Hindawi
Journal of Engineering
Volume 2022, Article ID 2449901, 19 pages
https://doi.org/10.1155/2022/2449901

Research Article

Modeling and Nonlinear Control of a Quadcopter for Stabilization and Trajectory Tracking

Ademola Abdulkareem[1], **Victoria Oguntosin**[1], **Olawale M. Popoola**[2] **and Ademola A. Idowu**[1]

[1] *Electrical and Information Engineering Department, Covenant University, Ota, Ogun State, Nigeria*
[2] *Centre for Energy and Electric Power, Tshwane University of Technology, Pretoria, South Africa*

Correspondence should be addressed to Victoria Oguntosin; victoria.oguntosin@covenantuniversity.edu.ng

Received 3 May 2022; Revised 12 August 2022; Accepted 22 September 2022; Published 10 October 2022

Academic Editor: Xiao He

This paper presents an adequate mathematical representation of a quadcopter's system dynamics and effective control techniques. A quadcopter is an unmanned aerial vehicle (UAV) that is able to do vertical take-off and landing. This study presents a nonlinear quadcopter system's mathematical modeling and control for stabilization and trajectory tracking. The mathematical model of the system dynamics of the quadcopter is derived using Newton and Euler equations with proper references to the appropriate frame or coordinate system. A PD control algorithm is developed for the nonlinear system for stabilization. Another nonlinear control technique called full state feedback linearization (FBL) using nonlinear dynamic inversion (NDI) is developed and implemented on the quadcopter system. However, there is a problem with the normal approach of the complete derivation of the full state FBL system using NDI as gathered from the literature review. In such an approach, the PD controller that was used for attitude stabilization was able to stabilize the angles to zero states, but the position variables cannot be stabilized because the state variables are not observable. Thus, a new approach where the position variables are mapped to the angle variables which are controllable so as to drive all states to zero stability was proposed in this study. The aim of the study was achieved but the downside is that it takes a longer time to achieve this stability so it is not efficient and should only be considered when absolute zero stability is the aim without considering time efficiency. The study further investigates the problem of nonlinear quadcopter system's mathematical modelling and control for stabilization and trajectory tracking using the feedback linearization (FBL) technique combined with the PD controller. The proposed control algorithms are implemented on the quadcopter model using MATLAB and analyzed in terms of system stabilization and trajectory tracking. The PD controller produces satisfactory results for system stabilization, but the FBL system combined with the PD controller performs better for trajectory tracking of the quadcopter system.

1. Introduction

A quadcopter, also known as a quadrotor, is a rotor-based aerial vehicle. They are multirotor aircrafts propelled by four rotors. To balance the torque, these rotors are built with two pairs of opposite rotors revolving clockwise and the other rotor pair moving anticlockwise. A quadcopter's dynamics are highly nonlinear; it is an underactuated system with six degrees of freedom and four control inputs which are the rotor velocities. The quadcopter is controlled by adjusting the rotors' angular speeds, which alters the quadcopter's torque and thrust characteristics. The quadcopter is designed with four rotors in cross-configuration, as shown in Figure 1. Two opposite rotors rotate in the same direction; the altitude and position of the quadcopter can be controlled by varying the rotor's angular speed. The quadcopter can be kept in a balanced position without spinning if the generated torque of the motors T1, T2, T3, and T4 is the same.

Thrust controls the quadcopter's altitude for ascending and descending and is achieved by increasing or reducing the rotational speed of motors 1, 2, 3, and 4, simultaneously. Roll is the movement of the quadcopter by tilting either left or right to allow side movements. Pitch is the movement of the quadcopter by tilting either front or back to allow

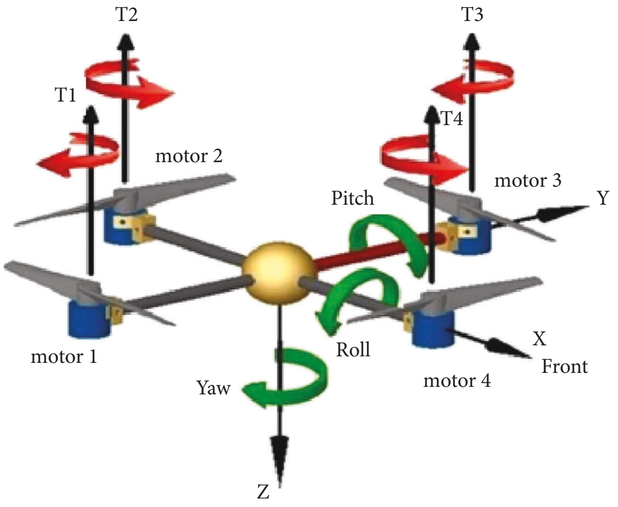


Figure 1: Quadcopter dynamics.

forward or backward movement. Yaw is movement in a clockwise or anticlockwise manner while staying level to the ground to change the quadcopter's direction. These flight motions can be achieved by controlling the rotational speeds of the four motors.

Quadcopter designs have recently been popular in UAV research. To control and stabilize the aircraft, these UAVs use an electrical control system and sensors [1]. The small unmanned aerial vehicles (SUAV) have become a reality, thanks to recent microcomputer technology, sensor technology, control systems, and dynamics theory advancements. Due to their compact size, low cost, and agility, these systems are being used in various applications.

Indoor quadcopters are usually relatively small in size and cannot utilize GPS for absolute positioning. They thrive in the absence of strong winds and relatively stable light conditions and their missions are usually shorter than those of an outdoor quadcopter [2, 3]. Outdoor quadcopters are relatively bigger in size than the indoor quadcopter and can utilize GPS for absolute positioning. They are more durable and can fly for longer periods of time than indoor quadcopters [2]. A DJI Phantom 4 is an example of an outdoor quadcopter [4].

In the 1920s and 1930s, a few manned designs existed. These vehicles were among the first heavier-than-air vertical take-off and landing (VTOL) vehicles to be successfully tested [1]. However, early prototypes performed poorly due to weak stability augmentation and restricted control authority, while latter versions required too much pilot effort. Historically, basic linear control techniques were used to ensure easy computation and flight stability, but due to improved modeling techniques and faster computational capabilities, it is now possible to run comprehensive nonlinear techniques in real-time [5]. The major components of a quadcopter are frame, propeller, motor, speed controller, flight controller, battery, transmitter, and receiver [6, 7]. Linear control techniques can be implemented on the controller of a quadcopter by linearizing the system about an equilibrium point, but such approximation may not preserve the dynamics of the system at every point and are not usually very effective in practical scenarios.

As the range and complexity of applications for quadcopters expand on a daily basis, the control techniques utilized on the system must also improve. This paper aims to develop a mathematical model of a quadcopter system and implement nonlinear control techniques on the derived model for stabilization and trajectory tracking of a quadcopter. The full state feedback linearization (FBL) system control based on dynamic inversion for a nonlinear system of a quadcopter is derived and computed in this paper without the use of small-angle approximation.

2. Literature Survey

The mathematical modelling and control of a quadcopter are presented in [8]. A matrix approach and the use of bond graphs to describe the dynamic model of a quadrotor are elaborated in [9]. A more detailed derivation of the system dynamics is given in [10]. The work in [8] shows the basics of quadcopter modelling and control to serve as a stepping stone for future research advancement. A detailed study of the mathematical model of the quadcopter dynamics is given, and the differential state equations of the system dynamics are derived from both the Newton–Euler and Euler–Lagrange equations. A simple mathematical model is presented in this paper, neglecting several aerodynamic effects and the modelling of the electric motor spinning the rotors. The developed model is simulated to analyze the behavior of the system, and control algorithms are implemented for stability and trajectory control of the system.

The PD controller is used to achieve this, and a heuristic method is developed for trajectory control of the quadcopter flight. Integration of the PD controller and the heuristic method is used to minimize the disturbance on the quadcopter system caused by external forces. The PD control algorithm was integrated into the heuristic method because the heuristic method development did not account for unmodelled disturbances (wind). A decline in the performance of the PD controller was observed if the parameter values were extremely small or high. The necessity for an actual experimental prototype was identified, so those realistic and accurate findings could be acquired and comparison between the simulated results and real-life measurements would be possible.

The PID controller may utilize the three controller terms of proportional, integral, and derivative gains to implement precise and optimum control output [11]. The controller attempts to minimize the error over time by adjusting the control variable $u(t)$. The proportional gain produces an output that is proportional to the error value $e(t)$. This output is obtained by multiplying the error value by a proportional constant. The integral gain considers the past error values and seeks to reduce the residual error by adding a control effect obtained from the integration of the cumulative previous errors. The derivative gain anticipates or predicts the future behavior of the error and exerts a control effect produced by the rate of change of the error with respect to time [12]. The linear quadratic regulator (LQR) is a popular linear control algorithm that provides optimally controlled feedback gains to enable stability and high-performance design of closed-loop systems. The LQR controller is implemented on a linearized system and makes use of a mathematical algorithm to minimize a cost function with when the weighting factors are provided. The cost function is often described as the sum of deviations of certain key measurements from their desired values [13].

Several control algorithms have been researched for quadcopters for attitude stabilization and trajectory tracking. The objective is to implement a control algorithm that permits the quadcopter's states to converge to an unpredictable set of time-varying reference states. It is possible to control the quadcopter by linearizing the system dynamics around a stable point and applying linear control techniques, as shown in [14, 15]. Nonlinear control algorithms such as backstepping, sliding mode [16, 17], and feedback linearization [16, 18] are effective for quadcopter control as better performances have been observed from their implementation. Numerous control algorithms for attitude control have been studied. These include the PID controllers [5, 8, 10], inverse dynamic control, sliding-mode controller [19], linear quadratic regulator [5], and feedback linearization control [18–20]. A hierarchical control approach is used in [10], where different levels (controlling rotor rotational speed, vehicle attitude, and trajectory position) form feedback loops.

A review of control algorithms for autonomous quadrotors is presented in [20]. Several control algorithms are analyzed to propose hybrid systems that would combine the advantages of more than one control algorithm. The focus of this paper was to implement the optimal control algorithm of the quadrotor manned aerial vehicle for game counting in the protected game reserves in Africa. The control algorithms analyzed in this paper are PID, linear quadratic regulator (LQR), sliding-mode control, backstepping control, adaptive control, and artificial neural networks. From the review, no particular control algorithm gives the best performance in all required features in terms of fast response, disturbance rejection, stability, robustness, adaptability, optimality, and tracking ability. A hybrid control system could have the best combination of some of these features but does not guarantee overall optimum performance. Certain features are more important depending on the application of the quadrotors; hence, compromises would have to be made to implement control algorithms that excel in such areas of application. A quadrotor to be used for game counting is designed to prioritize specific characteristics such as high endurance, high agility, high cruising, low noise, and vertical take-off and landing ability. This paper concluded that the designed quadrotor would be suitable for assisting nature conservationists in game counting and obtaining accurate animal statistics. A PID controller with provision for angular acceleration feedback is utilized in [21] for attitude stabilization. This causes significant gain increase resulting in higher bandwidth.

A comprehensive mathematical modelling of the kinematics and dynamics of a quadcopter system is described in [22]. The kinematic aspect of the modelling depicts the quadcopter motion without considering the forces acting on it, whereas the dynamics explain the forces causing the motion. This gives an overview of the response of quadcopters when under the influence of extraneous forces. Nonlinear state equations for the system were derived, and the manual method to obtain minimum steady-state error for PID tuning is presented. This research resulted in a quadcopter with a composite frame assembled and implemented in real-time for pitch, roll, and stabilization with PID control. The results yield adequate performance when the hardware configuration is at the hover point.

A thesis on quadcopter flight mechanics model and control algorithms is shown in [2]. The focus of this thesis was to develop a nonlinear model of quadcopter flight mechanics with suitable control algorithms for stabilization and implement in MATLAB/Simulink. The quadcopter dynamics were derived from the equation of motion and forces, and a nonlinear model was obtained based on these dynamics. Aerodynamic effects such as nonzero free steam and blade-flapping are ignored in the derivation of the system equations, but other forces such as air friction and drag forces are considered. For analysis of the system, the model is linearized at a stable hover point, and both the linear and nonlinear models are analyzed. The gradient descent method is used for controller tuning to obtain optimum control parameters, and the system is simulated for a given period of time. A better performance was observed from the automatically tuned PID controller (using the gradient descent method) than the manually tuned PID controller.

The design and control of quadrotors in relation to autonomous flying [1] are a thesis that entails the modelling, design, and control of small flying drones with a focus on vertical take-off and landing (VTOL) systems. In this thesis, an autonomous quadrotor called OS4 is modelled and simulated with various controllers. A more complex mathematical model is presented here as the system dynamics account for more realistic aerodynamic coefficients, sensor, and actuator models. The OS4 quadcopter model is simulated with five different control algorithms. The first one was based on Lyapunov theory and was implemented for attitude stabilization, but after simulation on OS4, it was observed that it was not stable enough to allow hover flight. The second control algorithm implemented was the PID controller, and it was suitable for attitude stabilization of the quadcopter system when flying near hover, but this was only possible in the absence of huge disturbances. The third control algorithm was an LQR controller that presented adequate stabilization, but this control algorithm was less adaptive than the PID. The fourth control algorithm was the backstepping controller, which achieved excellent stabilization and control in relatively large disturbances. The fifth control algorithm implemented was the sliding-mode technique, but this control algorithm did not give desirable results because the switching characteristics of the controller were not compatible with the dynamics of the quadcopter system. After the implementation and analysis of these five control algorithms, a suitable control algorithm was developed by integrating the PID and the backstepping algorithm called integral backstepping. This was used for altitude, attitude, position, and trajectory control, and the implementation of this control algorithm on the OS4 enabled the system to take-off, hover, land, and avoid collisions. The thesis noted that the OS4 was the first collision avoidance quadcopter system.

A fundamental drawback in quadrotor control is related to their high energy and power consumption [23]. Therefore, a controller was proposed with a power reduction methodology and cuckoo search algorithm (CSA) for tuning the controller gains to control the quadrotor position by a PID controller while the orientation control is achieved through a model-based controller [24]. A control strategy composed of a sensor fault diagnosis unit and a disturbance observer-based sliding-mode controller for quadrotor helicopters to simultaneously accommodate sensor faults and external disturbances were developed and analyzed [25]. Another fault-tolerant control scheme [26] was proposed for attitude control using adaptive sliding-mode backstepping control to accommodate the actuator faults, despite actuator saturation limitation and disturbances.

Classical controllers for UAVs and quadcopters involve the use of small-angle approximations [27]. For instance, a cascaded PID controller was designed for trajectory tracking but was made to hover at an altitude where the nonlinear model is linearized [28]. Sophisticated architectures such as genetic algorithm, neural networks, fuzzy logic [29], and adaptative control have to be integrated to classical controllers in order to achieve stability, robustness, optimal control, complex manoeuvres, and large angles. An example is a novel deep learning-based robust nonlinear controller that improves control performance of a quadrotor during landing by combining a nominal dynamics model with a deep neural network (DNN) that learns high-order interactions. A nonlinear feedback linearization controller was designed using the learned model and prove system stability with disturbance rejection [30]. A self-tuning PID controller based on fuzzy logic was designed to cater for payload weight variation. An adaptive robust control (ARC) was utilized to compensate for the parametric uncertainty for the UAVs altitude control [31]. The comparison between the conventional PID and self-tuning PID base on fuzzy logic shows the superiority of the self-tuning PID compared to the PID for variable payload weight [32]. The effectiveness of the PID controller was enhanced by the implementation of a genetic algorithm to autotune the controllers in order to adapt to changing conditions [33]. Five different control systems that include a traditional PID controller, two fuzzy-PD controllers, and two controllers autotuned using a GA were designed based on the developed dynamic model and their performance was compared. The fuzzy-PD controllers showed effectiveness with large angles and complex paths.

In this section, previous works on modelling and control of a quadcopter were discussed and reviewed. A precise overview of those topics was analyzed and examined to provide a more rounded understanding of the quadcopter system components, modelling, and control. From the papers reviewed, the importance of implementing a good control for a quadcopter is emphasized and different linear and nonlinear control algorithms are studied. In this research, the FBL system control based on dynamic inversion for the nonlinear system will be derived and computed without the use of small-angle approximation in order to achieve greater accuracy in tracking of trajectory. The small-angle conditions of less than $\pi/2$ are realistic for many applications of a quadcopter but extremely small-angle approximations cannot achieve extreme manoeuvers [34]. This paper will focus on presenting an adequate mathematical representation of the system dynamics and effective control techniques for a quadcopter.

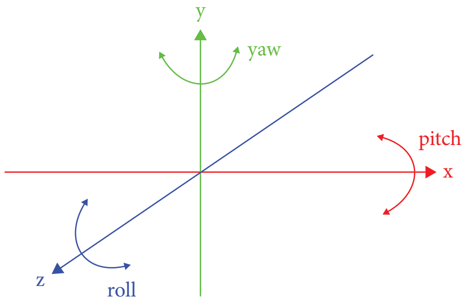


Figure 2: Six degrees of freedom.

3. Methodology

This section describes the mathematical modelling, system stabilization, and trajectory tracking of a quadcopter. A state-space model of the quadcopter in terms of rotational and translational dynamics is first developed. For the quadcopter system stabilization, the control objective is to control the attitude variable and the linear motion of the quadcopter. Feedback linearization is used to linearize the nonlinear system dynamics so that linear control methods can be used. Trajectory tracking to take the quadcopter system from its current position to the desired position using the quadcopter's rotor angular velocities is achieved using both PD control (using small-angle approximation) and FBL (without small-angle approximation) control.

3.1. Mathematical Modelling. A quadcopter is an aerial vehicle with four rotors that enable motion in different directions. Proper explanation of the system dynamics would require comprehension of the 6 degrees of freedom concept, which presents the position and orientation of the quadcopter in three dimensions (3-D).

The 6 degrees of freedom (Figure 2) describes the position of a body with six coordinates categorized in two reference frames. The first reference frame is the fixed coordinate system known as the inertial or Earth frame, which is depicted by the x, y, and z coordinates in the cardinal points of north, east, and down. The second reference frame is a mobile coordinate system known as the body frame, which is depicted by the ϕ, θ, and ψ angles with respect to the body center of gravity. The quadcopter is an underactuated nonlinear system because it has four inputs and six outputs.

The system is complex and to control it, the quadcopter is modelled on the following assumptions [1]:

(i) The structure is rigid
(ii) The structure is axis symmetrical
(iii) The center of gravity and the body-fixed frame origin coincide
(iv) The propellers are rigid
(v) Thrust and drag are proportional to the square of the propeller's speed.

3.1.1. Euler Angles. The Euler angles are three angles introduced by Leonhard Euler to describe the orientation of a rigid body in a coordinate system. They are also used to describe the relationship between two reference frames and convert the coordinates of a point in one reference frame to coordinates of the same point in another reference frame. Euler angles are denoted as ϕ, θ, and ψ for the roll, pitch, and yaw angles, respectively, and represent the rotations of a body about the axes of a coordinate system.

Any orientation of a rigid body can be achieved by the combination of the three basic Euler angles. The rotation matrices are given by the following equation [5]:

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\phi) & -s(\phi) \\ 0 & s(\phi) & c(\phi) \end{bmatrix},$$
$$R_y(\phi) = \begin{bmatrix} c(\phi) & 0 & s(\phi) \\ 0 & 1 & 0 \\ -s(\phi) & 0 & c(\phi) \end{bmatrix}, \tag{1}$$
$$R_z(\phi) = \begin{bmatrix} c(\psi) & -s(\psi) & 0 \\ s(\psi) & c(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

where $c(\phi) = \cos(\phi)$, $s(\phi) = \sin(\phi)$, $c(\theta) = \cos(\theta)$, $s(\theta) = \sin(\theta)$, $c(\psi) = \cos(\psi)$, $s(\psi) = \sin(\psi)$. The rotation matrix depicting the relationship between the inertial frame and the body frame is given as follows:

$$R = R_z(\psi) \times R_y(\theta) \times R_x(\phi),$$
$$R = \begin{bmatrix} c(\theta)c(\psi) & s(\phi)s(\theta)c(\psi) - c(\phi)s(\psi) & c(\phi)s(\theta)c(\psi) + s(\phi)s(\psi) \\ c(\theta)s(\psi) & s(\phi)s(\theta)s(\psi) + c(\phi)c(\psi) & c(\phi)s(\theta)s(\psi) - s(\phi)c(\psi) \\ -s(\theta) & s(\phi)c(\theta) & c(\phi)c(\theta) \end{bmatrix}. \tag{2}$$

R is a rotational matrix and is orthogonal such that $R^{-1} = R^T$.

3.1.2. Reference Frame Transformation. We assume $[\,x \;\; y \;\; z \;\; \phi \;\; \theta \;\; \psi\,]^T$ to be a vector of linear and angular positions in the inertial frame and let $[\,u \;\; v \;\; \omega \;\; p \;\; q \;\; r\,]^T$ be a vector of linear and angular velocities in the body frame.

Typically, the derivative gain of angular positions should give angular velocities, but the angular positions and velocities above are in a different frame, so we require some transformation matrix to convert from one reference frame to another.

Let

$$\xi = [\,x \;\; y \;\; z\,]^T, \qquad \eta = [\,\phi \;\; \theta \;\; \psi\,]^T. \tag{3}$$

Let

$$v_I = [\,\dot{x} \;\; \dot{y} \;\; \dot{z}\,]^T, \qquad \omega_I = [\,\dot{\phi} \;\; \dot{\theta} \;\; \dot{\psi}\,]^T. \tag{4}$$

Let

$$v_B = [\,u \;\; v \;\; \omega\,]^T, \qquad \omega_B = [\,p \;\; q \;\; r\,]^T. \tag{5}$$

From this statement, $v_I \neq v_B$ and $\omega_I \neq \omega_B$, instead

$$v_I = R \cdot v_B,$$
$$\omega_I = \omega_\eta^{-1} \cdot \omega_B,$$
$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & s(\phi)t(\theta) & c(\phi)t(\theta) \\ 0 & c(\phi) & -s(\phi) \\ 0 & \dfrac{s(\phi)}{c(\theta)} & \dfrac{c(\phi)}{c(\theta)} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \tag{6}$$

Conversely,

$$\omega_B = \omega_\eta \cdot \omega_I,$$
$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -s(\theta) \\ 0 & c(\phi) & s(\phi)c(\theta) \\ 0 & -s(\phi) & c(\phi)c(\theta) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}. \tag{7}$$

3.1.3. Rotational Motion. Assuming the quadcopter is a rigid body and using Euler's equations for rigid bodies, the dynamics equation in the body frame is given as follows:

$$I\dot{\omega}_B + \left[\omega_B \times (I\omega_B)\right] + \Gamma = \tau_B. \tag{8}$$

We also assume that the quadcopter has a symmetric structure with the four arms aligned with the body x- and y-axes. Therefore, the inertia matrix I is a diagonal matrix in which $I_{\mathrm{xx}} = I_{\mathrm{yy}}$.

$$I = \begin{bmatrix} I_{\mathrm{xx}} & 0 & 0 \\ 0 & I_{\mathrm{yy}} & 0 \\ 0 & 0 & I_{\mathrm{zz}} \end{bmatrix}. \tag{9}$$

The gyroscopic forces Γ are caused by the combined rotation of the four rotors and the quadcopter body such that the equation is as follows:

$$\Gamma = \sum_{i=1}^{4} J_r \left(\omega_B \wedge \hat{e}_3\right)(-1)^{i+1}\omega_{r_i}. \tag{10}$$

In matrix form,

$$\Gamma = J_r\omega_B\omega_r \quad \text{where}\omega_r = -\omega_1 + \omega_2 - \omega_3 + \omega_4. \tag{11}$$

The external torque

$$\tau_B = \left[\tau_\phi\tau_\theta\tau_\psi\right]^T. \tag{12}$$

The roll torque component τ_ϕ and the pitch torque component τ_θ are obtained from standard mechanics where $i = 1$ and $i = 3$ motors are arbitrarily chosen to be on the roll-axis while $i = 2$ and $i = 4$ are arbitrarily chosen to be on the pitch-axis.

$$\begin{aligned} \tau_\phi &= \sum r \times T = l\left(-k_t\omega_2^2 + k_t\omega_4^2\right) = lk_t\left(-\omega_2^2 + \omega_4^2\right), \\ \tau_\theta &= \sum r \times T = l\left(-k_t\omega_1^2 + k_t\omega_3^2\right) = lk_t\left(-\omega_1^2 + \omega_3^2\right). \end{aligned} \tag{13}$$

For the yaw-axis, the rotor axis is pointing in the z-direction in the body frame, and the torque created around the rotor axis is given as follows:

$$\tau_\psi = (-1)^{i+1}k_b\omega_i^2 + I_m\omega_i, \tag{14}$$

where $(-1)^{\mathrm{i}+1}$ is positive for the i^{th} propeller if the propeller is spinning clockwise and negative if it is spinning counterclockwise. The term $I_m\dot{\omega}_i$ can be ignored because it is in a steady-state, $\dot{\omega}_i \approx 0$.

Therefore, the total torque about the z-axis is given by the sum of all the torques from each propeller as follows:

$$\tau_\psi = k_b\left(-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2\right). \tag{15}$$

Therefore, the torque matrix can be written as follows:

$$\tau_B = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} lk_t\left(-\omega_2^2 + \omega_4^2\right) \\ lk_t\left(-\omega_1^2 + \omega_3^2\right) \\ k_b\left(-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2\right) \end{bmatrix}. \tag{16}$$

where k_t is the thrust coefficient, k_b is the drag coefficient, and l is the distance between the rotor and the center of mass of the quadcopter.

3.1.4. Translational Motion. Using Newtonian equation to model the linear dynamics, the extraneous forces acting on the quadcopter are given as follows:

$$\dot{m}v_1 = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + \mathrm{RT}_B + F_D. \tag{17}$$

Drag force F_D is the force acting on the system as a result of fluid friction (air resistance). A simplified equation form is used where friction is modelled as being proportional to the linear velocity in all directions.

$$F_D = -k_d v_I\,. \tag{18}$$

The angular velocity of the i^{th} rotor creates a force F_i in the direction of the rotor axis (z-direction). The combined forces create thrust T in the direction of the body z-axis such that

$$T = \sum_{i=1}^{4} F_i = k_t \sum_{i=1}^{4} \omega_i^2\,. \tag{19}$$

Since it acts in the z-axis,

$$T_B = \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} = k_t \begin{bmatrix} 0 \\ 0 \\ \sum i = 1\omega_i^2 \end{bmatrix}. \tag{20}$$

3.1.5. State-Space Model. From the rotational dynamics discussed, equation (8) can be rewritten as follows:

$$\begin{gathered} \dot{\omega}_B = I^{-1}\left(-\omega_B \times (I\omega_B) - \Gamma + \tau_B\right), \\ \dot{\omega}_B = I^{-1}\left(\begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} I_{xx}p \\ I_{yy}q \\ I_{zz}r \end{bmatrix} - J_r \begin{bmatrix} p \\ q \\ r \end{bmatrix}\omega_r + \tau_B\right), \\ \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \dfrac{\left(I_{yy} - I_{zz}\right)qr}{I_{xx}} \\ \dfrac{\left(I_{zz} - I_{xx}\right)pr}{I_{yy}} \\ \dfrac{\left(I_{xx} - I_{yy}\right)pq}{I_{zz}} \end{bmatrix} - J_r \begin{bmatrix} \dfrac{q}{I_{xx}} \\ \dfrac{-p}{I_{yy}} \\ 0 \end{bmatrix}\omega_r + \begin{bmatrix} \dfrac{\tau_\phi}{I_{xx}} \\ \dfrac{\tau_\theta}{I_{yy}} \\ \dfrac{\tau_\psi}{I_{zz}} \end{bmatrix}. \end{gathered} \tag{21}$$

From the linear dynamics discussed, equation (16) can be written out as follows:

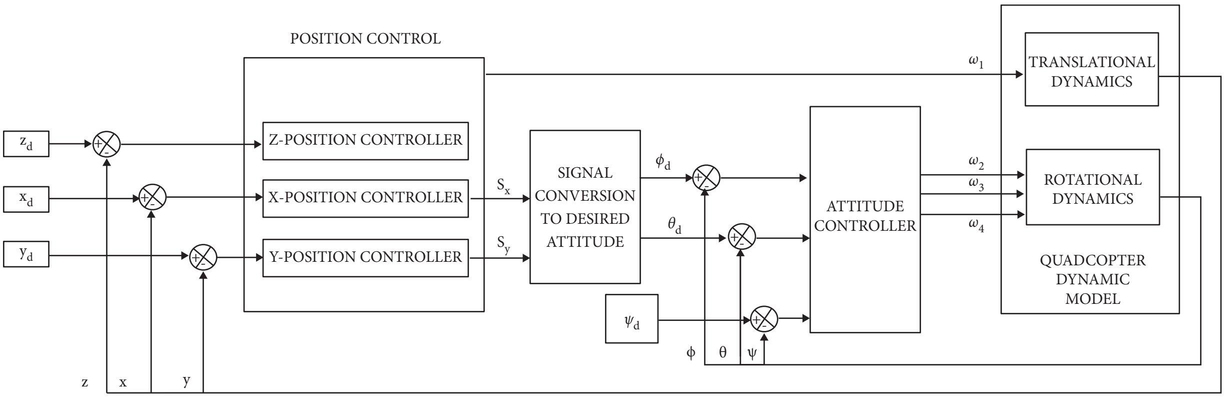


Figure 3: Control architecture.

$$[\ddot{x}\,\ddot{y}\ddot{z}] = -g\begin{bmatrix}0\\0\\1\end{bmatrix} + \frac{T}{m}\begin{bmatrix}c(\psi)s(\theta)c(\phi)+s(\psi)s(\phi)\\ s(\psi)s(\theta)c(\phi)+c(\psi)s(\phi)\\ c(\theta)c(\phi)\end{bmatrix}, \tag{22}$$

$$-\frac{1}{m}\begin{bmatrix}k_{dx} & 0 & 0\\ 0 & k_{dy} & 0\\ 0 & 0 & k_{dz}\end{bmatrix}\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\end{bmatrix}.$$

3.2. Quadcopter System Stabilization. A quadcopter has only four control inputs and primarily six outputs of interest $[\,x \;\; y \;\; z \;\; \phi \;\; \theta \;\; \psi\,]$ which makes it an underactuated system. This is resolved by separating it into two different control loops, with one working with attitude states and the other with position states. The quadcopter's angular motion is independent of the linear components, but the linear motion is determined by the angle variables. As a result, the goal is to control the attitude variable, which is independent of the linear motion and then control the linear motion. It is possible to integrate the attitude control with the trajectory controller once it is designed and optimized. The control architecture to be implemented on the quadcopter system is shown in Figure 3.

3.2.1. PID Control. The general form of a PID controller is given as follows:

$$e(t) = r(t) - y(t),$$
$$u(t) = K_P e(t) + K_I\int_0^t e(\tau)\mathrm{d}\tau + K_D\frac{d}{\mathrm{d}t}e(t), \tag{23}$$

where $u(t)$ represents the control input, $r(t)$ represents the desired state, and $y(t)$ is the current or actual state. K_P, K_I, and K_D are the gain parameters for the proportional, integral, and derivative gains of the PID controller.

The proportional and derivative gains would be used for the quadcopter control. The torque generated is proportional to the angular velocities; therefore, we set the torques to be proportional to the controller output such that

$$\tau = I \times u(t),$$
$$\begin{bmatrix}\tau_\phi\\ \tau_\theta\\ \tau_\psi\end{bmatrix} = \begin{bmatrix}I_{\mathrm{xx}}\big(k_{P\phi}(\phi_{\mathrm{des}}-\phi)+k_{D\phi}(\dot{\phi}_{\mathrm{des}}-\dot{\phi})\big)\\ I_{\mathrm{yy}}\big(k_{P\theta}(\theta_{\mathrm{des}}-\theta)+k_{D\theta}(\dot{\theta}_{\mathrm{des}}-\dot{\theta})\big)\\ I_{\mathrm{zz}}\big(k_{P\psi}(\psi_{\mathrm{des}}-\psi)+k_{D\psi}(\dot{\psi}_{\mathrm{des}}-\dot{\psi})\big)\end{bmatrix}. \tag{24}$$

The input to the quadcopter system is the angular velocity of the rotors. Recall equation (15)relates the torque to the square of the angular velocity of the rotors; there are three equations, but four unknowns. The reason for this singularity is the use of Euler angles to model the quadcopter dynamics rather than quaternions [35]; physical singularities also occur due to the limitation of underactuation in the dynamics of a quadcopter.

To allow simplification, the total thrust T which affects the acceleration in the z-direction, is set to be equal to mg. This constraint is enforced to keep the quadcopter flying. Converting this thrust equation to the appropriate reference frame and utilizing a PD to minimize the error in the z-axis,

$$T = \big(g + k_{\mathrm{Pz}}(z_{\mathrm{des}}-z) + k_{\mathrm{Dz}}(\dot{z}_{\mathrm{des}}-\dot{z})\big)\frac{m}{c(\phi)c(\theta)}. \tag{25}$$

Solving for the angular velocities of the rotor ω_1^2, ω_2^2, ω_3^2, ω_4^2 by computing equation (23) and equating it to equation (15),

$$\begin{bmatrix}\omega_1^2\\ \omega_2^2\\ \omega_3^2\\ \omega_4^2\end{bmatrix} = \begin{bmatrix}k_t & k_t & k_t & k_t\\ 0 & -lk_t & 0 & lk_t\\ -lk_t & 0 & lk_t & 0\\ -k_b & k_b & -k_b & k_b\end{bmatrix}^{-1}\begin{bmatrix}T\\ \tau_\phi\\ \tau_\theta\\ \tau_\psi\end{bmatrix}. \tag{26}$$

Simplifying equation (25) would give the equations as follows:

$$\begin{aligned}\omega_1^2 &= \frac{T}{4k_t} - \frac{\tau_\theta}{2lk_t} - \frac{\tau_\psi}{4k_b},\\ \omega_2^2 &= \frac{T}{4k_t} - \frac{\tau_\phi}{2lk_t} + \frac{\tau_\psi}{4k_b},\\ \omega_3^2 &= \frac{T}{4k_t} + \frac{\tau_\theta}{2lk_t} - \frac{\tau_\psi}{4k_b},\\ \omega_4^2 &= \frac{T}{4k_t} + \frac{\tau_\phi}{2lk_t} + \frac{\tau_\psi}{4k_b}.\end{aligned} \tag{27}$$

3.2.2. Feedback Linearization. Feedback linearization is a nonlinear control technique that has sparked a lot of interest recently. The main concept is to mathematically convert nonlinear system dynamics into (fully or partially) linear ones, allowing linear control methods to be used. The goal is to design a controller such that it exactly cancels out the system dynamics.

From equation (21), we can replace $\ddot{x} = k_{\mathrm{ax}}(\dot{x} - \dot{x}_d)$, $\dot{x} = k_{\mathrm{d}x}k_x(x - x_{\mathrm{d}})$, where k_{ax}, $k_{\mathrm{d}x}$, and k_x are gain values and replace $\ddot{y}, \dot{y}, \ddot{z},$ and $\dot{z}$ accordingly.

$$\frac{T}{m}\begin{bmatrix} c(\psi)s(\theta)c(\phi) + s(\psi)s(\phi) \\ s(\psi)s(\theta)c(\phi) + c(\psi)s(\phi) \\ c(\theta)c(\phi) \end{bmatrix} = \begin{bmatrix} k_{\mathrm{ax}}(\dot{x} - \dot{x}_d) \\ k_{\mathrm{ay}}(\dot{y} - \dot{y}_d) \\ k_{\mathrm{az}}(\dot{z} - \dot{z}_d) \end{bmatrix} + g\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \frac{1}{m}\begin{bmatrix} k_{\mathrm{d}x}k_x(x - x_d) \\ k_{\mathrm{d}y}k_y(y - y_d) \\ k_{\mathrm{dz}}k_z(z - z_d) \end{bmatrix}. \tag{28}$$

Expanding equation (26) would give the following equation:

$$\begin{aligned} k_{\mathrm{ax}}(\dot{x} - \dot{x}_d) + \frac{k_{\mathrm{dx}}k_x}{m}(x - x_d) &= \frac{T}{m}c(\psi)s(\theta)c(\phi), \\ k_{\mathrm{ay}}(\dot{y} - \dot{y}_d) + \frac{k_{\mathrm{d}y}k_y}{m}(y - y_d) &= \frac{T}{m}s(\psi)s(\theta)c(\phi) + c(\psi)s(\phi), \\ k_{\mathrm{az}}(\dot{z} - \dot{z}_d) + g + \frac{k_{\mathrm{dz}}k_z}{m}(z - z_d) &= \frac{T}{m}s(\psi)s(\theta)c(\phi) + c(\theta)c(\phi). \end{aligned} \tag{29}$$

Taking the square of both sides and adding equations together, we get the following equation:

$$\begin{aligned} \frac{T^2}{m^2} = {} & k_{\mathrm{ax}}^2(\dot{x} - \dot{x}_d)^2 + \frac{k_{\mathrm{dx}}^2k_x^2}{m}(x - x_d)^2 + \frac{2k_{\mathrm{ax}}k_{\mathrm{d}x}k_x}{m}(x - x_d)(\dot{x} - \dot{x}_d) \\ & + k_{\mathrm{ay}}^2(\dot{y} - \dot{y}_d)^2 + \frac{k_{\mathrm{dy}}^2k_y^2}{m}(y - y_d)^2 + \frac{2k_{\mathrm{ay}}k_{\mathrm{d}y}k_y}{m}(y - y_d)(\dot{y} - \dot{y}_d) \\ & + k_{\mathrm{az}}^2(\dot{z} - \dot{z}_d)^2 + \frac{k_{\mathrm{dz}}^2k_z^2}{m}(z - z_d) + g^2 + 2gk_{\mathrm{az}}(\dot{z} - \dot{z}_d) \\ & + \frac{2gk_{\mathrm{dz}}k_z}{m}(z - z_d) + \frac{2k_{\mathrm{az}}k_{\mathrm{dz}}k_z}{m}(z - z_d)(\dot{z} - \dot{z}_d). \end{aligned} \tag{30}$$

To obtain ϕ_d, from the x- and y-axis portion of equation (27), let

$$\begin{aligned} F_x &= c(\psi)s(\theta)c(\phi) + s(\psi)s(\phi), \\ F_y &= s(\psi)s(\theta)c(\phi) - c(\psi)s(\phi). \end{aligned} \tag{31}$$

From equation (30),we get the following:

$$s(\theta)c(\phi) = \frac{F_y + c(\psi)s(\phi)}{s(\psi)}. \tag{32}$$

Substituting equation (31) into equation (30),we get the following:

$$\begin{aligned} F_x &= \frac{c(\psi)}{s(\psi)}\left(F_y + c(\psi)s(\phi)\right) + s(\psi)s(\phi), \\ s(\phi) &= \frac{F_x - \left(F_y/t(\psi)\right)}{(c(\psi) + s(\psi)t(\psi)/t(\psi))}, \\ s(\phi) &= \frac{F_xt(\psi) - F_y}{c(\psi) + s(\psi)t(\psi)}, \\ \phi_d &= \sin^{-1}\left(\frac{F_x\tan(\psi) - F_y}{\cos(\psi) + \sin(\psi)\tan(\psi)}\right). \end{aligned} \tag{33}$$

Table 1: Quadcopter parameter values for simulation.

Parameter	Value	Unit
G	9.81	m/s^2
M	0.468	kg
L	0.225	m
J_r	3.357×10^{-5}	kg m^2
k_t	2.980×10^{-6}	
k_b	1.140×10^{-7}	
$\mathrm{I_{xx}}$	4.856×10^{-3}	kg m^2
$\mathrm{I_{yy}}$	4.856×10^{-3}	kg m^2
$\mathrm{I_{zz}}$	8.801×10^{-3}	kg m^2
$\mathrm{k_{dx}}$	0.25	
$\mathrm{k_{dy}}$	0.25	
$\mathrm{k_{dz}}$	0.25	

To obtain θ_d, from the x- and z-axis portion of equation (27), let

$$F_z = c(\theta)c(\phi),$$
$$c(\phi) = \frac{F_z}{c(\theta)}. \tag{34}$$

Substituting equations (32) and (33) into equation (30),

$$F_x = \frac{F_z c(\psi)s(\theta)}{c(\theta)} + \frac{F_x t(\psi)s(\psi) - F_y s(\psi)}{c(\psi) + s(\psi)t(\psi)},$$
$$F_z c(\psi)t(\theta) = F_x - \frac{F_x t(\psi)s(\psi) - F_y s(\psi)}{c(\psi) + s(\psi)t(\psi)}, \tag{35}$$
$$t(\theta) = \frac{F_x}{F_z c(\psi)} - \frac{F_x t(\psi)s(\psi) - F_y s(\psi)}{F_z c^2(\psi) + F_z s^2(\psi)}.$$

We recall

$\cos^2 x + \sin^2 x = 1,$

$$t(\theta) = \frac{F_x}{F_z c(\psi)} - \frac{F_x t(\psi)s(\psi) - F_y s(\psi)}{F_z},$$
$$t(\theta) = \frac{F_x - F_x s^2(\psi) - F_y s(\psi)c(\psi)}{F_z c(\psi)}, \tag{36}$$
$$t(\theta) = \frac{F_x c^2(\psi) - F_y s(\psi)c(\psi)}{F_z c(\psi)},$$
$$\theta_d = \tan^{-1}\left(\frac{F_x \cos^2(\psi) - F_y \sin(\psi)\cos(\psi)}{F_z \cos(\psi)}\right).$$

For equations (32) and (34), F_x, F_y, and F_z are given as follows:

$$F_x = \frac{\left(\mathrm{mk_{ax}}(\dot{x} - \dot{x}_d) + k_{\mathrm{d}x} k_x (x - x_d)\right)}{T},$$
$$F_y = \frac{\left(\mathrm{mk_{ay}}(\dot{y} - \dot{y}_d) + k_{\mathrm{d}y} k_y (y - y_d)\right)}{T}, \tag{37}$$
$$F_z = \frac{\left(\mathrm{mk_{az}}(\dot{z} - \dot{z}_d) + \mathrm{mg} + k_{\mathrm{d}z} k_z (z - z_d)\right)}{T}.$$

Table 2: Initial conditions for simulation.

Position	Value	State	Value
x	−1	Φ	10
y	2	Θ	−10
z	1	Ψ	5

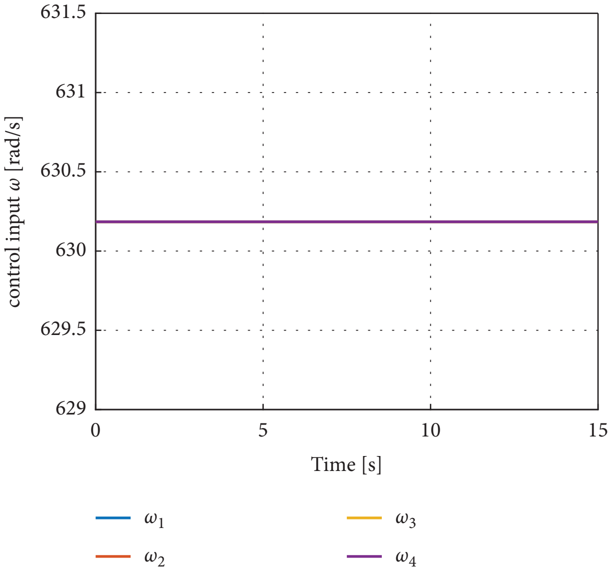


Figure 4: Control inputs for the angular velocities of the rotors.

From equations (6) and (23) in terms of desired angles, we can replace $\phi_d = k_\phi(\phi - \phi_d)$, where k_ϕ is the gain values, and replace θ_d and ψ_d accordingly:

$$\begin{bmatrix} p_d \\ q_d \\ r_d \end{bmatrix} = \begin{bmatrix} 1 & s(\phi_d)t(\theta_d) & c(\phi_d)t(\theta_d) \\ 0 & c(\phi_d) & -s(\phi_d) \\ 0 & \dfrac{s(\phi_d)}{c(\theta_d)} & \dfrac{c(\phi_d)}{c(\theta_d)} \end{bmatrix}^{-1} \begin{bmatrix} k_\phi(\phi - \phi_d) \\ k_\theta(\theta - \theta_d) \\ k_\psi(\dot{\psi} - \psi_d) \end{bmatrix},$$

$$\begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} I_{\mathrm{xx}} k_{P\phi}(p - p_d) \\ I_{\mathrm{yy}} k_{P\theta}(q - q_d) \\ I_{\mathrm{zz}} k_{P\psi}(r - r_d) \end{bmatrix} - \begin{bmatrix} \left(I_{\mathrm{yy}} - I_{\mathrm{zz}}\right) q_d r_d + J_r q_d \omega_r \\ (I_{\mathrm{zz}} - I_{\mathrm{xx}}) p_d r_d + J_r p_d \omega_r \\ \left(I_{\mathrm{xx}} - I_{\mathrm{yy}}\right) p_d q_d \end{bmatrix}. \tag{38}$$

The angular velocities ω_1, ω_2, ω_3, and ω_4 can be obtained using equation (25).

3.3. Trajectory Tracking.

3.3. Trajectory Tracking. Trajectory control aims to take the quadcopter system from its current position to the desired position by regulating the quadcopter's rotor angular velocities. Due to its complex dynamics, finding the best quadcopter trajectory is a huge challenge.

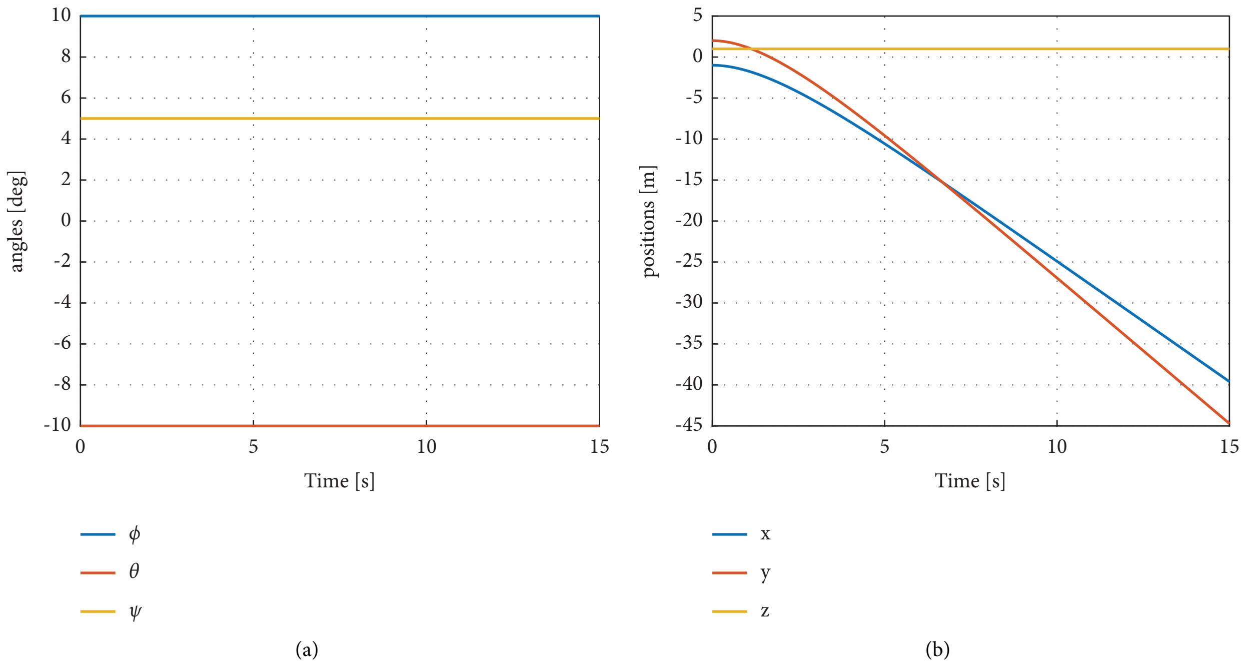


Figure 5: Control inputs for the positions x, y, and z and the angles ϕ, θ, and ψ.

Table 3: Gain values for PD controller.

Parameter	Value	Parameter	Value
$k_{P\phi}$	1.5	$k_{D\phi}$	2.6
$k_{P\theta}$	1.5	$k_{D\theta}$	2.6
$k_{P\psi}$	1.5	$k_{D\psi}$	2.6
k_{Pz}	6	k_{Dz}	1.5

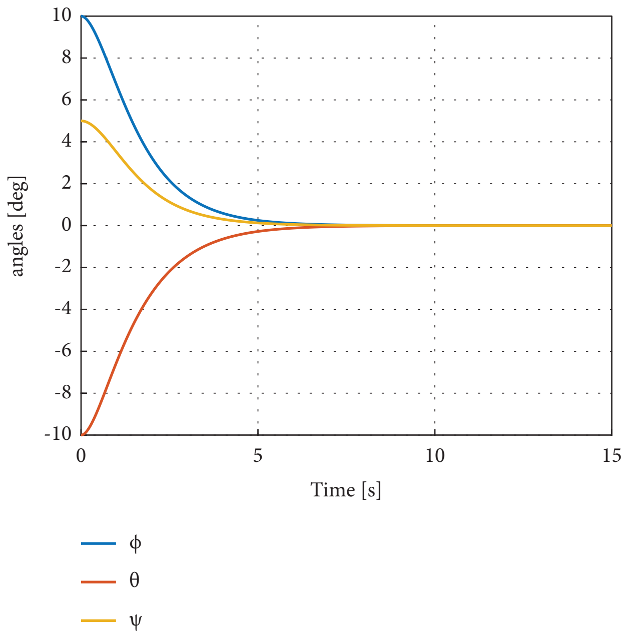


Figure 6: Result of angles ϕ, θ, and ψ as a function of time for implemented PD controller to drive the system states to zero.

3.3.1. Trajectory Tracking with PD Control. To relate the desired position x and y, which are not controllable to the desired angles of ϕ and θ that are controllable, we make two assumptions:

(i) Small-angle approximation such that $\sin(x) \approx x$ and $\cos(x) \approx 1$.

(ii) The desired ψ angle is zero.

From the above assumptions, equation (21) can be simplified as follows.

Equations (27) and (28) can be rewritten as follows:

$$\begin{aligned} \theta_{\mathrm{des}} &= \frac{1}{T}\left[m\ddot{x}_{\mathrm{des}} + k_{\mathrm{d}x}\dot{x}_{\mathrm{des}} + k_{Pxy}\left(x_{\mathrm{des}} - x\right)\right], \\ \phi_{\mathrm{des}} &= -\frac{1}{T}\left[m\ddot{y}_{\mathrm{des}} + k_{\mathrm{d}y}\dot{y}_{\mathrm{des}} + k_{Pxy}\left(y_{\mathrm{des}} - y\right)\right], \\ T_{\mathrm{des}} &= m\left(\ddot{z}_{\mathrm{des}} + g\right) + k_{\mathrm{d}z}\dot{z}_{\mathrm{des}}. \end{aligned} \tag{40}$$

θ_{des}, ϕ_{des}, and T_{des} represent the desired values of pitch angle, roll angle, and thrust.

3.3.2. Trajectory Tracking with FBL Control. To relate the desired position x and y, which are not controllable to the desired angles of ϕ and θ that are controllable using FBL control techniques, we make two assumptions:

(i) No small-angle approximation

(ii) he desired ψ angle is zero

4. Simulation and Results

The dynamic model and controllers are implemented in MATLAB 2021 for simulation with MATLAB programming language. For attitude stabilization, a PD controller is implemented on the nonlinear system to stabilize and drive the system states to zero. Simulation results for trajectory tracking using both PD and FBL control are presented.

The quadcopter model is simulated with parameter values, as shown in Table 1 [8]. The initial conditions assigned to the system for simulation are given in Table 2.

$$\begin{aligned} \ddot{x}_{\mathrm{des}} &= \frac{T}{m}\theta_{\mathrm{des}} - \frac{k_{\mathrm{d}x}}{m}\dot{x}_{\mathrm{des}}, \ddot{y}_{\mathrm{des}} = \frac{T}{m}\phi_{\mathrm{des}} - \frac{k_{\mathrm{d}y}}{m}\dot{y}_{\mathrm{des}}, \ddot{z}_{\mathrm{des}} \\ &= -g + \frac{T}{m} - \frac{k_{dz}}{m}\dot{z}_{\mathrm{des}}. \end{aligned} \tag{39}$$

The simulation progresses at 0.001-second intervals to a total time of 15 seconds. The control inputs (angular velocities of the rotors) are shown in Figure 4, and the positions x, y, and z and the angles ϕ, θ, and ψ are shown in Figure 5.

The control input remains constant, and hence, the angles remain the same, but the positions become unstable.

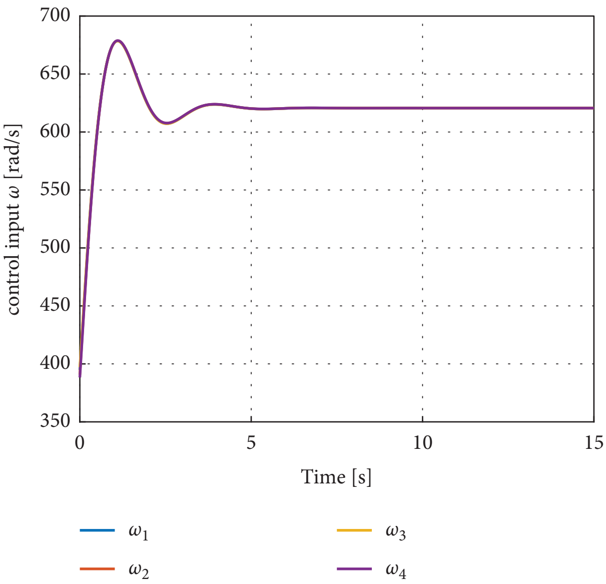


Figure 7: The angular velocities of the rotors as a function of time for implemented PD controller for attitude stabilization.

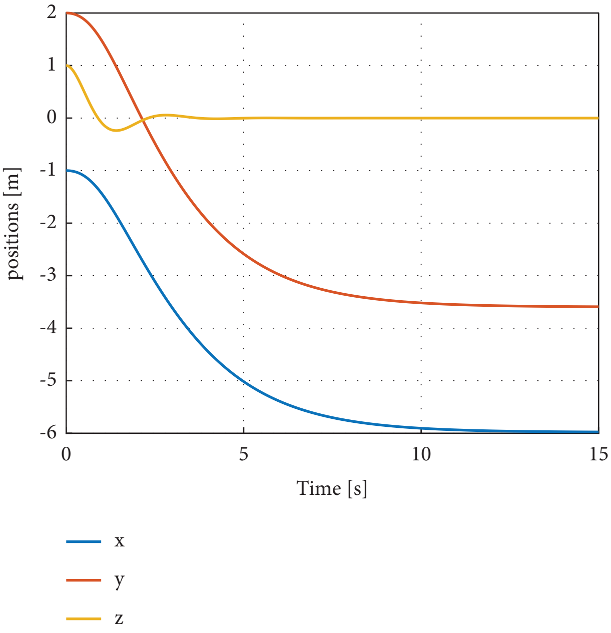


Figure 8: Positions x, y, and z as a function of time for implemented PD controller for attitude stabilization.

4.1. Attitude Stabilization. A PD controller is implemented on the nonlinear system to stabilize and drive the system states to zero. The initial conditions assigned for the system remain the same, as shown in Table 2. The control gain values for the PD controller are determined by manual tuning and are shown in Table 3.

The simulation progresses at 0.001-second intervals to a total time of 15 seconds. The angles ϕ, θ, and ψ are shown in Figure 6, the angular velocities of the rotors are shown in Figure 7, and the positions x, y, and z are shown in Figure 8. The angles are stabilized to zero after 10 seconds, and the position z is stabilized to zero after 5 seconds. The x and y positions do not stabilize to zero because the states are not observable.

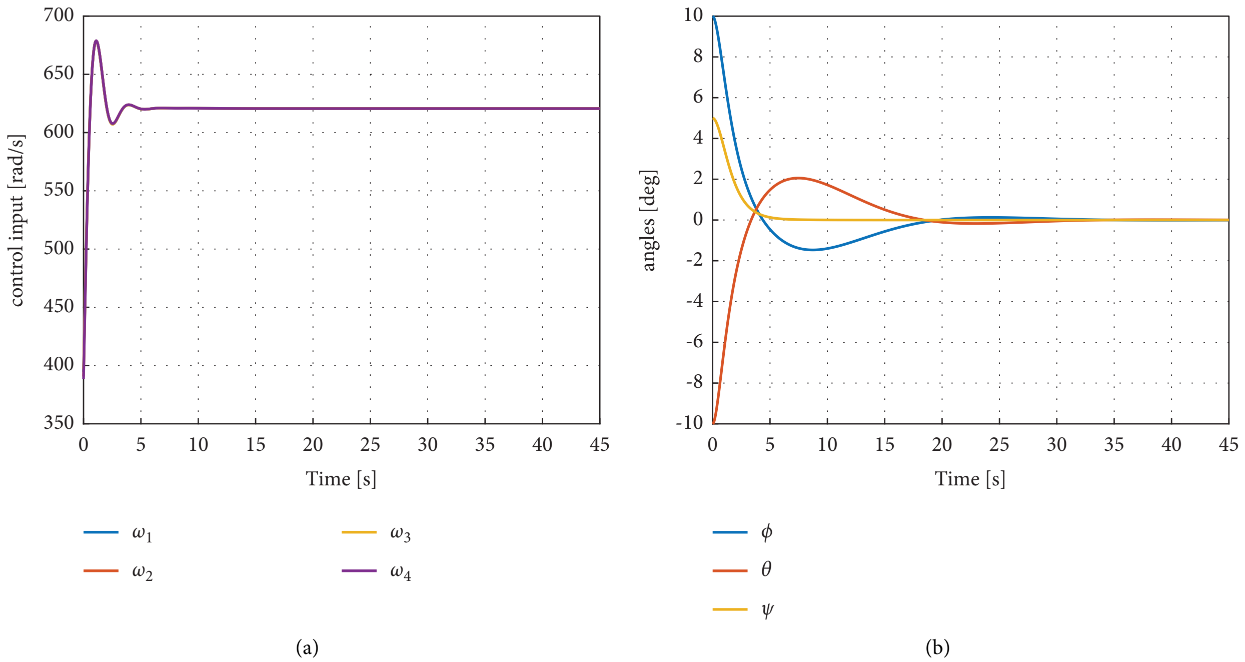


Figure 9: Control input and angles ϕ, θ, and ψ with an additional proportional controller gain k_{Pxy}, with a control gain of 0.04, implemented to drive the system states to zero.

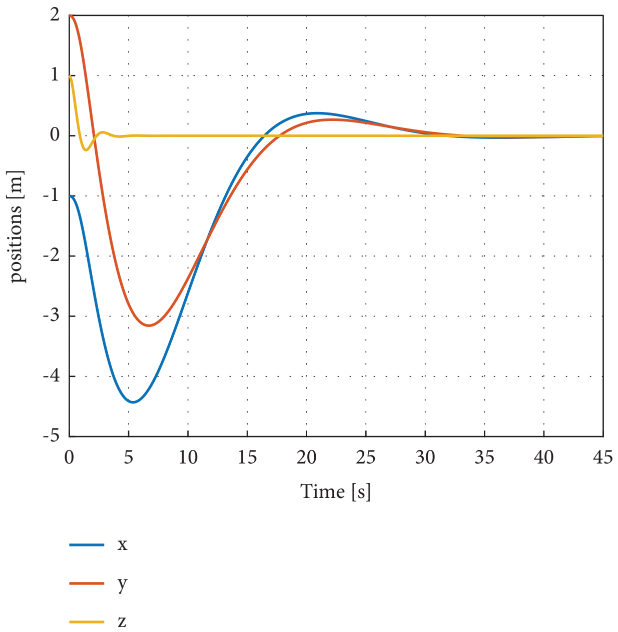


Figure 10: Positions x, y, and z with an additional proportional controller gain k_{Pxy}, with a control gain of 0.04, implemented to drive the system states to zero.

Using the concept of trajectory tracking, an attempt can be made to drive the x and y positions to zero. This can be done by mapping the zero desired states of x and y to the desired ϕ and θ states, which are controllable as shown in equations (28)–(30). An additional proportional controller gain k_{Pxy}, with a control gain of 0.04 is implemented to drive the system states to zero, and the simulation advances at 0.001-second intervals to a total time of 45 seconds. The angular velocities of the four rotors and the angles ϕ, θ, and ψ are shown in Figure 9; the modified positions x, y, and z are shown in Figure 10.

The angles ϕ, θ, and ψ are stabilized to zero after 35 seconds, and the positions x, y, and z are stabilized to zero after 45 seconds.

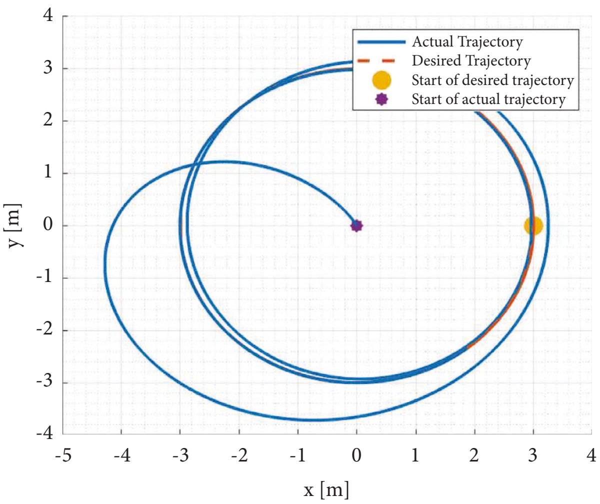
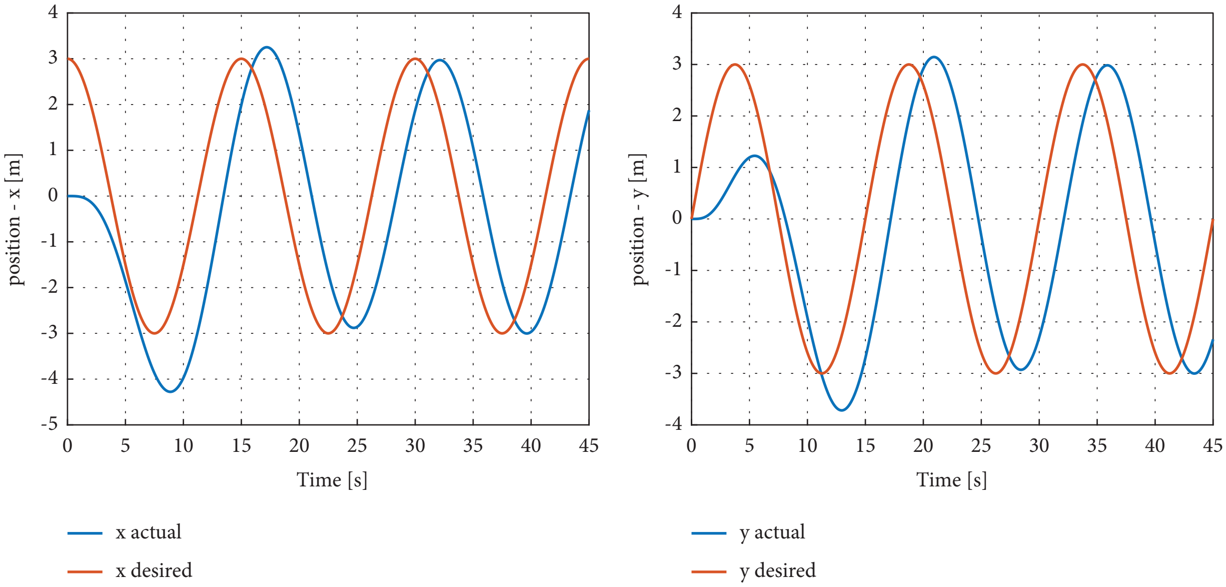


Figure 11: Desired and actual trajectories with PD Controller.



(a)

(b)

Figure 12: Coordinate-wise comparison between desired and actual trajectories of Figure 11.

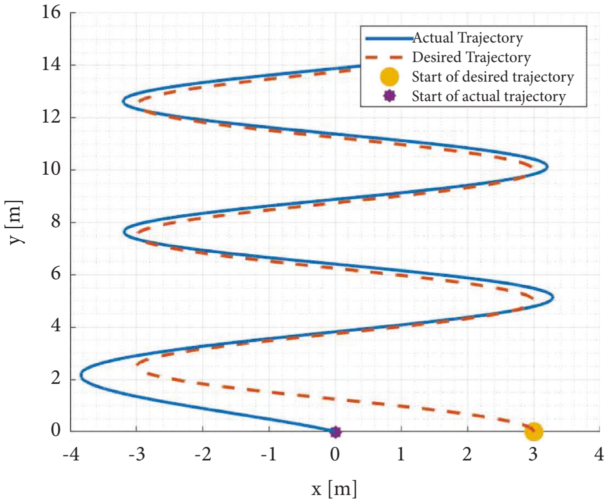


Figure 13: Desired and actual trajectories with PD controller.

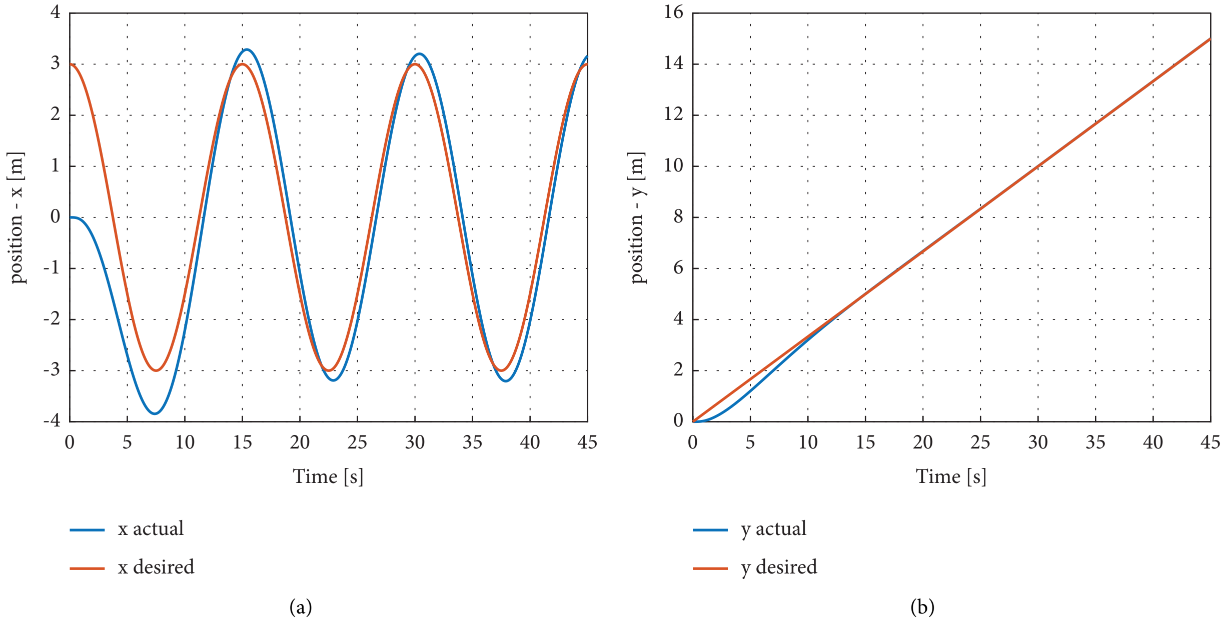


Figure 14: Coordinate-wise comparison between desired and actual trajectories of Figure 13.

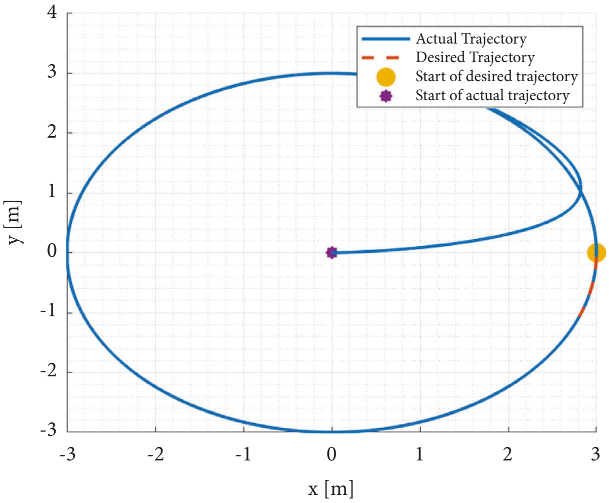


Figure 15: Desired and actual trajectories with feedback linearization and PD controller.

4.2. Trajectory Tracking. Two controls for trajectory tracking are considered and simulated for two different trajectories. The first control strategy is implemented with a PD controller, and the second strategy is a PD controller integrated with the linearized feedback system. The total time assigned for the simulation in both cases is 45 seconds.

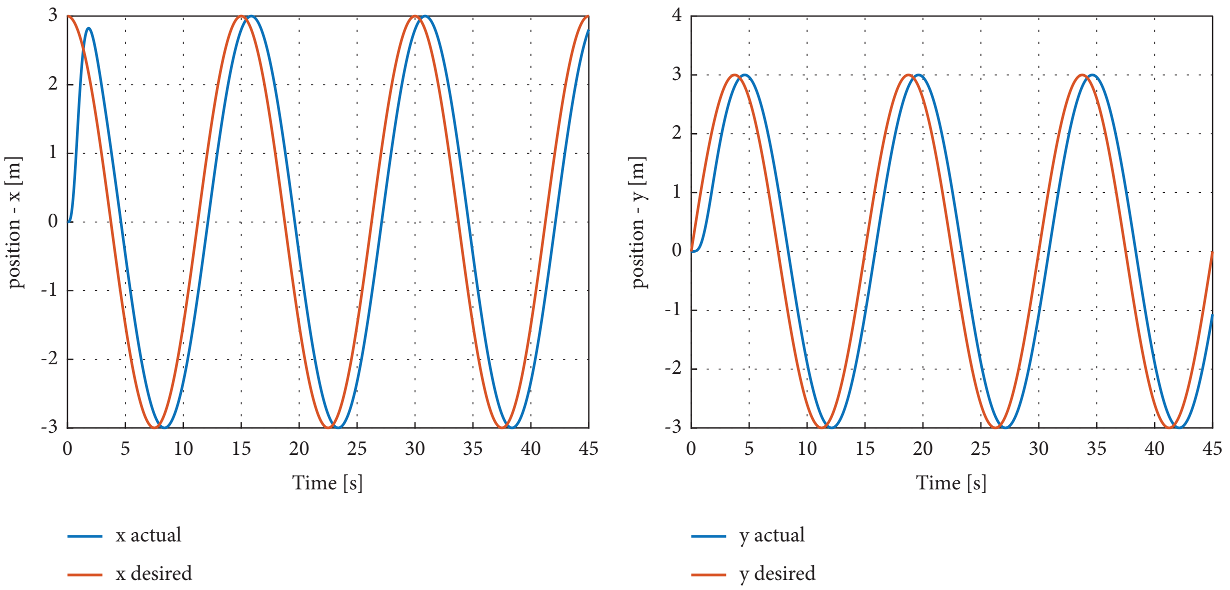
4.2.1. Trajectory Tracking with PD Controller. The comparison between the desired and actual trajectories for the first path is shown in Figure 11. Figure 12 shows the graph of the position variables against time.

Another trajectory path is simulated, and the result is shown in Figure 13. The graph of the position variables against time for this path is shown in Figure 14.

A similar result as shown in Figure 12 is also obtained; the quadcopter cannot follow the desired trajectory with precision.

4.2.2. Trajectory Tracking with Feedback Linearization and PD Controller. The feedback linearization technique is implemented on the quadcopter model and simulated with a PD controller. The desired trajectory and the actual trajectory for the first path are shown in Figure 15. Figure 16 shows the graph of the position variables against time.

Another trajectory path is simulated, and the result is shown in Figure 17. A similar result as shown in Figure 15 is also obtained; the quadcopter can lock on to the desired trajectory with more accuracy with this control.



(a)

(b)

Figure 16: Coordinate-wise comparison between desired and actual trajectories of Figure 15.

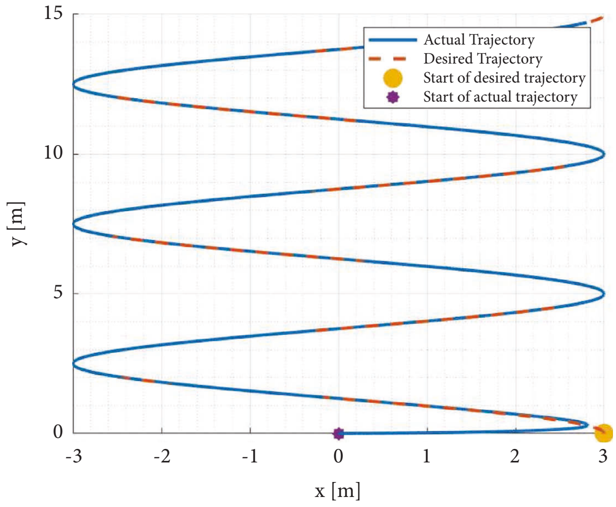


Figure 17: Desired and actual trajectories with feedback linearization and PD controller.

5. Result Analysis

Detailed analysis on the performance of the PD controller for stabilization and its integration with the FBL system is required to determine the most suitable control strategy. The results show that the feedback linearization technique with the PD control is more effective in trajectory tracking than the regular PD control.

5.1. Altitude Stabilization. As discussed earlier, the PD controller with control gains shown in Table 3 was able to stabilize the angles to the desired zero state after 10 seconds and drive the z position variable to zero in 5 seconds. The x and y positions are eventually stabilized, but they do not stabilize to the desired zero state. By mapping the desired states of x and y to the desired ϕ and θ states which are controllable, an additional proportional controller gain is used to drive all the system states to zero. However, this approach is inefficient because it takes too long for the angles and position to stabilize. The angles ϕ, θ, and ψ stabilize to zero after 35 seconds, and the positions x, y, and z stabilize to zero after 45 seconds. The ϕ, θ, x, and y states experience overshoots before stabilizing to zero. Side-by-side comparisons of the angle and position variables of the two approaches are shown in Figures 18 and 19.

5.2. Trajectory Tracking. The results of the two control strategies implemented for trajectory tracking were previously discussed. Better performance is observed in the feedback linearized system with a PD control. The

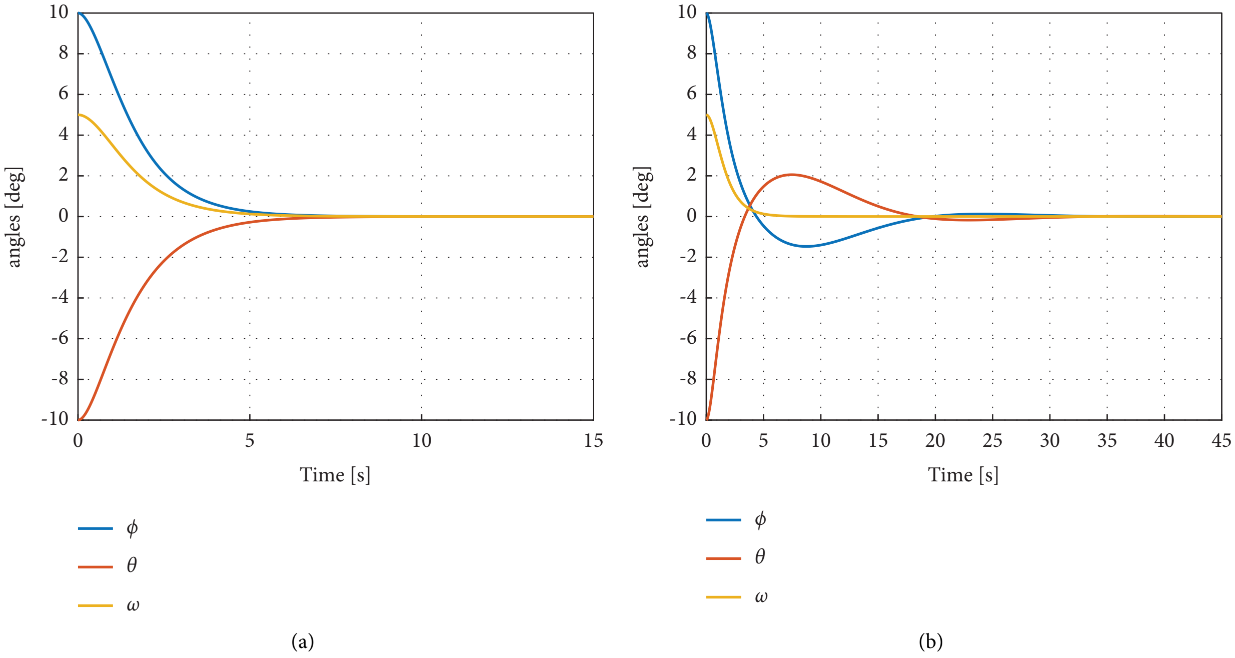


Figure 18: The angles ϕ, θ, and ψ stabilize to zero after 35 s; the ϕ and θ states experience overshoots before stabilizing to zero.

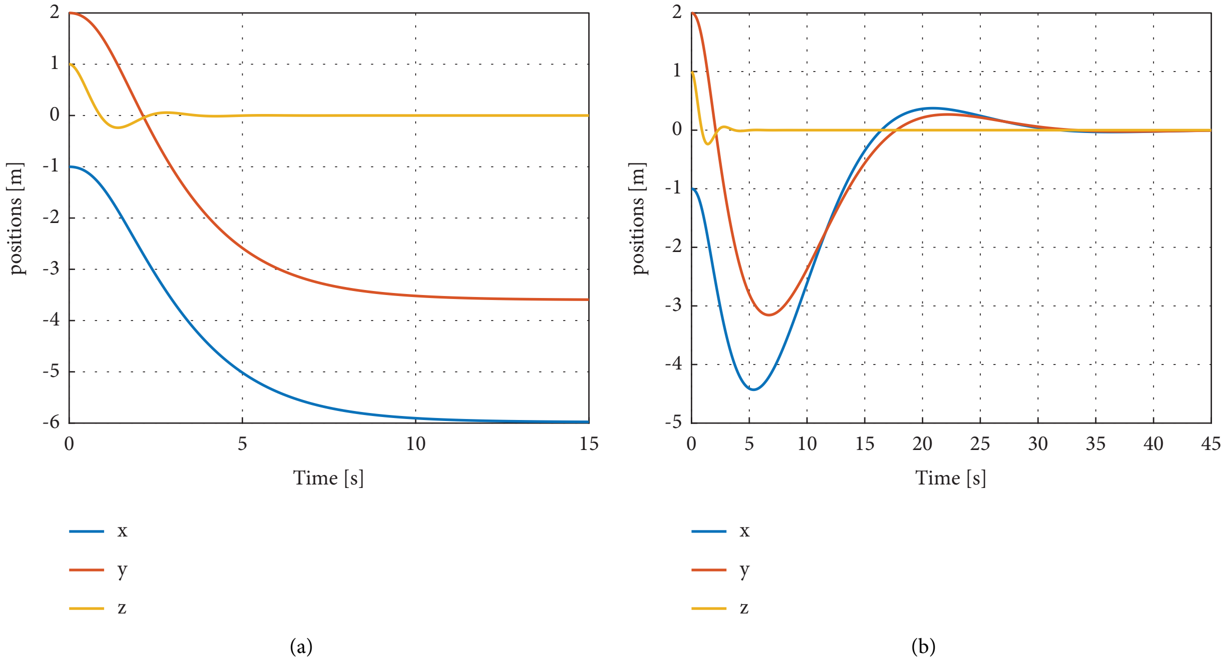
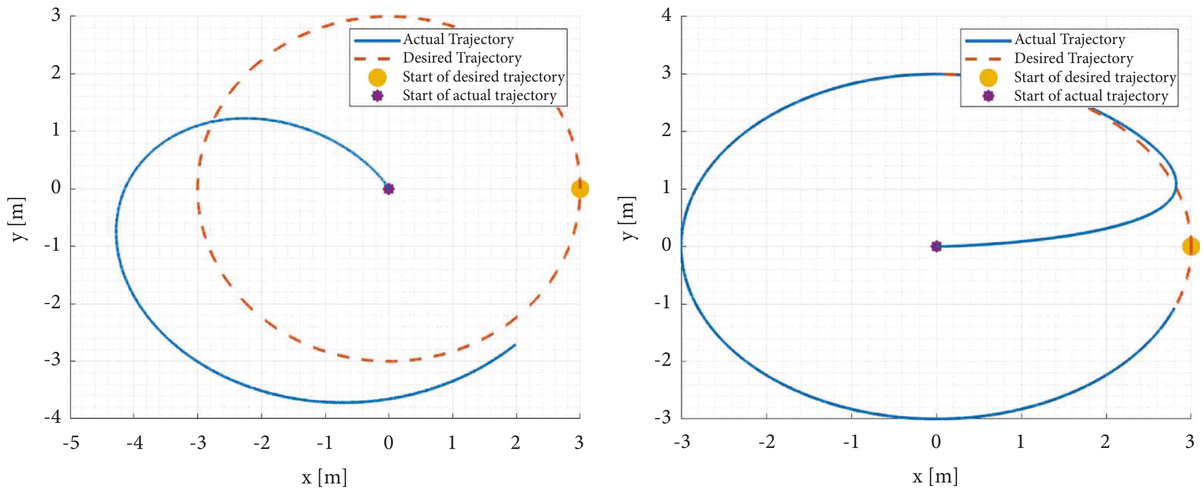


Figure 19: The position variables x, y, and z stabilize to zero after 45 s; the x and y states experience overshoots before stabilizing to zero.

quadcopter is able to lock on to the desired trajectory and follow the path with accuracy in a lesser time period. Figures 20–22 show the desired and actual trajectory plot after 15, 30, and 45 seconds, respectively, for both the PD control and the FBL with a PD control.

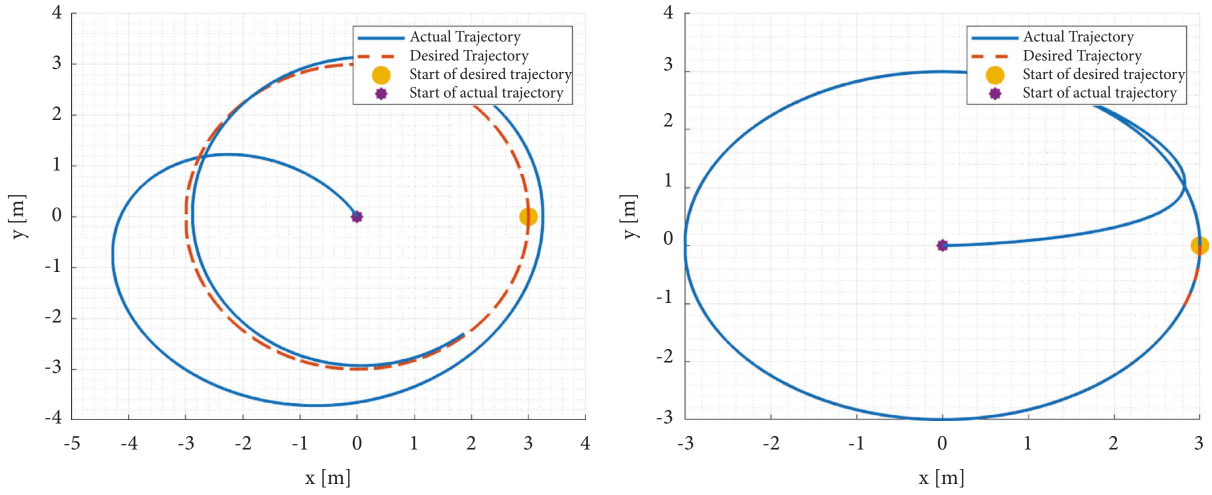
The quadcopter system simulated with the FBL and PD control is able to lock on to the desired trajectory after the first 15 seconds, while the system with only the PD control is unable to do that properly even after 30 seconds. This shows that the feedback linearization technique with the PD



(a)

(b)

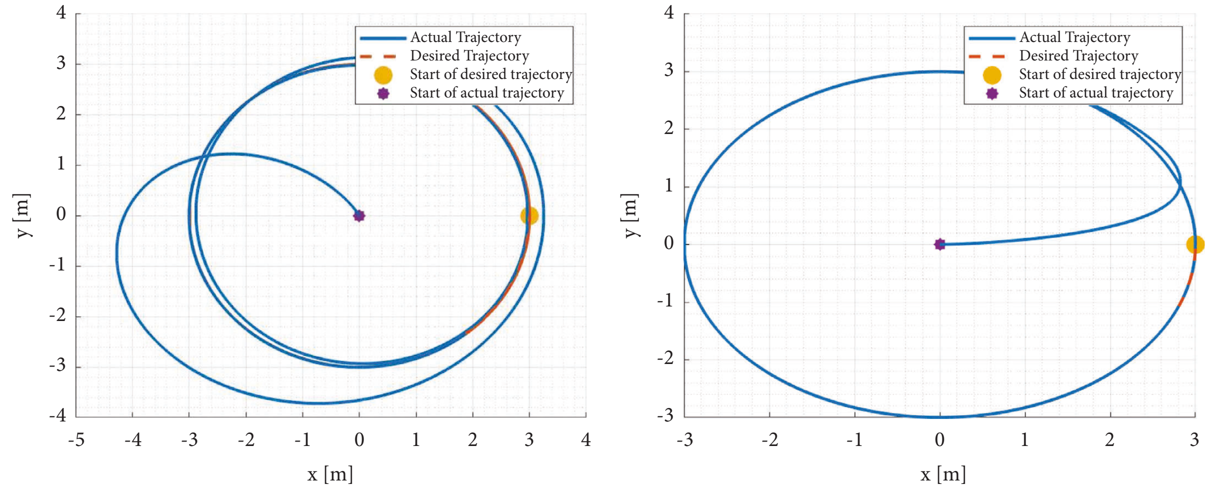
Figure 20: The trajectory for PD (a) and FBL with PD control (b) after 15 seconds.



(a)

(b)

Figure 21: The trajectory for PD (a) and FBL with PD control (b) after 30 seconds.



(a)

(b)

Figure 22: The trajectory for PD (a) and FBL with PD control (b) after 45 seconds.

control is more effective in trajectory tracking than the regular PD control.

6. Conclusion

The mathematical modeling of the quadcopter system dynamics has been developed based on a few standard assumptions. This was achieved by adequately understanding the different coordinate systems involved, Euler angles, and the Newtonian equation to express the various extraneous forces acting on the quadcopter body. The state-space model of the system dynamics was obtained using Newton's and Euler's laws. The model was tested by simulating the quadcopter flight on MATLAB with already established parameters.

Attitude stabilization and control of the quadcopter system have been achieved by using a PD controller. The simulation showed that the PD control algorithm stabilized the system states to the desired states. However, the PD controller could not stabilize the x and y positions to the desired state because the state variables are not observable. Another mapping approach was considered to drive the x and y positions to the desired zero state, but this was achieved after a more extended period of time.

Two nonlinear control strategies were implemented for trajectory tracking of the quadcopter system. The simulation showed that the PD control was inferior to the FBL with PD control. The hybridization of FBI and PD approach presented accurate tracking of the desired trajectory in a lesser period of time as reflected in Figure 22.

6.1. Recommendation to Improve the New Technique. The NDI technique used in feedback linearization of the system is implemented based on the assumption that all the states are available and measurable. Since this does not always seem to be the case, a further step in this control strategy can be the design of a suitable estimator or observer which can generate the possible outcomes of a full state from partial state variables. Another step forward in this research is the design of a controller to address the failure of one or more rotors. The quadcopter must be configured to apply the contingency algorithm once failure is detected based on inputs of the controller. Also, the proposed model and control techniques were evaluated using simulations only. Further studies can include the construction of the actual prototypes of a quadcopter to obtain more practical and accurate results.

Data Availability

No data were used in this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors acknowledge Covenant University for publication support funding.

References

[1] S. Bouabdallah, *Design and Control of Quadrotors with Application to Autonomous Flying*, Swiss Federal Institute of Technology Lausanne, Switzerland, 2007.

[2] E. Gopalakrishnan, "Quadcopter flight mechanics model and control algorithms," *Czech Technical University in Prague*, vol. 69, 2017.

[3] R. Salis, *Quadcopter Dynamics Analysis in Real Life and Simulation*, Universitat Politècnica de Catalunya, Barcelonatech, 2020.

[4] N. Intaratep, W. N. Alexander, W. J. Devenport, S. M. Grace, and A. Dropkin, "Experimental study of quadcopter acoustics and performance at static thrust conditions," in *Proceedings of the In 22nd AIAA/CEAS Aeroacoustics Conference*, Boston, MA, USA, 2016.

[5] F. Sabatino, "Quadrotor control: modeling, nonlinear control design, and simulation," 2015, https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A860649&dswid=9015.

[6] A. A. Najm and I. K. Ibraheem, "Nonlinear PID controller design for a 6-DOF UAV quadrotor system," *Engineering Science and Technology, an International Journal*, vol. 22, no. 4, pp. 1087–1097, 2019.

[7] Drone Tech Planet, "Drone arts overview," http://www.dronetechplanet.com.

[8] T. Luukkonen, "Modelling and control of quadcopter," *Independent Research Project in Applied Mathematics Espoo*, vol. 22, p. 22, 2011.

[9] R. Hossain, D. Rideout, and N. Krouglicof, "Bond graph dynamic modeling and stabilization of a quad-rotor helicopter," in *Proceedings of the 2010 Spring Simulation Multiconference*, San Diego, CA, USA, 2010.

[10] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: modeling, estimation, and control of quadrotor," *IEEE Robotics and Automation Magazine*, vol. 19, no. 3, pp. 20–32, 2012.

[11] H. A. Kiam, G. Chong, and Y. Li, "PID control system analysis, design, and technology," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 4, pp. 559–576, July 2005.

[12] A. Kaya and T. J. Scheib, "Tuning of PID controls of different structures," *Control Engineering*, vol. 35, pp. 62–65, 1988.

[13] H. Wang, H. Dong, L. He, Y. Shi, and Y. Zhang, "Design and simulation of LQR controller with the linear inverted pendulum," in *Proceedings of the 2010 International Conference on Electrical and Control Engineering*, Gangtok,India, 2010.

[14] J. C. Doyle, B. A. Francis, and A. R. Tannenbaum, *Feedback Control Theory*, , 2013.

[15] S. Bouabdallah, A. Noth, and R. Siegwart, "IEEE/RSJ international conference on intelligent robots and systems (IROS)," in *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2451–2456, Sendai, Japan, 2004.

[16] D. Lee, H. Jin Kim, and S. Sastry, "Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter," *International Journal of Control, Automation and Systems*, vol. 7, no. 3, pp. 419–428, 2009.

[17] R. Xu and U. Ozguner, “Sliding mode control of a quadrotor helicopter,” in *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, CA, USA, 2006.

[18] A. Das, F. Lewis, and K. Subbarao, “Dynamic inversion with zero-dynamics stabilisation for quadrotor control,” *IET Control Theory & Applications*, vol. 3, pp. 303–314, April 2009.

[19] I. Dikmen, A. Arisoy, and H. Temeltas, “Attitude control of a quadrotor,” in *Proceedings of the 2009 4th International Conference on Recent Advances in Space Technologies*, Istanbul, Turkey, 2009.

[20] A. Zulu and S. John, *A Review of Control Algorithms for Autonomous Quadrotors*, arXiv, Ithaca, NY, USA, 2016.

[21] G. Hoffmann, S. Waslander, and C. Tomlin, “Quadrotor helicopter trajectory tracking control,” in *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, Monterey, California, 2008.

[22] M. Etemadi, “Mathematical dynamics, kinematics modeling and PID equation controller of quadcopter,” *International Journal of Applied Operational Research*, vol. 7, no. 1, pp. 77–85, 2017.

[23] J. Moreno-Valenzuela, R. Pérez-Alcocer, M. Guerrero-Medina, and A. Dzul, “Nonlinear PID-type controller for quadrotor trajectory tracking,” *IEEE*, vol. 23, no. 5, pp. 2436–2447, 2018.

[24] R. Miranda-Colorado and L. T. Aguilar, “Robust PID control of quadrotors with power reduction analysis,” *ISA Transactions*, vol. 98, pp. 47–62, 2020 March.

[25] B. Wang, P. Huang, and W. Zhang, “A robust fault-tolerant control for quadrotor helicopters against sensor faults and external disturbances,” *Complexity*, vol. 2021, Article ID 6672812, 13 pages, 2021.

[26] N. P. Nguyen and S. K. Hong, “Active fault-tolerant control of a quadcopter against time-varying actuator faults and saturations using sliding mode backstepping approach,” *Applied Sciences*, vol. 9, no. 19, p. 4010, 2019.

[27] B. Erginer and E. Altug, “Modeling and PD control of a quadrotor VTOL vehicle,” in *Proceedings of the 2007 IEEE Intelligent Vehicles Symposium*, Istanbul, Turkey, 2007.

[28] S. Abdelhay and A. Zakriti, “Modeling of a quadcopter trajectory tracking system using PID controller,” *Procedia Manufacturing*, vol. 32, pp. 564–571, 2019.

[29] I. S. Leal, C. Abeykoon, and Y. S. Perera, “Design, simulation, analysis and optimization of PID and fuzzy based control systems for a quadcopter,” *Electronics*, vol. 10, no. 18, p. 2218, 2021.

[30] G. Shi, X. Shi, M. O’Connell et al., “Neural lander: stable drone landing control using learned dynamics,” in *Proceedings of the 2019 International Conference on Robotics and Automation (ICRA)*, Montreal, QC, Canada, 2019.

[31] O. Doukhi, A. R. Fayjie, and D. J. Lee, “Intelligent controller design for quad-rotor stabilization in presence of parameter variations,” *Journal of Advanced Transportation*, vol. 2017, Article ID 4683912, 10 pages, 2017.

[32] B. C. Min, J. H. Hong, and E. T. Matson, “Adaptive robust control (ARC) for an altitude control of a quadrotor type UAV carrying an unknown payload,” in *Proceedings of the 2011 11th International Conference on Control, Automation and Systems*, Gyeonggi-do, Republic of Korea, 2011.

[33] E. Abbasi, M. Mahjoob, and R. Yazdanpanah, “Controlling of quadrotor uav using a fuzzy system for tuning the pid gains in hovering mode,” in *Proceedings of the 10th International Conference of Advanced Computer Entertainment and Technology*, Tehran, Iran, 2013.

[34] G. Perozzi, D. Efimov, J. M. Biannic, and L. Planckaert, “Trajectory tracking for a quadrotor under wind perturbations: sliding mode control with state-dependent gains,” *Journal of the Franklin Institute*, vol. 355, no. 12, pp. 4809–4838, 2018.

[35] A. A. El-Badawy and M. A. Bakr, “Quadcopter aggressive maneuvers along singular configurations: an energy-quaternion based approach,” *Journal of Control Science and Engineering*, vol. 2016, Article ID 7324540, 10 pages, 2016.