3.1 | EENG307

Arael Anaya | Owen Black | Daniel Kim | Kyle Ha

Problem Definition and Goals

Unmanned aerial vehicles (UAV's) have been heavily researched and developed in hopes of achieving accurate aerial movements to be implemented by different robots. Achieving this level of movement could help revolutionize package delivery, prevent the death/injury of workers who work in places with high height, automate the agriculture industry, and unlock new styles of photography [1]-[2]. Our team's goal is to control the height of the UAV by controlling the force of the rotors, accounting for gravity and linear drag. The height of the UAV is a useful metric because it allows for stable control of the drone in 3D space.

Metrics

For the quadcopter altitude control system, we will test performance by tracking both a step change in height and a smooth sinusoidal height change. Essentially, $r_1(t) = Au(t)$ will be a step force input and $r_2(t) = f(t)$ where f(t) will be a time-varying force input. We will measure how quickly and smoothly the quadcopter responds, focusing on its overshoot, and the steady-state error. In addition, we will check how much motor output is required compared to the maximum (control effort) and how often the motors hit their limits.

References

[1] L. Mirto, "What's The Future of Drone Technology?," Americas, https://www.zuken.com/us/blog/whats-the-future-of-drone-technology/ (accessed Sep. 10, 2025).

[2] R. Bishoff, "10 future uses for Drones," COLLEGE OF ENGINEERING, https://engineering.osu.edu/news/2024/02/10-future-uses-drones (accessed Sep. 10, 2025).

3.2.1: Transfer Function Derivation

In our system, we are modeling the force of the rotors (f_r), the drag force from air resistance ($f_{drag} = b\dot{h}$), and the disturbance force of gravity (f_g). Using Newtons 2nd Law, the systems motion is described by:

$$m\ddot{h} = f_r(t) - b(\dot{h}) - f_g(t)$$

Next, we take the Laplace transform of the function. Solving this equation out, we derive that the transfer function $G(s)$ can be defined by:

$$H(s) = \frac{F_r(s)}{s^2m+sb} - \frac{F_g(s)}{s^2m+sb} \text{ , Where } G(s) = \frac{1}{s^2m+bs}$$

3.2.2: Paramater Estimates

The two transfer function parameters are the mass of the drone and the damping constant from air resistance. For consistency, we used the specifications of one of the most common commercial UAVs, the DJI Mavic 3 [2]. From [1], we are able to derive the following function for the dampening coefficient:

$$b = \beta D = 1.6 \times 10^{-4} \text{ N·s/m}^2 \cdot 347.5 \cdot 10^{-3} m = 5.56 \times 10^{-5} \text{ N·s/m}$$

Therefore, $m = 0.958$ kg and $b = 5.56 \times 10^{-5}$ N·s/m

3.2.3: Simulink Analysis

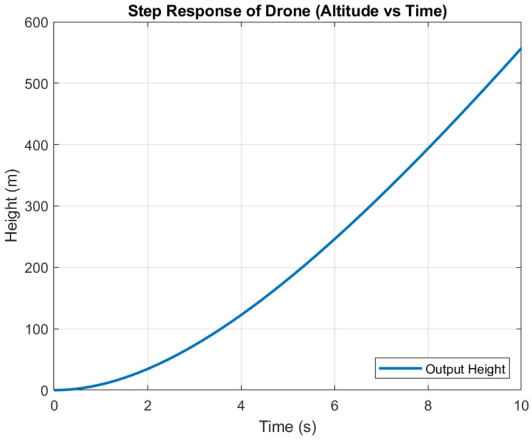


Figure 1: The rotor force F_r was modeled as a step input with a magnitude of $3mg$, while $F_g = mg$ acted as a constant disturbance. The resulting step response shows a continuously increasing height, matching the reference response from Section 8. Minor differences in slope are due to the specific parameter values of F_r, m, and b, which influence the rate of ascent.

[1] Classical Mechanics. Annai Academy Online School, Mar. 2025. [Online]. Available: https://annaiacademyonlineschool.com/wp-content/uploads/2025/03/classical-mechanics.pdf [Accessed: Oct. 8, 2025].

[2] “Mavic 3 Pro — Specifications,” DJI, 2025. [Online]. Available: https://www.dji.com/mavic-3-pro/specs [Accessed: Oct. 8, 2025].

3.3 Open Loop Plant Analysis

1. BIBO Stability

The transfer function of the open-loop plant is given by:

$$G(s) = \frac{1}{ms^2 + bs} = \frac{1}{s(ms + b)}$$

where $m = 0.958$ kg and $b = 5.56 \times 10^{-5}$ N·s/m

The poles of $G(s)$ are 0 and $-\frac{b}{m}$ (5.8×10^{-5}). Since there is a pole at 0, the system is not BIBO stable.

2. Simulation Behavior

The step response shown in Figure 1 does not settle to a steady-state value. Instead, the output continuously increases over time, which is consistent with the presence of a pole at the origin in the transfer function (type-1 plant), causing the step input to produce a ramp-like output. The second pole at $s = -\frac{b}{m}$ is very close to the origin, resulting in a slow transient before the output becomes nearly linear. Overall, the simulated behavior aligns with theoretical expectations based on the system's poles.

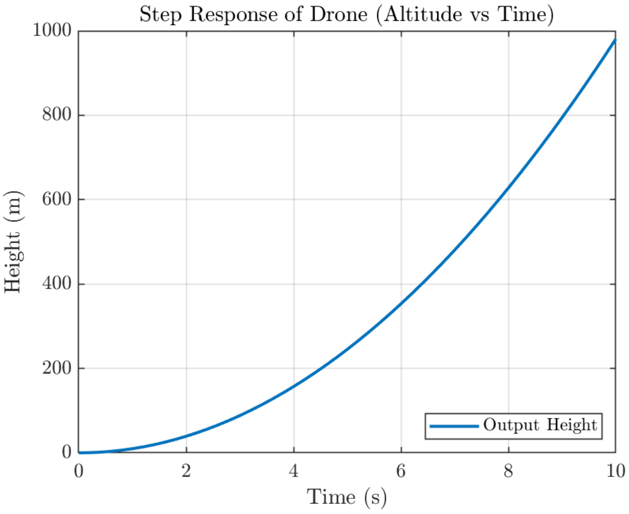


Figure 1: Step response of the open-loop plant $G(s)$.

3. Disturbance and Modeling

Yes, the plant will likely face external disturbances. For a drone's vertical motion, these include air turbulence, rotor thrust changes, or sudden payload shifts. Such disturbances alter the net force on the drone, affecting its acceleration and height. The configuration in Figure 2 models this accurately: disturbances enter the plant input, subtract from the control signal, and pass through G(s), effectively representing how external forces influence the dynamics of the real-world system.

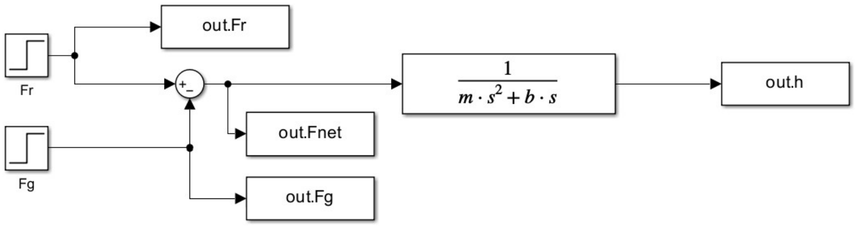


Figure 2: Simulink Block Diagram of Plant Function $G(s)$.

4. Additional Questions

- How sensitive is the system to uncertainty in mass (m) or the drag coefficient (b)?
- What is the steady-state error for standard inputs (step, ramp)?
- Which controller type will be most optimal for the system given the existing poles?

References:

- Kathryn Johnson, *EENG307: Semester Project*, Colorado School of Mines, Fall 2025.

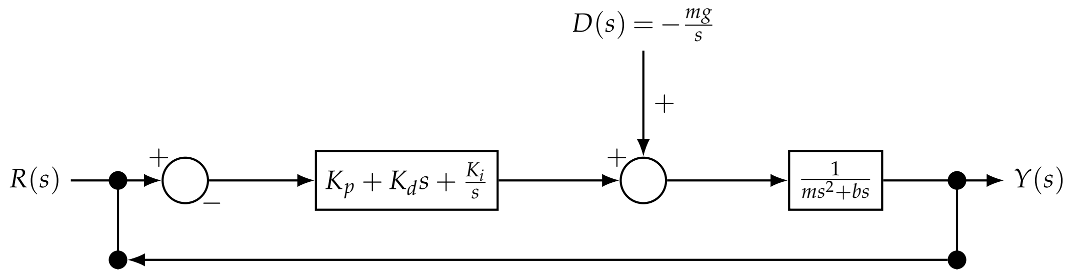
1. Reference Input

Our reference input is modeled as a unit step function scaled by a factor A, $r(t) = Au(t)$, where A corresponds to the desired hover height in meters. A step input is appropriate because our goal is for the drone to reach and maintain a constant altitude. This type of reference represents a sudden change in the altitude and is standard for evaluating performance of height controllers.

2. Performance Specifications

- Rise Time: For the transient responses, we expect the operator to fly between 0-100m. The DJI mavic 3 [1] has a vertical speed of 8 m/s, therefore, our rise time should be around 12.5 s.
- Steady State Error: We would like the steady state error 0 so that the the operators will be able to get the desired angles from the drone.
- Percent Overshoot: We would want the percent overshoot to not be higher than 1 percent this is so that the drone does not hit any ceilings or obstacles when being set to the correct area.

3.Block Diagram



4.Finding best controllers

We have decided that a PID Controller would help our system reach the desired specification of a quicker rise time, lower percent overshoot, and zero steady state error. The PI controller will cause the transfer function to be a type 1 function, and when our system takes a unit step, the steady state error is 0. The PD controller will shorten the rise time and percent overshoot. More details are shown in the block diagram.

5. Controller Transfer Function and Poles

$$C(s) = K_p + \frac{K_i}{s} + K_d s,$$

$$G(s) = \frac{1}{ms^2 + bs},$$

the closed-loop transfer function is

$$\frac{Y(s)}{R(s)} = \frac{K_d s^2 + K_p s + K_i}{ms^3 + (b + K_d)s^2 + K_p s + K_i}.$$

Using the controller designer and root locus analysis, the tuned gains were $K_p = 6$, $K_i = 2$, and $K_d = 4$. Substituting $m = 0.958$ and $b = 5.56 \times 10^{-5}$ gives the characteristic equation:

$$0.958s^3 + 4.00006s^2 + 6s + 2 = 0.$$

The resulting poles are located at $s = -1.86 \pm 1.05j$ and $s = -0.458$, indicating a stable response. The system is confirmed to be stable since all poles lie on the left-hand side of the imaginary axis. Overall, this configuration meets the required performance criteria for a controlled and well-behaved altitude response.

References:

[1] DJI, "Mavic 3 Pro - Specs." [Online]. Available: https://www.dji.com/mavic-3-pro/specs.

Project Part 3.5: Verification and Iteration

We are designing a controller for a drone to achieve stable altitude regulation by minimizing rise time, percent overshoot, and steady-state error in the height response.

Our plant is modeled as the transfer function

$$G(s) = \frac{1}{ms^2 + bs},$$

where $R(s)$ is the desired height, $Y(s)$ is the output height, $E(s) = R(s) - Y(s)$ is the error, $C(s)$ generates the control force $F_r(s)$, $D(s) = -\frac{mg}{s}$ represents gravity, and $G(s) = \frac{1}{ms^2+bs}$ maps the net force to the mass position. Our closed-loop system is modeled in Simulink as shown in Figure 1, where $R(s)$ is the reference input, $Y(s)$ is the output height, $E(s)$ is the error signal at the first summing junction, $F_r(s)$ is the controller-generated force, and $D(s)$ represents the gravity disturbance added at the second summing junction.

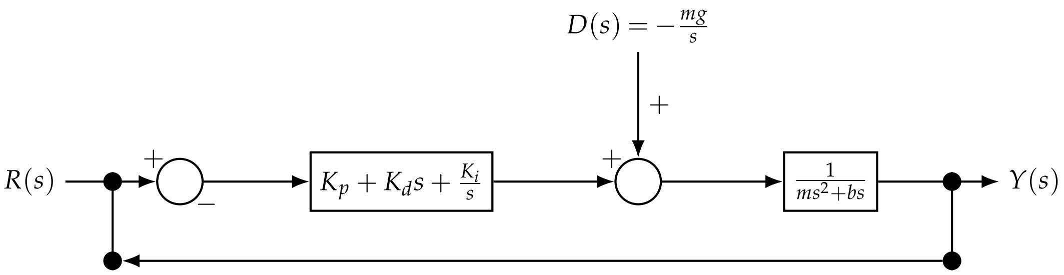


Figure 1: Block diagram of the PID-controlled mass–damper system with gravity disturbance input $D(s) = -\frac{mg}{s}$.

Our controller C(s) is

$$C(s) = K_p + \frac{K_i}{s} + K_d s$$

where K_p is the proportional gain, K_i is the integral gain, and K_d is the derivative gain. The final values for all 3 of these constants were 6, 2, and 4 respectively. This controller was developed by using MATLAB's Control System Designer by examining the open-loop transfer function, rise-time, and percent-overshoot targets on the root-locus plot, then adjusting pole and zero locations until the closed-loop step response met the required performance.

The results from our controller design are shown in Figure 2 - 4.

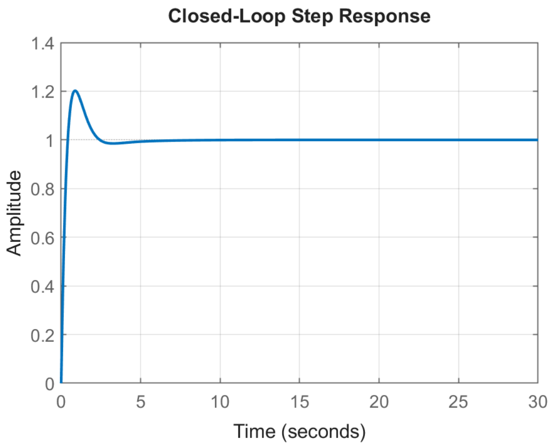


Figure 2: Closed-loop step response of the system.

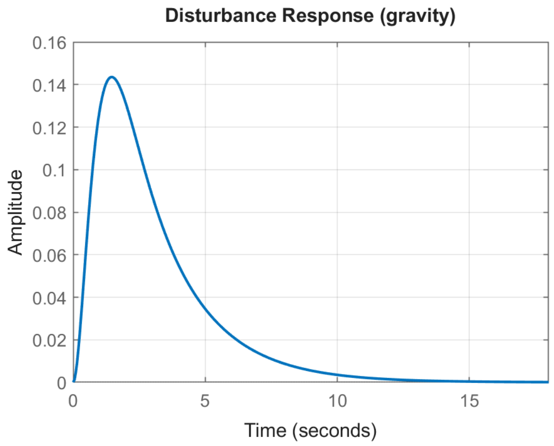


Figure 3: Response of the system to a gravity disturbance input.

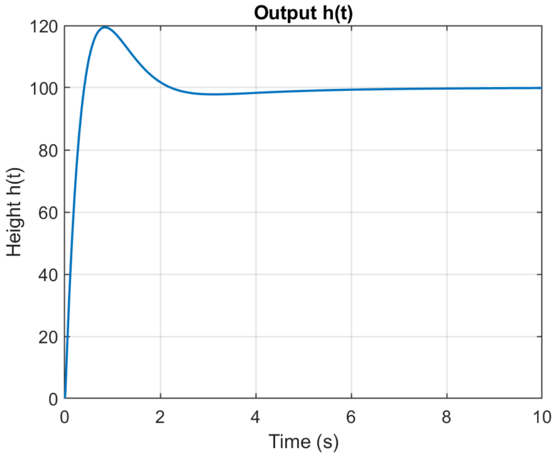


Figure 4: Output height $h(t)$ showing tracking of the reference signal.

The primary results are shown in Figures 2–4, which include the closed-loop step response, the disturbance rejection response, and the output height trajectory.

Figure 2 shows that the closed-loop step response meets the rise-time and overshoot specifications, settling quickly with minimal oscillation, indicating that the controller provides adequate damping and stability.

Figure 3 shows that the system effectively rejects the gravity disturbance, with the output force decaying smoothly to zero, demonstrating strong disturbance attenuation and proper integral action.

Figure 4 shows that the output height $h(t)$ tracks the reference without steady-state error and with well-behaved transient dynamics, confirming that the controller maintains accurate position control under the modeled conditions.

The closed-loop system is BIBO stable with poles at $s = -1.86 \pm 1.05j$ and $s = -0.458,$.