KFUPM
Journal of Undergraduate Research International

Diamond Open Access 74

Article Control and Instrumentation Engineering

Simulation-Based Modeling and PID control optimization for quadcopter UAV using genetic algorithm

Sajjad Al Mahmud and Ahmed Eltayeb*

Cite https://doi.org/10.64589/juri/221316

Submitted: November 16, 2025 **Revised:** February 25, 2026 **Accepted:** March 01, 2026

ABSTRACT

Quadcopters are widely used in modern applications; however, maintaining a stable hover and accurate path following remains challenging owing to their nonlinear dynamics. In this study, an efficient simulation-based control framework was developed, in which a nonlinear model was derived using the Newton–Euler method and subsequently linearized via feedback linearization. In MATLAB/Simulink, two proportional–integral–derivative tuning methods—manual tuning and a genetic algorithm (GA) with a population size of 30 and up to 30 generations—were employed to design the controller. Manual tuning achieved an integral square error (ISE) value of 0.03316, whereas the GA-based approach generated an ISE of 0.030056, corresponding to an average improvement of 13.9%. Furthermore, the GA-based tuning produced a smoother system response with reduced overshoot, thereby enhancing the quadcopter stability and trajectory tracking, and contributing to more reliable real-world flight performance.

Keywords: quadcopter UAV, PID controller, genetic algorithm (GA), feedback Linearization, trajectory tracking

1. INTRODUCTION

A quadcopter is an unmanned aerial vehicle (UAV) equipped with four rotors and is widely used in both military and civilian applications. Several methods have been employed to model quadcopters, including the Newton–Euler method, the Euler–Lagrange formulation, and aerodynamic modeling [1,2]. As a quadcopter operates with six degrees of freedom (6-DoF), stabilizing the system remains a challenging task. Its motion can be categorized into translational movement along the x, y, and z axes, and rotational movement about roll (ϕ), pitch (θ), and yaw (ψ) axes. The aerodynamic forces acting on a quadcopter during forward flight significantly influence its stability, as reported in [3]. A proportional-derivative (PD) controller combined with a feedback linearization strategy (FBL) has been used to control and stabilize quadcopter systems while achieving trajectory tracking [4]. A comprehensive analysis of quadcopter mathematical models, including both nonlinear and linear representations, forms the foundation for developing and implementing effective control strategies. Among various control techniques, the proportional–integral–derivative (PID) controller is most widely used for quadcopter systems [5,6]. The FBL technique transforms a nonlinear quadcopter system into an equivalent linear system, thereby simplifying controller design. Furthermore, FBL has been integrated with linear quadratic regulator (LQR) control to achieve optimal stabilization of the Euler angles and to minimize trajectory tracking [7].

A comparison of LQR, PID, and LQR-PID controllers in terms of vertical position and vertical speed, based on system performance characteristics such as overshoot, settling time, and rise time, indicates that the PID controller provides superior performance in achieving the optimal vertical speed. However, the LQR controller demonstrates the best capability for achieving optimal stability 8. Most existing studies treat nonlinear compensation and PID tuning as separate problems. In this study, feedback linearization and genetic algorithm (GA)-based PID tuning were combined and evaluated within a unified framework, with a direct comparison to manual tuning.

To enhance trajectory tracking and address hover instability, an LQR controller combined with Lyapunov theory was applied to achieve optimal longitudinal stability of the quadcopter, while minimizing a cost function associated with controller energy consumption [9]. The results reported in [9] demonstrate that the LQR controller effectively stabilizes the system and reduces energy consumption, even in the presence of disturbances.

Manual tuning of the PID parameters has been applied based on simulation results to improve system speed and accuracy [10]. The H-shaped quadcopter is modeled using the Newton–Euler method, as described below:

An additional strategy for optimizing PID parameters in H-shaped quadcopters is the use of a GA, which also aims to minimize the cost function [11]. In [12], PID parameters were optimized using two approaches: GA and ant colony (ACO) method. The study demonstrates the capability of GA to effectively select optimal parameters, resulting in improved settling time for Euler angles when using 60 generations and a population size of 50. However, the results in [12] indicate that ACO achieves a shorter settling time compared to GA. This comparison suggests that,

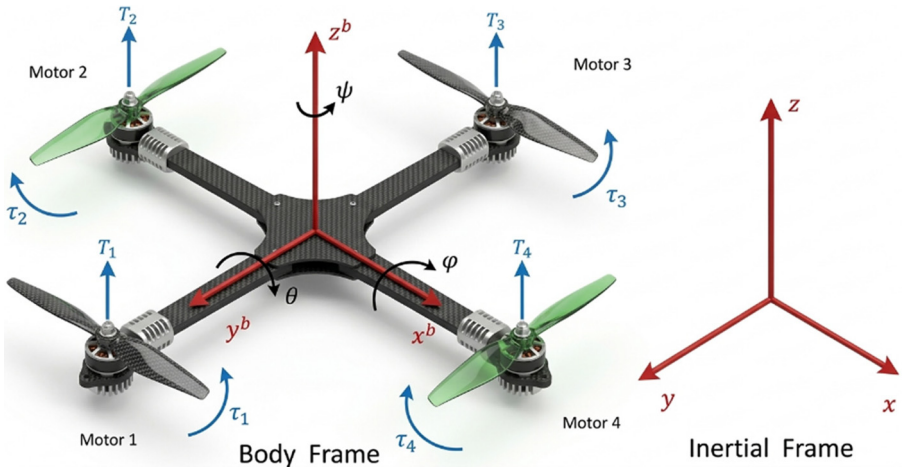


Figure 1. Quadcopter freebody diagram [13]

while GA is effective due to its global search capability, ACO may provide better performance in certain scenarios owing to its local search efficiency. In this context, the PID controller is commonly selected due to its simplicity and ease of implementation. By contrast, the LQR controller requires full-state measurements.

Many control techniques have been proposed for quadcopter systems. However, most existing studies either focus only on complex controller designs or do not provide a unified comparison between manual tuning and intelligent optimization methods. Therefore, developing a practical and efficient approach to improve stability and trajectory tracking remains a challenge. In this study, a GA is employed for PID tuning due to its capability for global search and its effectiveness in handling nonlinear optimization problems. The novelty of this study lies in the integration of feedback linearization with GA-optimized PID control, along with a direct comparison to manual tuning. The main objective is to enhance quadcopter stability and tracking performance through an efficient simulation-based control strategy.

This study contributed to the field of UAV control. This represents an improvement in the stability of the quadcopter by implementing a control strategy. Newton–Euler equations were applied to model the nonlinear dynamics of the system, as shown in Section 2, and linearized by FBL to enable a state-space representation suitable for controller synthesis.

This study focused on validating a UAV system by applying simulation methods to ensure the validity of the derived mathematical models and control strategies. The primary goal is to design a UAV system that performs perfectly in real life. A PID controller was designed and tested under different conditions to optimize the trajectory tracking and system robustness.

This study evaluated two tuning methods: manual tuning and GA tuning. The simulation results obtained under different situations show a smoother transient response of the GA and an enhancement of the hover stability and trajectory tracking precision.

This study seeks to provide an understanding of mathematical modeling, control design, and optimization methodologies to optimize quadcopter flying performance. Using traditional control methods with optimization methodologies, the proposed system aims to achieve precise control and maximum responsiveness to changing conditions.

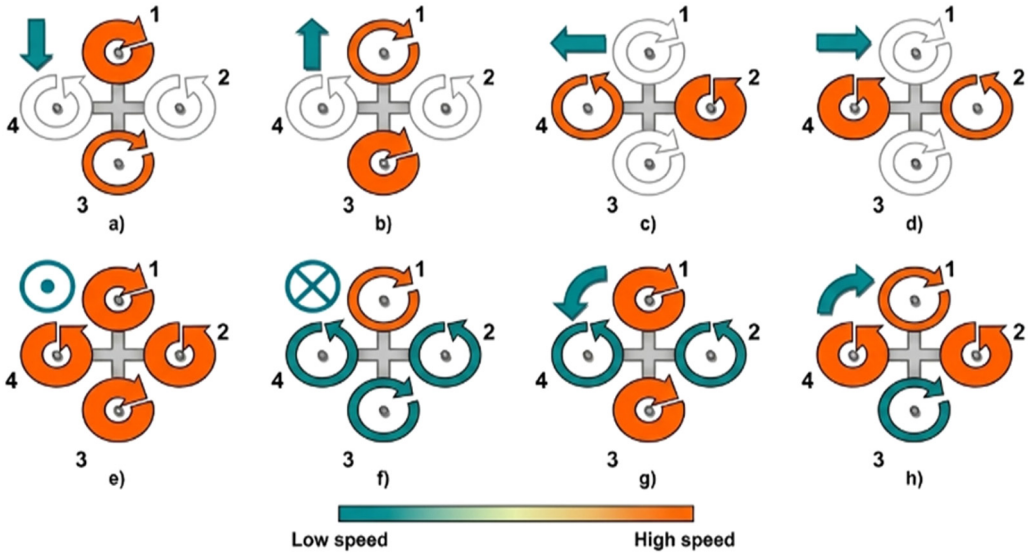


Figure 2. Quadcopter movements [15]

Table 1. *Physical parameters of the quadcopter* [14–17]

Parameter	Symbol	Value	Unit
Total mass	m	1.2	kg
Arm length (center to motor)	L	0.23	m
Moment of inertia (x-axis)	I_x	0.021	$kg \cdot m^2$
Moment of inertia (y-axis)	I_y	0.021	$kg \cdot m^2$
Moment of inertia (z-axis)	I_z	0.037	$kg \cdot m^2$
Thrust coefficient	b	2.98×10^{-6}	$N \cdot s^2$
Drag coefficient	d	1.14×10^{-7}	$N \cdot m \cdot s^2$
Gravitational acceleration	g	9.81	m/s^2

The main contributions of this study are the development of a quadcopter model by applying Newton–Euler and linearizing it using FBL, tuning PID gains using GA and manual tuning using the same Simulink block diagram, and comparing them based on integral square error (ISE) values. The remainder of this paper is organized into five sections. Section 2 describes the physical structure of a quadcopter. Section 3 begins by presenting nonlinear modeling and then linear modeling. The optimization of the best PID controller to improve the performance of the system is discussed in Section 4. The results are presented and discussed in Section 5 after a simulation using MATLAB and Simulink. Finally, a summary is presented in Section 6.

2. QUADCOPTER MODELING

The modeling of a quadcopter requires an analysis of the dynamic torques and forces of vehicle flight. The quadcopter moves freely in 6-DoF space and its motion can be characterized by 6-DoF: (x, y, z) are translations and (ϕ, θ, ψ) are rotations. These phenomena must be incorporated into modeling based on the correct physical principles for proper design and analysis of the control law.

To model the system, the Newton–Euler method, which works according to Newton's second law, must be applied [13].

As with any system modeling, some assumptions must be made for the quadcopter to simplify the model [13].

1. The quadcopter holds a symmetrical structure.
2. A proportional relationship exists between the squared value of the rotor speed and the thrust.
3. The analysis of one rotor is enough because all rotors are coinciding.

Additionally, small-angle assumption facilitated the linearization of the quadcopter model under hovering conditions, thereby reducing the multiple-input multiple-output (MIMO) nonlinear system to four independent single-input single-output (SISO) systems [14].

These assumptions simplify the modeling process while preserving the essential dynamics of the quadcopter. A symmetrical structure is assumed because most small UAV quadcopter frames are designed and built with equal arms and balanced layouts. The thrust produced by each rotor was assumed to increase with the square of its speed. This follows how real propellers work because spinning faster pushes much more air and creates much more lift. Because the quadcopter is built with a symmetric shape, all four rotors are identical. Therefore, analyzing a single rotor is sufficient.

The next step was to specify the physical parameters of the quadcopter and develop a mathematical model for the system based on its nonlinear and linear characteristics. Nonlinear systems exhibit a non- proportional response to applied inputs, making their behavior difficult to predict. However, a nonlinear system can be simplified using FBL to develop a practical controller. The following two sections describe the nonlinear model of the quadcopter UAV, followed by the linear model applied to develop the controller.

2.1. Physical Structure and Parameters. The most important step in understanding quadcopters is understanding their physical structures. It plays a crucial role in maintaining stability and overall performance. The quadcopter contained four rotors designed to be positioned at equal distances from the center of the mass of the UAV body. The rotors were divided into groups of two, with one group rotating counterclockwise and the other clockwise, driven by four motors, as shown in Figure

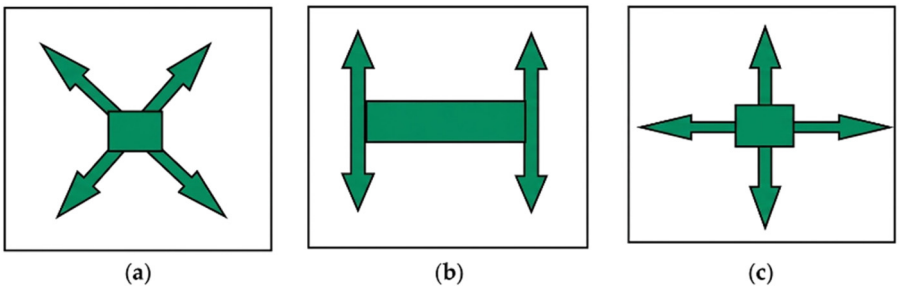


Figure 3. Different physical structures of quadcopter [16]

Table 2. *Variables applied in the nonlinear model and input mapping*

Variable	Definition	Units
$y, z; \dot{x}, \dot{y}, \dot{z}; \ddot{x}, \ddot{y}; \ddot{z}$	Quadcopter UAV position, velocity, and acceleration in inertial frame	m, m/s, m/s^2
$\varphi, \theta, \psi; \dot{\varphi}, \dot{\theta}, \dot{\varphi}, \ddot{\theta}, \ddot{\phi}$	Roll, pitch, and yaw angles, angular rates, and angular accelerations	rad, rad/s, rad/s^2
m, g	Vehicle mass and gravitational acceleration	kg, m/s^2
Ix, Iy, Iz, IR	Moments of inertia about body axes and rotor inertia	kg· m^2
L	Half motor-to-motor distance (arm length)	m
$u1$–$u4$	Control inputs: thrust ($u1$), roll ($u2$), pitch ($u3$), yaw ($u4$)	N, N· m
b, d	Thrust and drag (moment) coefficients	N· s^2, N· m· s^2
$\Omega 1 - \Omega 4, \Omega r$	Rotor angular speeds and net reaction term	rad/s
$\mathbf{U} = [U1U2U3U4]^\top$	Input vector (equivalent to $[u1, u2, u3, u4]^\top$)	–

1 . Studying the mechanical design of a quadcopter is important before deriving a mathematical model or applying a controller to a UAV system.

Figure 1 illustrates the physical structure of a quadcopter consisting of four rotors. To balance the quadcopter UAV flight, the two rotors rotate in opposite directions. As shown in Figure 1 , rotors 2 and 4 rotate counterclockwise, whereas rotors 1 and 3 rotate clockwise. When the rotors rotate, aerodynamic reaction forces generate torque on the quadcopter body.

As Figure 2 illustrates, there are eight possible cases for the quadcopter to move, resulting from differential rotor speeds. Differences in the rotor thrust lead to vertical motion and rotation around the roll, pitch, and yaw axes.

Figure 3 illustrates the three physical structures of the quadcopter: X, H, and +. Although these configurations differ in shape, they share the same basic symmetry used in the modeling assumptions of this study. Because the focus here is on control rather than mechanical design, the same control approach is applied to all configurations.

As this study was based purely on simulations, the physical parameters of the quadcopter were assumed using typical values commonly reported for small quadcopter UAVs in the literature, to ensure that the model represented a realistic quadcopter configuration and that the simulation results were meaningful and reproducible. The primary physical parameters used in this study are listed in Table 1. Furthermore, only physical parameters that directly influence the system dynamics and controller design were considered in this study.

2.2. Nonlinear Dynamic Model. The relationship between the forces and torque generated by the rotors and the motion of the quadcopter are presented using a nonlinear model of the quadcopter UAV.

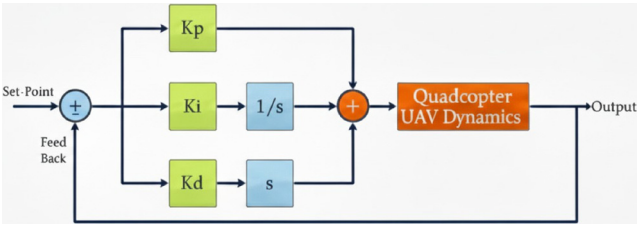


Figure 4. PID controller block diagram of quadcopter [14]

Abbasi & Mahjoob [17] started by focusing on the three Euler angles, which are the roll angle (ϕ), the pitch angle (θ), and the yaw angle (ψ). The final nonlinear equations of the quadrotor positions with Euler angles are written as follows:

$$\ddot{x} = -(\cos\varphi \sin\theta \cos\psi + \sin\varphi \sin\psi) \cdot \frac{u_1}{m} \tag{1}$$

$$\ddot{y} = -(\cos\varphi \sin\theta \sin\psi + \sin\varphi \cos\psi) \cdot \frac{u_1}{m} \tag{2}$$

$$\ddot{z} = g - (\cos\varphi \cos\theta) \cdot \frac{u_1}{m} \tag{3}$$

$$\dot{\varphi} = \theta\dot{\psi}\left(\frac{I_y - I_z}{I_x}\right) - \frac{I_R}{I_x}\dot{\theta}g(u) + \frac{L}{I_x}u_2 \tag{4}$$

$$\dot{\theta} = \dot{\varphi}\dot{\psi}\left(\frac{I_z - I_x}{I_y}\right) - \frac{I_R}{I_x}\dot{\varphi}g(u) + \frac{L}{I_y}u_3 \tag{5}$$

$$\dot{\psi} = \dot{\theta}\dot{\varphi}\left(\frac{I_x - I_y}{I_z}\right) + \frac{1}{I_z}u_4 \tag{6}$$

These equations describe the motion of the quadcopter in space. When the quadcopter tilts, a component of the thrust is redirected laterally, resulting in motion along x and y axes. The $z$equation shows that the drone moves up or down depending on the balance between the total thrust and gravity. The Euler angle equations describe the rotational dynamics of the quadcopter. When an imbalance in thrust occurs between opposing rotors, the quadcopter undergoes rolling or pitching motion. The yaw motion originates from the twisting effect created by the spinning rotors. The inertia terms indicate how difficult it is for a drone to change its rotation.

This approach has been applied to model a quadcopter [17]. The following equations show the input of a quadcopter UAV[2]:

$$\begin{aligned} u_1 &= b\left(\Omega_4^2 - \Omega_2^2\right) \\ u_2 &= b\left(\Omega_3^2 - \Omega_1^2\right) \\ u_3 &= d\left(\Omega_4^2 - \Omega_2^2 - \Omega_3^2 - \Omega_1^2\right) \\ u_4 &= b\left(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right) \end{aligned} \tag{7}$$

These inputs are derived from the rotor speed. Changing the motor speed changes the lift and creates a turning force. We used the square of the rotor speed because spinning faster pushes more air and produces greater thrust.

A quantitative link between the input (U) and the speed of the rotor (Ω) as well as the disturbance equation can be shown by the

following as[18]:

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ -b & -b & b & b \\ -b & b & b & -b \\ -d & d & -d & d \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \tag{8}$$

$$\Omega_r = \Omega_4 + \Omega_2 - \Omega_1 - \Omega_3 \tag{9}$$

This matrix shows how each rotor contributes to the quadcopter's movement. The opposite rotors work in pairs, and the signs indicate whether they help or oppose each other. The last row uses the drag coefficient because the yaw motion originates from the twisting effect of the propellers.

Table 2 defines each variable derived in the mathematical nonlinear model.

2.3. Linearization via Feedback Linearization. To facilitate the controller design, the nonlinear quadcopter model was transformed into an equivalent linear form using feedback linearization. To obtain identical results for the converted input and output, input–output linearization was used. It involves multiple differentiations of the output before the point where the input is found[19]. Other linearization methods exist, such as small-angle approximations or Jacobian-based linearization around the hover point; however, these methods only work in a very limited area and neglect several nonlinear effects. FBL was preferred because it avoids these limitations and offers a clearer input–output structure for later PID designs.

$$\dot{x} = f(x) + g(x)u \tag{10}$$

$$y = h(x) \tag{11}$$

$f(x)$ is a nonlinear system, $g(x)$ is the input dynamics, and the control input (u) is assigned to a state vector (x''). y represents the controlled variable (CV) with a nonlinear function $h(x)$. Eqs. 10–14 are applied to linearize the system, as presented in[17] and[19].

The following formula shows the frequent differentiation of the output variable, where y'' is defined in Table 2, and $L_f h(x)$ and $L_g h(x)$ represent the Lie derivatives of the output function $h(x)$ along the system dynamics $f(x)$ and the control input vector $g(x)$, respectively.

$$\dot{y} = L_f h(x) + L_g h(x)\, u \tag{12}$$

The formula of higher-order systems can be represented as follows:

$$y^{(r)} = L_f^r h(x) + L_g L_f^{r-1} h(x)\, u \tag{13}$$

Here, $y^(r)$ represents the required order of differentiation of the output for feedback linearization. The expression $L^r h(x)$ the repeated Lie derivative of $h(x)$ along $f(x)$ applied r times, while $L_g L^{r-1} h(x)$ denotes the Lie derivative of $h(x)$ along $g(x)$ after being differentiated $r-1$ times along $f(x)$, which forms part of the feedback linearization process.

To convert the system into a linear form, (14), which is presented as the input, must be substituted into (10) to eliminate nonlinearity.

$$u = \frac{1}{L_g L_f^{r-1} h(x)} \left(-L_f^r h(x) + \upsilon \right) \tag{14}$$

In the linearized model, υ denotes the transformed linear control input introduced through the FBL process.

The controlled variable can then be expressed as follows, where υ is the new linear control input:

$$y^{(r)} = \upsilon \tag{15}$$

$$y = \begin{bmatrix} Z \\ \varphi \\ \theta \\ \psi \end{bmatrix} \tag{16}$$

Linearized vertical dynamics:

$$\ddot{z} = \upsilon_z \tag{17}$$

υ_z corresponds to the new control input associated with the vertical motion along the z-axis. Linearized rotational dynamics as υ_ϕ, υ_θ, and υ_ψ represent their corresponding linear control inputs:

$$\ddot{\varphi} = \upsilon_\varphi, \quad \ddot{\theta} = \upsilon_\theta, \quad \ddot{\psi} = \upsilon_\psi \tag{18}$$

The previous approaches in Eqs. 15–18 were used in[19]. After linearizing the system, the state–space representation of the input can be expressed as follows[2]:

$$\dot{x} = Ax + Bu \tag{19}$$

A and B represent the state and input matrices, respectively.

The state space representation of the output as C is the output matrix:

$$y = Cx \tag{20}$$

The state variables are the following:

$$x = \begin{bmatrix} \varphi & \dot{\varphi} & \theta & \dot{\theta} & \psi & \dot{\psi} & z & \dot{z} \end{bmatrix}$$

$$A = \begin{bmatrix} 0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&0&0 \end{bmatrix} \quad B = \begin{bmatrix} 0&1&0&0 \\ 1&0&0&0 \\ 0&0&0&0 \\ 0&1&0&0 \\ 0&0&0&0 \\ 0&0&1&0 \\ 0&0&0&0 \\ 0&0&0&1 \end{bmatrix}$$

$$C = \begin{bmatrix} 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \end{bmatrix}$$

3. CONTROL SYSTEM DESIGN (PID)

After obtaining the mathematical model, the next step is to develop control strategies for the nonlinear dynamics of the

Table 3. *Variables applied in the PID controller design and Euler angle dynamics*

Variable	Definition	Units
$u(t)$	Control signal (PID controller output)	N· m
Kp Ki	Proportional gain constant Integral gain constant	$-_s-1$
K_d	Derivative gain constant	s
$e(t)$	Tracking error between desired and actual angle	rad
$desired(t)$	Desired Euler angle input	rad
$actual(t)$	Measured or actual Euler angle	rad
U_ϕ, U_θ, U_ψ	Control torques for roll, pitch, and yaw axes	N· m
$\Phi(s), \Theta(s), \Psi(s)$	Laplace-transformed roll, pitch, and yaw angles	rad
s	Complex frequency variable in Laplace domain	rad_s-1
b	Thrust coefficient (from rotor model)	N· s^2
Tx, Ty, Tz	Torque produced about each rotational axis	N· m
Ix, Iy, Iz	Moments of inertia about body axes (repeated here for clarity)	kg· m^2

Table 4. *Comparison between GA and DE*[24]

Feature	GA	DE	UAV Relevance
Optimization type	Population-based	Differential updates	Global search for nonlinear system
Convergence rate	Moderate, stable	Faster, consistent	GA avoids premature convergence
Reliability	High	Very high	GA reliable for dynamic models
Performance (MSE, IAE)	Nearly equal to DE	Slightly better in some cases	Both ensure low error
Parameter sensitivity	Low	High	GA easier to tune
Implementation	Simple	Parameter dependent	GA easier in Simulink
Overall suitability	Robust, flexible	Efficient, sensitive	GA best balance for quadcopter UAV PID

quadcopter UAV. Control methods such as PID control, and feedback linearization are widely used due to their effectiveness in handling system complexity and achieving high performance. These controllers stabilized the quadcopter under various flight conditions.

The PID controller is the most common controller used for a quadcopter owing to its effectiveness in controlling the roll, pitch, and yaw Euler angles.

Figure 4 illustrates the process of combining each part of the PID controller with the quadcopter in a feedback loop. The PID controller was utilized, as shown in the following equations[20]:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)\,dt + K_d \frac{de(t)}{dt} \tag{21}$$

In this study, the same PID gains were used for the roll, pitch, and yaw axes to maintain a clear and consistent comparison between the tuning methods. The use of identical gains enables the effect of the tuning approach to be observed without being affected by axis-specific adjustments. Although the axes have different inertial properties, this assumption is reasonable for a comparative analysis and makes the methodology straightforward.

where the error is represented as[20]:

$$e(t) = desired(t) - actual(t) \tag{22}$$

By controlling only the Euler angles, some terms in their equations were eliminated as follows:

$$\ddot{\varphi} = \frac{U_2}{I_{xx}} \tag{23}$$

$$\ddot{\theta} = \frac{U_3}{I_{yy}} \tag{24}$$

$$\ddot{\psi} = \frac{U_4}{I_{zz}} \tag{25}$$

Applying the Laplace transform to the Euler angles helps extract the characteristics of the system more easily using MATLAB.

$$\frac{\Phi(s)}{U_2(s)} = \frac{lb}{I_{xx}s^2} \tag{26}$$

$$\frac{\Theta(s)}{U_3(s)} = \frac{lb}{I_{yy}s^2} \tag{27}$$

$$\frac{\Psi(s)}{U_4(s)} = \frac{ld}{I_{zz}s^2} \tag{28}$$

To reduce the time required to tune the PID parameters, the quadcopter was divided into four axes: roll, pitch, yaw, and altitude. This division process facilitates independent tuning of parameters for each axis while neglecting the inherent coupling effects present in the real system. Additionally, it aids in analyzing the influence of the control gains on the system response. Subsequently, the PID gains were tuned. The quadcopter response

was evaluated using step and sinusoidal inputs, particularly for GA-based tuning. The goal was to achieve a smooth and stable response with minimal overshoot. The primary performance metric used for tuning was the ISE value, which quantifies the deviation of the system response from the reference signal. The settling time and overshoot were also verified to ensure that the controller did not react too slowly or aggressively. The same evaluation criteria were applied to both manual tuning and GA-based methods.

The PID gains were tuned to achieve a stable response with an acceptable rise time, settling time, and overshoot. Adjustments are made based on the system response to reference inputs to responsiveness and dampening rather than following a strict tuning sequence.

Applying a PID controller to the quadcopter establishes a closed-loop system that enables prediction of key performance characteristics, such as the peak time, rise time, settling time, and overshoot, which indicate the system's stability. Table 3 defines each variable applied to design the PID controller.

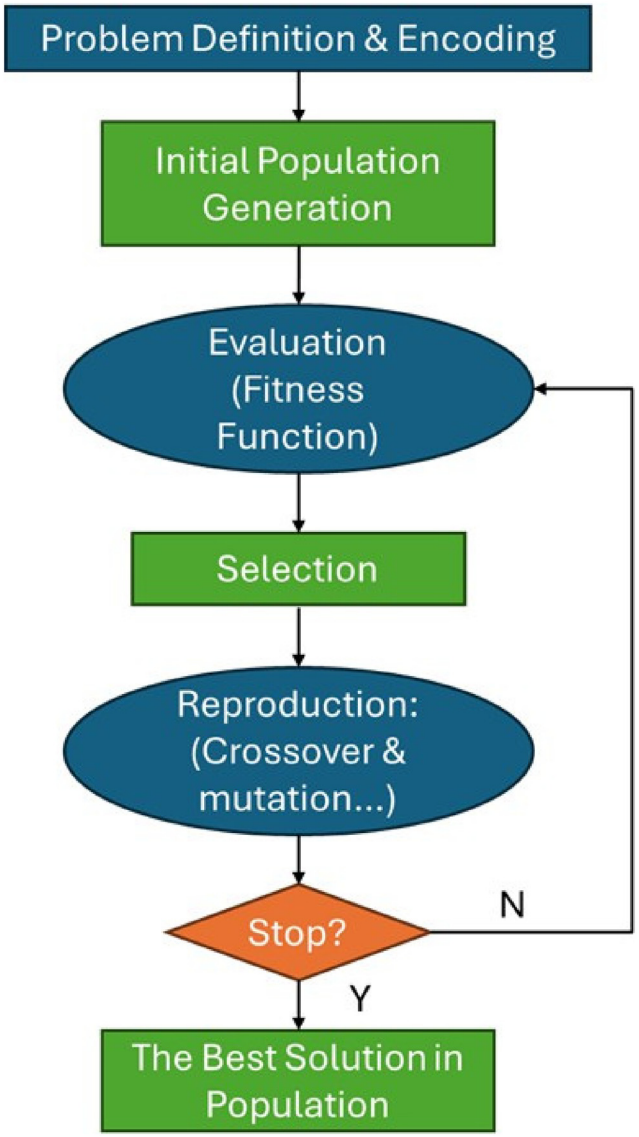


Figure 5. Genetic Algorithm flowchart [25]

4. GENETIC ALGORITHM FOR PID OPTIMIZATION

The GA is an evolutionary optimization technique inspired by natural selection. It is particularly suitable for designing PID controllers for complex systems and handling large nonlinear search spaces, as it reduces the likelihood of being trapped in local minima. Therefore, GA is an appropriate choice for such applications [21].

Several recent studies, as presented in [22], have also shown that GA-based tuning enhances quadrotor stability and tracking, which supports the approach used in this work; additionally, GA is an easy algorithm to apply and less sensitive to tuning settings than other evolutionary methods, which makes it a practical option for UAV PID tuning [23].

A comparison between two evolutionary algorithms, GA and differential evolution (DE), was performed to investigate their high capabilities in PID parameter tuning. As reported in [24], both methods demonstrated similar performance in minimizing the mean squared error (MSE) and integra of absolute error (IAE) for higher-order systems; however, the GA was simpler to apply and less sensitive to tuning parameters. Based on these features, GA is a practical choice for systems in which robustness and adaptability are critical.Table 4 summarizes the comparison investigated between GA and DE.

For the GA, the fitness function is the ISE. As mentioned in Section 3, the GA parameters were selected based on the ISE value to ensure stable and repeatable optimization results. The PID gains were constrained to positive values within bounded ranges to prevent unrealistic or unstable solutions, whereas the population size and number of generations were fixed throughout the optimization process.

The GA is a practical choice for quadcopter control, as it can search for a suitable combination of PID gains and handle the nonlinear nature of UAV dynamics. This demonstrates that it is a useful method for determining gain values that produces stable and smooth responses in simulations.

Figure 5 illustrates the five steps involved in the operation of the GA.

- Initialization: The start involves generating a chromosome that represents a collection of PID gain values.
- Selection: After operating the fitness function, which reduces the error of unreal PID gains, the best values were selected.
- Crossover: It is about using parents, which are a set of chromosomes, and mixing them to have offspring. This is similar to the biological process of merging, which offers a vast solution space for researchers.
- Mutation: The offsprings are manipulated randomly to discover new regions of solution space.
- Termination and Evaluation: The last step is to choose the optimal solution after a specified number of generations, after which the algorithm reaches the end and stops.

5. SIMULATION SETUP AND SCENARIO DESCRIPTION

Two tuning methods are used to optimize the PID parameters. The first method involves manual tuning, whereas the second employs GA.

Table 5. *Manual PID tuning values for three trials*

Trial	Kp (Manual)	Ki (Manual)	K_d (Manual)
1	1.5	3.9	16.0
2	1.9	3.6	17.5
3	2.5	2.9	18.5

Table 6. *Performance metrics of manual PID tuning*

Trial	Rise time T_r(s)	Settling time T_s(s)	Overshoot OS (%)
1	0.55	2.8	3.0
2	0.50	2.5	2.5
3	0.45	2.2	2.0

5.1. Manual PID Tuning. The outputs are generated from the control inputs processed by the linear model through its transfer function representation. The ODE45 solver provides a numerical solution to the state-space equations, enabling simulations of the system. This block diagram shows the feedback loop for every control variable, with each PID controller tuned manually by the operator such that the ISE decreases over time.

Figure 6 illustrates a closed loop of a quadcopter control system in Simulink. The entire system works based on four PID controllers developed to control the attitude and altitude. The error signal to the PID controller is defined according to the difference between the actual output and input signals.

Table 5 lists the values of the proportional (P), integral (I), and derivative (D) gains, which were applied to optimize the attitude and altitude performance. The PID parameters are tuned based on the stability of the system. Because the controlled variables are connected in the same closed loop, there is no difference in the values of the PID gains, and this equalization produces a similar response for the fourth controlled variable. However, each trial showed a different stability and tracking trajectory based on the overshoot performance.

Figure 7 illustrates trial 1, where a step input has been applied to the quadcopter UAV system and the PID gains, which are selected after several trials for Φ, Θ, Ψ, and Z, showed a high steady-state error as well as a failure to predicting the future error based on the rate of change of it. Additionally, there is a high overshoot in the system response, which significantly affects the stability of the quadcopter.

The same process was applied, as shown in Figure 8 ; however, different PID gain values were applied to improve the stability of the quadcopter system. It could be seen that a slight minimization occurred for the overshoot as well as the steady-state error. However, the optimal system performance has not yet been achieved.

In Trial 3, as Figure 9 demonstrates, the updated PID gains contribute to minimizing the current error as well as the steady-state error and predicting the future error based on its rate of change. Even though the simulation can be different from reality, the chosen values of the parameters result in good performance of the system with high stability and accuracy.

Table 6 shows that manual tuning is able to stabilize the system, but the response changes noticeably from one trial to another. While the rise time is generally short, the settling time and overshoot remain relatively high. This behavior reflects the trial-and-error nature of manual tuning, where small changes in gains can lead to different levels of performance.

5.2. GA-Based Tuning. In this section, the system faces a new sine input to test its ability to maintain stability with a low overshoot.

Table 7 lists the GA parameters used to tune the PID gains in the three trials. The mutation operation was adaptively handled using MATLAB's built-in mutation-adapt-feasible function of MATLAB, which automatically adjusts the mutation value during the optimization process instead of using a fixed probability value.

Figure 10 present the best solution of the ten-generation process, and they show a little oscillation in the system; however,

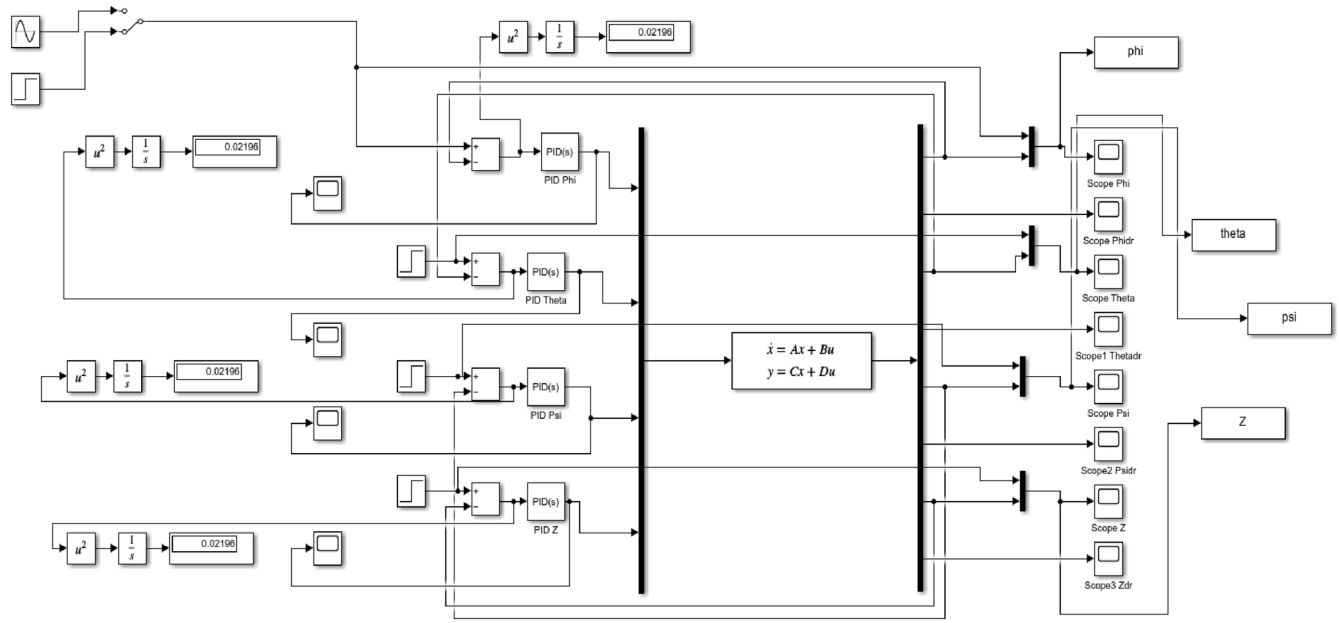


Figure 6. SIMULINK block diagram

Table 7. *Genetic Algorithm parameters used for PID optimization*

GA Parameters	Value
Population Size	30
Crossover Probability	0.8
Mutation	Adaptive (auto-adjusted)
Generations	10, 20, 30

it took less time compared to a large number of generations process.

Figure 11 shows the response of the control variable with respect to the number of generations, which was 20. Slight improvements in the overshoot and settling time performance of the system were observed between G = 10 and G = 20. It showed better results regarding the error; however, it took more time compared to ten generations.

A better result is illustrated in Figure 12 , which shows a more evolved solution than the previous two solutions. The optimal fitness value was reached as 0.030056. This solution is not the most optimal solution; however, it shows a better performance of the GA in tuning the PID gains instead of tuning them manually, which may require more human effort. In Figure 12 d, the quadcopter rises quickly to 1 m in nearly 1 s, then oscillates slightly after 6 s, but returns to the steady state smoothly. It could be seen from Figures 10a, 11a, and 12a, that the roll angle kept its performance similar to the other angles in each trial, even though it received a different input.

As listed in Table 8, the population value was fixed at 30, and the system performance was tested based on different generations (10, 20, and 30) to configure how the GA is evolving and gives better values of PID gains and ISE. Population represents the number of solutions per generation. Usually, an increase in the PID gain results in a lower efficiency and affects the response of the system owing to noise; however, in our case, the GA reached the optimal point, so a large increase in the PID parameters did not occur, as shown. The best fitness value was obtained in the last trial because the algorithm required a longer time to generate optimal results.

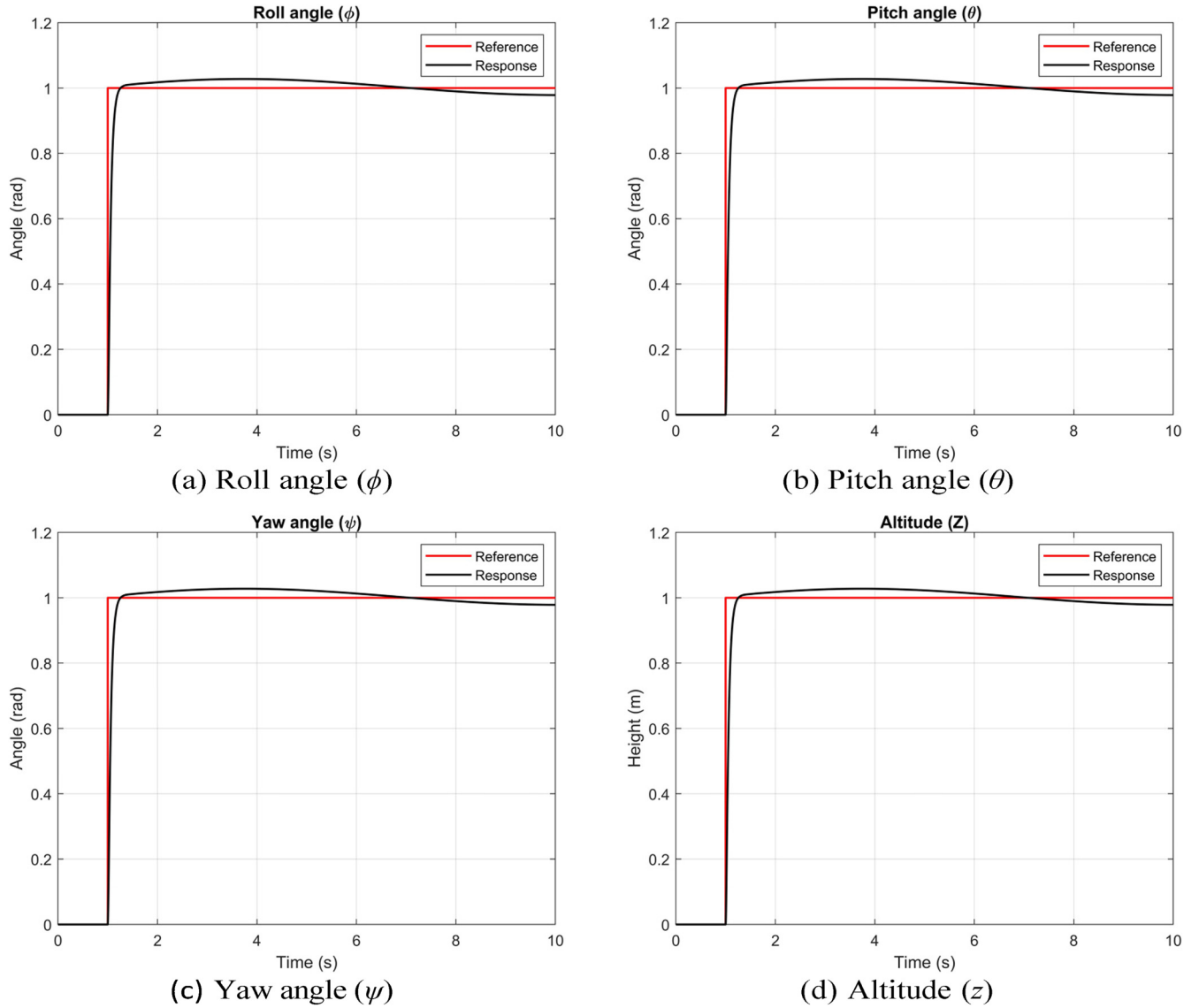


Figure 7. Quadcopter responses of attitude and altitude for trial 1

Table 8. *GA optimization results for different numbers of generations*

Trial	Population Size	Generations	K_p(All)	K_i(All)	K_d(All)
1	30	10	2.2142	3.5205	19.8439
2	30	20	2.9508	3.2705	19.9689
3	30	30	2.9801	2.9893	19.9913

Table 9. *Performance metrics of GA-tuned PID*

Trial	Rise time T_r(s)	Settling time T_s(s)	Overshoot OS (%)
1	0.35	1.5	1.5
2	0.30	1.3	1.3
3	0.28	1.1	1.0

Table 9 shows that, compared to the manual tuning method, the GA achieved a more stable response. It can be noticed that the rise time is reduced to nearly 0.3 s, and the settling time decreases to 1–1.5 s, which is almost half of the manual values. A less aggressive system response is produced as the overshoot value is reduced to 1% at the last trial. The enhancement of the system's performance demonstrates that the GA method results in better PID gains, which provide a faster and more stable performance for all four axes.

5.3. Disturbance Tests and Scenario Evaluations. To ensure that the GA performed better than the trial-and-error method, different scenarios and disturbances were applied to both methods.

Figure 13 illustrates a scenario in which the best manually tuned PID gains for altitude were tested by applying a rectangular disturbance. It could be seen that Figure 13 b shows the response, as it results in high overshoot and an unstable system.

Table 10 lists the values of T_r, T_s, and OS for manual tuning. High values were obtained when a disturbance was applied to the system.

In Figure 14 , four scenarios were applied to test the adaptation of the system while flying. Each figure shows the rising and landing times of the quadcopter. Figure 14 a shows a scenario in

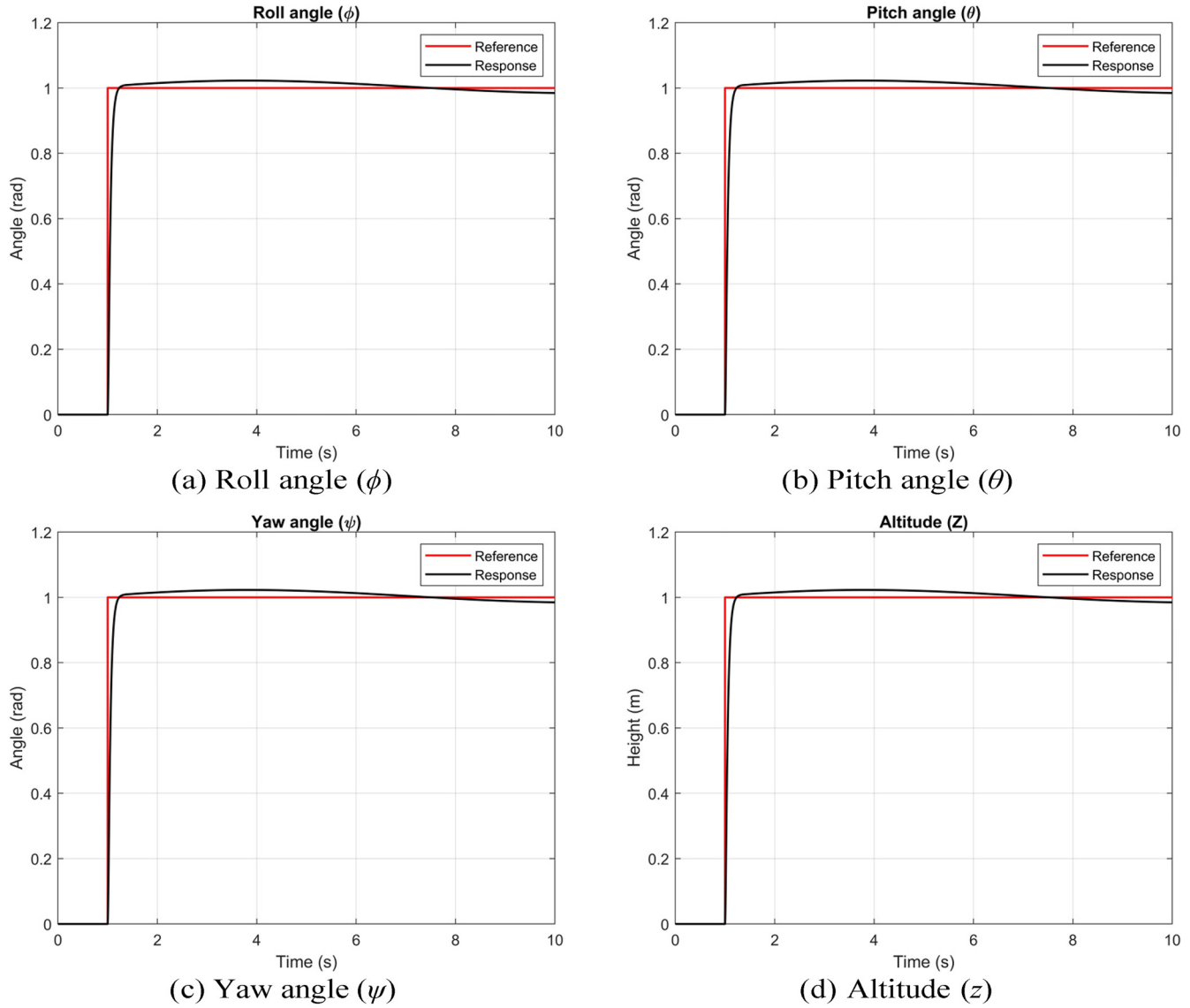


Figure 8. Quadcopter responses of attitude and altitude for trial 2

Table 10. *T_r, T_s, and OS for manual tuning*

Metric	Value
Rise time T_r(s)	0.35
Settling time T_s(s)	8.00
Overshoot (%)	8%

which the quadcopter UAV takes off at 1 s until it reaches 1 m, stays at the same level for 5 s, and then lands at ground level.

Another scenario is illustrated in Figure 14 b, where the quadcopter performs a two-time takeoff to reach a level of 1 m and then lands. The two takeoff and landing processes were repeated every 2 s and ended at 8 s.

Between takeoff and landing, the quadcopter remained at the same level for 2 s. In scenario 3, the quadcopter UAV takes off at 1 s to reach the level of 1 m and stays for 1 s more. It repeats the same process every 1 s until 10 s, however, every 1 s it flies almost 0.25 m more. Finally, it landed at 10 s, as presented in the Figure 14 c.

The last scenario in Figure 14 d consists of 5 phases: take-off (0 s–1 s), hovering at 1 m (1 s–3 s), climb to 2 m and staying at the same level (3 s–5 s), descent to 1 m (5 s–9 s), and landing to the ground level phase (9 s–10 s). Each scenario was applied to the system based on Trial 3, which exhibited the best performance of the three trials. For each scenario, a good performance was observed, despite a slight overshoot.

Once the scenarios and input disturbances are applied to the same PID gains of the manual tuning method, the GA-tuned PID gains are evaluated by applying the same process.

Figure 15 shows that the same input disturbance was applied to the best altitude PID gains for the GA. However, better performance for the GA altitude response was observed.

Table 11 summarizes that the GA under a disturbance shows lower values of T_r, T_s, and OS, confirming that the GA outperforms the manual tuning method.

Figure 16 illustrates four cases in which the system can adapt to different scenarios that were also applied to a manually tuned system. Each scenario is explained in section 5.1. Overall, there was

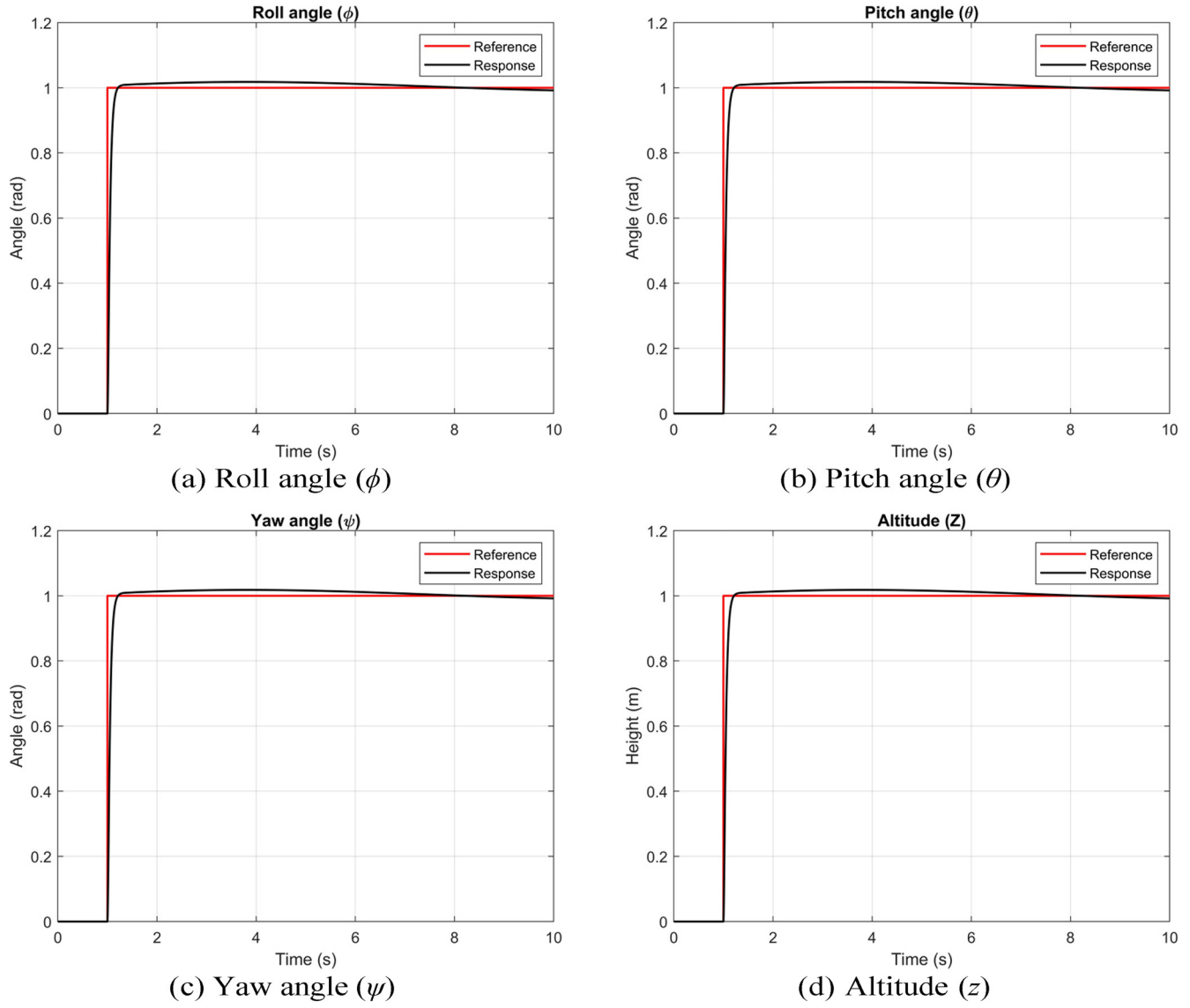


Figure 9. Quadcopter responses of attitude and altitude for trial 3

Table 11. *T_r, T_s, and OS for GA tuning under disturbance*

Metric	Value
Rise time T_r(s)	0.30
Settling time T_s(s)	7.50
Overshoot (%)	7%

no significant difference among the four cases regarding the performance of the system as it reached the optimal point. However, it was confirmed that increasing the number of generations to 30 assists in minimizing the overshoot and confirming the damping efficiency of the controller.

The four scenarios focused on the *z*direction; therefore, most of the changes were only in altitude. Although the roll, pitch, and yaw controllers remained stable, they were not fully challenged by the horizontal movement or path tracking. This means that the scenarios show how the quadcopter handles vertical changes, but do not reflect full trajectory tracking. To achieve a *x*–*y*path, tests are required to evaluate all axes together to achieve an x–y path.

Table 12. *Comparison between manual and GA tuning based on ISE performance*

Trial	ISE (Manual)	ISE (GA)	Improvement (%)
1	0.03919	0.03140	19.9
2	0.03543	0.03101	12.5
3	0.03316	0.030056	9.3
Average	–	–	**13.9**

Table 13. *Runtime comparison for best PID gains*

Method	Search effort	Best result obtained at	Total tuning time
Manual tuning	≈10 simulations	Trial 3	≈10 minutes
GA tuning (best run)	$P = 30, G = 30$ (≈900 evals)	≈30 minutes Trial 3	(or more)

5.4. Comparative Analysis. After obtaining the ISE values for both methods, a comparison was conducted as follows:

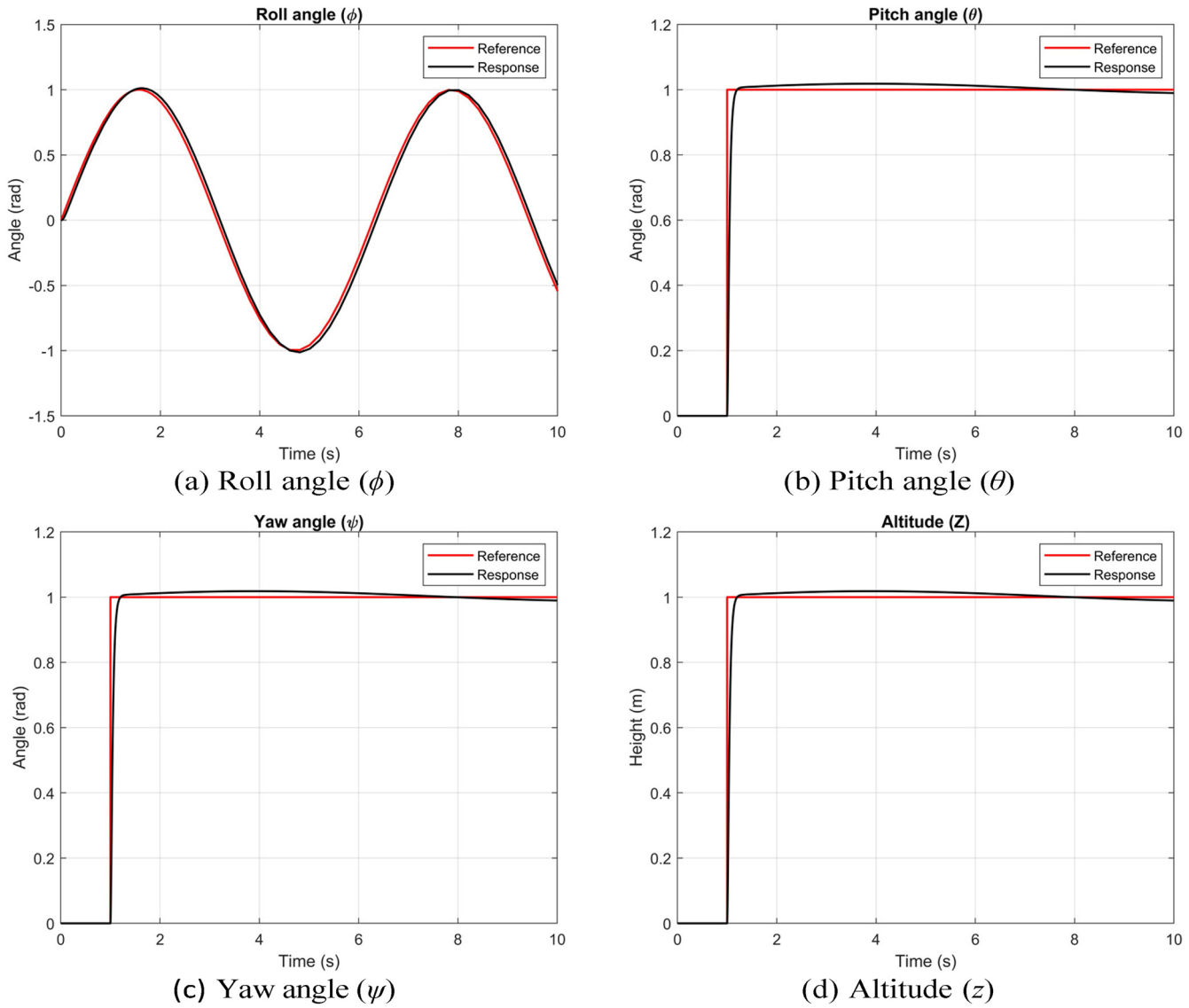


Figure 10. Quadcopter responses of attitude and altitude under G = 10

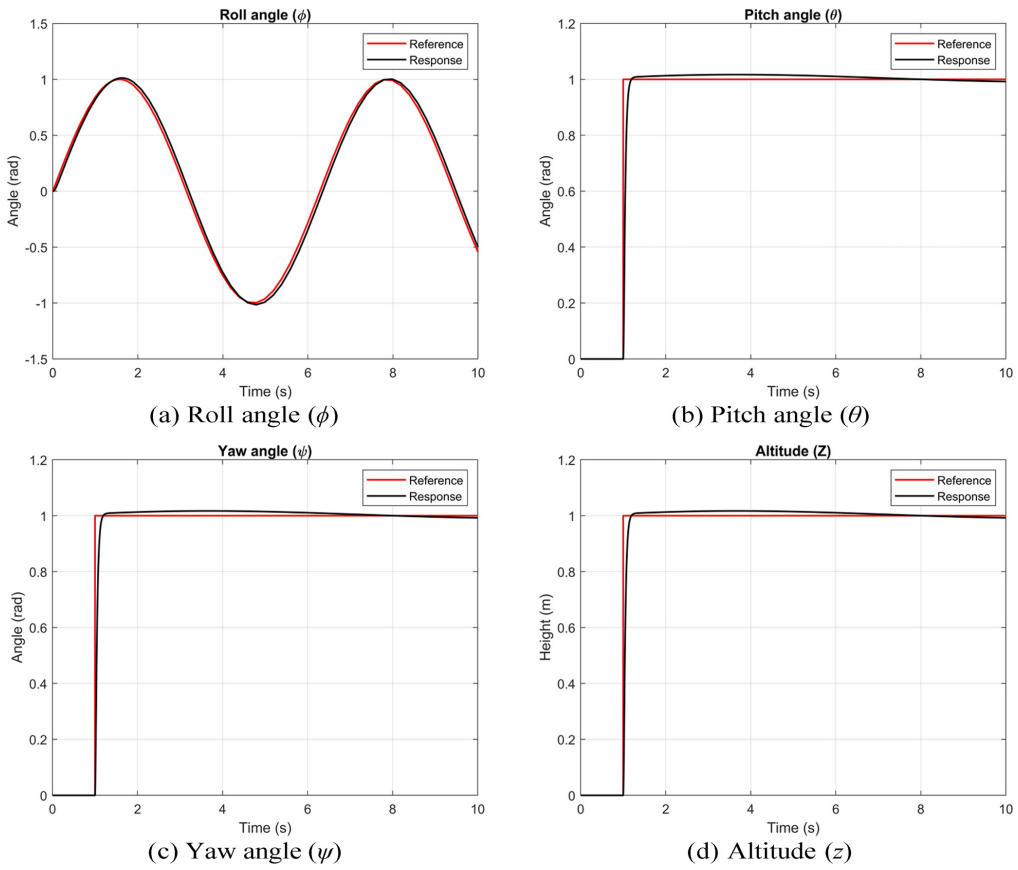


Figure 11. Quadcopter responses of attitude and altitude under G = 20

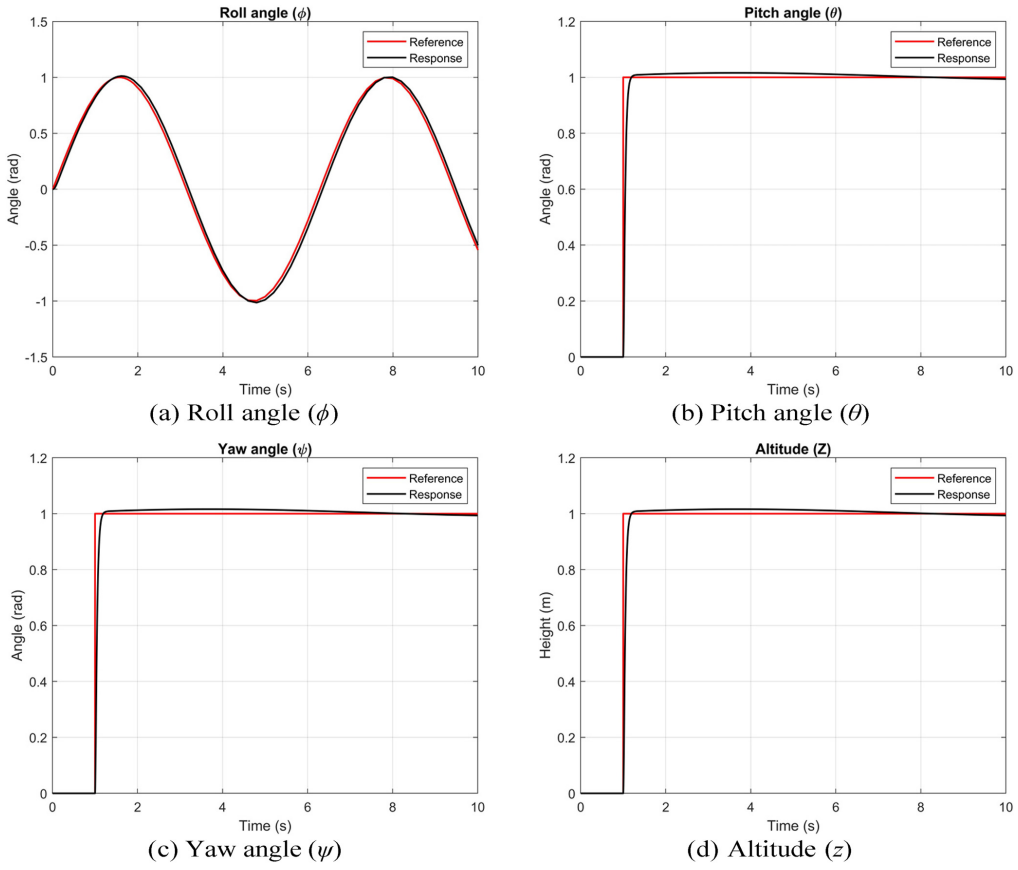
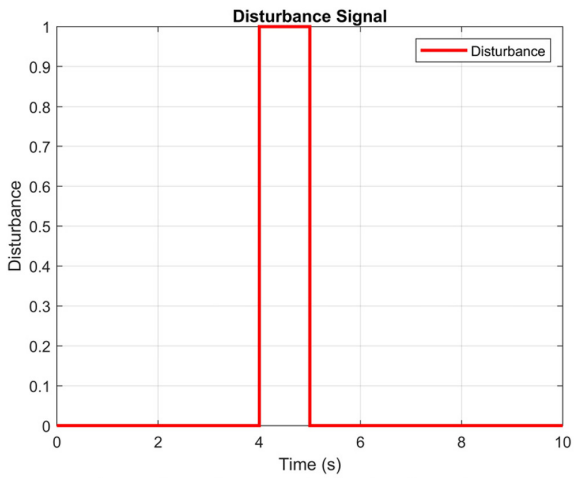
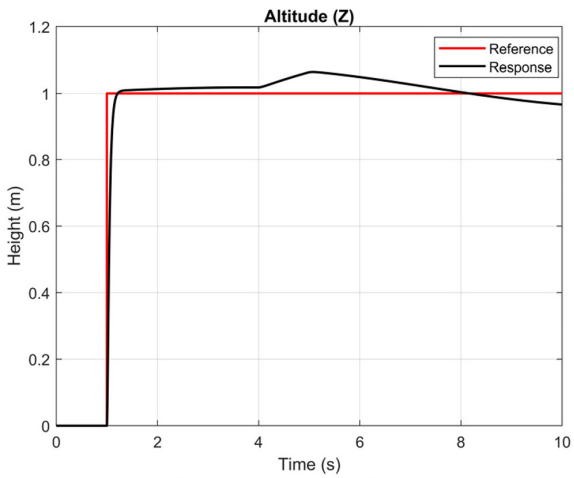


Figure 12. Quadcopter responses of attitude and altitude under G = 30

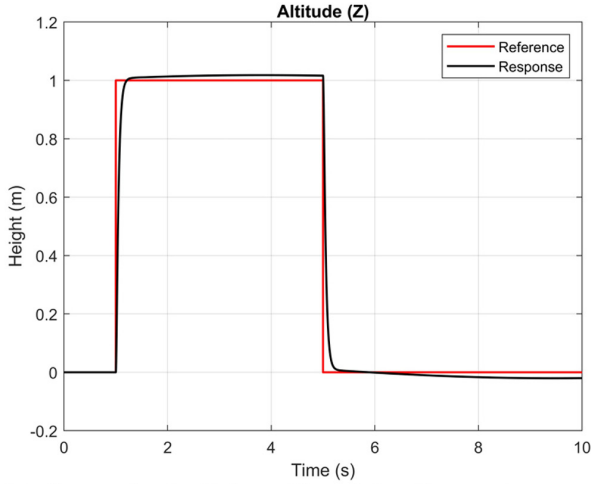


(a) Disturbance (Input) signal.

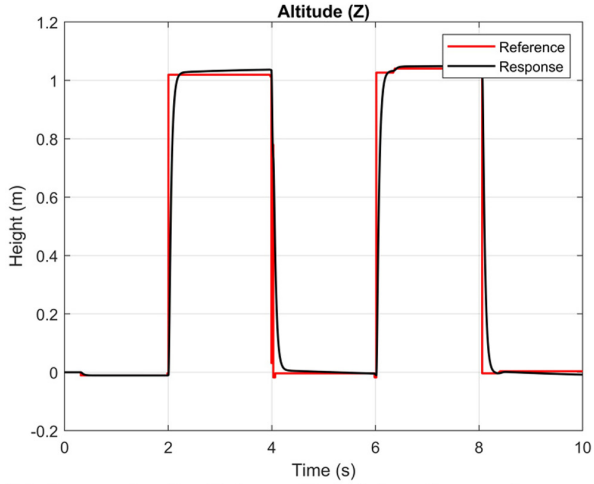


(b) Manual tuning under disturbances.

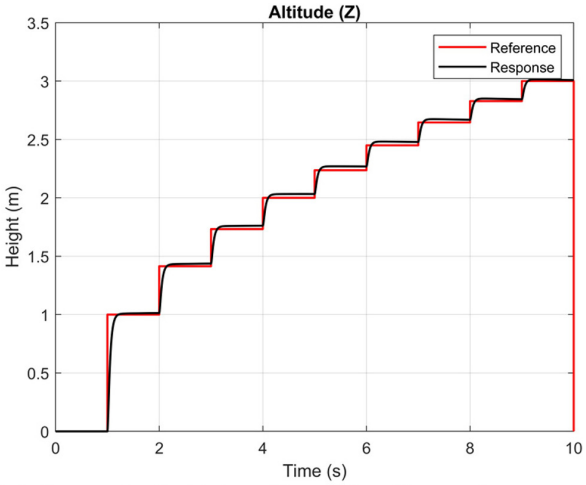
Figure 13. Manual tuned quadcopter response of altitude tested by disturbance input



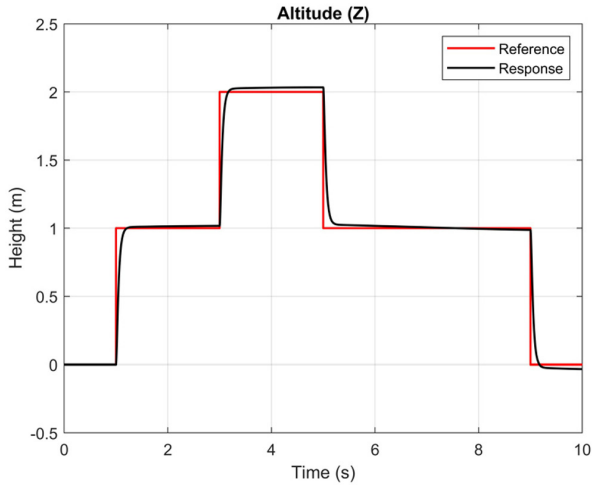
(a) Scenario 1: Take-off and landing using PID controller (Altitude z)



(b) Scenario 2: Take-off and landing using PID controller (Altitude z)



(c) Scenario 3: Takeoff and landing using a PID controller (altitude z)



(d) Scenario 4: Takeoff and landing using a PID controller (altitude z)

Figure 14. Quadcopter responses of altitude tested by four scenarios

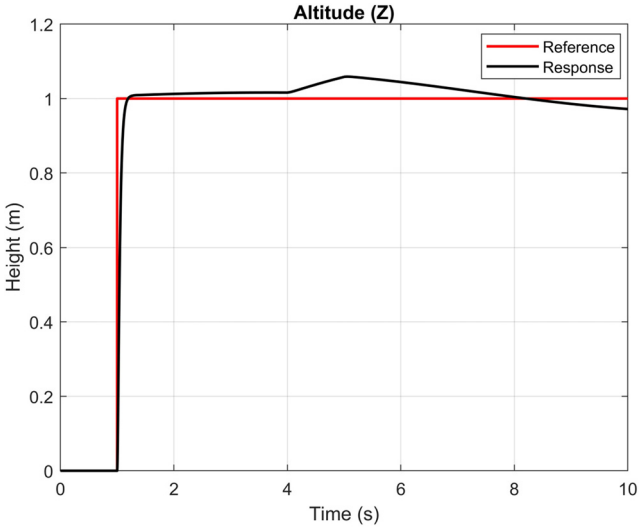
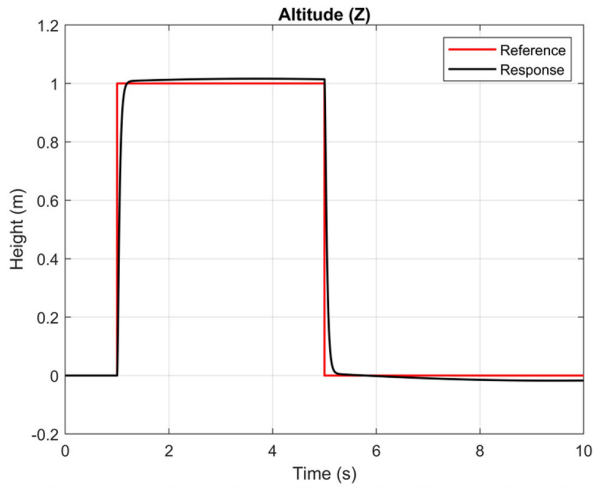


Figure 15. GA tuning under disturbance

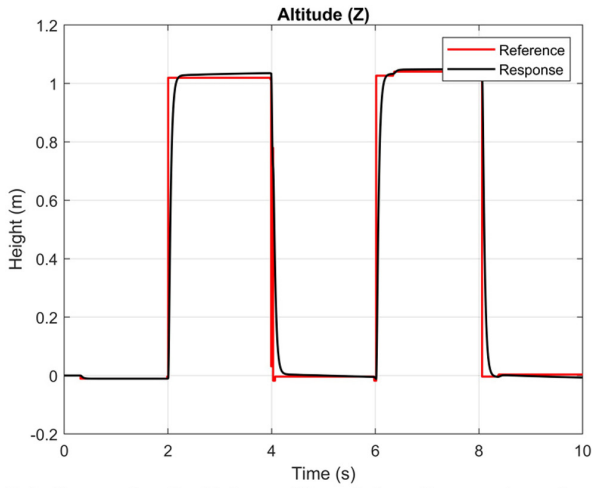
Table 12 shows that the GA-tuned controller consistently achieves lower ISE values than manual tuning across all trials. This indicates that the GA-selected gains allow the system to follow the reference signal more closely and with less variation between runs. The improvement is observed in all cases, which suggests that GA provides a more reliable tuning approach than manual adjustment. Notably, the GA-derived PID parameters are specific to this quadcopter model and the simulation scenarios applied to it, so they might not fit a different frame or flight regime. However, the tuning procedure is general. Changing the model parameters with the GA regenerating PID gains will show the same workflow that is applied to obtain suitable PID gains.

Figure 17 shows the GA convergence curve for the best ISE value in Trial 3. The ISE increases rapidly in the early generations, from 0.0315 to nearly 0.0302, and then stabilizes after 15–20 generations. By the end of the run, the best ISE value of 0.030056 was reached, indicating near convergence.

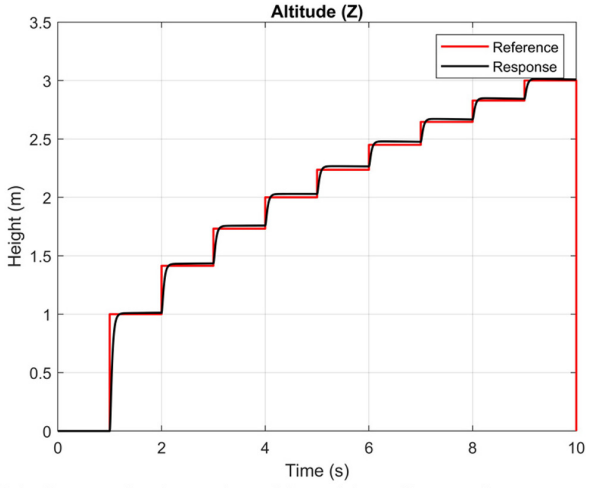
Table 13 shows that the computational time of GA-based PID tuning is evaluated by its fitness. Moreover, GA spends a longer time than the manual tuning method to generate the best PID gains because of its 900 evaluations. Conversely, the manual-



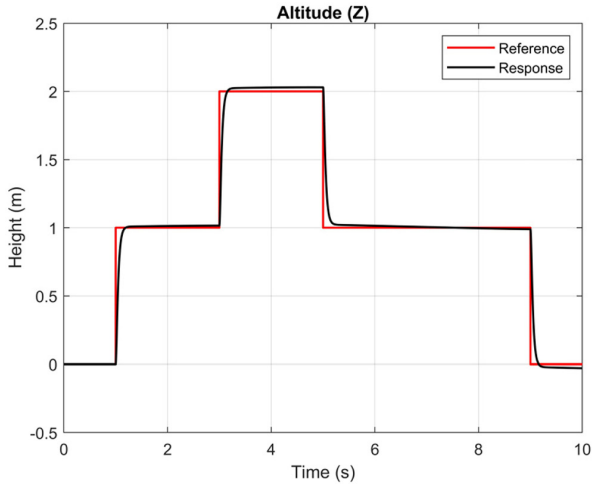
(a) Scenario 1: Take-off and landing using the PID controller (altitude z)



(b) Scenario 2: Take-off and landing using the PID controller (altitude z)



(c) Scenario 3: Takeoff and landing using a PID controller (altitude z)



(d) Scenario 4: Take-off and landing using the PID controller (altitude z)

Figure 16. Altitude responses of the quadcopter under GA-PID control for G = 30

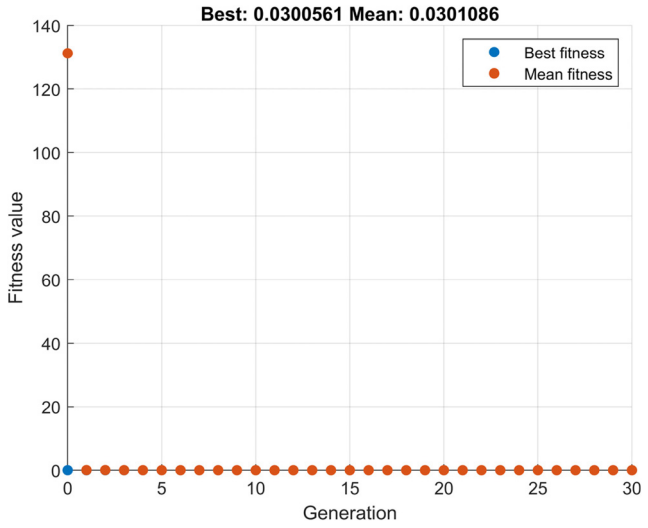


Figure 17. GA convergence curve showing best and mean ISE versus generation

based PID tuning does not use generations, but rather depends on trial-and-error, which is the reason for it to be faster than the GA. Although the GA is slower, it offers a more systematic and repeatable way to tune the controller than manual tuning.

6. CONCLUSION

This study focused on modeling and controller design of a quadcopter system to enhance hover stability and trajectory tracking performance. The nonlinear dynamics were derived by modeling the quadcopter system using the Newton–Euler method and further applying the FBL to obtain a suitable state-space representation for the input and output.

In MATLAB/Simulink, two PID tuning methods were applied: manual and GA. The comparison shows that GA-based tuning yields smoother responses and more consistent behavior than manual tuning. This suggests that a systematic optimization approach can reduce the uncertainty associated with trial-and-error adjustments.

Importantly, the results presented in this work are based on simulations and do not account for practical effects, such as real external disturbances. In addition to being simulation based, this study relied on simplified assumptions. For example, inter-axis coupling and sensor noise are not explicitly modeled, and parameters such as mass and inertia are assumed to be constant, which may affect the applicability of the tuned gains to real flight conditions. Therefore, future work should focus on the real-time implementation and experimental validation of an actual quadcopter platform.

AFFILIATIONS AND AUTHOR DETAILS

Undergraduate Author

Sajjad Al Mahmud – *Department of Control & Instrumentation Engineering, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia;*
Email: s202031380@kfupm.edu.sa

Corresponding Author

Ahmed Eltayeb – *Research Mentor, Researcher, Interdisciplinary Research Center for Smart Mobility and Logistics, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia;* 0000-0002-9151-4116
Email: ahmedtaha@kfupm.edu.sa

ACKNOWLEDGEMENTS

The authors would like to acknowledge the support provided by the Interdisciplinary Research Center for Smart Mobility and Logistics under Project No. INML2404 at King Fahd University of Petroleum and Minerals. The authors also extend their appreciation to the technical staff and reviewers for their valuable time and comments.

REFERENCES

(1) Luukkonen, T. Modelling and control of quadcopter. *Independent Research Project in Applied Mathematics*, **22**, 1–24 (2011).
(2) Rahmat, M. F., Eltayeb, A. & Basri, M. A. M. Adaptive feedback linearization controller for stabilization of quadrotor UAV. *International Journal of Integrated Engineering* **12**, 1–17 (2020).
(3) Ye, J., Wang, J., Song, T., Wu, Z. & Tang, P. Nonlinear modeling the quadcopter considering the aerodynamic interaction. *IEEE Access* **9**, 134716–134732 (2021).
(4) Abdulkareem, A., Oguntosin, V., Popoola, O. M. & Idowu, A. A. Modeling and nonlinear control of a quadcopter for stabilization and trajectory tracking. *Journal of Engineering* **2022**, 2449901 (2022).
(5) Abdelhay, S. & Zakriti, A. Modeling of a quadcopter trajectory tracking system using PID controller. *Procedia Manufacturing* **32**, 564–571 (2019).
(6) Praveen, V., Pillai, S. *et al.* Modeling and simulation of quadcopter using PID controller. *International Journal of Control Theory and Applications* **9**, 7151–7158 (2016).
(7) Kuantama, E., Tarca, I. & Tarca, R. Feedback linearization LQR control for quadcopter position tracking. In *2018 5th International Conference on Control, Decision and Information Technologies (CoDIT)*, 204–209 (2018).
(8) Argentim, L. M., Rezende, W. C., Santos, P. E. & Aguiar, R. A. PID, LQR and LQR-PID on a quadcopter platform. In *2013 International Conference on Informatics, Electronics and Vision (ICIEV)*, 1–6 (2013).
(9) Setlak, L. & Kowalik, R. Analysis of the LQR algorithm in the field of testing the dynamics of quadcopter movement. *Feedback* **7**, 9 (2020).
(10) Baharuddin, A. & Basri, M. A. M. Trajectory tracking of a quadcopter UAV using PID controller. *ELEKTRIKA-Journal of Electrical Engineering* **22**, 14–21 (2023).
(11) Alkamachi, A. & Ercelebi, E. Modelling and genetic algorithm based-PID control of h-shaped racing quadcopter. *Arabian Journal for Science and Engineering* **42**, 2777–2786 (2017).
(12) Olorunnisola, A. & Dahunsi, O. Optimization of quadcopter PID controller gains using ant colony optimization and genetic algorithm. *Journal of Computational Mechanics* **7** (2024).
(13) Musa, S. Techniques for quadcopter modeling and design: A review. *Journal of Unmanned System Technology* **5**, 66–75 (2018).
(14) Romero, L. E., Pozo, D. F., Rosales, J. A. *et al.* Quadcopter stabilization by using PID controllers (2014).
(15) Anudeep, M., Diwakar, G. & Katukam, R. Design of a quad copter and fabrication. *International Journal of Innovations in Engineering and Technology* **4**, 59–65 (2014).

(16) Balayan, A. *et al.* Optimal design of quadcopter chassis using generative design and lightweight materials to advance precision agriculture. *Machines* **12**, 187 (2024).

(17) Abbasi, E., Mahjoob, M. & Yazdanpanah, R. Controlling of quadrotor UAV using a fuzzy system for tuning the PID gains in hovering mode. In *10th International Conference on Advanced Computational Entertainment Technology*, 1–6 (2013).

(18) Kingry, N. *et al.* Design, modeling and control of a solar-powered quadcopter. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 1251–1258 (2018).

(19) Suresh, H., Sulficar, A. & Desai, V. Hovering control of a quadcopter using linear and nonlinear techniques. *International Journal of Mechatronics and Automation* **6**, 120–129 (2018).

(20) Baharuddin, A. & Basri, M. A. M. Self-tuning PID controller for quadcopter using fuzzy logic. *International Journal of Robotics and Control Systems* **3**, 728–748 (2023).

(21) Nagaraj, B. & Murugananth, N. A comparative study of PID controller tuning using GA, EP, PSO and ACO. In *2010 International Conference on Communication Control and Computing Technologies*, 305–313 (2010).

(22) Zareb, M., Nouibat, W., Bestaoui, Y., Ayad, R. & Bouzid, Y. Evolutionary autopilot design approach for UAV quadrotor by using GA. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering* **44**, 347–375 (2020).

(23) Shima, T., Rasmussen, S. J. & Sparks, A. G. UAV cooperative multiple task assignments using genetic algorithms. In *Proceedings of the 2005 American Control Conference*, 2989–2994 (2005).

(24) Saad, M. S., Jamaluddin, H. & Darus, I. Z. PID controller tuning using evolutionary algorithms. *WSEAS Transactions on Systems and Control* **7**, 139–149 (2012).

(25) Hu, Y. & Yang, S. X. A knowledge based genetic algorithm for path planning of a mobile robot. In *IEEE International Conference on Robotics and Automation (ICRA)*, 4350–4355 (2004).