real-time application. IEEE Trans. Ind. Electron. **66**(2), 1264–1272 (2019). https://doi.org/10.1109/TIE.2018.2831191
21. Eltayeb, A.; ad Rahmat, M.F.; Basri, M.A.M.; Eltoum, M.A.M.; El-Ferik, S.: An improved design of an adaptive sliding mode controller for chattering attenuation and trajectory tracking of the quadcopter UAV. IEEE Access **8**, 205968–205979 (2020). https://doi.org/10.1109/ACCESS.2020.3037557

22. Reizenstein, A.: Position and trajectory control of a quadcopter using PID and LQ controllers. Linköping Univ., p. 81 (2017), [Online]. Available: http://urn.kb.se/resolve?urn=urn%3Anbn%3Ase%3Aliu%3Adiva-139498
23. Abdelhay, S.; Zakriti, A.: Modeling of a quadcopter trajectory tracking system using PID controller. Procedia Manuf. (2019). https://doi.org/10.1016/j.promfg.2019.02.253
24. González-Vázquez, S.; Moreno-Valenzuela, J.: A new nonlinear PI/PID controller for quadrotor posture regulation. In: 2010 IEEE Electronics, Robotics and Automotive Mechanics Conference CERMA 2010, pp. 642–647 (2010). https://doi.org/10.1109/CERMA.2010.78
25. Moreno-Valenzuela, J.; Perez-Alcocer, R.; Guerrero-Medina, M.; Dzul, A.: Nonlinear PID-Type controller for quadrotor trajectory tracking. IEEE/ASME Trans. Mechatron. **23**(5), 2436–2447 (2018). https://doi.org/10.1109/TMECH.2018.2855161
26. Lopez-Sanchez, I.; Montoya-Cháirez, J.; Pérez-Alcocer, R.; Moreno-Valenzuela, J.: Experimental parameter identifications of a quadrotor by using an optimized trajectory. IEEE Access **8**, 167355–167370 (2020). https://doi.org/10.1109/ACCESS.2020.3023643
27. Braik, M.; Sheta, A.; Al-Hiary, H.: A Novel Meta-Heuristic Search Algorithm for Solving Optimization Problems: Capuchin Search Algorithm, Vol. 33, p. 7. Springer, London (2021) https://doi.org/10.1007/s00521-020-05145-6
28. Wang, C.-F.; Liu, K.: A novel particle swarm optimization algorithm for global optimization. Comput. Intell. Neurosci. **2016**, 1–9 (2016). https://doi.org/10.3390/math6120287
29. Noordin, A.; Basri, M.A.M.; Mohamed, Z.; Abidin, A.F.Z.: Modelling and PSO fine-tuned PID control of quadrotor UAV. Int. J. Adv. Sci. Eng. Inf. Technol. **7**(4), 1367–1373 (2017). https://doi.org/10.18517/ijaseit.7.4.3141
30. Madruga, S.P; de Holanda Barreto Martins Tavares, A; Basso, G.F.; do Nascimento, T P.; Brito, A.: A PSO-based tuning algorithm for quadcopter controllers. In: Proceedings XXII Congresso Brasileiro de Automática (2018). https://doi.org/10.20906/cps/cba2018-0064
31. Rendón, M.A.; Martins, F.F.: Path following control tuning for an autonomous unmanned quadrotor using particle swarm optimization. IFAC-PapersOnLine **50**(1), 325–330 (2017). https://doi.org/10.1016/j.ifacol.2017.08.054
32. Cárdenas, J.A.; Carrero, U.E.; Camacho, E.C.; Calderón, J.M.: Optimal PID ø axis control for UAV quadrotor based on multi-objective PSO. IFAC-PapersOnLine **55**(14), 101–106 (2022). https://doi.org/10.1016/j.ifacol.2022.07.590
33. Keskin, B.; Keskin, K.: Position control of quadrotor using firefly algorithm. El-Cezeri J. Sci. Eng. **9**(2), 554–566 (2022). https://doi.org/10.31202/ecjse.975718
34. Canal, I.P.; Martin, M.; Reimbold, P.; De Campos, M.: Ziegler–Nichols customization for quadrotor attitude control under empty and full loading conditions (2020). https://doi.org/10.32604/cmes.2020.010741
35. Sonugür, G.; Gökçe, C.O.; Koca, Y.B.; Inci, ŞS.; Keleş, Z.: Particle swarm optimization based optimal Pid controller for quadcopters. C. R. L'Academie Bulg. Sci. **74**(12), 1806–1814 (2021). https://doi.org/10.7546/CRABS.2021.12.11
36. Domingos, D.; Camargo, G.; Gomide, F.: Autonomous fuzzy control and navigation of quadcopters. IFAC-PapersOnLine (2016). https://doi.org/10.1016/j.ifacol.2016.07.092
37. S. PhD T. Bouabdallah, Design and Control of Quadrotors with Application to Autonomous Flying, vol. 3727, no. Lausanne, EPFL (2007), [Online]. Available: https://infoscience.epfl.ch/record/95939/files/EPFL_TH3727.pdf
38. Iyer, A.; Bansal, H.O.: Modelling, simulation, and implementation of PID controller on quadrotors. In: 2021 International Conference on Computer Communication and Informatics, ICCCI 2021 (2021). https://doi.org/10.1109/ICCCI50826.2021.9402301
39. Voos, H.: Nonlinear control of a quadrotor micro-UAV using. In: IEEE International Conference on Mechatronic, no. April, pp. 4–9 (2009)
40. Sheta, A.; Braik, M.; Maddi, D.R.; Mahdy, A.; Aljahdali, S.; Turabieh, H.: Optimization of PID controller to stabilize quadcopter movements using meta-heuristic search algorithms. Appl. Sci. (2021). https://doi.org/10.3390/app11146492
41. Sahib, M.A.; Ahmed, B.S.: A new multiobjective performance criterion used in PID tuning optimization algorithms. J. Adv. Res. **7**(1), 125–134 (2016). https://doi.org/10.1016/j.jare.2015.03.004