



Article

Attitude and Altitude Control Design and Implementation of Quadrotor Using NI myRIO

Jun-Yao Hong, Po-Jui Chiu , Chun-Da Pong and Chen-Yang Lan * 

Department of Mechanical Engineering, National Taiwan University of Science and Technology, Taipei 10607, Taiwan

* Correspondence: jimmylan@mail.ntust.edu.tw

Abstract: Multi-rotor vehicles have demonstrated great potential in many applications, from goods delivery to military service. Its simple structure has drawn much attention in control research, with various feedback control design methods being tested on it. In this study, a quadrotor is constructed with National Instrument (NI) myRIO for its flight controller. Linear controllers are synthesized with a simple model and with a more detailed model, which also considers the actuator dynamic and system time delay, to exploit the limitations and trade-offs posted by hardware and its influence on linear control design and implementation. The simulation results match the real response better if the detailed model is used in the control design. In addition, the linear–quadratic–Gaussian (LQG) control provides the best response in the control design in this study. The constraints posted by the actuator and time delay are clearly observed in the control synthesis process and the experiment result. These constraints also led to poor control performance or even instability if not considered in the control implementation in advance.

Keywords: quadrotor; model identification; compensator; LQG



Citation: Hong, J.-Y.; Chiu, P.-J.; Pong, C.-D.; Lan, C.-Y. Attitude and Altitude Control Design and Implementation of Quadrotor Using NI myRIO. *Electronics* **2023**, *12*, 1526. <https://doi.org/10.3390/electronics12071526>

Academic Editor: Cecilio Angulo

Received: 7 February 2023

Revised: 13 March 2023

Accepted: 17 March 2023

Published: 23 March 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

A quadrotor is generally constructed by a symmetric cross frame with a motor and propeller at each end. By adjusting each motor's rotation speed, a thrust force and rotating moment are generated to maneuver the vehicle. In addition to its low manufacturing cost and simple construction, the ability of vertical take-off and landing and hovering provides great potential in many applications, such as aerial photography, delivery, surveillance, reconnaissance, semantic simultaneous localization, and mapping [1]. Flying through narrow windows has also been demonstrated with excellent maneuverability by Loianno et al. [2]. In the drone market report [3], a global market of 30.6 billion USD for drones is estimated in 2022 with a market compound annual growth rate of 7.8% until 2030. The market potential is massive and thus attracts many global competing brands, such as DJI from China, Dragan Flyer from Canada, and Pixhawk and Ardupilot from Switzerland.

In this article, a quadrotor is constructed, and its flight controller developed based on NI myRIO with the objective of providing a unified process and discussing the design and implementation of a quadrotor's flight control. Therefore, flight control is developed according to the conventional steps in feedback control design, including dynamic modeling and parameter identification, state measurement and estimation, and feedback control design.

The dynamic model for quadrotors, assumed as a rigid body in a three-dimensional space, has been derived from different approaches, such as Newton's Law [4], Euler-Lagrange equation [5], and Hamiltonian [6]. In most cases, kinematics and kinetics are considered, while aerodynamics is generally simplified with just a thrust factor and a drag factor corresponding to its propeller portfolio. Thus, the physical parameters required to describe the dynamics of a quadrotor comprise the mass and inertia, the radius of the frame, the thrust factor, the drag factor, and the motor constants. These parameters can

be identified from flight data [7] and experiment measurements [8]. The mass and inertia can also be estimated from Solid Work drawing, with its material properties assigned [9] or measured by the bifilar pendulum's period [10]. In [11], Derafa et al. applied the Levenberg–Marquardt optimization method to minimize the system response from the compass to identify inertia in pendulum experiments. This approach allows a bigger swinging angle in helping to excite the targeted swinging mode more easily.

The position, altitude, and attitude information of the quadrotor are required for feedback control. Although many quantities are measurable, they usually suffer from noise and disturbance. Thus, the estimator is adapted to provide a filtering function to reduce noise in its attitude and heading reference system (AHRS) and inertial navigation system (INS). Accelerometers, gyroscopes, and magnetometers are installed for AHRS. The measured value is then processed by different filters, such as the complementary filter, Mahony filter [12], Madgwick filter [13], and Extended Kalman Filter [14,15]. Moreover, Global Positioning System (GPS) sensors, cameras, and surrounding video systems are commonly used to provide position information for INS in conjunction with an Extended Kalman filter or Unscented Kalman filter [16]. In addition, Wang et al. [17] used a nonlinear signal-correction observer for measured signals that are non-Gaussian at different levels from the GPS and accelerometer. Berkane et al. [18] used a non-linear observer with an inertial measurement unit (IMU) and generic position information from GPS or a camera for state estimation and has been shown to be exponentially stable and validated with experiments.

The flight control of a quadrotor consists of attitude control, altitude, and position tracking control, while position tracking is achieved through the control of quadrotor attitude angles and thrust. Linear control design methods, non-linear control design methods, and even learning-based methods are all commonly seen in the literature. In the case that the requirement in maneuver and trajectory tracking is not aggressive and only a small attitude angle is needed, linear control design methods, such as Proportional-Integral-Derivative (PID) and compensator [19–21], linear-quadratic-regulator (LQR) and linear-quadratic-Gaussian (LQG) [22,23], H-infinity [24,25], and quantitative feedback theory (QFT) [26], are applied with a satisfactory result. On the other hand, if aggressive maneuver and complex trajectory tracking are desired, sophisticated control design methods, such as sliding mode control [27], adaptive control [28], backstepping [29], feedback linearization [30], and model predictive control (MPC) [31], would provide better performance. Moreover, the reinforced-learning algorithm has also been applied in quadrotor attitude control [32–34] and has been observed with better control performance.

The development of a quadrotor and its flight control is usually a foundation for an autonomous vehicle system for various tasks. Under such missions, reliable and safe operation is critical and is usually achieved through high-level control and navigation. Geofencing technology is usually used to steer the multirotor to avoid obstacles and illegal zones. The cascaded control is improved through the constrained control scheme in geofencing in [35]. A quadratic programming-based cascaded control architecture was demonstrated with control barrier functions in [36]. Experiments have demonstrated the feasibility of those proposed approaches. In most works of literature, the motor dynamics and system time delay are usually omitted, such as in [19,22], and thus the performance achieved might not be optimum due to this assumption. For example, the altitude settling time is 9 s, and the yaw settling time is about 35 s, with a steady state error in [22]. Therefore, this study explores this potential by considering a more detailed dynamics model that incorporates motor response and system time delay in the control design. Classical linear controllers, PID controllers, and LQR and LQG controllers are synthesized and implemented for comparison. It is desirable in feedback control research to develop its own controller. Thus, an NI myRIO-based flight controller is developed that provides better flexibility and computation capability than commercial-ready control boards. This hardware is used to address the potential and limitations of attitude control.

The result has demonstrated that motor dynamics have less impact on the controller performance if the closed loop poles are far enough from the poles of the motor. Nonetheless, this implies that the motor limited the achievable closed loop control performance. Moreover, among the four examined controllers, LQG design has been observed with the smoothest tracking performance and the smallest overshoot under a smaller control effort.

The contributions of this study are highlighted as follows:

- The flight controller of a quadrotor is realized using NI myRIO. The system development for constructing the drone and the ground station for tasks is simplified under one developing frame. In addition, the loop execution time is easily observed in the control, compared to other Microprocessor Control Unit (MCU)-based controllers. This facilitates the investigation of different feedback control schemes or autonomous systems.
- The motor dynamic and system time delay are identified and explicitly considered in the feedback control design. The constraints due to hardware are presented and compared using different control design approaches.
- Model parameters are identified through experiments. The parameter values in a quadrotor dynamic model are estimated according to the least square results of experiments.

The remainder of this article is organized as follows. Section 2 presents the system architecture and devices in constructing the quadrotor and introduces the methods used in the modeling, sensing, and control of the developed quadrotor. Section 3 describes the experiments and the results in parameter identification, state estimation, and control. The control results are discussed in Section 3.3.7. The conclusions of the presented material are then given in Section 4.

2. Materials and Methods

In this article, a flight control design following the conventional feedback control design approach is developed using linear control design methods. A dynamic model is first established, with its parameters identified via experiments and measurements for the constructed quadrotor. The motor response to the command and the system time delay are also incorporated and compared in the control design. State information, which is required for feedback control, is measured by deployed sensors or estimated via the Kalman filter. Finally, linear control is synthesized using a basic lead-lag compensator, PID, and LQG for the cases with or without the consideration of the motor response and system time delay, both in simulations and in experiments.

2.1. System Architecture

The quadrotor is constructed in this study using commercially available components with glass-fiber reinforced polyamide nylon frames and 3D-printed plastic parts. The system architecture is shown in Figure 1. The bill of materials is listed in Table 1. The main system is depicted within the dashed line box. It has a NI myRIO at the center as the controller and four sets of motors and propellers, which are controlled by an electronic speed controller, IMU sensor, range finder, GPS, camera, and a radio receiver. The controller, NI myRIO, can communicate with the PC through WiFi or receive commands from the remote controller by the radio receiver. It has to be emphasized that the PC currently serves as a ground station and acquires data transmitted from myRIO through WiFi. The conditions of the quadrotor are monitored in the PC interface. The data from the experiments are stored in a memory stick on the myRIO. In attitude control experiments, the desired attitude angle is given in the myRIO program for flight testing. The remote control is used to give attitude angle commands and to steer the quadrotor under realized attitude control.

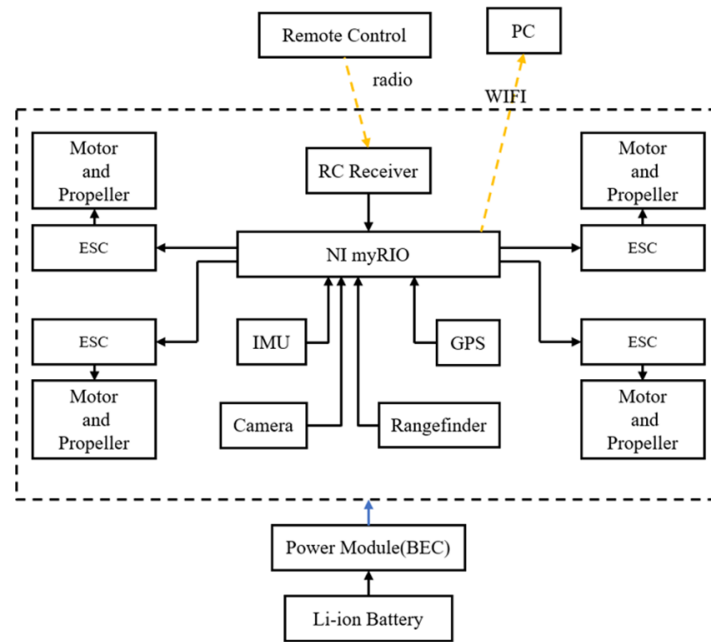


Figure 1. System architecture of the developed quadrotor.

Table 1. Bill of material for the constructed quadrotor.

Item	Name
Frame	DS-HJ-450
Propeller	P11x3.7
Motor	MN3508-16 KV700
Controller	MyRIO-1900
Battery	3S 60C 6500mAh
Battery Eliminator Circuit	PM07
ESC	AIR 40A
GPS	NEO-M8N
IMU	BNO055
Camera	OpenMV Cam H7 Plus
LiDAR	TF Mini LiDAR

The battery is connected to a battery eliminator circuit (BEC) to provide stable dc voltage to the main system. The frame is 450 mm in length, which gives a radius of $r_f = 0.225$ m. The total weight measured is $m = 1.789$ kg.

2.2. Modeling

As derived by Sabatino in [4], the kinematics of the quadrotor are derived as in Equations (1) and (2) by assigning the earth coordinate using the North-East-Down system and a body coordinate centered at the quadrotor’s geometrical center, as shown in Figure 2.

$$V = R \cdot V_B \tag{1}$$

$$\omega_B = T^{-1} [\dot{\phi} \quad \dot{\theta} \quad \dot{\psi}]^T \tag{2}$$

where

$V = [\dot{x} \quad \dot{y} \quad \dot{z}]^T$ is the translation velocity in the earth coordinate,

$V_B = [u \ v \ w]^T$ is the translation velocity in the body coordinate,

$$R = R_{zyx}(\phi, \theta, \psi) = \begin{bmatrix} c(\psi)c(\theta) & -s(\psi)c(\phi) + c(\psi)s(\theta)s(\phi) & s(\psi)s(\phi) + c(\psi)s(\theta)c(\phi) \\ s(\psi)c(\theta) & c(\psi)c(\phi) + s(\psi)s(\theta)s(\phi) & -c(\psi)s(\phi) + s(\psi)s(\theta)c(\phi) \\ -s(\theta) & c(\theta)s(\phi) & c(\theta)c(\phi) \end{bmatrix}$$

is the transformation matrix of translation velocity from the body coordinate to the earth coordinate,

$\omega_B = [p \ q \ r]^T$ is the angular velocity in the body coordinate,

$$T = \begin{bmatrix} 1 & s(\phi)t(\theta) & c(\phi)t(\theta) \\ 0 & c(\phi) & -s(\phi) \\ 0 & \frac{s(\phi)}{c(\theta)} & \frac{c(\phi)}{c(\theta)} \end{bmatrix}$$

is the transformation matrix from the angular velocity

vector in the body coordinate to the rate of change of attitude angles,

ϕ is the roll angle,

$\dot{\phi}$ is the rate of change in the roll angle,

θ is the pitch angle,

$\dot{\theta}$ is the rate of change of pitch angle, ψ is the yaw angle,

ψ is the yaw angle,

$\dot{\psi}$ is the rate of change of the yaw angle,

$c(\cdot) = \cos(\cdot)$ and $s(\cdot) = \sin(\cdot)$.

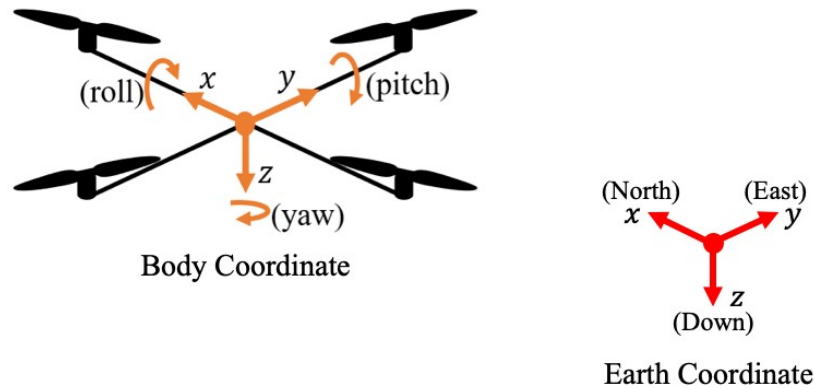


Figure 2. Coordinate assigned in deriving dynamic model.

The kinetics is obtained in Equations (3) and (4).

$$\dot{V}_B = -\omega_B \times V_B + gR^T \hat{e}_z + \frac{1}{m}(-f_t \hat{e}_z + f_w) \tag{3}$$

$$\dot{\omega}_B = I^{-1}(-\omega_B \times (I \cdot \omega_B) + \tau_B + \tau_w) \tag{4}$$

where

$$I = \begin{bmatrix} I_{xx} & & \\ & I_{yy} & \\ & & I_{zz} \end{bmatrix}$$

is the moment of inertia,

m is the mass of the vehicle.

g is the gravitational acceleration,

f_t is the trust provided by motors and propellers,

$f_w = [f_{wx} \ f_{wy} \ f_{wz}]^T$ is the force caused by external disturbance,

$\tau_B = [\tau_x \ \tau_y \ \tau_z]^T$ is the moment provided by motors and propellers,

$\tau_w = [\tau_{wx} \ \tau_{wy} \ \tau_{wz}]^T$ is the moment caused by an external disturbance,

and $\hat{e}_z = [0 \ 0 \ 1]^T$ is the z-direction unit vector of the body coordinate.

The thrust force and induced torque caused by the propeller are considered, while the horizontal force and its angular moment due to gyroscopic effect are ignored. The thrust is proportional to the square of the rotation speed of the propeller by a thrust factor, as in Equation (5). The induced torque is proportional to the square of the rotation speed of the propeller by a drag factor, as in Equation (6).

$$|T| = K_T \omega^2 \tag{5}$$

$$|Q_d| = K_d \omega^2 \tag{6}$$

In this study, the relationship between motor angular speed ω and motor command M_{cmd}^* is described in Equation (7).

$$\omega = b_\omega M_{cmd}^* \tag{7}$$

where b_ω is a gain value.

It is worth mentioning that $\sqrt{M_{cmd}} = M_{cmd}^*$ is the parameter used in Equations (8) and (9) to describe the motor-generated thrust force and induced torque, and ω_m is the cut-off frequency.

$$\frac{T}{M_{cmd}} = \frac{b_\omega^2 K_T}{\frac{s}{\omega_m} + 1} = \frac{b_T}{\frac{s}{\omega_m} + 1} \tag{8}$$

$$\frac{Q_d}{M_{cmd}} = \frac{b_\omega^2 K_d}{\frac{s}{\omega_m} + 1} = \frac{b_d}{\frac{s}{\omega_m} + 1} \tag{9}$$

Since the motor command affects the thrust and the torques simultaneously, mixing transformation (10) is used to allocate the required force and torques from the control command.

$$\begin{bmatrix} M_{1cmd} \\ M_{2cmd} \\ M_{3cmd} \\ M_{4cmd} \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 & -1 \\ 1 & 0 & -1 & 1 \\ 1 & 1 & 0 & -1 \\ 1 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} f_{tCMD} \\ \tau_{xCMD} \\ \tau_{yCMD} \\ \tau_{zCMD} \end{bmatrix} \tag{10}$$

This leads to Equation (11), describing the actuator dynamics. $f_{tCMD}, \tau_{xCMD}, \tau_{yCMD}, \tau_{zCMD}$ are the normalized thrust and torque commands.

$$\begin{bmatrix} \dot{f}_t \\ \dot{\tau}_x \\ \dot{\tau}_y \\ \dot{\tau}_z \end{bmatrix} = \begin{bmatrix} -\omega_m & & & \\ & -\omega_m & & \\ & & -\omega_m & \\ & & & -\omega_m \end{bmatrix} \begin{bmatrix} f_t \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} + \begin{bmatrix} 4\omega_m b_T & & & \\ & 2\omega_m b_{Tr} & & \\ & & 2\omega_m b_{Tr} & \\ & & & 4\omega_m b_d \end{bmatrix} \begin{bmatrix} f_{tCMD} \\ \tau_{xCMD} \\ \tau_{yCMD} \\ \tau_{zCMD} \end{bmatrix} \tag{11}$$

Finally, the quadrotor's dynamics are linearized around the condition stated by Equations (12) and (13).

$$\phi = \theta = \psi = 0 \tag{12}$$

$$f_{tCMD} = \frac{mg}{4b_T} \tag{13}$$

The linearized dynamics model is decoupled for translation and rotation and is described by the transfer function of Equations (14) to (19). The transfer function is later used in attitude and altitude control in this study.

$$\frac{\phi}{\tau_{xcmd}} = \frac{2b_T r / I_{xx}}{s^2 \left(\frac{s}{\omega_m} + 1 \right)} \tag{14}$$

$$\frac{\theta}{\tau_{ycmd}} = \frac{2b_T r / I_{yy}}{s^2 \left(\frac{s}{\omega_m} + 1 \right)} \tag{15}$$

$$\frac{\psi}{\tau_{zcmd}} = \frac{4b_d / I_{zz}}{s^2 \left(\frac{s}{\omega_m} + 1 \right)} \tag{16}$$

$$\frac{z}{f_{tcmd}} = \frac{-4b_T / m}{s^2 \left(\frac{s}{\omega_m} + 1 \right)} \tag{17}$$

$$\frac{x}{\theta} = \frac{-g}{s^2} \tag{18}$$

$$\frac{y}{\phi} = \frac{g}{s^2} \tag{19}$$

2.3. Parameter Identification

The dynamics equations describing a quadrotor have many parameters to be identified, as shown in Figure 3, including radius, mass, inertia, thrust factor, drag factor, motor gain, and cut-off frequency. The quadrotor developed in this study utilizes the material in Table 1.

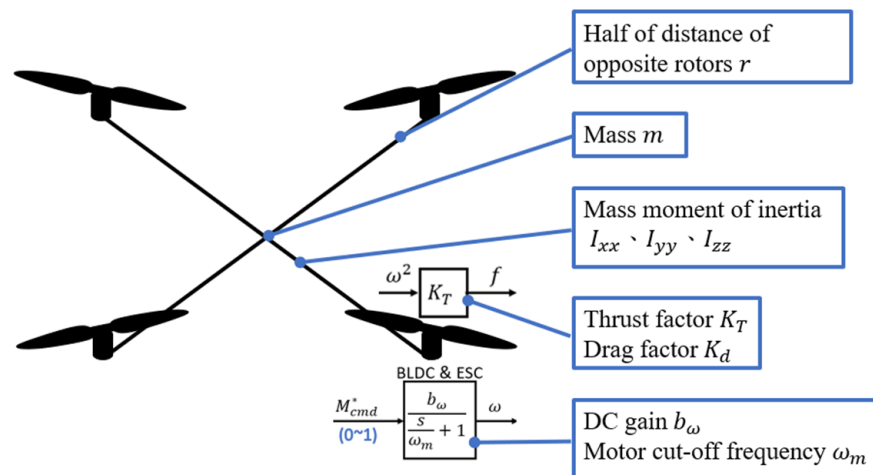


Figure 3. Model parameters to be identified.

A simple pendulum swinging at a small angle is used to estimate the moment of inertia for the roll and the pitch motion. The moment of inertia is calculated through the length of the pendulum and the period of the pendulum using Equation (20) and is depicted in Figure 4. The Δd represents the wire length to the fixing point of the quadrotor and d₀ is the distance from the fixing point to the center of gravity.

$$T^2 = \frac{m\Delta d^2 + 2md_0\Delta d + I_{xx,yy} + md_0^2}{\frac{mg}{4\pi^2}(\Delta d + d_0)} \tag{20}$$

Bilinear pendulum swinging is used to estimate the moment of inertia for the yaw motion. It is calculated using Equation (21) and is depicted in Figure 5. The Δh represents

the wire length to the fixing point of the quadrotor and h_0 is the distance from the fixing point to the plane of the center of gravity.

$$T^2 = \frac{16\pi^2 I_{zz}}{mgD^2} (\Delta h + h_0) \quad (21)$$

The schematic diagram of the pendulum experiment is shown in Figures 4 and 5. In the experiments, the length of the pendulum, Δd or Δh , is increased and recorded with the measured swinging period. The moment of inertia is then found by curve fitting using the least square approximation.

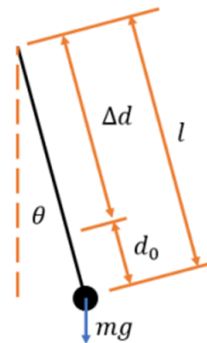


Figure 4. Simple pendulum swinging experiment.

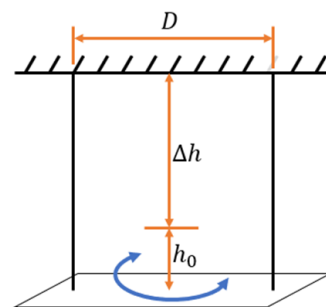


Figure 5. Bifilar pendulum swinging experiment.

When the speed of the motor increases, the thrust and torque will also increase, as described in Equations (10) and (11). Thus, the slope values found by fitting the speed with respect to the thrust and the torque curve are the thrust factor and the drag factor, respectively.

The parameters in the motor model include a DC gain b_ω and a cut-off frequency. Since the motor command M_{cmd}^* is proportional to the motor speed ω , b_ω can be obtained by linear curve fitting.

The motor dynamic models, as in Equations (13) and (14), take the modified motor command M_{cmd} as the input and the thrust force or the induced torque as the output. By measuring both input and output data, the TFest function from the system identification toolbox of MATLAB is used to obtain the motor dynamic model in frequency domain fitting to find the cut-off frequency.

2.4. Measurement and Estimation

The attitude state of the quadrotor is usually measured by accelerometers, gyroscopes, and magnetometers, and the position state is obtained by a distance range finder, GPS, and cameras. According to the modeling, the measurement states required to describe the dynamic behavior include attitude angle, angular velocity, angular acceleration, position, translation velocity, and translation acceleration. Nonetheless, in real implementation, the attitude angle is not directly measured using sensors. It is obtained with the Extend Kalman Filter that fuses the accelerometer, gyroscope, and magnetometer measurements

from the IMU. In addition, the IMU does not provide the data of angular acceleration, and the rangefinder does not provide the velocity of altitude. Since all this state information is required in the full state feedback control, a Kalman Filter is generally used to estimate all state values if not all states are measurable.

The IMU BNO055 offers an estimated attitude value, but no estimation of angular acceleration. To satisfy the state feedback requirement, the estimation model, including noise, is adapted to fill the requirement. The estimation models used are shown in Equations (22) to (24).

$$\begin{aligned} \dot{X}_\phi &= A_\phi X_\phi + B_\phi u_\phi + G_\phi w_\phi \\ y_\phi &= C_\phi X_\phi + v_\phi \end{aligned} \tag{22}$$

$$\begin{aligned} \dot{X}_\theta &= A_\theta X_\theta + B_\theta u_\theta + G_\theta w_\theta \\ y_\theta &= C_\theta X_\theta + v_\theta \end{aligned} \tag{23}$$

$$\begin{aligned} \dot{X}_\psi &= A_\psi X_\psi + B_\psi u_\psi + G_\psi w_\psi \\ y_\psi &= C_\psi X_\psi + v_\psi \end{aligned} \tag{24}$$

where $X_{\phi,\theta,\psi} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$ include angle, angular speed, angular acceleration, $u_{\phi,\theta,\psi}$ is torque

input, $w_{\phi,\theta,\psi}$ is input noise, $v_{\phi,\theta,\psi}$ is measurement noise, $A_\phi = A_\theta = A_\psi = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\omega_m \end{bmatrix}$,

$B_\phi = G_\phi = \begin{bmatrix} 0 \\ 0 \\ 2rb_T\omega_m / I_{xx} \end{bmatrix}$, $B_\theta = G_\theta = \begin{bmatrix} 0 \\ 0 \\ 2rb_T\omega_m / I_{yy} \end{bmatrix}$, $B_\psi = G_\psi = \begin{bmatrix} 0 \\ 0 \\ 4b_d\omega_m / I_{zz} \end{bmatrix}$,

$C_\phi = C_\theta = C_\psi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$, $w_\phi = w_\theta \in \mathbb{R}$, $w_\psi \in \mathbb{R}$, $v_\phi = v_\theta \in \mathbb{R}^{2 \times 2}$, $v_\psi \in \mathbb{R}^{2 \times 2}$.

In addition to attitude estimation, the quadrotor requires altitude estimation to regulate the vehicle distance to the ground. However, the IMU only provides an accelerometer and rangefinder for the z-axis distance; the state of the climbing rate is not measurable. Thus, the Kalman Filter is again employed to estimate the altitude information for feedback. At the same time, the filter provides a reduction in the high-frequency noise due to motor vibration.

$$\begin{aligned} \dot{X}_Z &= A_Z X_Z + B_Z u_Z + G_Z w_Z \\ y_Z &= C_Z X_Z + v_Z \end{aligned} \tag{25}$$

where $X_Z = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$ include altitude, velocity, acceleration, u_Z is torque input, w_Z is input

noise, v_Z is measurement noise, $A_Z = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\omega_m \end{bmatrix}$, $B_Z = G_Z = \begin{bmatrix} 0 \\ 0 \\ -4b_T\omega_m / m \end{bmatrix}$, $C_Z =$

$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, $w_Z \in \mathbb{R}$, $v_Z \in \mathbb{R}^{2 \times 2}$.

2.5. Feedback Control Design

The linearized model is decoupled for each attitude angle control and altitude control. The attitude and altitude control is achieved by using an inner loop control, while the x-y position control is accomplished by an outer loop control, as shown in Figure 6. The attitude and altitude controller provides a thrust and moment command to the actuator based on the desired value and estimated state values to achieve feedback control.

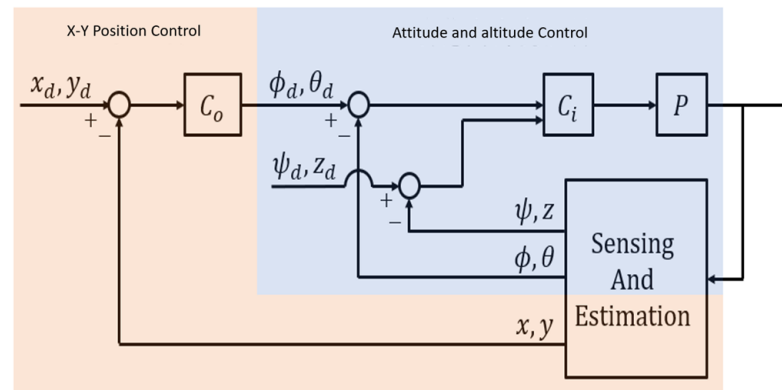


Figure 6. Flight Control Structure.

Several linear control synthesizing methods are performed in the control design in this study. First, a cascade PID controller based on online tuning is synthesized. It is constructed with an inner loop for angular velocity control and an outer loop for angle control. The cascade control structure is shown in Figure 7. C_i and C_o are in the format of the PID controller.

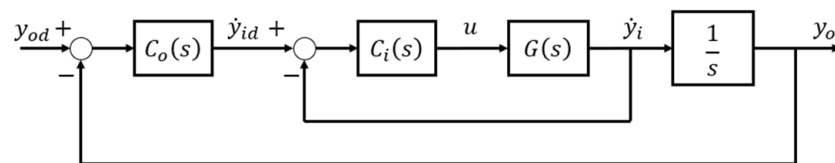


Figure 7. Cascade Control Structure.

Online tuning of PID control requires no modeling of the target system. However, its control performance is limited by its controller order and the number of iterations. Thus, C_i and C_o are augmented to higher order pole-zero-gain structure using compensator design in the classical control design method. While the classical linear control design requires a model description of the target system, this study evaluates the influence of a more detailed model, including the actuator dynamic and system time delay, for this control design.

Moreover, LQ design is also performed in this attitude and altitude control, as shown in Figure 8. The Kalman filter in LQG is used to estimate the state of motor dynamics, while the LQR controller can be synthesized if the motor dynamic is not considered since the attitude angle and the angle velocity could be measured. An integrator internal model, as shown in Figure 8, as servo, is used to form an augmented system in the LQ control design so that the steady-state error is eliminated.

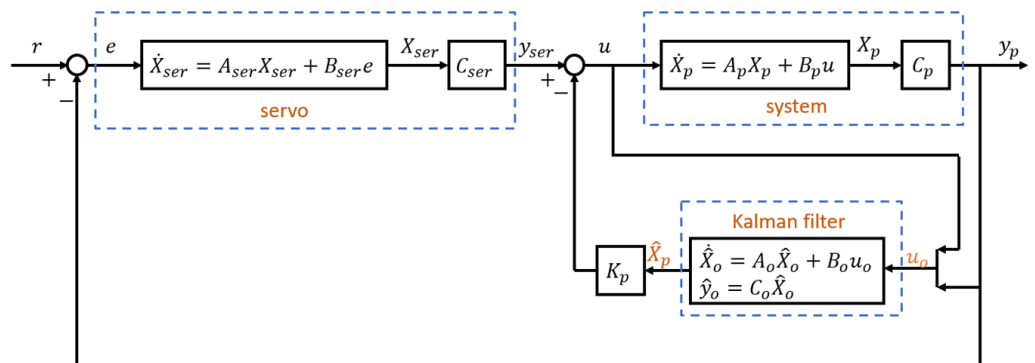


Figure 8. Linear-quadratic-Gaussian (LQG) servo control structure.

3. Results

3.1. System Parameters

The experiment steps to identify system parameters and their results are described in this subsection.

3.1.1. Mass and Frame Length

The mass of the whole vehicle m is measured as 1.789 kg. The arm radius r_f of quadrotor is 0.225 m.

3.1.2. Moment of Inertia

In the experiment, the IMU BNO055 is used to measure the period of a pendulum swinging through the zero-crossing time. The initial pendulum length d_0 is unknown, and the Δd is increased by 25 mm each time from 25 mm to 200 mm, as shown in Figure 9. In finding the I_{zz} , the suspension is replaced by wires so that the quadrotor has a rotation degree of freedom in the z-axis, as shown in Figure 10.



Figure 9. Moment of inertia experiment for I_{xx} and I_{yy} .

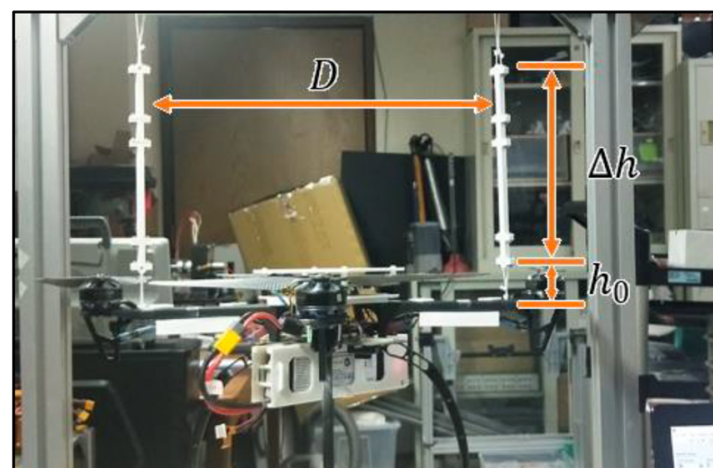


Figure 10. Moment of inertia experiment for I_{zz} .

Since the developed quadrotor is assumed to be symmetric, I_{yy} is thus equal to I_{xx} . From the fitting result of I_{xx} as in Figure 11, I_{xx} is found to be $23.5 \times 10^{-3} \text{ kg}\cdot\text{m}^2$, and the distance d_0 is 41.9 mm. The fitting R^2 value is 99.39%.

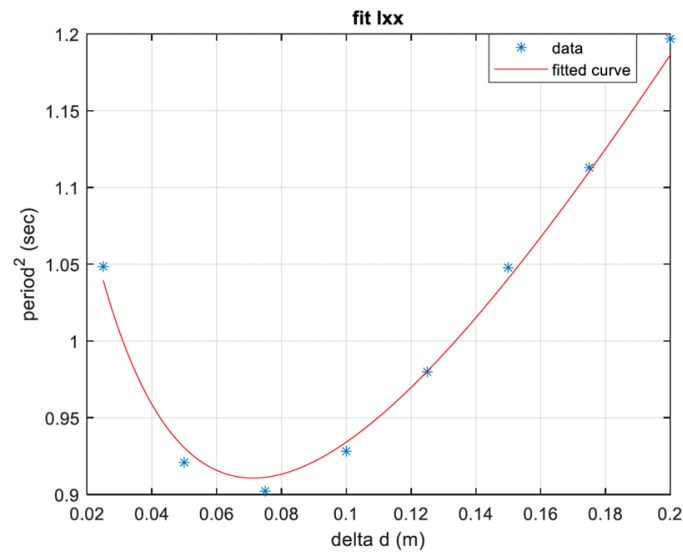


Figure 11. Experiment data fitting result of I_{xx} .

The fitting result of I_{zz} is shown in Figure 12. I_{zz} is then found to be $36.2 \times 10^{-3} \text{ kg}\cdot\text{m}^2$, and distance d_0 is 56.3 mm. The fitting R^2 value is 99.91%. Although there is a difference between the two d_0 estimations, the result is acceptable because the error might be due to the pendulum wiring.

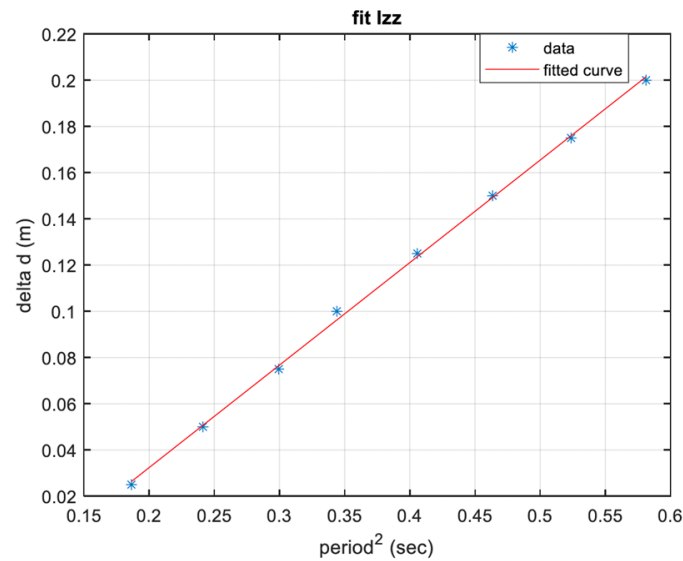


Figure 12. Experiment data fitting result of I_{zz} .

3.1.3. Thrust Coefficient and Drag Coefficient

A testing structure is used to hold onto the motor and propeller set and to identify the thrust coefficient and drag coefficient. Two load cells, one for thrust force and the other for torque measurement, are mounted on the testing structure. An optocoupler is used to measure propeller rotation speed. The experimental arrangement is shown in Figure 13. The supply voltage is 12 V, and an electrical speed controller (ESC) is used to drive the motor at different rotation speeds. The sampling rate of the load cell is set at 1 kHz. The motor command is described as a value between 0 and 1 that generates a PWM signal to the ESC at different speeds.



Figure 13. Equipment for thrust and drag coefficient measurement.

As shown in Figure 14, the gain of the motor command to the motor angular speed is $b_\omega = 756.6 \text{ rad/s} \cdot M_{cmd}^*$. The motor command is observed to be linear to the motor angular speed.

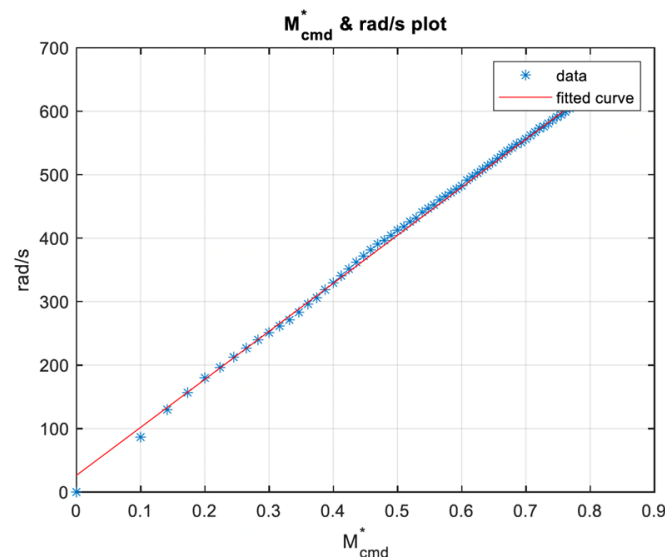


Figure 14. Motor command versus rotation speed.

The thrust coefficient K_T is obtained by curve fitting the measure torques and the square value of the motor angular speed, as shown in Figure 15. The result gives a K_T of $14.769 \times 10^{-6} \text{ kg} \cdot \text{m} / \text{rad}^2$. Figure 15 also shows that the square value of the rotation speed and thrust have a linear relationship.

In Figure 16, the relationship of the DC gain b_T from modified motor command M_{cmd} , the square of the motor command, to the thrust is shown. The result identifies b_T in Equation (8) as a value of $8.8381 \text{ N} / M_{cmd}$. b_T can also be calculated by multiplying the square of b_ω and K_T which gives a $b_\omega^2 K_T$ equal to $8.4546 \text{ N} / M_{cmd}$. This calculation leads to a 4.34% error in the fitting result. This error might be the sum of the fitting error of two estimations. Thus, the direct fitting result for DC gain b_T is considered.

The drag coefficient is estimated using the same approach. The fitting result of drag coefficient K_d is $2.1758 \times 10^{-7} \text{ kg} \cdot \text{m}^2 / \text{rad}^2$. The estimation of DC gain from modified motor command M_{cmd} to the torque, b_d , is $0.12973 \text{ N} \cdot \text{m} / M_{cmd}$. Due to the same reason for the fitting error, the calculated DC gain $b_\omega^2 K_d$ is $0.12455 \text{ N} \cdot \text{m} / M_{cmd}$ and has a 4% error to fitting result. Thus, the direct fitting result is considered. The experimental results are shown in Figure 17.

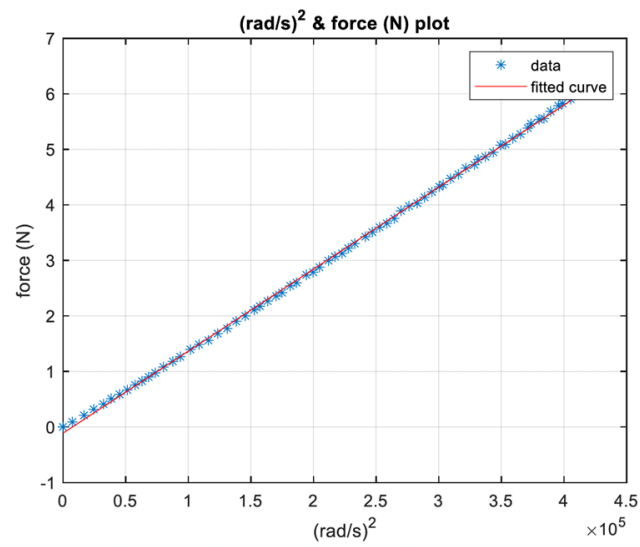


Figure 15. Motor rotation speed versus thrust.

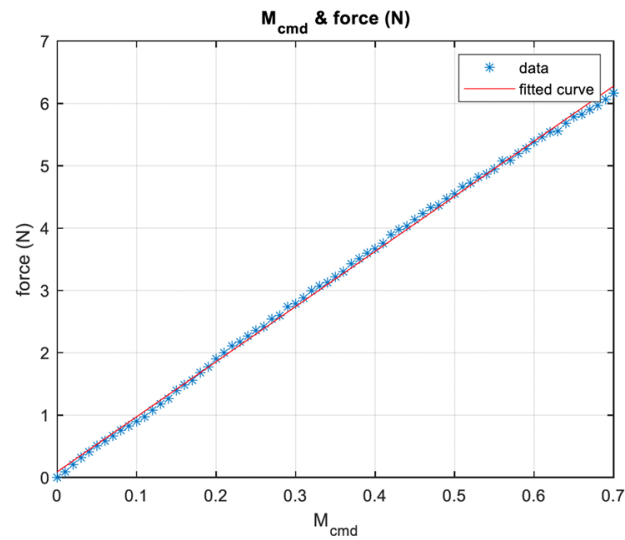
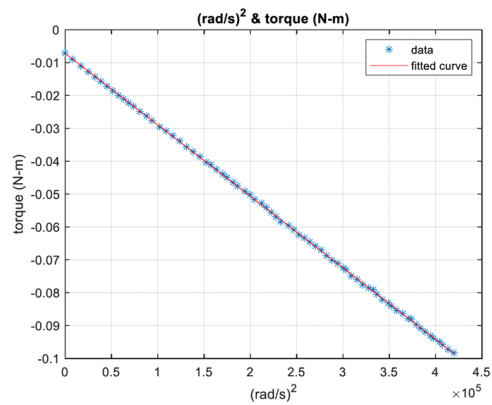
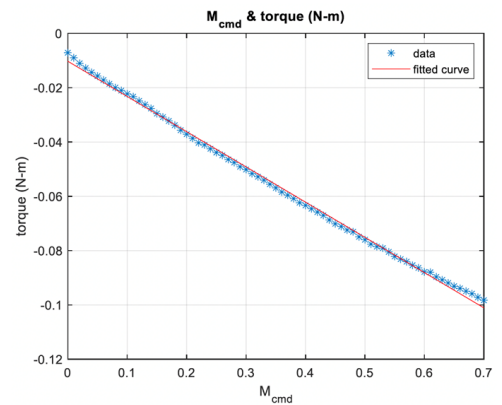


Figure 16. Modified motor command M_{cmd} versus thrust plot.



(a)



(b)

Figure 17. (a) Motor rotation speed versus torque; (b) Modified motor command M_{cmd} versus torque.

3.1.4. Motor Dynamic

To model the dynamics of the motor, a chirp signal is used to excite the system. A load cell is used to measure the output thrust due to the motor and propeller. Considering the thrust coefficient estimated earlier, the nominal operation value in the command is set at 0.5. The mean magnitude region from 0.48 to 0.54 is divided into 4 levels, including 0.48, 0.5, 0.52, and 0.54. After the chirp signal excitation experiment for model identification, a square chirp signal is used as input to validate the model. The chirp signal and square signal are shown in Table 2.

Table 2. Chirp signals.

Parameter	Excitation Signal	Validation Signal
Signal Type	Chirp	Square wave chirp
Mean Value (M_{cmd})	0.48, 0.5, 0.52, 0.54	0.49
Amplitude (M_{cmd})	0.05, 0.1	0.0725
Frequency (Hz)	0~10	0~10
Experiment Time (s)	60	60

Model fitting result in the Bode diagram is shown in Figure 18 for different experiments. The semi-transparent lines are the fitting model results from each experiment. The black line is selected as the nominal model for all fitting results.

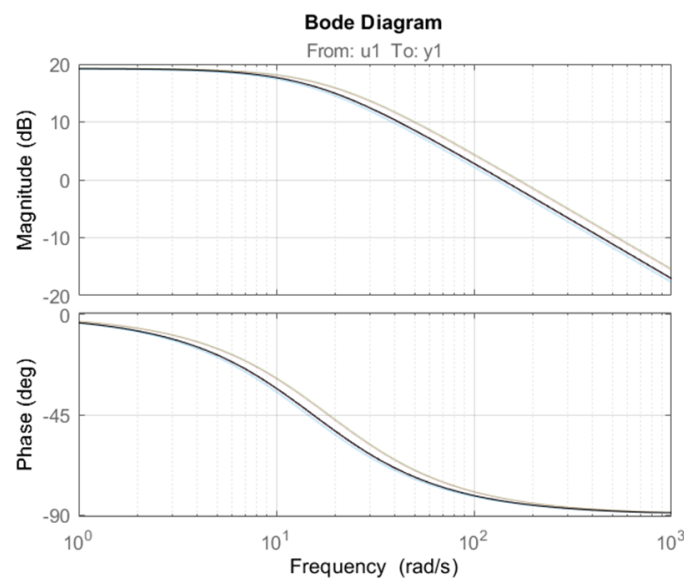


Figure 18. Bode diagram of motor model identification.

As an example, the input and output response data of the case with 0.5 mean M_{cmd} and 0.1 value amplitude are shown in Figure 19. The model estimation R^2 is 67.25%. The R^2 in the response around 43 s, as in Figure 20, gives the same property with a zoomed view. The estimated motor trust model is shown in Equation (26).

$$\frac{T}{M_{cmd}} = \frac{b_T}{\frac{s}{\omega_m} + 1} = \frac{9.134}{\frac{s}{15.36} + 1} \tag{26}$$

A square wave chirp is then used to validate the model. The cutoff frequency ω_m is 15.36 rad/s. The DC gain b_T is 9.134. The estimation R^2 is 73.38%. The results shown in Figure 21 validate the model.

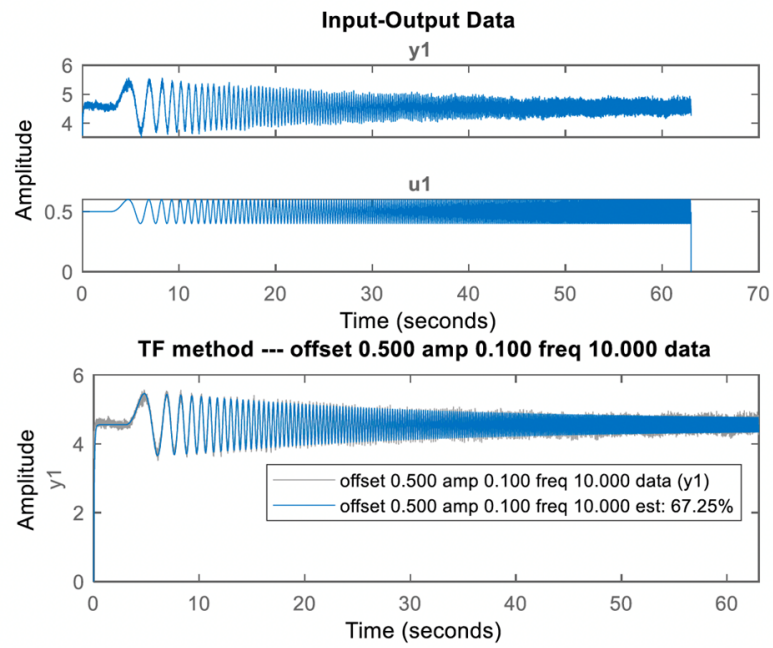


Figure 19. Input output data comparison.

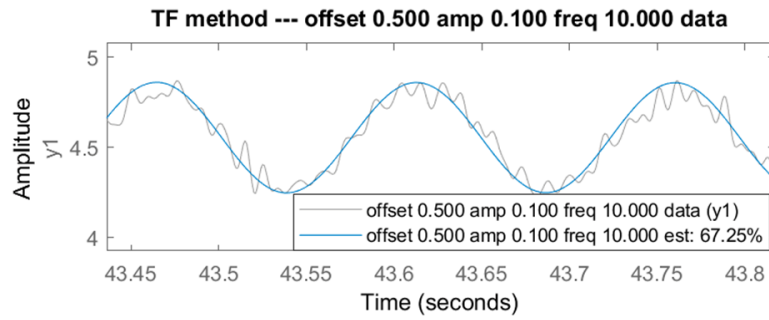


Figure 20. Output data and estimated output enlarged at 43 s.

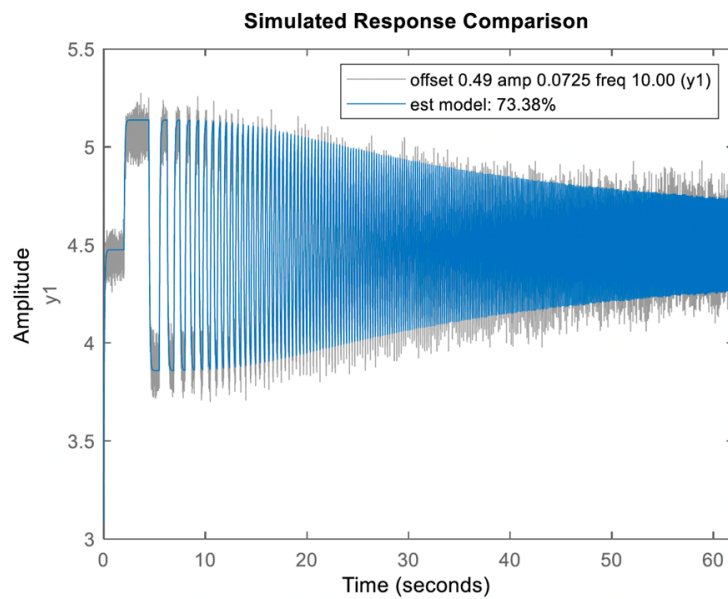


Figure 21. Square wave chirp signal validation.

The fitting result of DC gain b_T from static fitting has a small 3.2% error compared to the value found by dynamic model estimation. The b_d static fitting value is used in motor torque modeling rather than in more dynamic experiments. In addition, the cutoff frequency is assumed to be the same as in the thrust response. The motor torque model is thus determined as Equation (27).

$$\frac{Q_d}{M_{cmd}} = \frac{b_d}{\frac{s}{\omega_m} + 1} = \frac{0.13}{\frac{s}{15.36} + 1} \tag{27}$$

3.1.5. Time Delay

The system time delay of a quadrotor is due to input delay, measurement delay program loop, running delay, etc. Figures 22–24 show the time delay observed in each axis from the command given to the start of the output response. The time delay of the roll and yaw angle motion is 60 ms. On the other hand, a time delay in altitude response is not observed. Therefore, the delay in altitude motion is ignored.

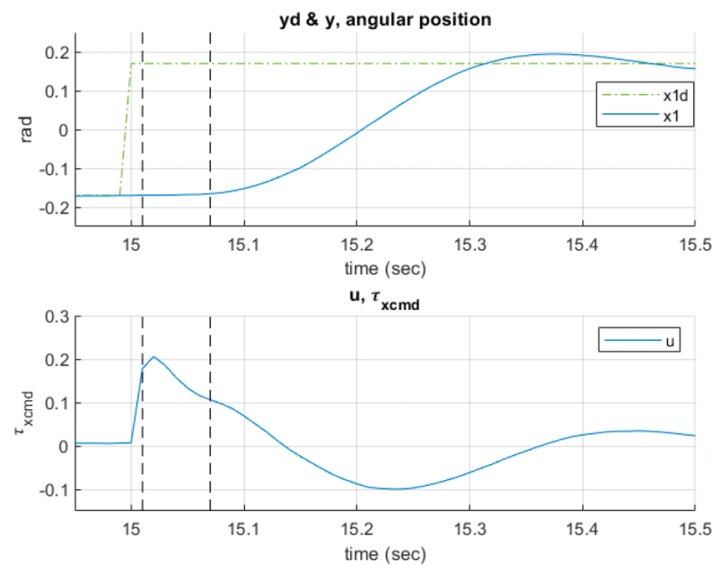


Figure 22. Roll angle time delay.

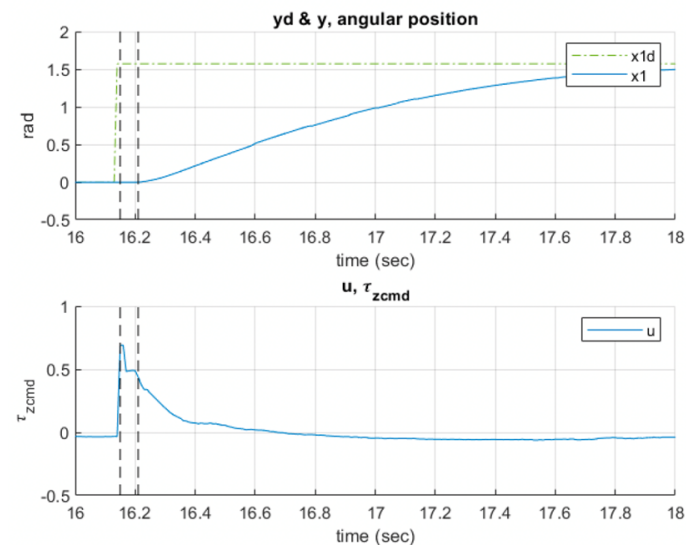


Figure 23. Yaw angle time delay.

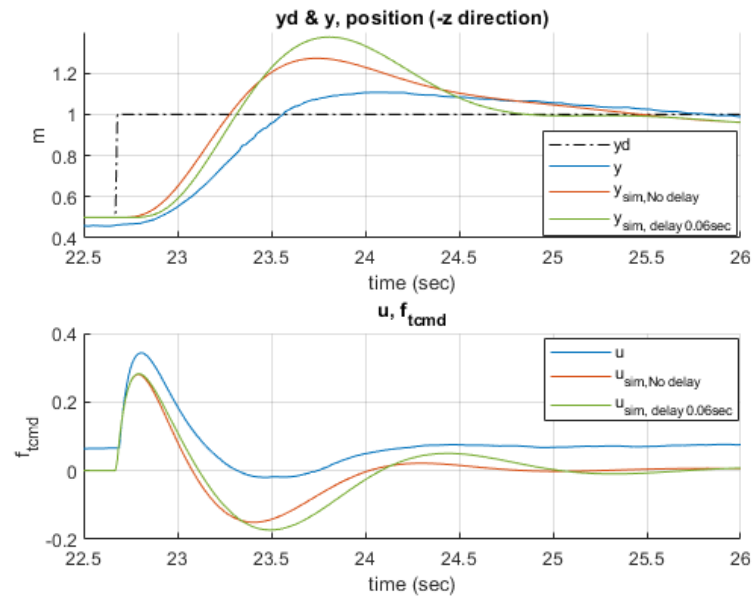


Figure 24. Altitude time delay in the -z direction.

3.1.6. Linearized Model

Once all parameters are found, the quadrotor model is obtained for the control design and described in Equations (28) to (31).

$$\frac{\phi}{\tau_{xcmd}} = \frac{2rb_T\omega_m / I_{xx}}{s^2(s + \omega_m)} = \frac{2687}{s^2(s + 15.36)} \tag{28}$$

$$\frac{\theta}{\tau_{ycmd}} = \frac{2rb_T\omega_m / I_{yy}}{s^2(s + \omega_m)} = \frac{2687}{s^2(s + 15.36)} \tag{29}$$

$$\frac{\psi}{\tau_{zcmd}} = \frac{4b_d\omega_m / I_{zz}}{s^2(s + \omega_m)} = \frac{220.6}{s^2(s + 15.36)} \tag{30}$$

$$\frac{z}{f_{tcmd}} = \frac{-4b_T\omega_m / m}{s^2(s + \omega_m)} = \frac{-313.5}{s^2(s + 15.36)} \tag{31}$$

3.2. State Estimation

This subchapter presents the Kalman Filter variance measurement and the Q and R matrix synthesis in estimator design. The state estimation thus satisfies the state feedback control purpose.

3.2.1. Variance Measurement and Tuning

In Kalman Filter design, Q and R matrices are determined by input and measurement noise. Since input noise is hard to measure, the Q matrix is tuned by the trial-and-error method. The objective is to achieve convergence in quadrotor state measurement with stable flight. On the other hand, the R matrix is determined by the sensor datasheet and variance experiments.

The noise in attitude measurement is given by sensor characteristics. From the datasheet, the measurement error affected by temperature rising by 10^{-4} rad/K. Moreover, the maximum output noise is 5.236×10^{-3} rad/s. Since the temperature is kept steady for a short period of time, the temperature effect is ignored. The variance is then measured by the error between the encode and the IMU. The R value is calculated using the variance of measurement error and selected to be $R = \begin{bmatrix} 2.5 \times 10^{-4} & 0 \\ 0 & 5 \times 10^{-2} \end{bmatrix}$. The results are shown in Figure 25.

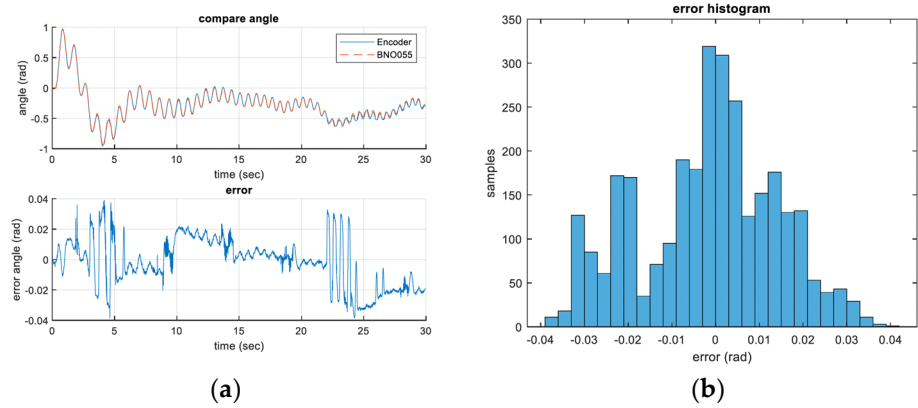


Figure 25. (a) Difference between the encoder and inertial measurement unit (IMU) measurement; (b) Histogram of measurement error.

The altitude is estimated using measured data from the accelerometer and rangefinder. The R matrix is determined by the measurement noise of these two sensors. The rangefinder was tested under a fixed distance to find the variance, which is 3.385×10^{-5} . The results are shown in Figure 26.

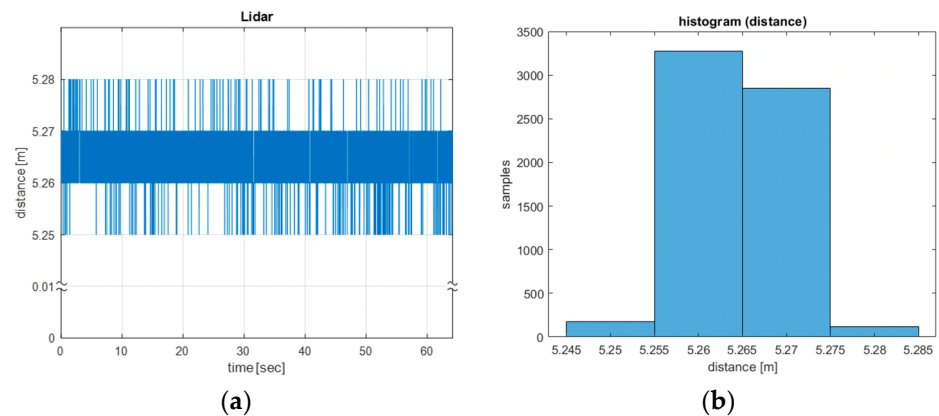


Figure 26. (a) Rangefinder measurement; (b) Rangefinder measurement histogram.

The noise of the accelerometer is measured under steady flight conditions. The variance of the accelerometer on the z-axis is 0.505. The results are shown in Figure 27.

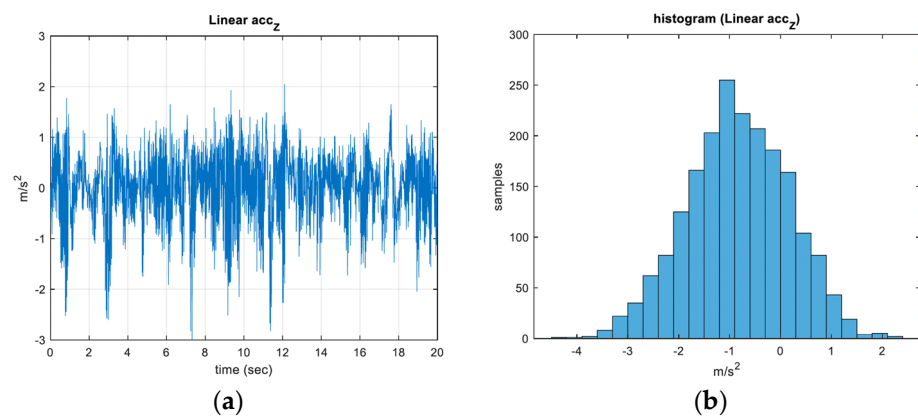


Figure 27. (a) Accelerometer z measurement; (b) Accelerometer z measurement histogram.

3.2.2. Estimator Design

In the attitude estimation, the tuned Q and R matrices are shown in Table 3. The Kalman Filter poles are smaller than -10 and specified far away from quadrotor closed loop poles. From Figures 28 and 29, the estimation results are observed to be smoother with less noise than simple sensor reading.

Table 3. Attitude Kalman Filter parameter.

System	Q	R	Kalman Filter Poles
Roll and Pitch	10^{-2}	$\begin{bmatrix} 2.5 \times 10^{-4} & 0 \\ 0 & 5 \times 10^{-2} \end{bmatrix}$	$-14.2, -24.6 \pm 24.4i$
Yaw	10^{-2}	$\begin{bmatrix} 2.5 \times 10^{-4} & 0 \\ 0 & 5 \times 10^{-2} \end{bmatrix}$	$-15.2, -6.9 \pm 6.7i$

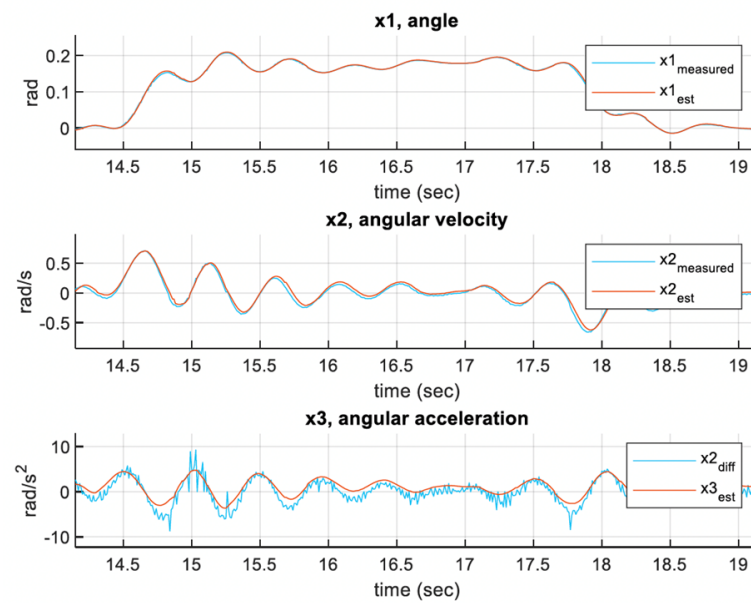


Figure 28. Roll angle estimation (compared with simple measurement reading).

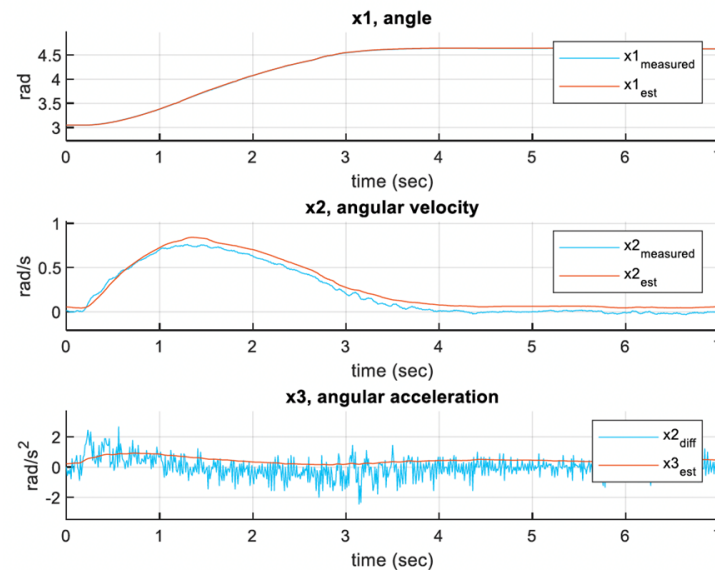


Figure 29. Yaw angle estimation (compared with simple measurement reading).

In the altitude estimation, the measurement noise is also reduced, especially in acceleration value. The Kalman Filter design parameters are shown in Table 4. The estimation result for altitude is shown in Figure 30.

Table 4. Altitude Kalman Filter parameter.

System	Q	R	Kalman Filter Poles
Altitude	3×10^{-3}	$\begin{bmatrix} 3.385 \times 10^{-5} & 0 \\ 0 & 0.505 \end{bmatrix}$	$-28.8, -6.9 \pm 7.4i$

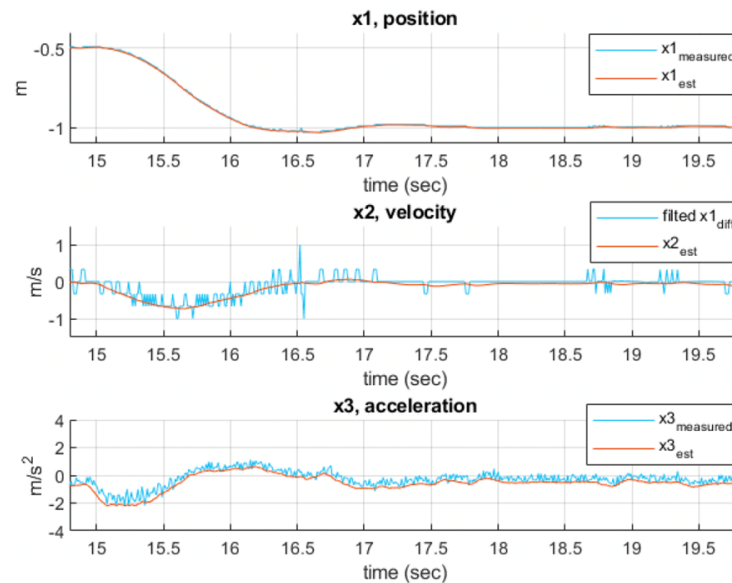


Figure 30. Altitude estimation (compared with simple measurement reading).

3.3. Attitude Control

3.3.1. Control Specifications

The feedback control response dominates the overall performance of the quadrotor. However, the motor settling time is around 0.3 s, and the maximum control command value is manually limited to 0.5. Thus, the fastest possible response and maximum control command effort are set to avoid control saturation with an acceptable response. The desired control performance is described in Table 5.

Table 5. The desired control performance specifications.

System	Settling Time	Overshoot	Control Effort
Roll and Pitch control	0.7~1	<5%	<0.3
Yaw control	1~3	<5%	<0.3
Height control	1~3	<5%	<0.3

3.3.2. Attitude Experiment Set-Up

Attitude experiments are performed on an in-house platform that only allows rotation in a single axis in each experiment, as shown in Figure 31.

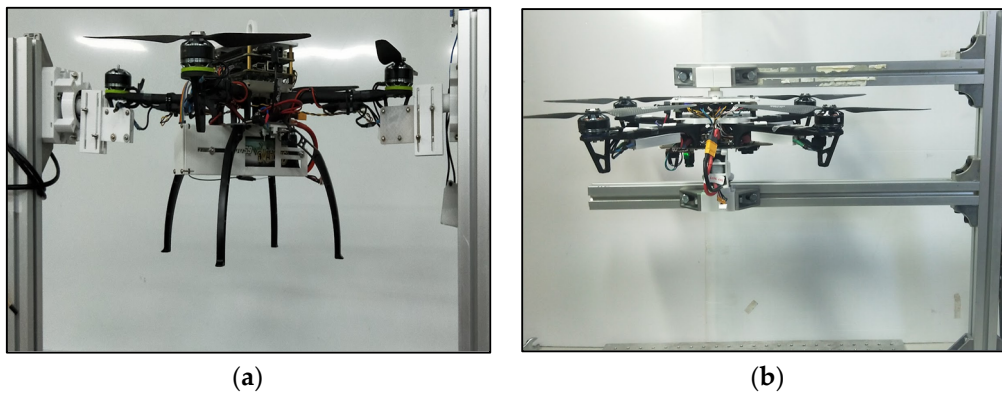


Figure 31. Attitude control experiments. (a) Roll and pitch testing platform; (b) Yaw testing platform.

3.3.3. PID Control Design and Experiment Result

The PID gain is tuned empirically from the inner loop controller and then from the outer loop controller. The velocity performance due to inner loop control is tuned to achieve the expected response first. After that, an outer loop K_p value is selected so that the quadrotor’s desired attitude angle is reached quickly. The final gain values for roll angle control are listed in Table 6 and for yaw angle control are listed in Table 7. The control responses are shown in Figures 32 and 33.

Table 6. Proportional-Integral-Derivative (PID) gains for roll angle control.

Loop	K_p	K_i	K_d
Inner loop	0.066	0.23	0.0023
Outer loop	4	0	0

Table 7. PID gains for yaw angle control.

Loop	K_p	K_i	K_d
Inner loop	0.1	0.045	0.003
Outer loop	1	0	0

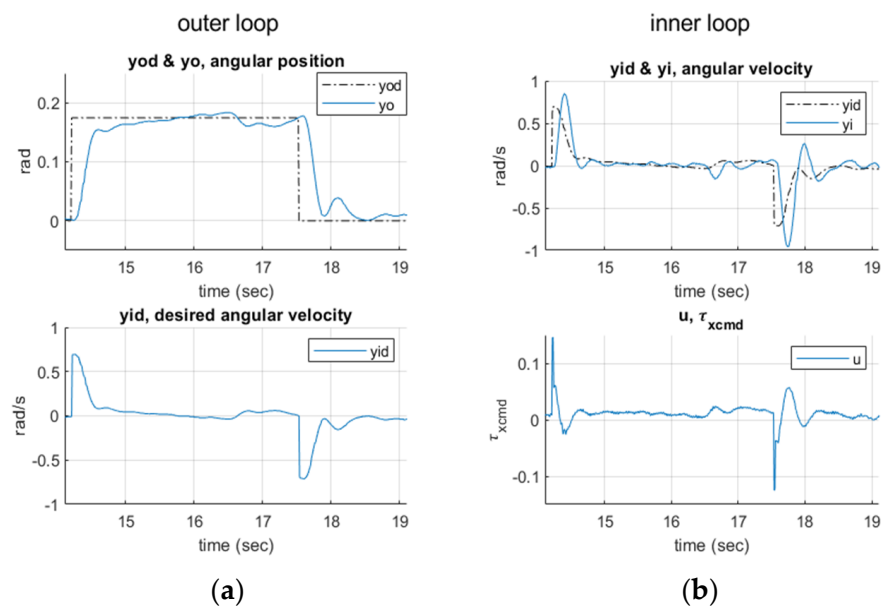


Figure 32. Cascaded Proportional-Integral-Derivative (PID) control for the roll angle. (a) Outer loop tracking performance; (b) Inner loop tracking performance.

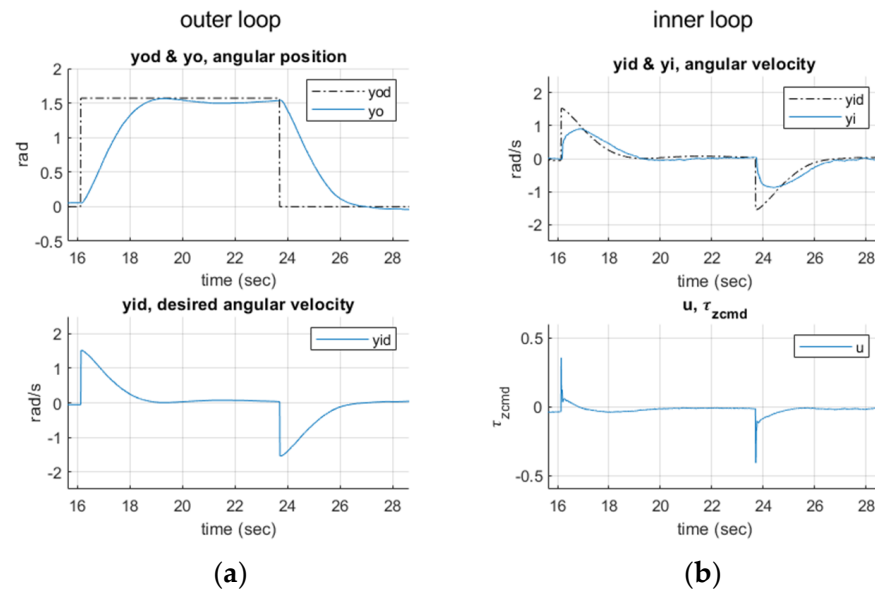


Figure 33. Cascaded PID control for the yaw angle. (a) Outer loop tracking performance; (b) Inner loop tracking performance.

The closed loop performance meets the control specifications. The yaw angle control response exhibits overdamped behavior, and the angular velocity converges to zero in the inner loop. High-frequency interference of the sensor is observed to cause a small oscillation phenomenon in the roll control. The control in the yaw is observed with a peak value close to the pre-defined saturation limit.

3.3.4. Compensator Control Design and Experiment Result

The cascaded control structure shown in Figure 7 is used again in the compensator design. C_i and C_o are now allowed in the format of a higher-order lead-lag compensator. The inner loop compensator is designed with a higher-order structure due to the complex dynamic of plant G , while the outer loop compensator is kept simple since there is only an additional integral block. The system is observed with a time delay of 0.06 s in testing. This system time delay is approximated with 2nd order Padé approximation in the root locus design.

The root locus design plots are shown in Figure 34 for C_i and C_o , respectively. It is observed that the system delay causes NMP zero in the right-half plane, which limits the compensator gain value for a stable closed loop response. The final selected C_i and C_o are shown in Equations (32) and (33), respectively. The designed compensators are simulated with the identified system model, which is shown in Figure 35. It is observed to reach the final value with fluctuation. This is confirmed by the loci plot that the design has a small damping value in its dominating pole pair.

$$C_i(s) = \frac{26(s+4)(s+15)(s+18)}{s(s+50)^3} \quad (32)$$

$$C_o(s) = 3 \quad (33)$$

The yaw control is designed in the same manner. Nonetheless, the yaw model has a smaller DC gain compared to roll and pitch. Therefore, it requires a larger gain in the controller for tracking performance. The loci plot is shown in Figure 36 for C_i and C_o . Since the closed loop poles for the outer loop are required to be around -1.3 , the closed loop poles for the inner loop should be designed further into the left-half plane. The final selected C_i and C_o for yaw control are shown in Equations (34) and (35), respectively. The

yaw control simulation plot is shown in Figure 37. It reaches the final value with fluctuation due to the small damping value in its dominating pole pair.

$$C_i(s) = \frac{100(s + 3)(s + 16)^2}{s(s + 50)^3} \tag{34}$$

$$C_o(s) = 1 \tag{35}$$

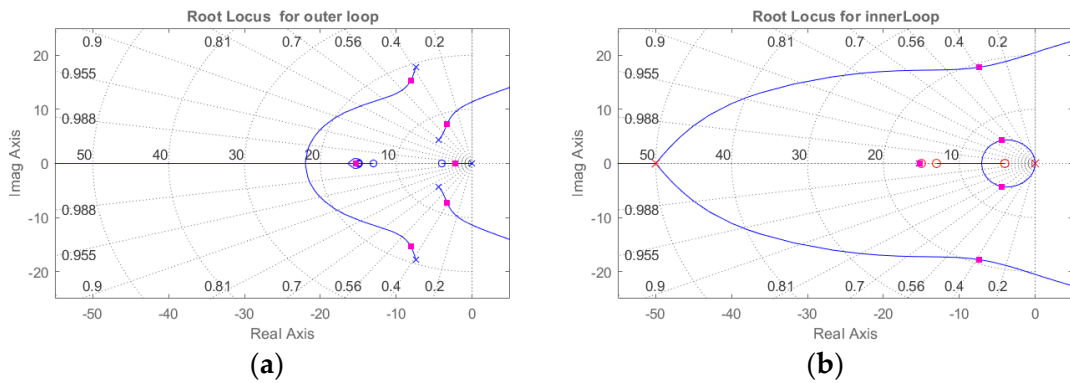


Figure 34. Root locus of the roll angle controller. (a) Outer loop root loci; (b) Inner loop root loci.

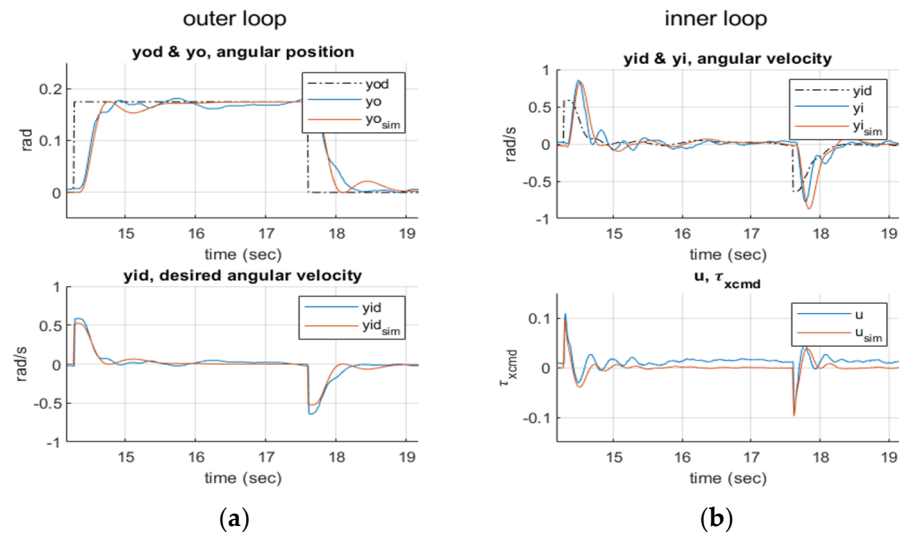


Figure 35. Cascaded compensator control for the roll angle controller. (a) Outer loop tracking performance; (b) Inner loop tracking performance.

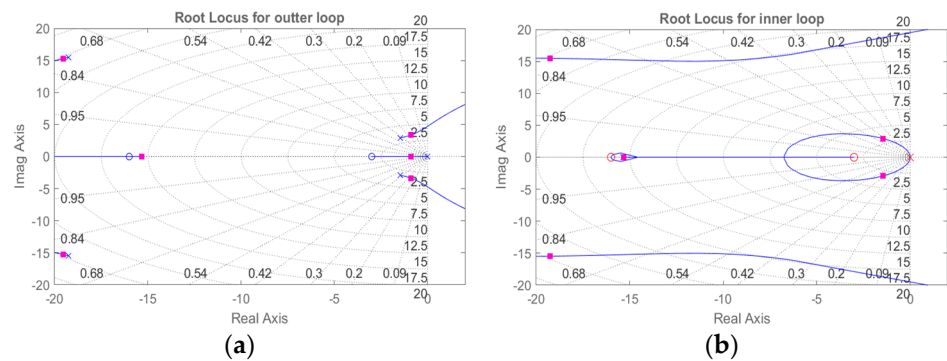


Figure 36. Root locus of the yaw angle controller. (a) Outer loop root locus; (b) Inner loop root locus.

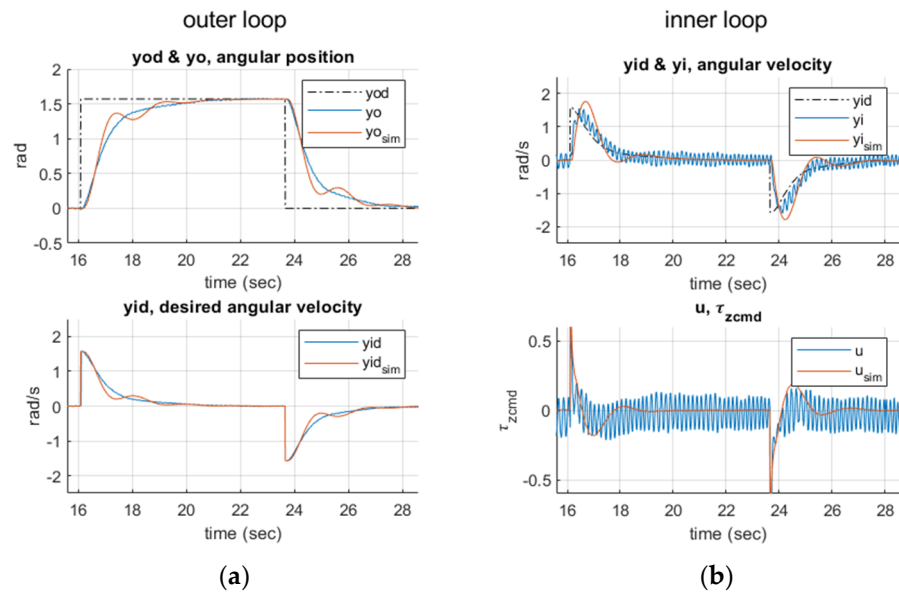


Figure 37. Cascaded compensator control for the yaw angle controller. (a) Outer loop tracking performance; (b) Inner loop tracking performance.

Fast oscillation in the experiment result is observed in the inner loop response, which is also not found in other control simulation setups that do not consider the system time delay and motor dynamics. However, the simulation and the experiment results tend to represent a similar response without this oscillation.

Figure 38 presents the simulation result of the roll control inner loop response under different model considerations. The blue line indicates the recorded measurement, while the red line represents the simulation result using a detailed model, the green line using a model with no time delay, and the yellow line with no time delay and no motor dynamics. The responses are quite different from the recorded measurement for those cases in which the time delay is not considered. If the time delay is not considered, the implemented closed loop system might diverge in the response or even go unstable without notice in the design stage, since the closed loop poles move toward the right half plane under large feedback gain in the implementation.

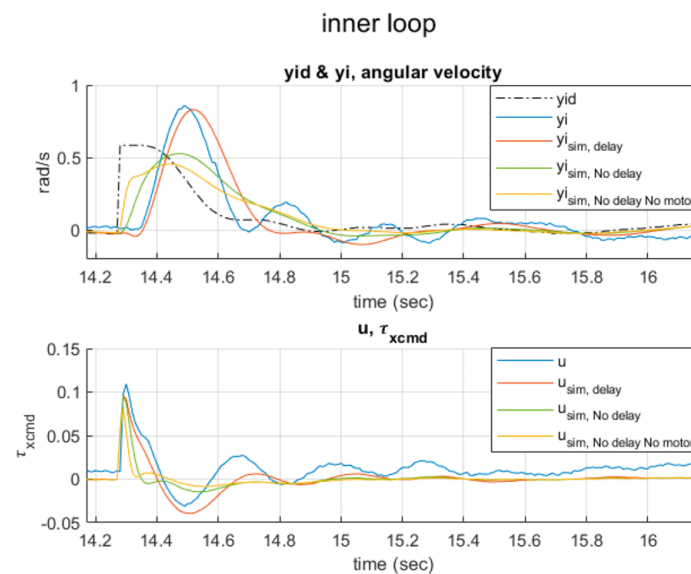


Figure 38. Compensator control simulations of the inner loop with different considerations.

3.3.5. LQR and LQG Control Design and Experiment Results

The system states corresponding to the Q matrix in the LQR design are angular position, angular velocity, and cumulative error. If the weight of the cumulative error has increased, the importance of control in the cumulative error will be increased, and the convergence speed will become faster. If the weight of the angular position is increased, the importance of control in the corresponding angular position will increase, and the convergence speed will be slower. The increase in the weight of the angular velocity and the R weighting matrix corresponding to the control amount have a similar effect, which will also slow down the convergence speed. Since the feedback gain is affected by the relative values among elements in the matrices of Q and R , R is initially set to 1000, and then the design process continues to adjust all other weighting parameters. Compared with adjustments in the angular velocity weighting value, an increase in the weighting of the angular position can effectively reduce the amount of overshoot and avoid excessive feedback gain. Therefore, the Q matrix is mainly used to adjust the cumulative error weighting for convergence speed. Angular position weighting is tuned for the reduction in the overshoot, and the rest weighting values are set to 0.

The designed LQR and LQG control weighting matrices and the closed loop poles are shown in Tables 8 and 9. In Figure 39, the simulated response diagrams are shown. It can be observed that the responses of the two designs are similar.

Table 8. Linear-quadratic parameters for roll and pitch angle control.

Controller	Q	R	Closed Loop Poles
LQG	diag([30 0 800])	1000	$-3.6 \pm 4.8i, -4.4$
LQR	diag([80 0 0 3000])	1000	$-4.3 \pm 6.1i, -5.4, -15.6$

Table 9. Linear-quadratic parameters for yaw angle control.

Controller	Q	R	Closed Loop Poles
LQG	diag([40 0 800])	1000	$-1.1 \pm 1.5i, -1.3$
LQR	diag([40 0 0 3000])	1000	$-1.1 \pm 1.5i, -1.3, -15.4$

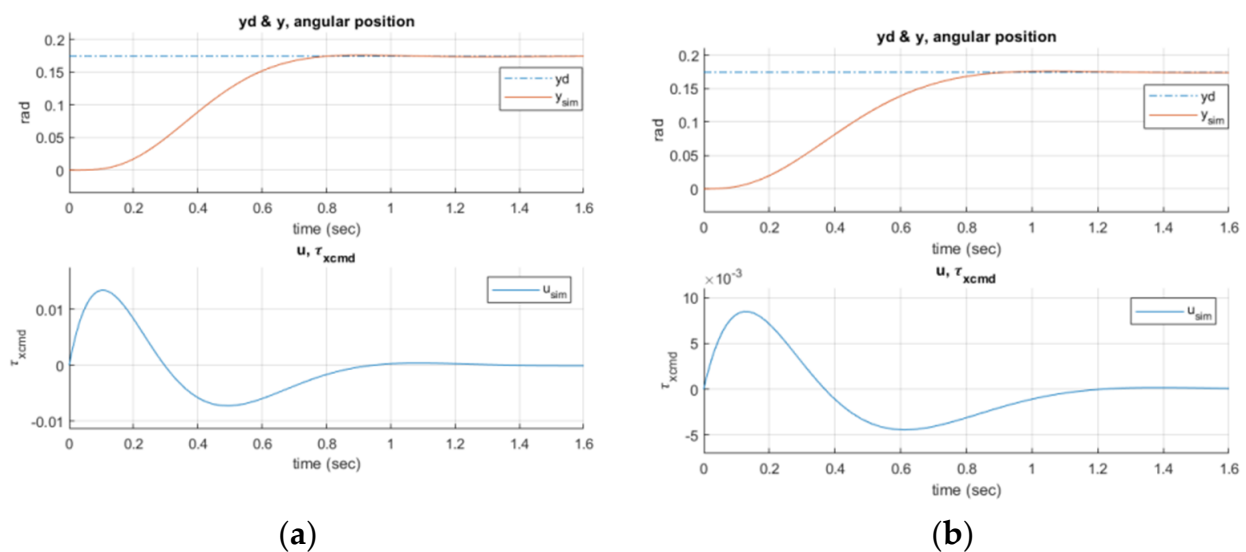


Figure 39. Roll angle control response simulation. (a) LQR design; (b) LQG design.

In general, the LQR design has better robustness than the LQG design. However, if the system model used in the design is very different from the actual system, the performance

of the LQR will be degraded or even diverge. Figures 40 and 41 are the control experiment results of roll and yaw angle compared to its simulation. The performance of the LQR and LQG designs is similar in the yaw control because the motor poles are located 15 times left to the LQR closed loop poles, causing almost no influence in the closed loop performance.

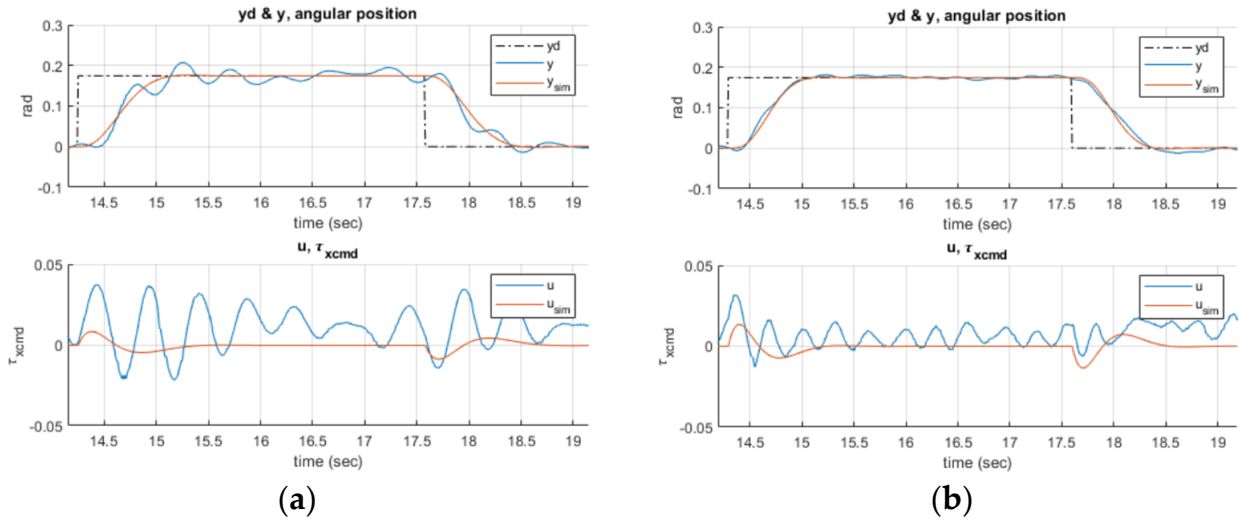


Figure 40. Roll angle control results in real flight tests. (a) LQR design; (b) LQG design.

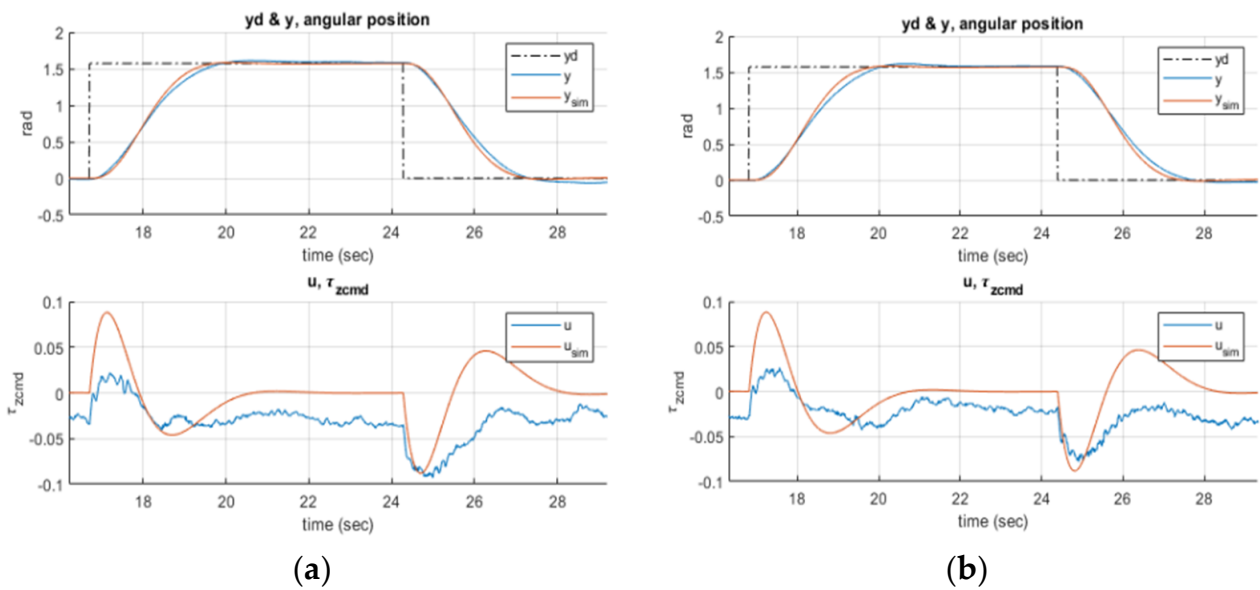


Figure 41. Yaw angle control results in real flight tests. (a) LQR design; (b) LQG design.

3.3.6. Altitude Control

For the altitude control, the control performance of the compensator design and of the LQG design is shown in Figures 42 and 43, respectively. The simulation agrees with the experimental response. Overshooting is observed in both design approaches, while the LQG control response is smoother and faster.

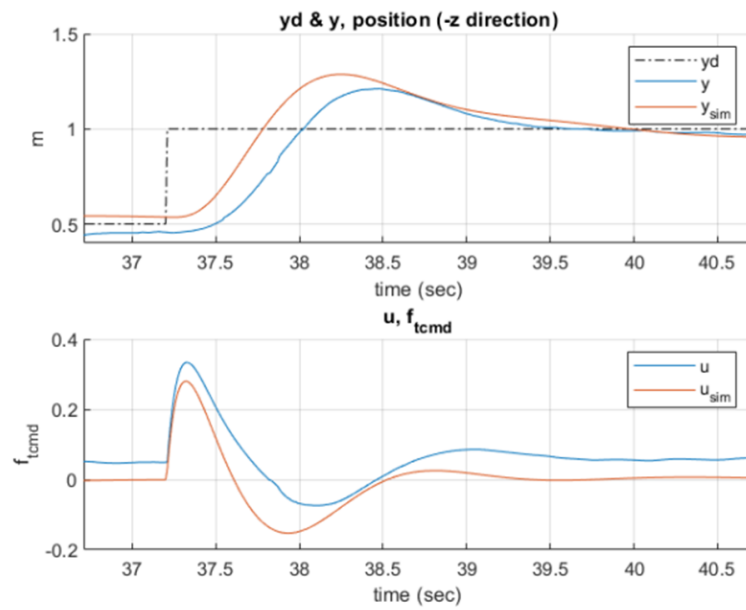


Figure 42. Compensator control for altitude.

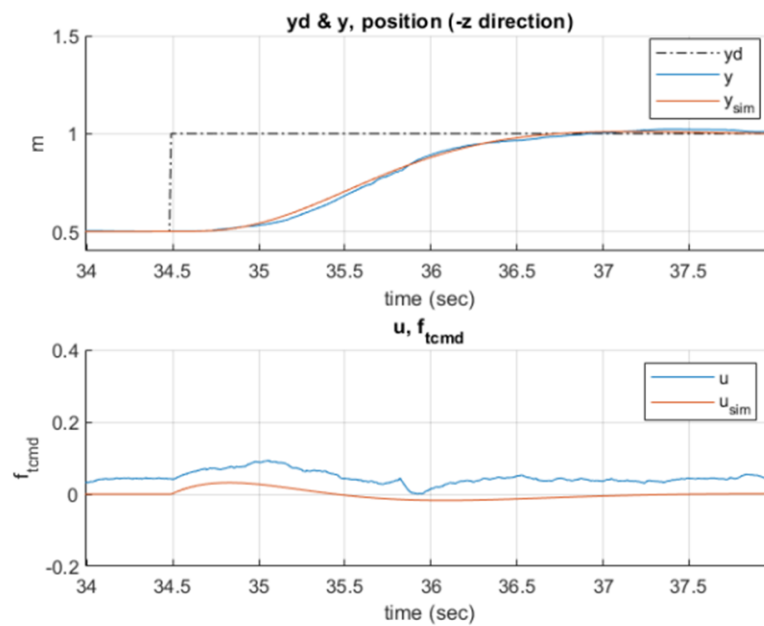


Figure 43. LQG control for altitude.

3.3.7. Comparison of Different Controller Output Response

The comparison plots of different controllers in roll angle and yaw angle output response are shown in Figures 44 and 45, respectively. It is observed that LQG has a smaller overshoot and less oscillation than the other three controllers in the roll control. In addition, it is observed with a smaller peak in the control amount when the angle command is switched for the LQG design, compared to other designed controllers.

For the altitude control, a comparison between the LQG and the lead-lag compensator is shown in Figure 46. The performance of the LQG is better than that of the lead-lag compensator since the control amount of the LQG controller is smoother so that it is kept away from control saturation with a similar settling time.

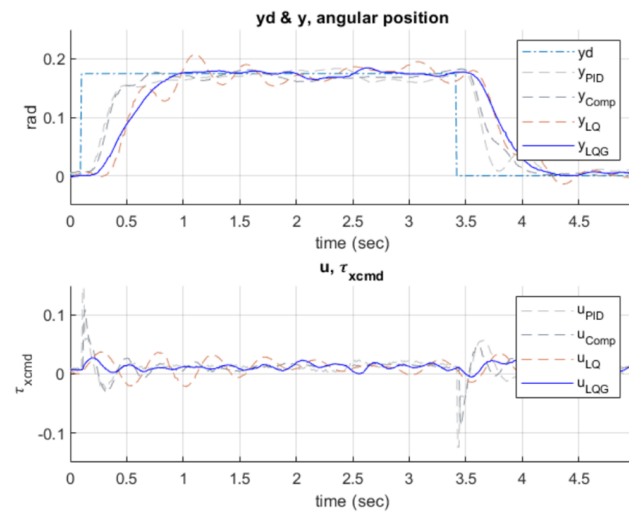


Figure 44. Comparison of different controller output responses of the roll angle.

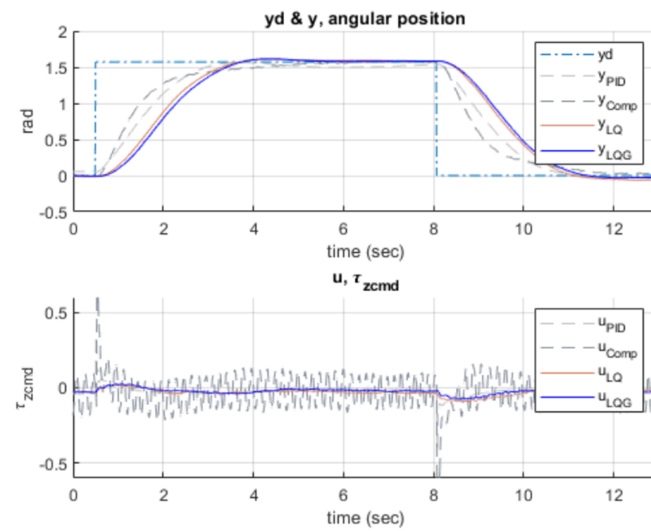


Figure 45. Comparison of different controller output responses of yaw angles.

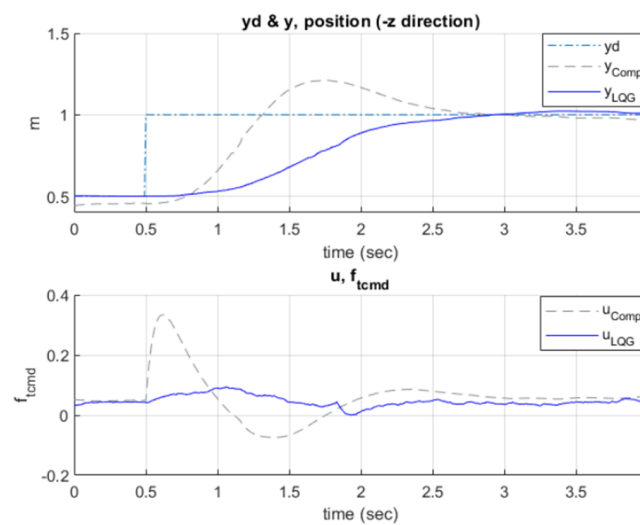


Figure 46. Comparison of different controller output responses of altitude control.

3.4. Position Tracking Control

An LQG controller is further designed to realize x-y position control of the quadrotor with achieved attitude and altitude control design. The flight test result is degraded due to GPS precision and the achieved attitude response time.

3.4.1. Control Design

Since the settling time of achieved quadrotor attitude control is about 1 s, the settling time of position control is set to around 4~6 s for a 1-m step, and the angle command is limited to less than 0.17 rad when giving a step response of 1 m to fulfill the requirements of the system linearization. The LQG parameters for the x-y position tracking control are selected as in Table 10. It utilizes the same feedback control structure by augmenting an integrator to eliminate position steady state error, as in Figure 8.

Table 10. LQG parameters for position control.

Controller	Q	R	Closed Loop Poles
LQG without integrator	diag([10 8])	1000	$-0.83 \pm 0.55 i$
LQG with integrator	diag([13 0 10])	1000	$-0.7 \pm 0.89 i, -0.77$

3.4.2. Experiment Environment

The flight test is conducted on a lawn outdoors with a length and width of 35 m to reduce the effect of surrounding buildings on GPS signal receiving. The steps of the test are to fly the quadrotor to the center of the lawn at a 1-m fixed height accomplished by the attitude and altitude control operated by the user and then hold the position for 10 s to confirm that the GPS signal converges to the current position of the quadrotor as much as possible. After that, the position control is switched on to fly the quadrotor in a predefined 1-m square trajectory. Position data are captured using myRIO with a sampling frequency of 100 Hz.

3.4.3. LQG Position Control Design Experiment

An integrator is in the position control loop to eliminate position steady-state error. Position tracking performed well along the edge trajectory under wind. However, it is observed with a circling phenomenon at each corner position, where the quadrotor is asked to hover from the next motion path execution. This may be due to the GPS precision and the slow response in attitude control. The experiment results are shown in Figures 47–49.

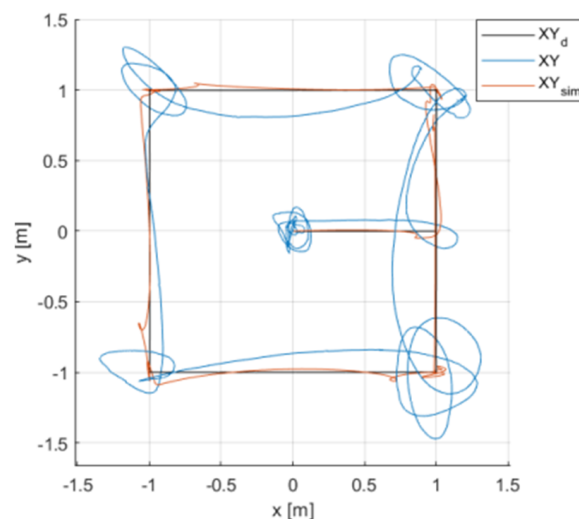


Figure 47. LQG control record with integrator position tracking performance in Earth coordinates.

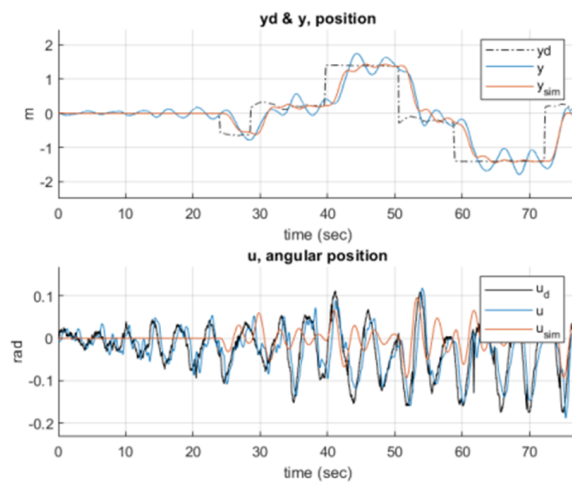


Figure 48. LQG design x-axis response in body coordinates.

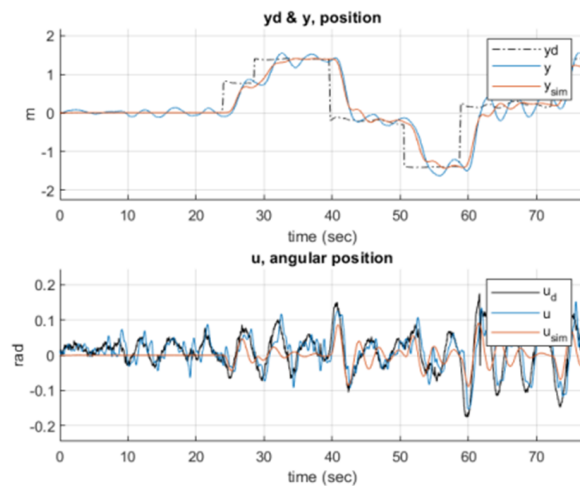


Figure 49. LQG design y-axis response in body coordinates.

4. Conclusions

In the design and implementation of feedback control, there is always a trade-off among actuator effort, sensor noise, system time delay, model complexity, and control performance. Nonetheless, these constraints are not usually considered in advance in the control design, which sometimes causes a gap in simulation and implementation.

In this study, the limitations caused by hardware in the feedback control design are discussed and clearly observed. These hardware properties cause different levels of impact when using different design approaches and have limited the performance achievable in reality. The motor used in this study has a settling time of 0.3 s; therefore, it is not feasible to require a control response faster than that. Moreover, the control performance tends to be poorer in implementation if these constraints due to hardware are not considered in the control design and sometimes can lead to instability, especially system time delay.

PID trial and error tuning is a method of synthesizing the control based on the system response online. Therefore, it inherently considers the actual hardware properties in the design process. However, limited by its structure and number of design iterations, its performance is restricted in this study. For the compensator design, the actuator dynamic and the system time delay cause more issues if not incorporated into the design process. Without the consideration of motor dynamics, the real response deviates from the simulation and performs worse in the implementation. Without the consideration of system time delay, the closed system tends to be unstable with increased high gain in the design.

On the other hand, the LQ control design provides a relatively smooth ramping in control effort, which is less likely to cause actuator saturation. The LQG design is also observed with less discrepancy in the simulation and real response and provides better performance. In addition, the consideration of system time delay has been observed with less influence in the LQ design compared to compensator design, as the LQ automatically synthesizes an optimum control structure to achieve desired performance rather than compensator design, which has to assign the control structure manually in an elaborate way. The motor dynamic has less influence in the design if the system closed loop poles were designed to the right and far away from the motor poles, as in the case of LQR design in this study. Although the attitude and altitude LQG control performance is acceptable in the implementation, the x-y position tracking is still not favorable due to the poor GPS precision and the slow response in the achieved attitude control.

An attitude and altitude flight controller is realized by NI myRIO in this study using a linear control design method and serves as a foundation for the development of autonomous multi-rotors. It demonstrates the limitations post by hardware to obtain agile maneuverability regardless of the control design methods. The quadrotor can be further improved by a more powerful actuator, higher performance embedded controller, and more accurate sensor or estimating technology to investigate advanced control strategies and to achieve better maneuverability or even aggressive flying. With the flexibility offered by NI myRIO and its software developing interface, an intelligent system of drones and ground stations for various applications is to be achieved, such as for building safety inspections, unmanned cargo transport, aerial photography, and geographic mapping.

Author Contributions: Conceptualization, J.-Y.H. and C.-Y.L.; data curation, J.-Y.H.; formal analysis, J.-Y.H.; funding acquisition, C.-Y.L.; investigation, J.-Y.H., P.-J.C., C.-D.P. and C.-Y.L.; methodology, J.-Y.H. and C.-Y.L.; project administration, C.-Y.L.; resources, C.-Y.L.; software, J.-Y.H. and P.-J.C.; supervision, C.-Y.L.; writing—original draft preparation, J.-Y.H.; writing—review and editing, J.-Y.H., P.-J.C., C.-D.P. and C.-Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Science and Technology, Taiwan, R.O.C. under Grant No. MOST 111-2221-E-011-121-.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available upon request to the corresponding author.

Acknowledgments: The authors would like to express their gratitude to the funding support by the Ministry of Science and Technology, Taiwan, R.O.C. Their thanks extend to the members of Electro-Mechanical Systems Integration Laboratory and Multivariable Control Systems Laboratory, National Taiwan University of Science and Technology, for their hardware and technical support during the experiments.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

LQG	Linear–quadratic–Gaussian
LQR	Linear-quadratic-regulator
GPS	Global positioning system
IMU	Inertial measurement unit
PID	Proportional-Integral-Derivative
QFT	Quantitative feedback theory
MPC	Model predictive control
MCU	Microprocessor control unit
BEC	Battery eliminator circuit

References

1. Liu, X.; Nardari, G.V.; Ojeda, F.C.; Tao, Y.; Zhou, A.; Donnelly, T.; Qu, C.; Chen, S.W.; Romero, R.A.F.; Taylor, C.J.; et al. Large-Scale Autonomous Flight with Real-Time Semantic SLAM Under Dense Forest Canopy. *IEEE Robot. Autom. Lett.* **2022**, *7*, 5512–5519. [CrossRef]
2. Loianno, G.; Brunner, C.; McGrath, G.; Kumar, V. Estimation, Control, and Planning for Aggressive Flight with a Small Quadrotor with a Single Camera and IMU. *IEEE Robot. Autom. Lett.* **2016**, *2*, 404–411. [CrossRef]
3. Schroth, L. The Drone Market Size 2020–2025: 5 Key Takeaways. Drone Industry Insights. Available online: <https://droneii.com/the-dronemarket-size-2020-2025-5-key-takeaways> (accessed on 21 May 2022).
4. Sabatino, F. Quadrotor Control: Modeling, Nonlinear control Design, and Simulation. Master's Thesis, School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden, 2015.
5. Walid, M.; Slaheddine, N.; Mohamed, A.; Lamjed, B. Modeling and control of a quadrotor UAV. In Proceedings of the 2014 15th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), Hammamet, Tunisia, 21–23 December 2014; pp. 343–348.
6. Wu, Y.; Hu, K.; Sun, X.-M. Modeling and Control Design for Quadrotors: A Controlled Hamiltonian Systems Approach. *IEEE Trans. Veh. Technol.* **2018**, *67*, 11365–11376. [CrossRef]
7. Six, D.; Briot, S.; Erskine, J.; Chriette, A. Identification of the Propeller Coefficients and Dynamic Parameters of a Hovering Quadrotor from Flight Data. *IEEE Robot. Autom. Lett.* **2020**, *5*, 1063–1070. [CrossRef]
8. Sun, S.; de Visser, C. Aerodynamic model identification of a quadrotor subjected to rotor failures in the high-speed flight regime. *IEEE Robot. Autom. Lett.* **2019**, *4*, 3868–3875. [CrossRef]
9. Sun, Y. Modeling, Identification and Control of a Quad-Rotor Drone Using Low-Resolution Sensing. Master's Thesis, University of Illinois at Urbana-Champaign, Champaign, IL, USA, 2012.
10. Lu, S. Modeling, Control and Design of a Quadrotor Platform for Indoor Environments. Master's Thesis, Arizona State University, Tempe, AZ, USA, 2018.
11. Derafa, L.; Madani, T.; Benallegue, A. Dynamic modelling and experimental identification of four rotors helicopter parameters. In Proceedings of the 2006 IEEE International Conference on Industrial Technology, Mumbai, India, 15–17 December 2006; pp. 1834–1839.
12. Mahony, R.; Hamel, T.; Pflimlin, J.-M. Complementary filter design on the special orthogonal group SO_3 . In Proceedings of the 44th IEEE Conference on Decision and Control, Seville, Spain, 15 December 2005; pp. 1477–1484.
13. Madgwick, S. An Efficient Orientation Filter for Inertial and Inertial/Magnetic Sensor Arrays. *Rep. x-Lo Univ. Bristol (UK)* **2010**, *25*, 113–118.
14. Marins, J.L.; Yun, X.; Bachmann, E.R.; McGhee, R.B.; Zyda, M.J. An extended Kalman filter for quaternion-based orientation estimation using MARG sensors. In Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the Next Millennium (Cat. No. 01CH37180), Maui, HI, USA, 29 October–3 November 2001; Volume 4, pp. 2003–2011.
15. Lee, H.; Lee, C.; Jeon, H.; Son, J.J.; Son, Y.; Han, S. Interference-Compensating Magnetometer Calibration with Estimated Measurement Noise Covariance for Application to Small-Sized UAVs. *IEEE Trans. Ind. Electron.* **2019**, *67*, 8829–8840. [CrossRef]
16. Omotuyi, O.; Kumar, M. UAV Visual-Inertial Dynamics (VI-D) Odometry using Unscented Kalman Filter. *IFAC-PapersOnLine* **2021**, *54*, 814–819. [CrossRef]
17. Wang, X.; Wang, W. Nonlinear Signal-Correction Observer and Application to UAV Navigation. *IEEE Trans. Ind. Electron.* **2018**, *66*, 4600–4607. [CrossRef]
18. Berkane, S.; Tayebi, A.; de Marco, S. A nonlinear navigation observer using IMU and generic position information. *Automatica* **2021**, *127*, 109513. [CrossRef]
19. Kahili, K.; Bouhali, O.; Khenfri, F.; Rizoug, N. Robust Intelligent Self-tuning PID Controller for the Body-rate Stabilization of Quadrotors. In Proceedings of the IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society, Lisbon, Portugal, 14–17 October 2019; Volume 1, pp. 5281–5286.
20. Liu, Y.; Bei, H.; Li, W.; Huang, Y. Design of Altitude Control System for Quadrotor UAV Based on PID and LADRC. In Proceedings of the 2021 4th International Conference on Pattern Recognition and Artificial Intelligence (PRAI), Yibin, China, 20–22 August 2021; pp. 301–305.
21. Idrissi, M.; Salami, M.; Annaz, F. Modelling, simulation and control of a novel structure varying quadrotor. *Aerosp. Sci. Technol.* **2021**, *119*, 107093. [CrossRef]
22. Outeiro, P.; Cardeira, C.; Oliveira, P. Multiple-model control architecture for a quadrotor with constant unknown mass and inertia. *Mechatronics* **2021**, *73*, 102455. [CrossRef]
23. Cohen, M.R.; Abdulrahim, K.; Forbes, J.R. Finite-Horizon LQR Control of Quadrotors on $SE_2(3)$. *IEEE Robot. Autom. Lett.* **2020**, *5*, 5748–5755. [CrossRef]
24. Noormohammadi-Asl, A.; Esrafilian, O.; Arzati, M.A.; Taghirad, H.D. System identification and H_∞ -based control of quadrotor attitude. *Mech. Syst. Signal Process.* **2020**, *135*, 106358. [CrossRef]
25. Rekabi, F.; Shirazi, F.A.; Sadigh, M.J.; Saadat, M. Nonlinear H_∞ Measurement Feedback Control Algorithm for Quadrotor Position Tracking. *J. Frankl. Inst.* **2020**, *357*, 6777–6804. [CrossRef]

26. Mardan, M.; Esfandiari, M.; Sepehri, N. Attitude and position controller design and implementation for a quadrotor. *Int. J. Adv. Robot. Syst.* **2017**, *14*, 1729881417709242. [[CrossRef](#)]
27. Xiong, J.; Pan, J.; Chen, G.; Zhang, X.; Ding, F. Sliding Mode Dual-Channel Disturbance Rejection Attitude Control for a Quadrotor. *IEEE Trans. Ind. Electron.* **2021**, *69*, 10489–10499. [[CrossRef](#)]
28. Tian, B.; Cui, J.; Lu, H.; Zuo, Z.; Zong, Q. Adaptive Finite-Time Attitude Tracking of Quadrotors with Experiments and Comparisons. *IEEE Trans. Ind. Electron.* **2019**, *66*, 9428–9438. [[CrossRef](#)]
29. Ou, T.-W.; Liu, Y.-C. Adaptive backstepping tracking control for quadrotor aerial robots subject to uncertain dynamics. In Proceedings of the 2019 American Control Conference (ACC), Philadelphia, PA, USA, 10–12 July 2019; pp. 1–6.
30. Martins, L.; Cardeira, C.; Oliveira, P. Feedback Linearization with Zero Dynamics Stabilization for Quadrotor Control. *J. Intell. Robot. Syst.* **2020**, *101*, 7. [[CrossRef](#)]
31. Zanelli, A.; Horn, G.; Frison, G.; Diehl, M. Nonlinear model predictive control of a human-sized quadrotor. In Proceedings of the 2018 European Control Conference (ECC), Limassol, Cyprus, 12–15 June 2018; pp. 1542–1547.
32. Zhao, W.; Liu, H.; Wang, B. Model-free attitude synchronization for multiple heterogeneous quadrotors via reinforcement learning. *Int. J. Intell. Syst.* **2021**, *36*, 2528–2547. [[CrossRef](#)]
33. Mysore, S.; Mabsout, B.; Saenko, K.; Mancuso, R. How to Train Your Quadrotor: A Framework for Consistently Smooth and Responsive Flight Control via Reinforcement Learning. *ACM Trans. Cyber-Phys. Syst.* **2021**, *5*, 1–24. [[CrossRef](#)]
34. Rubí, B.; Morcego, B.; Pérez, R. A Deep Reinforcement Learning Approach for Path Following on a Quadrotor. In Proceedings of the 2020 European Control Conference (ECC), St. Petersburg, Russia, 12–15 May 2020; pp. 1092–1098.
35. Hermand, E.; Nguyen, T.W.; Hosseinzadeh, M.; Garone, E. Constrained Control of UAVs in Geofencing Applications. In Proceedings of the 26th Mediterranean Conference on Control and Automation (MED), Zadar, Croatia, 19–22 June 2018; pp. 217–222.
36. Khan, M.; Zafar, M.; Chatterjee, A. Barrier Functions in Cascaded Controller: Safe Quadrotor Control. In Proceedings of the 2020 American Control Conference (ACC), Denver, CO, USA, 1–3 July 2020; pp. 1737–1742.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.