

## Research Article

# Model-Based Design, HIL Testing, and Rapid Control Prototyping of a Low-Cost POC Quadcopter with Stability Analysis and Control

Abdullah Irfan , Muhammad Gufran Khan , Arslan Ahmed Amin, Syed Ali Mohsin, Muhammad Adnan , and Adil Zulfiqar 

*Department of Electrical Engineering, National University of Computer and Emerging Sciences, Chiniot-Faisalabad Campus, Islamabad 44000, Pakistan*

Correspondence should be addressed to Abdullah Irfan; [f180868@nu.edu.pk](mailto:f180868@nu.edu.pk)

Received 15 October 2021; Revised 29 January 2022; Accepted 22 February 2022; Published 28 March 2022

Academic Editor: Chao Liu

Copyright © 2022 Abdullah Irfan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Unmanned aerial vehicles (UAVs), particularly quadcopters, have several medical, agriculture, surveillance, and security applications. However, the use of this innovative technology for civilian applications is still very limited in low-income countries due to the high cost, whereas low-cost controllers available in the market are often tuned using the hit and trial approach and are limited for specific applications. This paper addresses this issue and presents a novel proof of concept (POC) low-cost quadcopter UAV design approach using a systematic Model-Based Design (MBD) method for mathematical modeling, simulation, real-time testing, and prototyping. The quadcopter dynamic model is developed, and controllers are designed using Proportional Integral, and Derivative (PID), Pole Placement, and Linear Quadratic Regulator (LQR) control strategies. The stability of the controllers is also checked using Lyapunov stability analysis. For verification and validation (V&V) of the design, Software-in-the-Loop, Processor-in-the-Loop, Hardware-in-the-loop testing, and Rapid Control Prototyping have been performed. The V&V methods of the MBD approach showed practically valid results with a stable flight of the quadcopter prototype. The proposed low-cost POC quadcopter design approach can be easily modified to have enhanced features, and quadcopters with different design parameters can be assembled using this approach for a diverse range of applications.

## 1. Introduction

Unmanned aerial vehicles (UAVs) are quite popular as they are small and can be flown without a pilot, either remotely or through autonomous algorithms. UAVs have applications in several domains like agriculture, military, rescue missions, etc. [1]. Several types of UAVs have been introduced; their classification can be done according to size and payload, aerodynamic configuration, applications, level of autonomy, or their range of action [2].

The quadcopter/quadrotor is a type of UAV compact and possesses diverse capabilities like better maneuverability, stationary hovering, and vertical takeoff and landing (VTOL) [3]. A quadcopter consists of a frame/skeleton responsible for supporting its components. It has four

propellers attached to Brushless DC (BLDC) motors to control the motion of the quadcopter [4]; each motor speed varies according to the control strategy to maintain its stability. It also contains electronic speed controllers (ESCs), which provide PWM signals to the motors, a flight controller, and a battery. The rotation of the motors is paired as clockwise and anticlockwise to keep the total angular momentum zero.

During the flight, the quadcopter experiences external forces like gravity, viscous friction, propellers, thrust and drag forces, etc. and thus makes the gyroscope output of the quadcopter nonlinear [5]. Such nonlinear behavior and the mechanical structure of the system make the quadcopter quite complex to control [6]. To counter these challenges, the controller must be designed in such a way that it can keep the

quadcopter stabilized under different real-world conditions. Among other control techniques, Linear Quadratic Regulator (LQR), Model Predictive Control, Proportional Integral, and Derivative (PID), backstepping, and neural network are quite prominent [7] in this field.

Many researchers have proposed different design techniques to develop mathematical models of the quadcopter. A dynamic control model is proposed by [8], using linear and nonlinear control techniques combined to create algorithms that stabilize the altitude, attitude, heading, and position in space. The proposed model includes the effects of rotor and aerodynamics with two control techniques, i.e., PD and nonlinear sliding mode controller. A limitation of this work is that the designed Multiple Input Multiple Output (MIMO) system is not controllable, so a proper MIMO controller for this model could not be developed. Reference [9] developed a dynamic model using Euler and Newton equations, and the modeling for the environmental effects was done with high accuracy to validate the model under different circumstances. The model performance results are quite good and show quite realistic results. Still, the designed model is not suitable to be used for MIMO controller design since the nonlinearities in the model are not dealt with efficiently.

Reference [10] has introduced a mathematical design for the quadcopter and developed a simulation environment to validate the software development. The developed model performed exceptionally well, but the approach used to validate the model had hardware limitations. Reference [11] proposed a sliding mode controller technique with state estimation through linear observer by employing the residual-based method for control configuration. The model can also distinguish between disturbances and faults, thus theoretically making it quite suitable for use. However, the processing cost of the proposed technique is quite extensive, thus making it nonfeasible for practical use. Reference [12] proposed dynamic modeling of the system in the MathWorks Simulink environment. PID control strategy is built for single-axis control, and STM32 is used for practical validation of the system, which showed promising results. Reference [13] proposed a cascaded PID control strategy and compared classical and cascaded PID controller performances. However, the conventional hit and trial method is employed to get PID gains rather than designing a proper autotuning setup for the system.

Apart from this, reference [14] proposed a tiltrotor model with H-configuration, which allowed relatively the undefined weight to be pulled easily theoretically. Reference [15] proposed a variable pitch-based quadrotor model, which showed disturbance compensation during a wind gust. Reference [16] proposed a model with the capability to change shape during flight by compensating for the dynamic center of gravity and continuous variation in a moment of inertia.

The Model-Based Design (MBD) approach has become quite popular in recent years due to its ability to design and test plant and controller models iteratively and detect errors early in the design process. Software In Loop (SIL) and

Processor In Loop (PIL) simulations are an essential part of the MBD approach [17]. It integrates visual and mathematical methods to address the complex problems related to signal processing, communication systems, and embedded software. It provides a mutual design environment that helps in data analysis, general communication, and system verification. It lets engineers locate and remove early errors in the system, minimizing the time and financial impact of the system [18]. The HIL-based model and testing have been proposed in [19], in which a quadcopter model setup is developed, and indoor testing for hardware is done.

Moreover, the designed system can be upgraded/modified easily as per the requirements due to the modular approach. After successfully validating SIL and PIL simulations, Hardware In Loop (HIL) testing is adopted by deploying an actual controller to identify real-world problems and issues. After the successful real-time HIL testing, rapid control prototyping is done using plant and controller hardware, and the complete system is validated in the real world. MBD has become a standard approach for system design in many industries such as aerospace, automotive, and industrial equipment manufacturing. The designed model can also be utilized for wireless-powered communication to improve energy efficiency in a distributed nonorthogonal multiple access PSN [20].

This paper presents a novel end-to-end approach of quadcopter system design through systematic design and implementation strategy, mathematical modeling, control algorithm, and real-time testing of a low-cost quadcopter by employing the MBD approach. Previously, this kind of system design and prototyping approach having similar features with low-cost components is not available in the literature to the best of our knowledge. The following are the key contributions of the paper:

- (1) Mathematical modeling a quadcopter with multiple control techniques using a model-based design approach
- (2) Defining criteria for practical implementation of the validated theoretical system
- (3) Validation of proposed system by practical testing of the prototype.

Apart from these, the controllers can be efficiently designed and adapted according to required tasks; e.g., a controller designed for speed can be modified to carry heavy loads without moving to any hit and trial approach. This increases work efficiency and cost optimization by adopting the system as per requirements.

The remainder of the paper is organized as follows: Section 2 comprises mathematical modeling of the plant, while the mathematical modeling of the controller is presented in Section 3. Section 4 discusses the results obtained from SIL and PIL simulations. Section 5 presents HIL testing on different microcontrollers, while Section 6 comprises rapid control prototyping of the proposed system, Section 7 discusses the cost and computational complexity of the system, and finally, Section 8 concludes the paper.

## 2. System Modelling

In system modeling, the mathematical models of the system through transfer functions or state-space equations are derived and developed. Primarily derived systems are nonlinear, and to design a controller for them, they need to be linearized. Newton Euler equations are used for quadcopter dynamic modeling to develop the quadcopter frames. To limit the variable constraints, some assumptions are made as follows:

- (1) The body of the quadcopter is rigid and symmetrical
- (2) The center of gravity of the quadcopter is in the center of the body
- (3) Quadcopter rotors are inflexible

$$R_{xyz} = \begin{bmatrix} \cos \theta \cos \psi & \sin \phi \sin \theta \cos \psi & \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ \cos \theta \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \theta \cos \psi & \cos \phi \sin \theta \sin \psi + \sin \theta \cos \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix}. \quad (1)$$

Quadcopter motion velocities are generated from the Inertial Measurement Unit (IMU) sensor model (embedded in the equation), and the body coordinated (Euler) rates ( $\eta$ ) are converted into angular rates ( $\omega$ ) by using the following relation:

$$\omega = R_T \dot{\eta}. \quad (2)$$

Here,  $R_T$  is given as

$$R_T = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \sin \phi \cos \theta \\ 0 & -\sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix}. \quad (3)$$

The system is decomposed into two parts to generate a dynamic mathematical model of the quadcopter. One is rotational (along  $\phi$ ,  $\theta$ , and  $\psi$  axes), and the other is the translation (along  $x$ ,  $y$ , and  $z$  axes). Here, the rotational part is actuated; however, the translational part is underactuated. Gyroscopic moments derive rotational equations of motions because the desired inertial matrix should be independent of time. The rotational equation of motion for the system is given as

$$J\dot{\omega} + \omega \times J\omega + \omega [0 \ 0 \ J_r \ \omega_r]^T. \quad (4)$$

Here,  $J_r$  is rotor inertia,  $J$  is quadcopter's diagonal inertia matrix,  $\omega$  is angular body rate, and  $\omega_r$  is rotors speed ( $\omega_r = \omega_2 - \omega_1 + \omega_4 - \omega_3$ ). As quadcopter is in symmetry, so inertia matrix is a diagonal matrix given by

$$J = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix}, \quad (5)$$

where  $I_x$ ,  $I_y$  and  $I_z$  are the moment of inertial of the system along the three perpendicular axes.

Generally, two frame references are used to express the quadcopter position: one is Earth Frame Reference (EFR), presented as  $e_x, e_y, e_z$  and the other is body frame reference (BFR), presented as  $x, y, z$ . The lists of abbreviations and symbols with descriptions have been mentioned in Tables 1 and 2, respectively.

In EFR, an initial point on ground/plane/Earth is taken as a reference, while, in the case of BFR, the center of the quadcopter is treated as a reference point, where the  $z$ -axis points downwards/ground. Roll, pitch, and yaw are used to represent the quadcopter's orientation. EFR and BFR conversion is required for some states, so the rotation matrix, given below, is used:

Individual axis moments are derived and combined using the right-hand rule. The forces impacting the quadcopter are derived to estimate each motor rotation impact on the system. After adding these two forces, the obtained equations are

$$B_M = \begin{bmatrix} F_A l (\omega_4^2 - \omega_2^2) \\ F_A l (\omega_3^2 - \omega_1^2) \\ F_M (\omega_2^2 - \omega_1^2 + \omega_4^2 - \omega_3^2) \end{bmatrix}. \quad (6)$$

where  $B_M$  is turning force acting on a quadcopter with respect to its body frame,  $F_M$  is moment acting on quadcopter model (it is the constant derived from the product of the quadcopter's air density, propeller area, and the radius of the propeller [21]),  $F_A$  is aerodynamic force, and  $l$  is propeller length, which can be modified as per the available system.

Quadcopter in hovering state exerts force in a downward direction only, which is equal to the thrust force generated by the system, i.e., against the gravitational force. The  $x$ - and  $y$ -axis movements will be zero in the hovering state because pitch and yaw angles are zero. The hovering state forces are given by

$$B_F = \begin{bmatrix} 0 \\ 0 \\ F_A (\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \end{bmatrix}. \quad (7)$$

The air friction causes drag forces given by  $D_F$  and their impact is in the opposite direction to the moment of quadcopter body, i.e.,  $B_M - D_F$ .

State-space modeling of the system helps in MIMO controller design. Quadcopter degrees of freedom (DOF) are mapped into state vector  $X$  presented as

TABLE 1: List of abbreviations.

Abbreviations	Description
EFR	Earth frame reference
BFR	Body frame reference
IMU	Inertial measurement unit
MIMO	Multiple input multiple output
LQR	Linear quadratic regulator
SIL	Software in the loop
COG	Center of gravity
MBD	Model based design
HIL	Hardware in the loop
RCP	Rapid control prototyping
ESC	Electronic speed controller
BLDC	Brushless DC motor
PPM	Pulse position modulation

TABLE 2: List of symbols.

Symbols	Description
$e_x$	X position w.r.t BFR
$e_y$	Y position w.r.t BFR
$e_z$	Z position w.r.t BFR
$x$	X position w.r.t EFR
$Y$	Y position w.r.t EFR
$Z$	Z position w.r.t EFR
$\phi$	Angular position phi
$\theta$	Angular position theta
$\psi$	Angular position psi
$\eta$	Eta - viscosity
$R_T$	Rotation matrix
$J_r$	Rotor inertia
$J$	Diagonal inertia matrix
$\omega$	Angular rate
$I_x$	Moment of inertial along the x-axis
$I_y$	Moment of inertial along the y-axis
$I_z$	Moment of inertial along the z-axis
$I_{xx}, I_{yy}, I_{zz}$	Moment of inertia for body frame principle axes
$A_x, A_y, A_z$	Drag coefficients
$F_A$	Aerodynamic force
$F_M$	Moment force
$B_M$	Quadcopter moment
$D_F$	Drag force
$k_M$	Moment constant
$k_f$	Aerodynamic force constant
$P$	Pitch
$T$	Thrust
$D$	Diameter
$Kv_{require d}$	Required $Kv$
$V_{battery}$	Battery voltage
$Kv_{nominal}$	Nominal $Kv$
$P_{out}$	Output power
$B_C$	Battery capacity
$I_{nominal}$	Nominal current
$E$	Endurance
$e$	Efficiency

$$X = [\phi \dot{\phi} \theta \dot{\theta} \psi \dot{\psi} x \dot{x} y \dot{y} z \dot{z}]' \quad (8)$$

$$= [x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 x_{10} x_{11} x_{12}]'$$

Input vectors are presented as  $u = [u_1 u_2 u_3 u_4]$ . Moment constant as  $k_M$  and aerodynamic force constant as

$k_f$  are considered for the derivation of frame forces. Thus, the system state-space model becomes

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{k_f(\omega_4^2 - \omega_2^2)}{I_{xx}} \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= \frac{k_f(\omega_3^2 - \omega_1^2)}{I_{yy}} \\ \dot{x}_5 &= x_6 \\ \dot{x}_6 &= \frac{k_M(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2)}{I_{zz}} \\ \dot{x}_7 &= x_8 \\ \dot{x}_8 &= A_x x_1 \\ \dot{x}_9 &= x_{10} \\ \dot{x}_{10} &= A_y x_3 \\ \dot{x}_{11} &= x_{12} \\ \dot{x}_{12} &= A_z x_5 - g + \left(\frac{1}{m}(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2)\right) \end{aligned} \quad (9)$$

where  $I_{xx}$ ,  $I_{yy}$ , and  $I_{zz}$  are the area moments of inertia about the principal axes of quadcopter body frame,  $m$  is mass, and  $g$  is the gravitational force.  $A_x$ ,  $A_y$ , and  $A_z$  are the drag coefficients taken from the dynamic behavior of the system of [22].

The developed state-space model is then transformed into  $A$ ,  $B$ , and  $C$  matrices and used for controller design. However, a nonlinear model is used as a plant model for testing purposes. One significant benefit of this design is that  $m$  variable can be varied as per load requirements.

### 3. Controller Design

The main objective of the state-space modeling of the quadcopter is to minimize the complexity of control strategies as MIMO systems can be efficiently dealt with in this form. Three control strategies are designed in this work: PID that is capable of altitude control, pole placement controller (designed through full state feedback), and LQR technique that offers position control of quadcopter.

**3.1. PID Controller Design.** A PID controller combines three actions to control the signal. They are responsible for delivering zero error between the feedback signal (process variable) and the desired output (setpoint). Proportional (P) control calculates error and multiplies with P constant to generate output. P-controller provides stable operation but

never reaches steady-state; thus, it needs a manual reset or basing. I-controller holds error value time until error becomes zero; this resolves the P-controller steady-state problem. A nonlinear plant can cause integral output to increase event at zero error state, also known as an integral windup condition, so a limiter is required at the output of integral action. Also, I-controller cannot predict future errors and only operates when the setpoint is changed. D-controller can predict future system behavior by considering the rate of change of error with time. If the controller observes any rate change, it acts on output increasing system response [23]. A general PID can be presented using

$$u(t) = K_p \left[ e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right], \quad (10)$$

where  $K_p$  is the gain.

A PID can only do Single Input Single Output (SISO) operation. A quadcopter has at least four controllable states (orientation with altitude control), so individual PID is required for each state. Before placing in the system, all four PIDs are tuned separately using the LabVIEW PID tuning module. PID controller in the proposed model is only designed for position control. System states are defined using LabVIEW subVIs, and indicators show results.

**3.2. Full State Feedback Control Design.** A system can have multiple degrees of freedom. A closed-loop characteristic equation for a system can be expressed as

$$s^n + a_{n-1}s^{n-1} + \dots + a_1s + a_0 = 0. \quad (11)$$

Poles in the closed-loop system can be set at the desired location by selecting the  $n$  value. A typical feedback control

system has its output  $y$  fed/added to the input  $u$ . In a full state feedback control system, again  $k$  is introduced; if the system has multiple inputs/outputs, it can be a vector or matrix that multiplies with output  $y$  and then fed/added to the input  $u$  [24]. Thus, (11) can be modified to (11), where higher orders are divided into smaller single order equations.

$$\dot{x} = Ax + Bu(-Kx + r) = (A - BK)x + Br; y = Cx \quad (12)$$

Gains  $k$  of the closed-loop system are designed by equating characteristic equations of the closed-loop system, equating desired characteristic equation. Then, these equations are compared to get gain values. Typically, in state-space representation, a system can be represented as either phase variable form or canonical form [25]. Phase variable form is defined as

$$\begin{aligned} A &= \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -a_0 & -a_1 & -a_2 & \dots & -a_{(n-1)} \end{bmatrix}; \\ B &= \begin{bmatrix} 0 \\ \vdots \\ 1 \end{bmatrix}; \\ C &= [c_1 \dots c_n]. \end{aligned} \quad (13)$$

The system modeling from the beginning is performed in phase variable form; thus, the controller design of the developed model is also done in phase variable form. It is done by introducing  $K$  (vector or matrix) in the closed-loop system of  $A - BK$  as shown in the following equation:

$$A - BK = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -(a_0 + k_1) & -(a_1 + k_2) & -(a_2 + k_3) & \dots & -(a_{(n-1)} + k_n) \end{bmatrix}. \quad (14)$$

Thus, the closed-loop characteristic equation becomes

$$\det(sI - (A - BK)) = s^n + (a_{(n-1)} + k_n)s^{(n-1)} + \dots + (a_0 + k_1) = 0. \quad (15)$$

The desired characteristic equation is given in

$$s^n + d_{(n-1)}s^{(n-1)} + \dots + s_2s^2 + d_1s + d_0 = 0. \quad (16)$$

Since the gain matrix is desired from the characteristic equation, thus by subtracting (15) from (16), we get the desired gain values  $k$ .

$$k_{(i+1)} = d_i + a_i. \quad (17)$$

The controller for closed-loop systems cannot be directly designed as the system may have one or more than one uncontrollable state. Thus, to ensure that every state of the

system is controllable, a controllability matrix is designed and then evaluated by checking its rank and determinants. The controllability matrix is generated by putting the A and B matrices of the system into (17). This causes the highest order nonsingular square submatrix, which gives the system rank, and then the determinant of the matrix is evaluated as

$$C_c = [B \ AB \ A^2B \ \dots \ A^{n-1}B]. \quad (18)$$

For this system, poles are taken at  $-2 \pm 1.96j$  to keep overshoot at less than 4%, while poles are paired at -1.5, -2, -2.5, -3, and -3.5. The pole placement controller design has  $K$  gain in the matrix form. The matrix is multiplied on the feedback terminal, i.e., from output to input.

**3.3. Linear Quadratic Regulator Design.** Optimal control theory is used to operate the dynamic system at lower costs. Linear Quadratic Controller (LQR) is a type of optimal

control that allows user custom cost/performance ratio control strategy design. LQR controller has better accuracy because, in contrast to pole placement where eigenvalues set location is specified in the designing process, LQR has performance weighting matrices that define eigenvalues set location guaranteeing system stability. So, LQR is an automated state feedback controller [26]. The cost function of the system is given as

$$J = \int_0^{\infty} (x^T Q z + u^T R u) dt. \quad (19)$$

Here,  $Q$  penalizes system performance, while  $R$  penalizes controller efforts. In this work, an iterative approach is used where the cost function is used to calculate the controller's gains.  $Q$  and  $R$  values are predefined, gains are calculated, and system response is noted for the designed controller. Since it is an iterative approach, it is repeated over 1000 times. Briefly, the steps of this approach are as follows:

- (1) Define the value of  $Q$  and change  $R$  over iteration
- (2) Design controller over this value and store system response
- (3) Change  $Q$  value by a step and change  $R$  over iteration
- (4) Design controller over this value and store system response
- (5) Repeat steps 1 to 4 over the defined iterative range of  $Q$  and  $R$
- (6) Compare performance matrices of this system to shortlist the best performing controller

After comparing results of different iterations, the selected weight matrix  $Q$  and quantity weight matrix  $R$  are  $100 \times I \times (12)$  and  $0.1 \times I \times (12)$ .

**3.4. Stability Analysis.** The Lyapunov stability analysis of the stable system is performed by using controlled  $A$  matrix obtained from  $A - BK$  with Lyapunov equation with  $Q = I_{12}$ . The generic  $P$  matrix is defined in the following equation:

$$P = \begin{bmatrix} P_{11} & P_{12} & P_{13} & \cdots & P_{1n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ P_{1n} & P_{2n} & P_{3n} & \cdots & P_{nm} \end{bmatrix}. \quad (20)$$

Unknown values of  $P$  are estimated using  $A^T P + P A = -Q$ . By solving the matrix  $P$  values, eigenvalues are calculated to be positive definite, thus proving that the system is stable.

## 4. Simulation Implementation and Results

The model-based design approach is used to develop the plant and controller models. The system is tested in two phases, SIL and PIL simulation.

**4.1. Software in-the-Loop (SIL) Simulation.** Initially, the system is designed for SIL simulation where plant and controller models are designed and simulated, and an interface is developed to monitor and control the system response. The front panel of LabVIEW is created where the 3D GUI of the quadcopter is designed, as shown in Figure 1; the object is shaped as a quadcopter in a plane environment. The position and orientation of the quadcopter with respect to the plane are set through inputs defined in the block window in the form of  $x$ ,  $y$ ,  $z$ ,  $\phi$ ,  $\theta$ , and  $\psi$

Latitude ( $X$ ) is the  $x$ -axis, longitude ( $Y$ ) is the  $y$ -axis, and attitude ( $Z$ ) is the  $z$ -axis reference point for the designed model. In contrast,  $\phi$ ,  $\theta$ , and  $\psi$  reference points are set zero as system stability on angular axes is required. Figure 2 shows the block diagram of the implemented system. Here, the system in mathematical form is defined using modules with multiple pellets (e.g., CDSim pellet, programming pellet, etc.). PID controller is designed for position control along with altitude control.

Simulations are run for thirty seconds setting the reference altitude to one meter, reference angle ( $\phi$ ,  $\theta$ , and  $\psi$ ) to zero, and environmental conditions are set as per Table 3.

Figure 3 shows the angular response of the plant model; it offers a very low angular change. The zoomed view shows the slight difference in  $\phi$ , and  $\theta$  angles to attain the required altitude of 1 meter within seconds (the altitude response is shown in Figure 4).

Although  $\psi$  response takes around 30 seconds to become zero completely, it is negligible since its value is relatively low. Pole placement and LQR controller are also capable of performing position control. Simulation for this scenario is run for thirty seconds. The reference positions of  $x$ ,  $y$  and  $z$  are set to one meter, reference angles ( $\phi$ ,  $\theta$ , and  $\psi$ ) are set to zero, and model conditions are set as per Table 1.

Figure 5 is the results comparison graph for two controllers where 5(a) shows the system linear response using LQR and pole placement controller. Figure 5(b) shows the angular response of the system using LQR and pole placement controller, where  $\phi$ ,  $\theta$ , and  $\psi$  changes can be observed minimizing as the required  $x$  and  $y$  positions are being approached. The angular response becomes zero as soon as the required position is reached.

Figure 6 shows the angular response of the quadcopter for  $\psi$  at  $45^\circ$  while the  $x$  and  $y$  are at 1 meter and altitude movement to 20 meters. The response of LQR is relatively stable since there is not much deviation with respect to the reference. The error between the reference and original position is obtained, and the mean error is calculated; the controller showed 94% accuracy. However, the accuracy is reduced if the position of the quadcopter is changed randomly and abruptly over small time steps.

**4.2. Processor in-the-Loop (PIL) Simulation.** After the system stability is verified in SIL simulation in the MBD approach, it is tested in a PIL simulator. The plant model is set in a LabVIEW simulator (without any controller) in PIL, and control techniques are deployed on the myRIO controller.

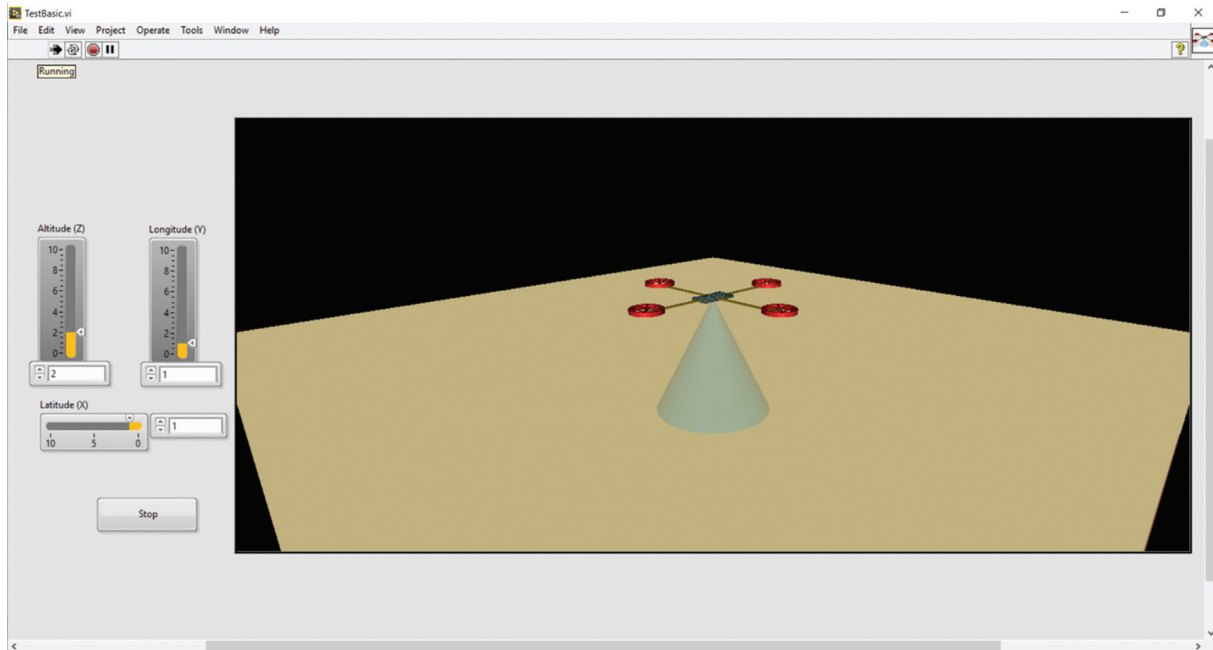


FIGURE 1: LabVIEW GUI of quadcopter flight simulation.

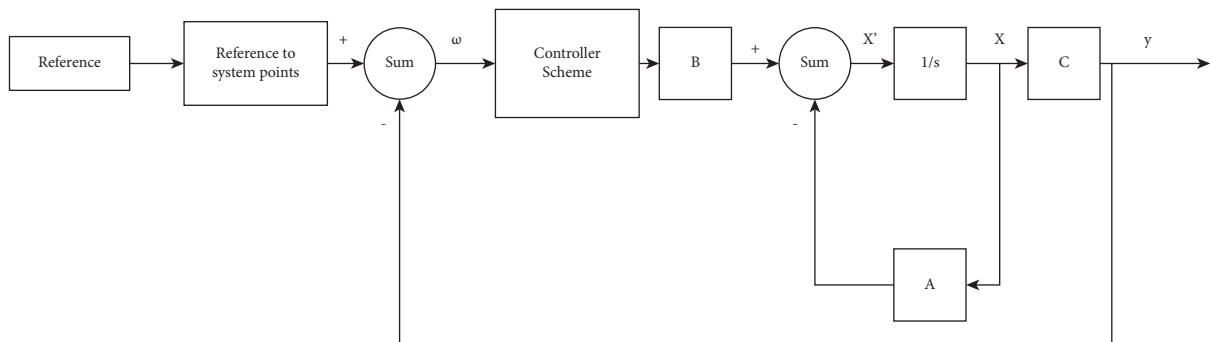


FIGURE 2: LabVIEW block diagram of SIL simulation.

TABLE 3: Model parameters.

Parameter	Value
Mass (m)	1.316 kg
Length of rotor from COG (l)	0.5 m
Lift coefficient (b)	$1.14e^{-7} \text{ N rad}^{-1} \text{ s}^{-1}$
Aerodynamic frictions	$0.45 \text{ N m}^{-1} \text{ s}^{-1}$
Body inertia	$3.57e^{-6} \text{ kgm}^2$
Gravity acceleration (g)	$9.81 \text{ m s}^{-2}$
Drag coefficient (k)	$2.98e^{-6} \text{ Nm rad}^{-1} \text{ s}^{-1}$

myRIO is a real-time embedded evaluation board made by National Instruments and is used to develop applications that utilize a microcontroller. myRIO is programmed using LabVIEW, and for PIL simulation, it is used with the simulator (where the plant model is operational). For communication between controller and simulator, there is the network stream where a local network is defined and myRIO is connected using an IP address. The designed system modules from the Virtual Instrument Software

Architecture (VISA) palette enable communication in the plant model. Thus, the plant model structure is almost the same as that in Figure 2, but the controller structure has changed since it was deployed on myRIO. Figure 7 shows the block diagram of the controller designed and deployed on myRIO.

In PIL simulation, the system showed similar results as SIL results. Besides, the model is simulated in an infinite loop to monitor its continuous response for around 48 hours, and it ran without any errors, thus justifying the controller's stability. Table 4 shows the PID gains.

## 5. Hardware-in-the-Loop Testing

The second last step for system verification in the MBD system design approach is HIL testing. To perform HIL real-time simulation and testing, motors with sensors are deployed on the fixed frame to estimate their response under different conditions. Moreover, the controller is mounted on the prototype, while the plant model is still deployed on the simulator.

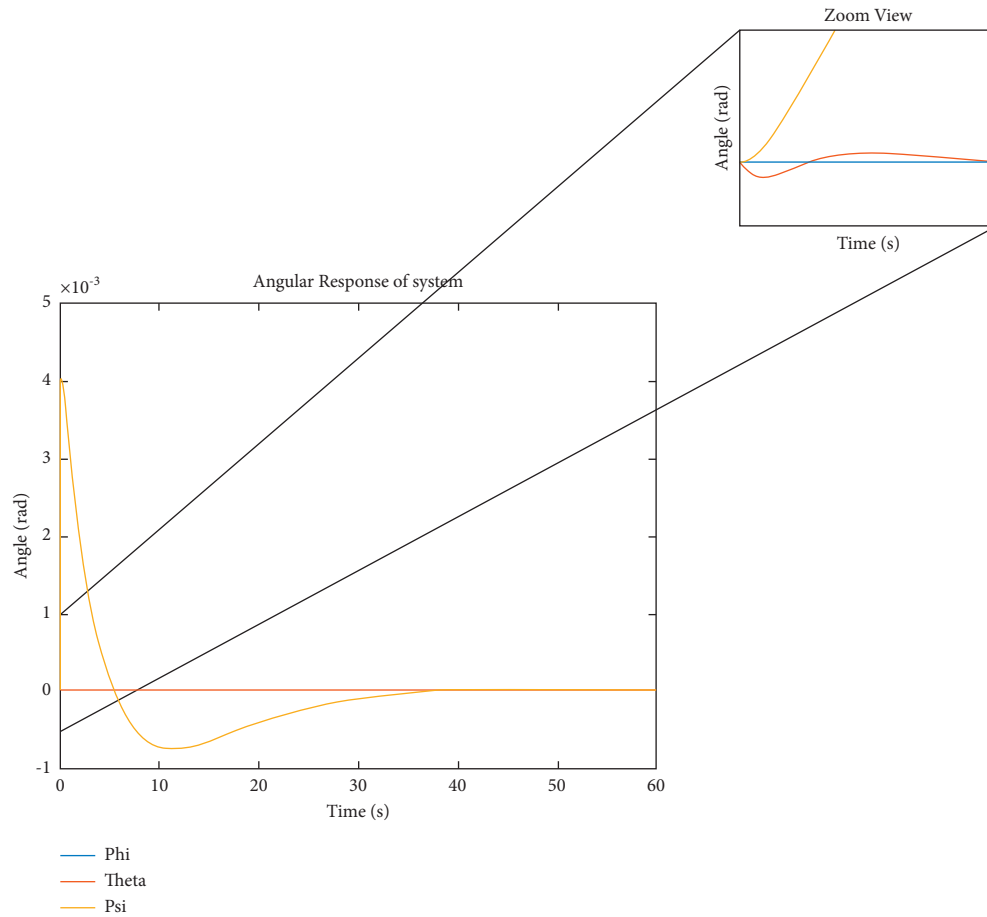


FIGURE 3: PID Angular Response of SIL simulation.

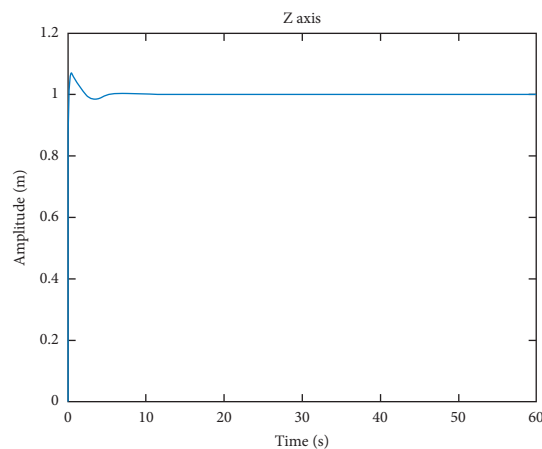


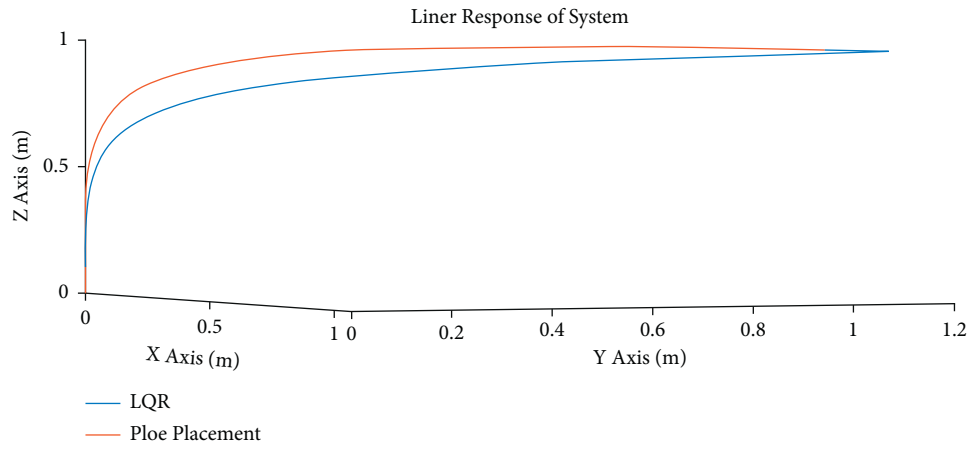
FIGURE 4: PID Altitude Response of SIL simulation.

The basic frame is wood with a 1.5-meter height and 1-meter width (on both sides). The quadcopter frame is set in the middle through the ropes with maximum flexibility of approximately  $20^\circ$  between the string and the quadcopter frame. This ensures that the system's behavior is observable, while the propeller would not hit any part of the bindings. The limitation of this fixed/bounded prototype is that the system behavior for limited orientation angles (along with

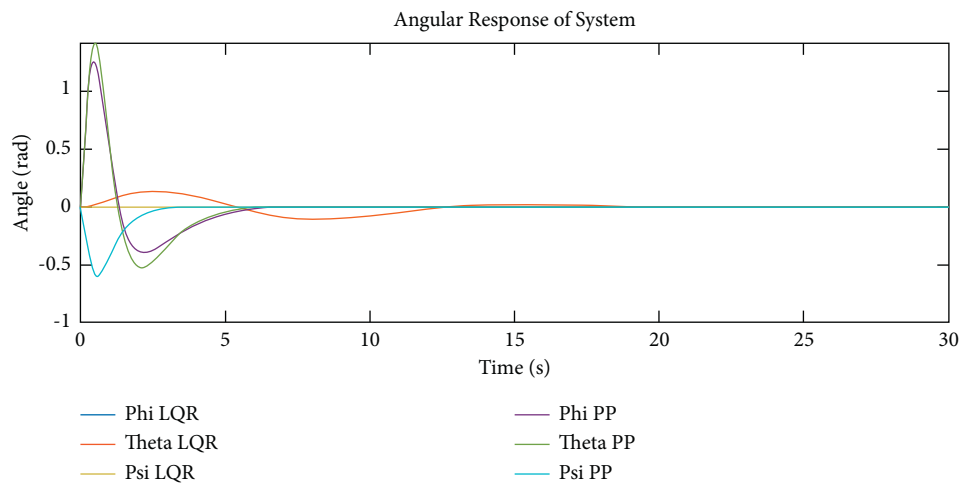
pitch and yaw) and hovering state are measurable only. The quadcopter frame is set at 1-meter height (from the base of the frame) and centered in between. The quadcopter stability was tested by using myRIO and ATmega 328p.

The controller model developed in PIL testing is further improved by defining motors' output. A complementary filter is designed to enhance the signal-to-noise ratio (SNR) (noise in the plant model is generated using additive white



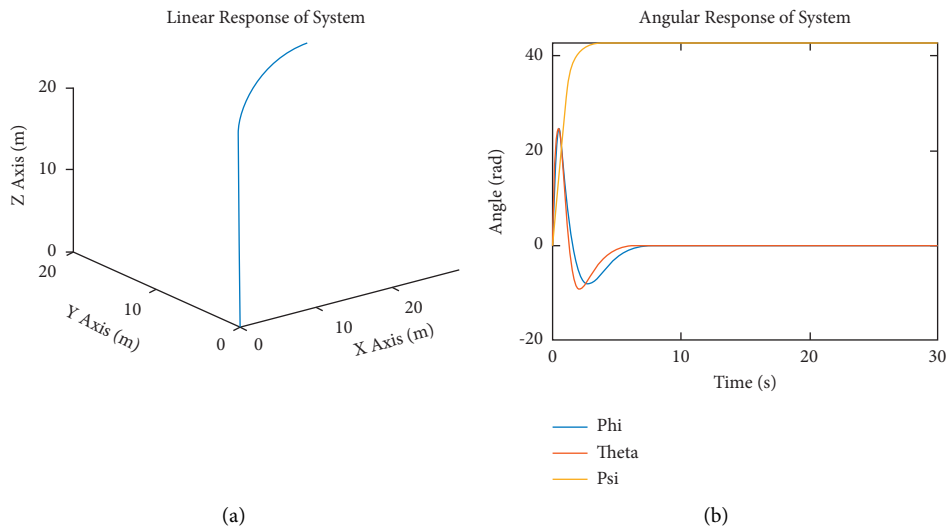


(a)



(b)

FIGURE 5: Pole Placement and LQR Controllers response comparison for system. (a) Liner Response of system. (b) Angular Response of system.



(a)

(b)

FIGURE 6: System response for psi at 45° and altitude 20 meters.

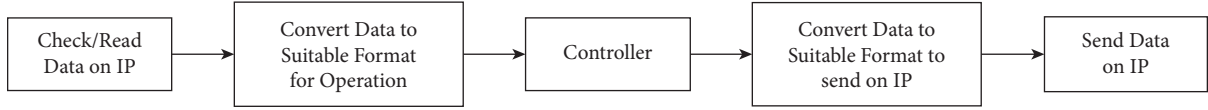


FIGURE 7: Block diagram for PIL simulation in simulator.

TABLE 4: Gains from PIL testing.

Axis	$K_p$	$K_i$	$K_d$
$\phi$	0.55	0.4	0.1
$\theta$	0.55	0.4	0.1
$\psi$	0.55	0.4	0.1
Z-axis/altitude	0.75	1.75	0.1

Gaussian noise). The complementary filters consist of two basic filters: a low pass filter (for filtering accelerometer data) and a high pass filter (for filtering gyroscope data). The signal is then passed to the controller gain, and the resultant values are serially sent to the plant model deployed in the LabVIEW simulator.

Pole placement gains and LQR controller showed better results, and the response observed from the motors with respect to variations in input was also satisfactory. The response of the PID controller to small changes was rather abrupt; after some manual adjustment of the derivative gain, the controller showed significant improvement. Table 5 shows the revised PID gains.

ATmega code to serially receive/send data from LabVIEW is developed (as shown in Figures 8 and 9).

Figure 10 shows the data obtained (without and with filter) from ATmega only as both controller strategies have shown identical responses.

## 6. Rapid Control Prototyping (RCP)

For the core of the quadcopter, i.e., the structure of the system, DJI 450 frame is selected to be used in the designed system. Propellers are responsible for propelling the quadcopter in the upward direction, and they consist of radiating blades with revolving hubs. The propellers selected for the frame are 1045. The correction factor from [27] is 1.45 (it is a unitless quantity).

The system's total weight is calculated by adding the values of different components used in the quadcopter (as given in Table 6). Here, the weight of motors, battery, and ESC is supposedly taken as an approximate value.

Force (upward exerted by the quadcopter) to weight is termed as thrust to weight ratio. It is calculated to estimate other requirements of the quadcopter. For the default chosen system, the weight from Table 6 is 1316 grams. Thus, its totally required thrust is the product of the desired thrust and the system's total weight. And for individual motors, the individual motor thrust is calculated by dividing the total thrust with total motors.

The motors' desired revolutions per minute (RPM) are computed using (24) [24] to calculate the motors and battery required for the system. The case of full throttle (i.e., motors running at the maximum possible speed) is considered to

ensure system durability. The quadcopter's already defined pitch, diameter, and desired thrust are used here. Battery voltage and capacity are taken to be 11.1 V and 5.2 A (ratings taken from the available batteries in the market). To estimate the motor RPM, the parameters of required thrust, propeller pitch, and diameter are used in the following equation:

$$\text{RPM} = \frac{2362.39}{P} \times \sqrt{\frac{(T \times D)/0.098102 \times \alpha^{1/3}}{(0.0254 \times D)^2 \times (D/(P * 3.29546))^{1.5}}}, \quad (21)$$

Here,  $T$  is the thrust generated by the quadcopter,  $D$  is the diameter of propellers, and  $P$  is the pitch (the distance the propeller would move forward in one rotation if it were moving through a soft solid).

Propellers are responsible for propelling the quadcopter in the upward direction, and they consist of radiating blades with a revolving hub. The propellers selected for the frame are 1045, so its dimensions are as follows: diameter is 10 inches, the pitch is 4.5 inches, and the correction factor from [27] is used to be 1.45 (unitless quantity).

Using (24) and putting the defined parameter values as required, the RPM value is 8281. In motors, the constant velocity is given by  $K_v$  and is measured by the number of RPMs that a motor turns when 1 V (one volt) is applied with no load attached to that motor [28].

Now, (21 and 22) are used to get the nominal battery voltage value of the required motor specs. Then, with respect to the estimated efficiency (in this scenario, according to the datasheet, it is taken as 0.8), the minimum required  $K_v$  of the motor is estimated to be 753, while the nominal  $K_v$  is 903.6.

$$K_{v_{\text{required}}} = \frac{\text{RPM}}{V_{\text{battery}}} = 753K_v, \quad (22)$$

$$K_{v_{\text{nominal}}} = K_{v_{\text{required}}} (1 + (1 - e)). \quad (23)$$

Here,  $K_{v_{\text{required}}}$  is required  $K_v$ ,  $K_{v_{\text{nominal}}}$  is nominal  $K_v$ ,  $V_{\text{battery}}$  is battery voltage, and  $e$  is efficiency.

The thrust force generation (electrical power for it) depends on the diameter and pitch of propellers and the estimated RPM of the motor. Thus, the electrical power consumption is estimated using Equation (23) to be 136 W. For nominal estimation, the efficiency is estimated to be 0.8,

TABLE 5: Gains from HIL testing.

Axis	$K_p$	$K_i$	$K_d$
$\phi$	0.15	0.1	0.01
$\theta$	0.15	0.1	0.01
$\psi$	0.15	0.1	0.01
Z-axis/altitude	0.75	1.5	0.1

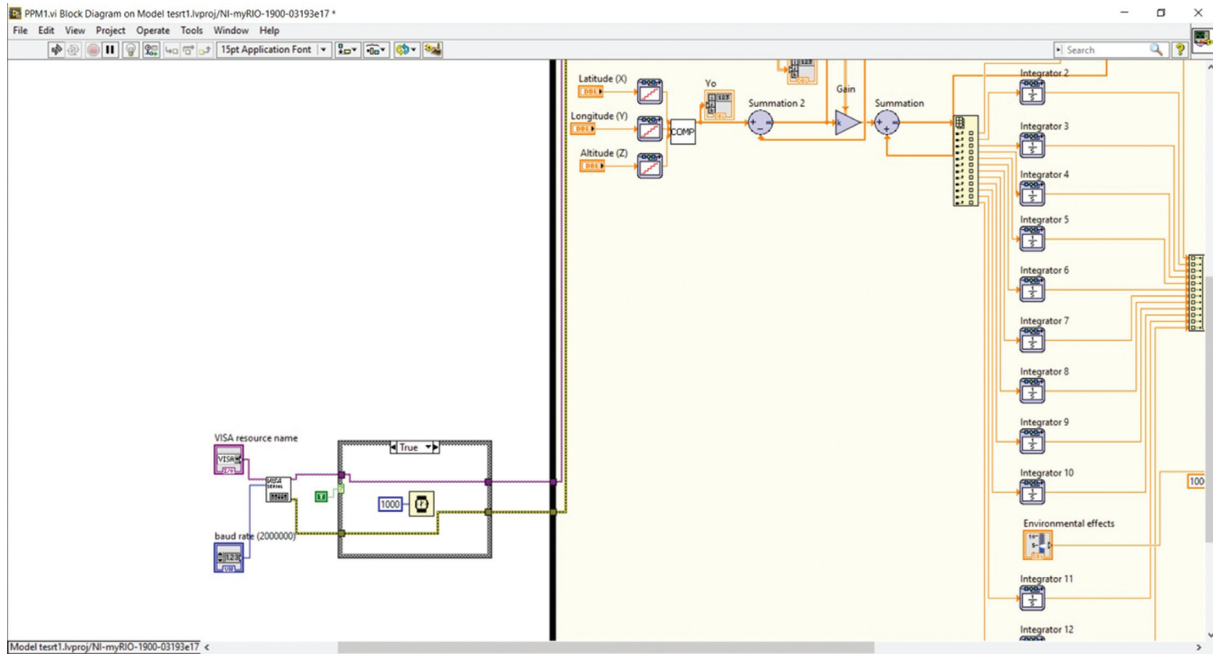


FIGURE 8: myRIO Serial Receiving Setup for ATmega Microcontroller.

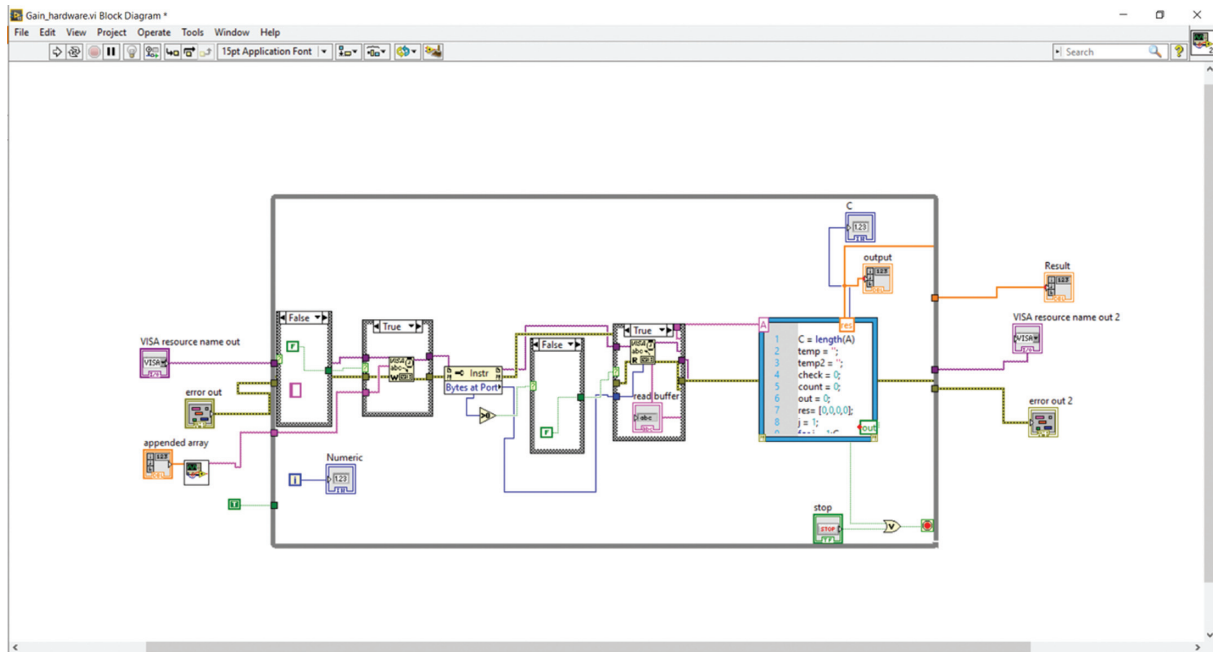


FIGURE 9: myRIO Serial Receiving and conversion from ASCII to Float Array.

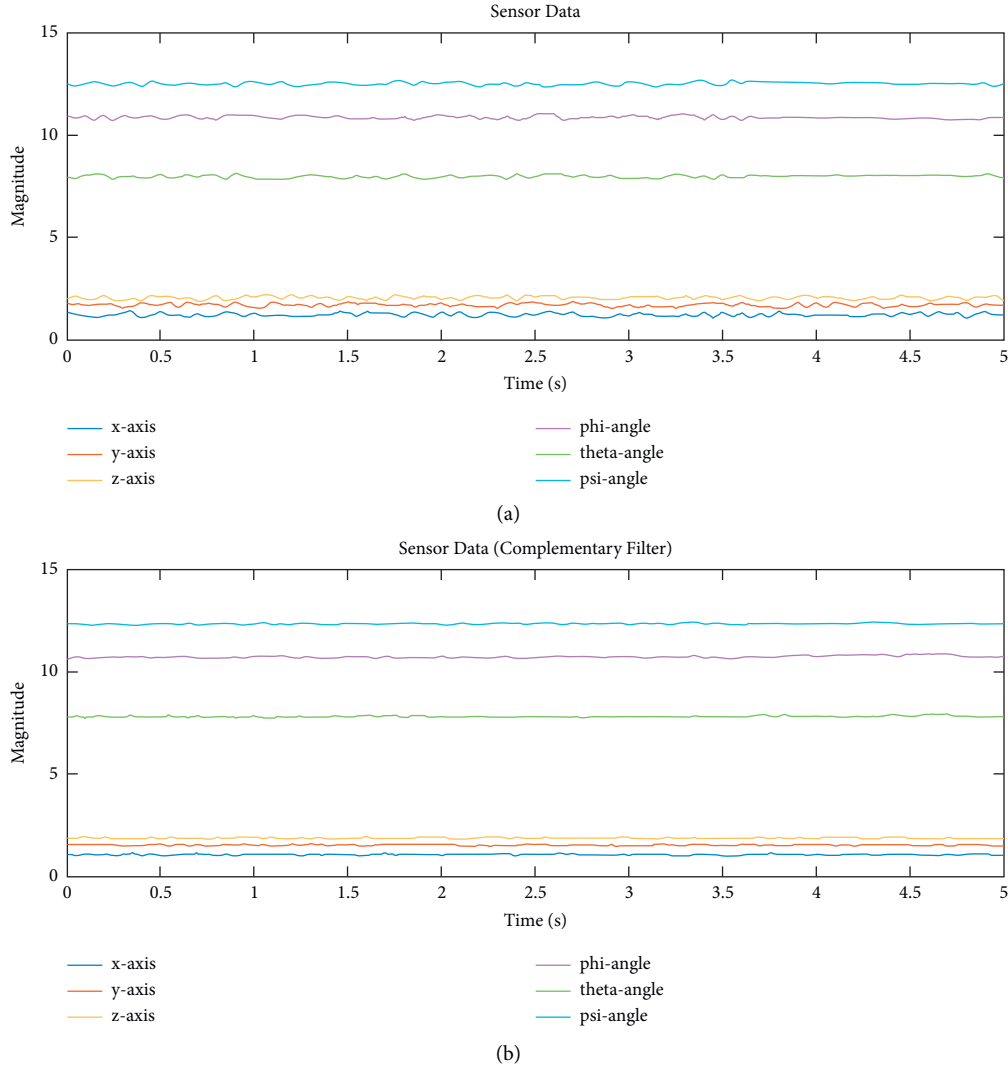


FIGURE 10: IMU Sensor Data on Inclined Surface with and without filter.

TABLE 6: System weight calculations.

Sr. no.	Parameter	Weight (grams)
1	Frame	282
2	Motor	75
3	ESC ( $\alpha$ )	25
4	Battery	374
5	Miscellaneous ( $\alpha$ )	260
Total		1316

which gives the desired power of 163.2 W. The required current for this system is estimated to be 18 A with an offset of 20%. Now, for the desired battery rating/discharge rate, (24) is used; it gives a rating of 14 C of discharge rate.

$$P_{\text{out}} = \frac{D^4 \times P \times \text{RPM}_{\text{required}}}{1.4 \times 10^{17}} \times 746. \quad (24)$$

Usually, a quadcopter starts to hover at 35–45% thrust of the total thrust generated by the motor. Updating the estimation in equations (20)–(28) shows that the new desired

power for the individual motor is approximately 4 A. System endurance depicts the amount of time a quadcopter can take flight and is calculated using equation (25). Estimated system endurance at full throttle is 4.16 minutes, while, in the hovering state, it is 15.6 minutes, thus giving an average flight time of around 9.88 minutes.

$$E = \frac{B_C \times e \times 60}{I_{\text{nominal}} \times 4}. \quad (25)$$

Here,  $e$  is endurance,  $B_C$  is battery capacity, and  $I_{\text{nominal}}$  is nominal current.

Considering the requirements and prototype availability in the local market, a 5200 mAh LiPo battery with 30C discharge rate, A2212 BLDC motor with specs given in Table 7, and ESCs of 30 A are used.

After calculating the required prototype specifications for the Inertial Measurement Unit (IMU), sensor GY-521 (Chinese variant of MPU-6050) is used. It has an accelerometer and gyroscope, is operable on 3.3–5V, and allows I2C communication. For GPS, Neo-M8N is selected; it offers an

TABLE 7: Summarized system.

Sr. no.	Parameter	Value	Value (weight)	Unit
1	Required RPM FT	8281	10985	RPM
2	Required kv FT	753	990	Kv
3	Required nom. kV FT	904	1188	Kv
4	<i>P<sub>out</sub></i> FT	136	318	W
5	<i>P<sub>in</sub></i> FT	163.2	380	RPM
6	Motor ampere	15	35	A
7	Required ESC	18	42	A
8	Battery rating	14	32.3	C
9	Endurance FT	4.16	1.78	Minutes
10	Required RPM H	5233	6947	RPM
11	<i>P<sub>out</sub></i> H	34.4	80.4	W
12	<i>P<sub>in</sub></i> H	41.28	96.4	W
13	Endurance H	15.6	9	Minutes
14	Flight time (average)	9.88	4.35	Minutes

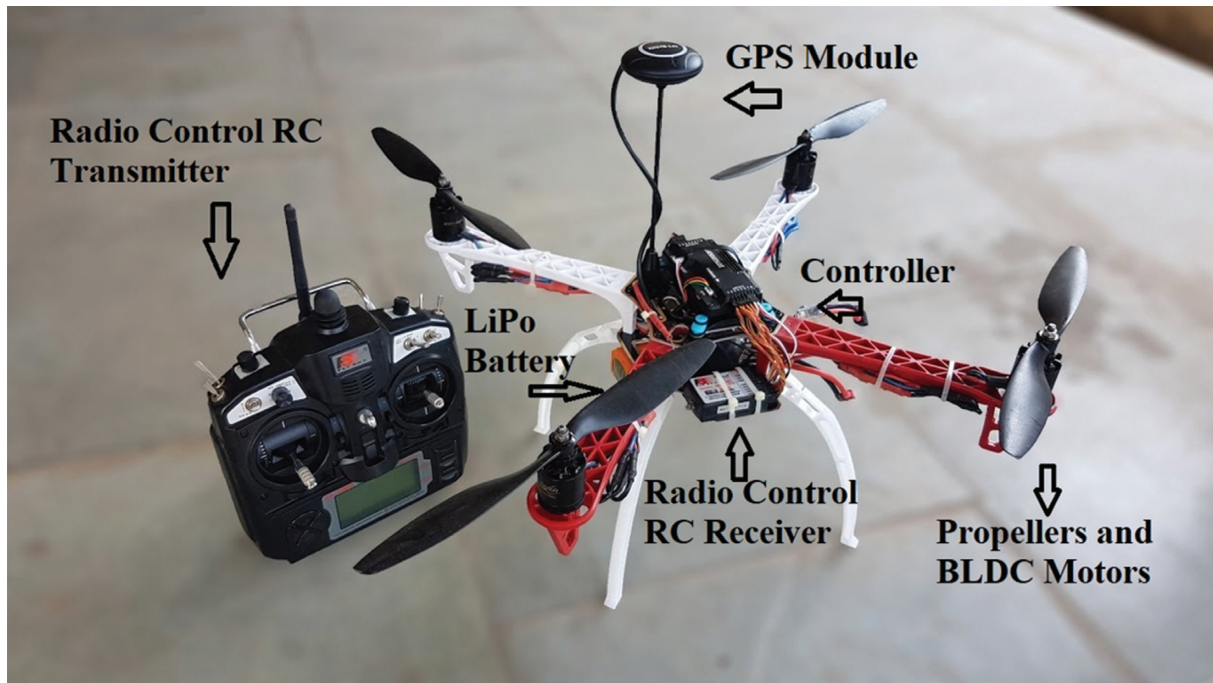


FIGURE 11: Assembled quadcopter prototype.

accuracy of 0.6–0.9 meters and supports a 10 Hz update rate; however, later, it was not deployed due to practical limitations. As for manual control of the quadcopter, Fly-Sky i6 is used with a TGY-iA6B receiver that supports six channels at 2.4 GHz and Pulse Position Modulation (PPM) data receiving. 2 Bluetooth (HC-05) devices are used in master-slave configuration to transmit data from the controller to the computer.

Figure 11 shows the assembled prototype with labeled hardware components. The assembled prototype is tested in two different conditions: initially in a room with constraints to minimize the possible damage, and then in an open environment for final testing. The quadcopter showed a stable response within the confined room, but results collection was quite tricky due to area limitations. Therefore, tests were conducted in the open area after some practice

and obtained relatively good results. Flight response in terms of attitude and position for around 70 seconds is shown in Figure 12.

The quadcopter took a stable takeoff and flew at low altitude; however, due to wind (the speed of the wind was around 10–14 km/h), the position of the quadcopter was not maintained. Still, due to robust orientation control, the quadcopter maintained its stability. Multiple flights are carried out to estimate battery drainage, and the average flight time, including takeoff and landing, was 9 minutes and 7 seconds, which is quite close to the estimated time.

Although there are several other prebuilt options available in the market (which are relatively way expensive), however, cost evaluation is done between the cheapest solutions available in the market to the proposed model. The calculations show that the proposed model is more than 25%

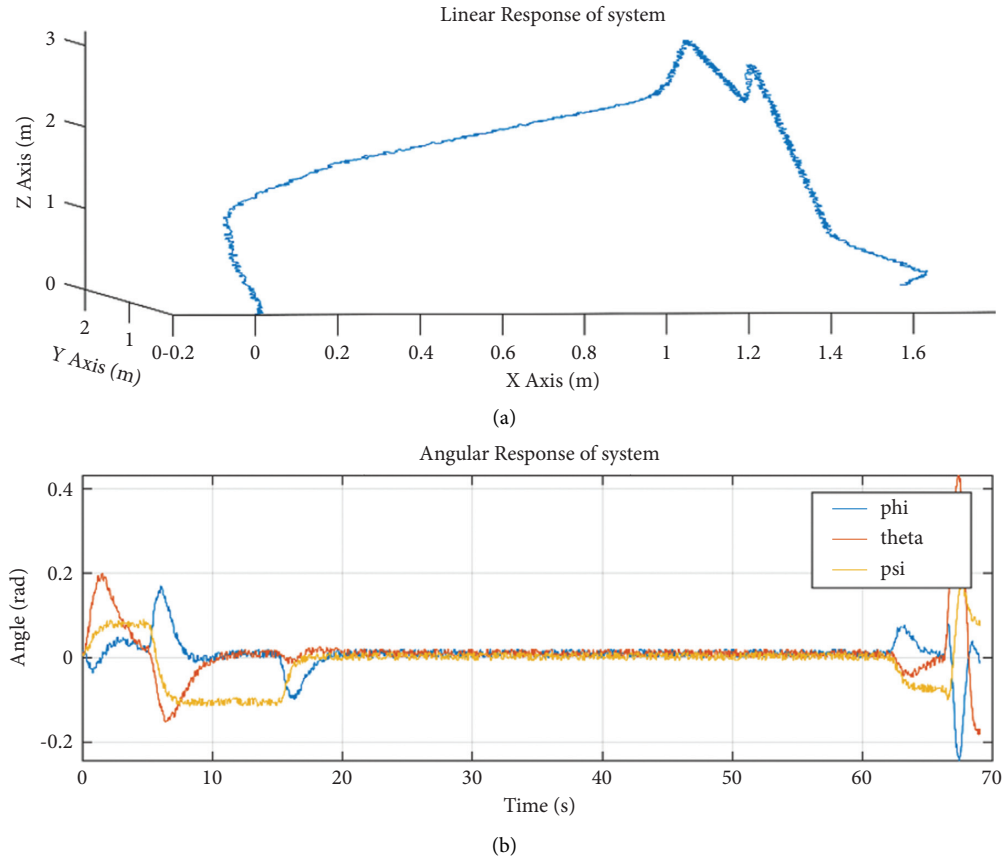


FIGURE 12: Prototype linear and angular response.

TABLE 8: Cost comparison.

Sr. no.	Component name	Price (PKR)	Prototype	Generic
1	DJI 450 frame	1700	Yes	Yes
2	Landing gear	1300	Yes	Yes
3	Fly-Sky Tx/Rx	10000	Yes	Yes
4	Pixhawk	10000	No	Yes
5	ATmega with PCB	800	Yes	No
6	GY-521	300	Yes	No
7	ESCs (4)	3200	Yes	Yes
8	Motors and propellers (4)	4400	Yes	Yes
9	Battery Li-Po	4500	Yes	Yes
Total			26200	35100

less expensive than the most affordable solution available in the market; details are given in the next section.

## 7. Computational Complexity and Cost Analysis

The computational cost analysis is done on the code by analyzing the function calls on different operations. The base initialization of parameters is considered constant  $c$ , thus making its complexity to  $O(n)$ ; after that, receiving data from IMU and Rx (ground transceiver) also has the complexity of  $O(n)$ . However, the control and stabilization part has double nested loops, thus increasing the complexity to  $O(n^2)$ , making the overall operation complexity  $O(n^2)$ . As

for cost analysis, Table 8 gives a detailed cost analysis of the system, where the prototype costs PKR 26,200, and the generic model costs around PKR 35100. The professionally developed systems sold in the market cost from PKR 70,000 to PKR 200,000.

## 8. Conclusion

A cost-effective end-to-end quadcopter design, development, and validation framework is presented using model-based design, HIL testing, and a rapid control prototyping approach. Firstly, a mathematical model of the dynamic behavior of the quadcopter is developed, and PID, LQR, and



full state feedback controllers are designed and compared. Moreover, Lyapunov stability analysis, SIL, PIL, and HIL testing are performed to evaluate the stability of all the controllers. The LQR controller showed an accuracy of 94% with the limitation of a simple path. The accuracy decreased for abrupt path proving the validity of the designed model. Finally, the RCP approach has been used to implement a prototype and test the stability under different real-time scenarios. The developed POC prototype quadcopter took a stable flight and proved the idea of an affordable quadcopter design approach. It is concluded that the proposed framework and design approach can be employed with low-cost components and different design parameters for various quadcopter use cases and applications.

Future work includes a ground control station for the autonomous operation of the quadcopter to enable additional features like preflight planning, real-time flight observation, and data collection from onboard sensors during the flight.

## Data Availability

No external data is used in this work.

## Conflicts of Interest

The authors declare no conflicts of interest in preparing this paper.

## Acknowledgments

This research was funded by Higher Education Commission Pakistan, grant number NRPU-5939.

## References

- [1] S. T. Elliott and T. W. Carr, "Robust nonlinear control of BLDC motor in quadcopter applications," *American Journal of Undergraduate Research*, vol. 12, no. 2, 2016.
- [2] H. Menouar, I. Guvenc, K. Akkaya, A. S. Ulugac, A. Kadri, and A. Tuncer, "UAV-enabled intelligent transportation systems for the smart city: applications and challenges," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 22–28, 2017.
- [3] A. Intwala and Y. Parikh, "A review on vertical take off and landing (vtol) vehicles," *International Journal of Innovative Research in Advanced Engineering (IJIRAE)*, vol. 2, no. 2, pp. 187–191, 2015.
- [4] V. Praveen and S. Pillai, "Modeling and simulation of quadcopter using PID controller," *International Journal of Control Theory and Applications*, vol. 9, no. 15, pp. 7151–7158, 2016.
- [5] A. Zenkin, I. Berman, K. Pachkouski, I. Pantiukhin, and V. Rzhnevskiy, "Quadcopter simulation model for research of monitoring tasks," in *Proceedings of the 2020 Twenty Sixth Conference of Open Innovations Association (FRUCT)*, pp. 449–457, IEEE, Yaroslavl, Russia, April 2020.
- [6] H. Mo and G. Farid, "Nonlinear and adaptive intelligent control techniques for quadrotor UAV - a survey," *Asian Journal of Control*, vol. 21, no. 2, pp. 989–1008, 2019.
- [7] Y. Wang, A. Ramirez-Jaime, F. Xu, and V. Puig, "Nonlinear model predictive control with constraint satisfactions for a quadcopter," in *Journal of Physics: Conference Series* vol. 783, no. 1, IOP Publishing, Article ID 12025, 2017.
- [8] Z. Samadikhoshkho, S. Ghorbani, and F. Janabi-Sharifi, "Modeling and control of aerial continuum manipulation systems: a flying continuum robot paradigm," *IEEE Access*, vol. 8, Article ID 176883, 2020.
- [9] G. Feng, "A survey on analysis and design of model-based fuzzy control systems," *IEEE Transactions on Fuzzy Systems*, vol. 14, no. 5, pp. 676–697, 2006.
- [10] P. Wang, Z. Man, Z. Cao, J. Zheng, and Y. Zhao, "Dynamics modelling and linear control of quadcopter," in *Proceedings of the 2016 International Conference on Advanced Mechatronic Systems (ICAMEchS)*, pp. 498–503, IEEE, Melbourne, Australia, December 2016.
- [11] M. Alhaddad, M. Tyupkin, and L. Aeshina, "Modeling and attitude control OF a quadcopter using linear quadratic regulator," vol. 15, pp. 173–176, 2016.
- [12] M. Agha, K. Kanistras, M. J. Rutherford, and K. P. Valavanis, "Mathematical model derivation of an unmanned circulation control aerial vehicle UC2AV," *Control Theory and Technology*, vol. 18, no. 1, pp. 1–18, 2020.
- [13] F. Sharifi, M. Mirzaei, B. W. Gordon, and Y. Zhang, "Fault tolerant control of a quadrotor UAV using sliding mode control," in *Proceedings of the 2010 Conference on Control and Fault-Tolerant Systems (SysTol)*, pp. 239–244, IEEE, Nice, France, October 2010.
- [14] R. B. Anderson, J. A. Marshall, and A. L'Afflitto, "Constrained robust model reference adaptive control of a tilt-rotor quadcopter pulling an unmodeled cart," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 1, pp. 39–54, 2021.
- [15] S. Nekoo Rafee, J. Ángel Acosta, G. Heredia, and A. Ollero, "A benchmark mechatronics platform to assess the inspection around pipes with variable pitch quadrotor for industrial sites," *Mechatronics*, vol. 79, Article ID 102641, 2021.
- [16] R. Kumar, A. M. Deshpande, J. Z. Wells, and M. Kumar, "Flight Control of Sliding Arm Quadcopter with Dynamic Structural Parameters," in *Proceedings of the 2020 IEEE/R SJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1358–1363, Las Vegas, NV, USA, January 2020.
- [17] D. Gheorghiuță, I. Vintu, L. Mîrea, and C. . Brăescu, "Quadcopter control system," in *Proceedings of the 2015 Nineteenth International Conference on System Theory, Control and Computing (ICSTCC)*, pp. 421–426, IEEE, Cheile Gradistei, Romania, October 2015.
- [18] A. Irfan, M. Gufran Khan, and S. Ali Mohsin, "Quadcopter dynamic modeling and stability control design using hardware in loop," in *Proceedings of the Accepted to IEEE International Conference on Robotics, Automation, Artificial-Intelligence and Internet-Of-Things (RAAICON)*, Dhaka, Bangladesh, December 2021.
- [19] E. Paiva, J. Soto, J. Salinas, and W. Ipanaque, "Modeling, simulation and implementation of a modified PID controller for stabilizing a quadcopter," in *Proceedings of the 2016 IEEE International Conference on Automatica (ICA-ACCA)*, pp. 1–6, IEEE, Curico, Chile, October 2016.
- [20] D. Sikeridis, E. E. Tsiropoulou, M. Devetsikiotis, and S. Papavassiliou, "Wireless powered Public Safety IoT: a UAV-assisted adaptive-learning approach towards energy efficiency," *Journal of Network and Computer Applications*, vol. 123, pp. 69–79, 2018.
- [21] A. Nagaty, S. Saeedi, C. Thibault, M. Seto, and H. Li, "Control and navigation framework for quadrotor helicopters," *Journal of Intelligent & Robotic Systems*, vol. 70, no. 1, pp. 1–12, 2013.

- [22] T. Luukkonen, "Modelling and control of quadcopter." Independent research project in applied mathematics," *Espoo*, vol. 22, p. 22, 2011.
- [23] E. Vinodh Kumar and J. Jerome, "Robust LQR controller design for stabilizing and trajectory tracking of inverted pendulum," *Procedia Engineering*, vol. 64, pp. 169–178, 2013.
- [24] J.-S. Song and X.-H. Chang, "H $\infty$  controller design of networked control systems with a new quantization structure," *Applied Mathematics and Computation*, vol. 376, Article ID 125070, 2020.
- [25] M. Aoki, *State Space Modeling of Time Series*, Springer Science & Business Media, Berlin, Germany, 2013.
- [26] D. E. Rivera, M. Morari, and S. Skogestad, "Internal model control: PID controller design," *Industrial and Engineering Chemistry Process Design and Development*, vol. 25, no. 1, pp. 252–265, 1986.
- [27] Singfaye, "Aircraft Propeller Calculator," Singfaye.co.uk, 2020, <http://singfaye.co.uk/docs/e9d2ca-aircraft-propeller-calculator>.
- [28] RotorDrone, "Understanding Kv Ratings - RotorDrone," RotorDrone, 2020, <https://www.rotordronepro.com/understanding-kv-ratings/>.