Quadcopter Attitude and Altitude control

Project Course Report

Submitted by:

Harshada Kale

Roll No: **23110154**

ME 399: Project Course

Indian Institute of Technology Gandhinagar

April 2025

Quadcopter Attitude rate and Altitude control

Harshada Kale[1]

[a]*Prof.Vineet Vashista*

Contents

1 Abstract ii
2 Objective ii
3 Methodology ii
3.1 Derivation of Equations of Motion Using Euler-Lagrange ii
3.2 Drone Physical Parameters iii
4 Battery and Motor Modeling iii
4.1 Battery Specs iii
4.2 Propeller Thrust and Torque iii
4.3 Translational Dynamics iv
4.4 Rotational Dynamics v
4.5 Moments of Inertia v
4.6 Summary of Forces and Moments in Simulation . . v
5 Implementation in MATLAB/SIMULINK v
6 Results and Discussions v
6.1 Angular Velocity (Roll , Pitch and Yaw rates) Responsev
6.2 Non zero attitude rates vi
6.3 Altitude Control vii

1. Abstract

This project presents the modeling, linearization, and simulation of a quadcopter's dynamics using the Euler-Lagrange formalism and its subsequent implementation in Simulink. The quadcopter is modeled as a rigid body with six degrees of freedom, and its equations of motion are derived using the Euler-Lagrange approach. The nonlinear equations are then linearized around a hover condition to facilitate control design and simulation. The full system, including motor dynamics, aerodynamic drag, and control allocation, is implemented in Simulink using MATLAB function blocks. Simulation results demonstrate the model's response to control inputs and disturbances, providing insights into the quadcopter's behavior and validating the effectiveness of the modeling approach.

2. Objective

The primary objectives of this project are:

- To implement the equations of motions of Quadcopter and system components in Simulink for simulation.
- To analyze the simulation results, for attitude (Roll, pitch and yaw) and Altittude control

3. Methodology

3.1. Derivation of Equations of Motion Using Euler-Lagrange

3.1.1. System Definition

The quadcopter is modeled as a rigid body with six degrees of freedom:

- **Translational coordinates:** x, y, z (inertial frame)
- **Rotational coordinates:** ϕ (roll), θ (pitch), ψ (yaw)
- **Velocities:** $\dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi}$

The inputs to the system are the thrusts F_1, F_2, F_3, F_4 and torques $\tau_1, \tau_2, \tau_3, \tau_4$ generated by the four rotors.

3.1.2. Kinetic and Potential Energy

Potential Energy

$$V = mgz \tag{1}$$

where g is the gravitational acceleration.

Kinetic Energy

$$T = \frac{1}{2}m\left(\dot{x}^2 + \dot{y}^2 + \dot{z}^2\right) + \frac{1}{2}\boldsymbol{\omega}^T I \boldsymbol{\omega} \tag{2}$$

where m is the mass, $\boldsymbol{\omega}$ is the angular velocity vector in the body frame, and I is the inertia matrix.

3.1.3. Lagrangian Formulation

The Lagrangian L is defined as the difference between the kinetic and potential energy:

$$L = T - V \tag{3}$$

The equations of motion are derived using the Euler-Lagrange equation for each generalized coordinate q_i:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = Q_i \tag{4}$$

where Q_i represents the generalized forces, including thrust, rotor moments, aerodynamic drag, and disturbances.

3.1.4. Force and Moment Allocation

The total thrust generated by the rotors is:

$$F_T = F_1 + F_2 + F_3 + F_4 \tag{5}$$

The moments about the body axes are:

$$M_x = \frac{l}{2}\left[(F_3 + F_4) - (F_1 + F_2)\right] \tag{6}$$

$$M_y = \frac{l}{2}\left[(F_3 + F_2) - (F_1 + F_4)\right] \tag{7}$$

$$M_z = \tau_4 - \tau_1 + \tau_2 - \tau_3 \tag{8}$$

where l is the distance from the center to each rotor.

3.1.5. Translational Dynamics (Inertial Frame)

$$m\ddot{x} = -F_T\left(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi\right) \tag{9}$$

$$m\ddot{y} = -F_T\left(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi\right) \tag{10}$$

$$m\ddot{z} = F_T\cos\phi\cos\theta - mg \tag{11}$$

where:

- F_T is the total thrust generated by the rotors,
- m is the mass of the quadcopter,
- g is the acceleration due to gravity,
- (ϕ, θ, ψ) are the roll, pitch, and yaw angles, respectively.

3.1.6. Rotational Dynamics (Body Frame)

$$I_x \dot{p} = (I_y - I_z)qr + M_x \tag{12}$$
$$I_y \dot{q} = (I_z - I_x)pr + M_y \tag{13}$$
$$I_z \dot{r} = (I_x - I_y)pq + M_z \tag{14}$$

where:

- p, q, r are the angular velocity components about the body x, y, and z axes, respectively,
- I_x, I_y, I_z are the moments of inertia about the respective axes,
- M_x, M_y, M_z are the moments (torques) generated by the rotors.

Relationship Between Body Rates and Euler Angle Rates The angular velocities (p, q, r) are related to the time derivatives of the Euler angles $(\dot{\phi}, \dot{\theta}, \dot{\psi})$ via:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{15}$$

These six equations of motion describe the evolution of the quadcopter's position (x, y, z) and orientation (ϕ, θ, ψ) in three-dimensional space. The translational dynamics are expressed in the inertial (world) frame, while the rotational dynamics are typically expressed in the body frame.

3.2. Drone Physical Parameters

- **Diagonal length:** $L_{\text{diag}} = 335$ mm
- **Weight:** $m = 0.743$ kg
- **Frontal Area:** $A = 0.0197\ \text{m}^2$
- **Propeller Size:** 8×4 inches
- **Air Density:** $\rho = 1.225\ \text{kg/m}^3$ (sea level)

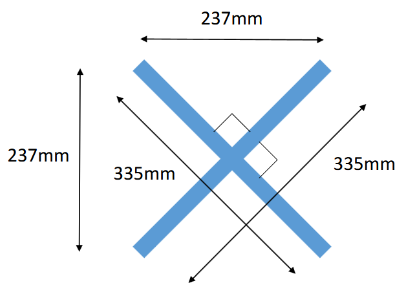

Figure 1. drone visualization

4. Battery and Motor Modeling

4.1. Battery Specs

$$V_{\text{nominal}} = 3.8 \times 3 = 11.4\ \text{V}$$
$$I_{\max} = 3.830\ \text{Ah} \times 20 = 77\ \text{A}$$

4.1.1. Motor RPM Equation

The loaded RPM for a given voltage V is modeled as:

$$\text{RPM} = -2.6931 \cdot V^3 + 1400 \cdot V$$

This accounts for a non-linear drop in RPM with increasing voltage due to loading effects.

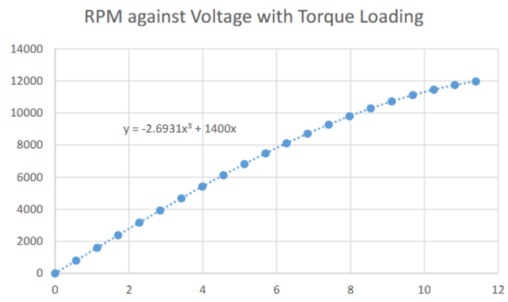

Figure 2. RPM vs Voltage [1]

4.2. Propeller Thrust and Torque

Thrust Calculation

Thrust generated by a propeller is given by :

$$\text{Thrust} = C_T \cdot \rho \cdot n^2 \cdot D^4$$

where:

- C_T = coefficient of thrust (function of RPM)
- n = revolutions per second
- D = propeller diameter (m)

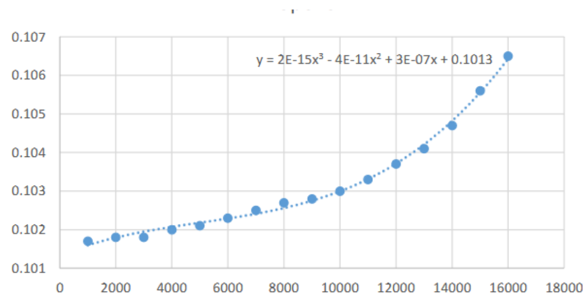

Figure 3. Thrust coefficient

The coefficient of thrust is modeled as [1]:

$$C_T = 2{\times}10^{-15}{\cdot}\text{RPM}^3 - 4{\times}10^{-11}{\cdot}\text{RPM}^2 + 3{\times}10^{-7}{\cdot}\text{RPM} + 0.1013$$

Torque Calculation

Torque required to overcome drag at a given RPM is obtained from experimental data and fitted polynomials. For example:

$$\text{Torque} = f(\text{RPM})$$

where f is a polynomial fit to the measured data [1].

Motor current and Torque

The torque constant is:

$$K_T = \frac{1}{K_V} = \frac{1}{1400} = 0.000714\ \text{Nm/A}$$

Thus,

$$\text{Torque} = K_T \cdot I \implies I = \frac{\text{Torque}}{K_T}$$

$$\text{Current} = \text{Torque} \cdot 1400$$

$$I = 1400{\cdot}(4 \cdot 10^{-14} \cdot RPM^3 + 8 \cdot 10^{-12} \cdot RPM^2 + 3 \cdot 10^{-6} \cdot RPM) \tag{16}$$

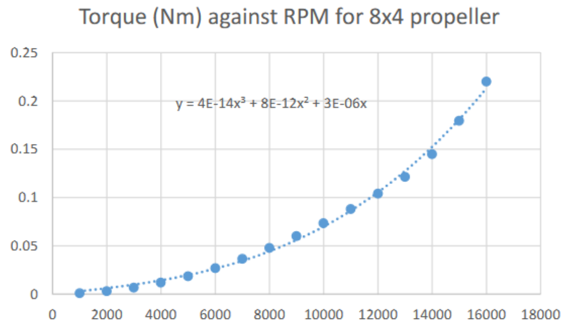

Figure 4. Torque vs RPM

4.3. Translational Dynamics

The translational motion follows Newton's second law:

$$F_x = ma_x$$
$$F_y = ma_y$$
$$F_z = ma_z$$

The net force in each direction is the sum of thrust, gravity, and aerodynamic drag:

$$F_x = F_{\text{prop},x} - \text{Drag}_x$$
$$F_y = F_{\text{prop},y} - \text{Drag}_y$$
$$F_z = F_{\text{prop},z} - mg - \text{Drag}_z$$

Thrust Vector Components for Roll and Pitch

The thrust vector is projected into the inertial frame based on the drone's orientation:

$$F_{\text{prop},x} = \sin\theta\cos\phi(F_1 + F_2 + F_3 + F_4)$$
$$F_{\text{prop},y} = \sin\phi\cos\theta(F_1 + F_2 + F_3 + F_4)$$
$$F_{\text{prop},z} = \cos\phi\cos\theta(F_1 + F_2 + F_3 + F_4)$$

where ϕ is roll, θ is pitch.

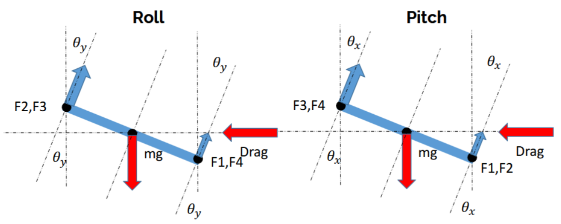

Figure 5

Yaw Thrust Vector Equations

Yaw motion describes the rotation of the drone about its vertical (z) axis, changing the drone's heading. Unlike roll and pitch, which tilt the thrust vector to produce horizontal motion, yaw does not directly alter the direction of the thrust vector in the inertial frame but rather rotates the body frame itself.

Yaw Angle and Thrust Vector Projection The yaw angle, denoted θ_z (or commonly ψ), determines the orientation of the drone's body axes relative to the world frame. When the drone yaws, the x and y components of the total thrust in the world frame are rotated by the yaw angle.

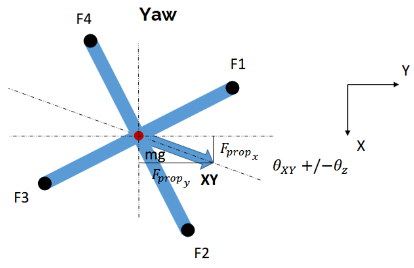

Figure 6

Given:

$$\theta_{XY} = -\,\text{atan2}\left(\frac{F_{\text{prop},x}}{F_{\text{prop},y}}\right)$$

$$XY_{2D} = \sqrt{F_{\text{prop},x}^2 + F_{\text{prop},y}^2}$$

The thrust components in the x and y directions, accounting for yaw, are:

$$F_{\text{prop},x} = XY_{2D} \cdot \sin(\theta_{XY} \pm \theta_z)$$
$$F_{\text{prop},y} = XY_{2D} \cdot \cos(\theta_{XY} \pm \theta_z)$$

where:

- $F_{\text{prop},x}$ and $F_{\text{prop},y}$ are the x and y components of the total thrust in the world frame,
- XY_{2D} is the magnitude of the thrust vector in the xy-plane,
- θ_{XY} is the initial orientation angle of the thrust vector in the xy-plane,
- θ_z is the yaw angle (rotation about z).

- As the drone yaws, the direction of the projected thrust vector in the xy-plane rotates by θ_z.

The yaw angle is incorporated into the thrust vector projection equations to ensure that the drone's horizontal thrust components are always correctly oriented in the world frame, regardless of the drone's heading.

Aerodynamic Drag

Drag is modeled as:

$$\text{Drag} = \frac{1}{2}\rho V^2 A C_d$$

where C_d is the drag coefficient (typically $C_d = 1.0$ for a cube).

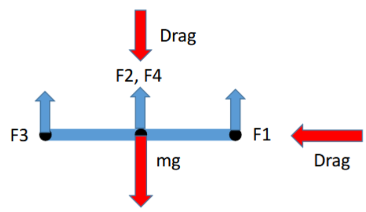

Figure 7. Drag Disturbance

4.4. Rotational Dynamics

The rotational motion is governed by:

$$M_x = I_x\ddot{\phi}$$
$$M_y = I_y\ddot{\theta}$$
$$M_z = I_z\ddot{\psi}$$

where M_x, M_y, M_z are the moments about the principal axes, and I_x, I_y, I_z are the moments of inertia.

Moments from Propellers

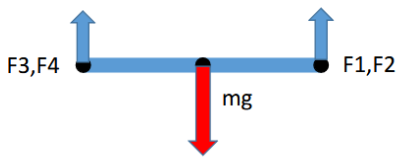

Figure 8

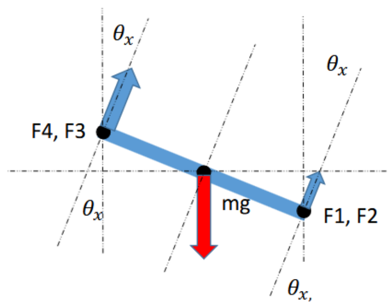

Figure 9

$$M_x = \frac{(F_3 + F_4)d - (F_1 + F_2)d}{2}$$
$$M_y = \frac{(F_3 + F_2)d - (F_4 + F_1)d}{2}$$
$$M_z = (T_4 - T_1 + T_2 - T_3)$$

where d is the arm length from the center to each propeller, and T_i is the torque from each propeller.

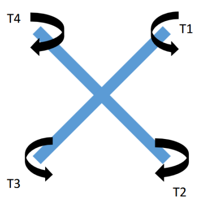

Figure 10. Torque

4.5. Moments of Inertia

The drone is modeled as four rods (arms) of equal mass. For a rod about its end:

$$I = \frac{1}{3}ML^2$$

For the quadrotor:

$$I_z = \frac{1}{3}\frac{m}{4}\left(\frac{L_{\text{diag}}}{2}\right)^2 \times 4$$
$$I_{x,y} = \frac{1}{3}\frac{m}{4}\left(\frac{L_{\text{side}}}{2}\right)^2 \times 4$$

where L_{diag} is the diagonal length, L_{side} is the side length.

4.6. Summary of Forces and Moments in Simulation

- **Thrust:** Generated by each propeller, calculated from RPM and C_T.
- **Torque:** Due to propeller drag, calculated from RPM and experimental fit.
- **Gravity:** mg acts downward.
- **Aerodynamic Drag:** Opposes motion, calculated using projected area, drag coefficient, and velocity.
- **Moments:** Generated by differential thrust and torque from the four rotors.

5. Implementation in MATLAB/SIMULINK

All these equations are implemented as blocks or functions in the SIMULINK model:

- Motor and propeller blocks calculate thrust and torque based on voltage and RPM.
- Dynamics blocks compute translational and rotational accelerations.
- Integration blocks update velocities and positions/angles.
- Drag and disturbance blocks apply environmental effects.

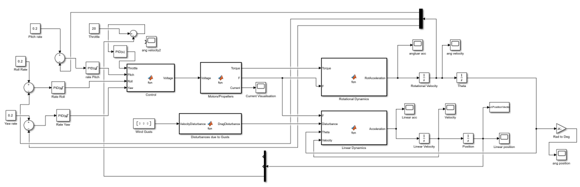

Figure 11. Simulink model

6. Results and Discussions

After tuning the PID gains for roll, pitch, yaw rate, and throttle, I ran the simulation with zero attitude rate inputs and an altitude input of 20. The following plots show the results.

6.1. Angular Velocity (Roll , Pitch and Yaw rates) Response

Following figure shows the angular velocities for roll, pitch, and yaw. All three rotational velocities stay at zero throughout the simulation. This means the attitude rate controller is working well and the drone isn't rotating or drifting when the rate commands are zero.

Altitude (Position) Response

Above figure shows the altitude response. The altitude tracks the setpoint of 20, but there are noticeable oscillations around the setpoint. The P and I gains are high

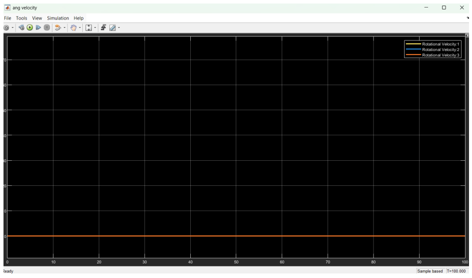

Figure 12. Rotational velocities (roll, pitch, yaw rates) over time. All rates remain at zero, showing stable attitude rate control.

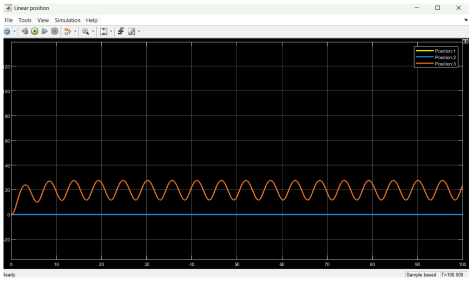

Figure 13. Altitude (Position) response. The altitude oscillates around the setpoint of 20.

enough to reach the target, but the low D gain means there isn't much damping, so the oscillations don't die out quickly but by increasing the D gain simulation crashes.

Angular Position Response

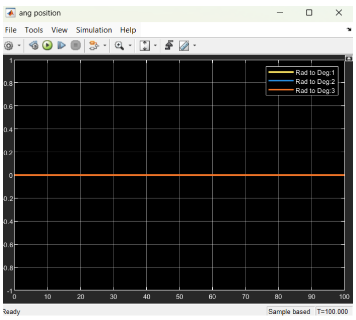

Figure 14. Angular positions (roll, pitch, yaw) over time. All angles remain at zero.

Above figure show the angular positions (roll, pitch, yaw). All the angles remain at zero for the entire simulation, which means the attitude controller is holding the drone steady and not letting it tilt or rotate.

Summary Table

Controller	P	I	D	N	Performance
Altitude	3	1	0.001	100	Oscillates around target value.
Attitude Rate	10	5	0	100	Holds orientation steady.

Table 1. Summary of PID gains and observed performance.

Overall, the attitude rate controller is working well and keeps the drone stable. The altitude controller works but need more tuning to reduce oscillations.

6.2. Non zero attitude rates

For this test, I wanted to see if the tuned PID controllers could handle nonzero attitude rate commands. I set the roll rate and pitch rate commands both to 10, kept the yaw rate at 0, and gave an altitude input of 20.

Case 1: Roll Rate = 10, Pitch Rate = 10, Yaw Rate = 0

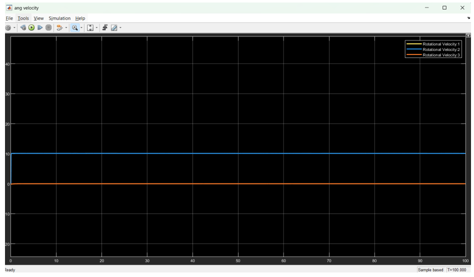

Figure 15. Rotational velocities (roll, pitch, yaw rates) for nonzero attitude rate commands.

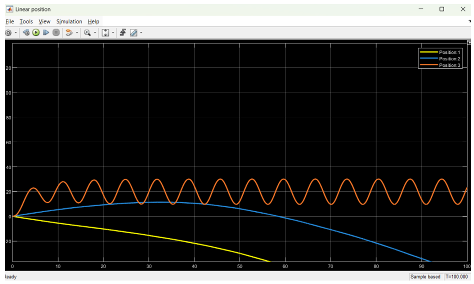

Figure 16. Altitude response with roll and pitch rates set to 10 and altitude input of 30.

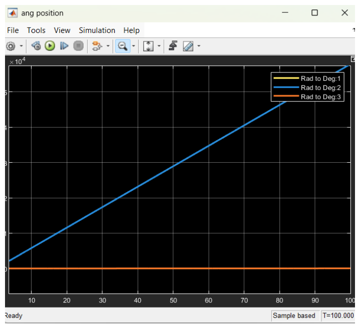

Figure 17. Angular positions (roll, pitch, yaw) over time for nonzero attitude rate commands.

The Figure 15 and Figure 17 show that the rotational velocities and angular positions are held steady at zero for yaw, while the roll and pitch rates are commanded as expected. This means the attitude rate controller is working correctly for these commands, and the drone is able to follow the desired roll and pitch rates without any instability.

The Figure 15 shows the altitude. The drone's altitude oscillates around the setpoint of 20. The oscillation means the altitude PID is still not perfectly tuned, but it is able to keep the drone close to the target value most of the time.

Overall, the controller is able to track the commanded roll and pitch rates of 10, and the altitude controller keeps the drone near the setpoint, though with some oscillation.

This shows the control system is stable for symmetric, high attitude rate commands.

Case 2: Roll Rate = 2, Pitch Rate = 10 (Simulation Crash)

When I tried asymmetric commands (roll rate = 2, pitch rate = 10), the simulation crashed and gave a solver error about non-finite derivatives and minimum step size violation.

Possible Cause:

- **Mixer Saturation and Physical Limits:** In a quadrotor, the four motors must share the work of generating the desired roll, pitch, and yaw rates while also maintaining enough total thrust to keep the drone in the air. When the roll and pitch rate commands are not balanced, the control allocation (or mixer) might try to assign negative thrust to some motors or exceed their maximum possible thrust. This is physically impossible, and can lead to numerical instability or singularities in the simulation. A saturation block needs to be added avoid this.
- **Numerical Instability:** When the mixer tries to solve for impossible motor commands, the state derivatives can become undefined (infinite or NaN), which causes the Simulink solver to fail and stop the simulation.

The PID controllers work well for symmetric, high roll and pitch rate commands, keeping the drone stable and tracking the desired rates. The altitude controller still shows some oscillation. When the attitude rate commands are unbalanced, the simulation becomes unstable and crashes, most likely because the mixer or controller tries to command motor thrusts that are not physically possible.

6.3. Altitude Control

For this test, I set the altitude PID gains to P = 1, I = 1, D = 2.5, N = 5. For the attitude rate controllers, the gains were P = 10, I = 5, D = 0.1, N = 20 for roll and pitch, and P = 5, I = 0.5, D = 0.1, N = 20 for yaw rate.

The roll and pitch rates were 70 and 30, with yaw rate at 0, and set the altitude input to 10.

Altitude

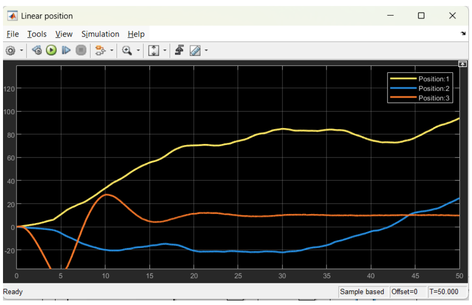

Figure 18. Altitude response. The altitude tracks the setpoint and remains stable.

The altitude plot (Figure 18) shows that the drone holds the altitude setpoint of 10 very well with some oscillations in beginning. There is no noticeable oscillation or drift after the altitude stabilizes.

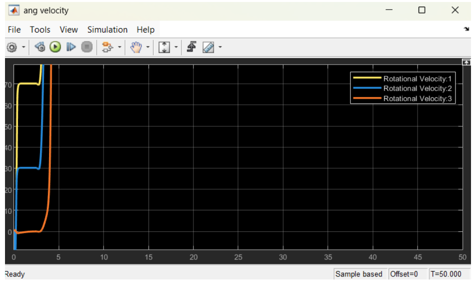

Figure 19. Angular velocities (roll, pitch, yaw rates) over time. The rates are initially stable but spike later.

Roll, Pitch, and Yaw Rate Control

The angular velocity plot (Figure 19) shows that the roll and pitch and yaw rates are initially stable at their commanded values, but they spike suddenly and become unstable.

Summary Table

Controller	Performance
Altitude	Tracks setpoint well
Attitude Rate (Roll/Pitch)	Initially tracks input, but rates spike and become unstable

Table 2. Summary of observed performance.

The altitude controller is able to hold the setpoint stably, even when large roll and pitch rate commands are given. However, the roll and pitch rate controllers become unstable after a short time, with the rates spiking suddenly. To improve stability, it may help to limit the maximum rate commands, or further tune the gains for input scenarios.

References

References

[1] APC Propellers, "Performance Data," 2025. [Online]. Available: https://www.apcprop.com/technical-information/performance-data/ [Accessed: May 2, 2025].

[2] E. Wertheimer, "Lecture by Eliott Wertheimer," 2025. [Online]. Available: https://drive.google.com/file/d/1-QdK_z4iUL6aR-xDvku-iKhh7Y9cVCSx/view?usp=drive_link [Accessed: May 2, 2025].

[3] K. C. Craig, "Quadrotor Equations of Motion and Control KCC Final 4 2023 Video," YouTube, Apr. 16, 2023. [Online]. Available: https://www.youtube.com/watch?v=xCoFaTyn5dg [Accessed: May 2, 2025].