Control Systems EGE 416-01

Dr. Julio Gonzalez

May 12, 2025

Name	Major
John Urban Quezada	EE
Simon Maranga	EE

Introduction

An engineering firm (Professor Gonzalez) has asked us to design a controller that regulates the altitude of a drone. For simplicity, the effects of pitch, yaw, and horizontal displacement will be ignored.

Before any analysis could be done, the variables that affect the drone's performance must be considered.

- **y(t)** denotes the drone's vertical displacement. The reference altitude of the drone will be denoted as y(t) = 0 (i.e the drone is hovering at constant altitude when y = 0)
- **F(t)** denotes the total lift thrust force exerted by the propellers, **W** denotes the weight of the drone, $\mathbf{b_F}$ denotes constant friction, **M** denotes the mass of the drone, and **g** denotes the gravitational constant of Earth. Together, these 4 variables are related by the equation:

$$(1) \quad M \frac{d^2 y(t)}{dt^2} = F(t) - b_F \frac{dy(t)}{dt} - Mg$$

- The propellor motor torque is denoted by $\boldsymbol{\tau_p(t)}$ and it proportional to its current **i(t)**. The relationship can be seen in the following equation:

$$(2) \quad \tau_p(t) = K_\tau i(t)$$

- The propeller's moment of inertia is denoted by $\mathbf{J_p(t)}$ and it is subject to a friction force. The moment of inertia can be described with the following equation, where $\mathbf{w_p(t)}$ is the angular velocity of the propellers and $\mathbf{b_p}$ is the propellers' friction coefficient:

$$(3) \quad J_p(t) \frac{dw(t)}{dt} = \tau_p(t) - b_p w_p(t)$$

- Assuming that the total lift thrust force is proportional to the propellers', the following equation results:

$$(4) \quad F(t) = K_F w_p(t)$$

The values of all of these constants are shown in Table 1 below.

Name	Value	Unit	Name	Value	Unit	Name	Value	Unit
M	0.5	Kg	K_τ	2	$\mathrm{N\,m\,A^{-1}}$	K_F	1×10^{-2}	N s
b_F	5	$\mathrm{Nm^{-1}s}$	J_p	2×10^{-3}	$\mathrm{Kg\,m^2}$			
g	9.8	$\mathrm{m\,s^{-2}}$	b_p	4×10^{-3}	N m s			

Table 1: Values for constants used in equations 1-4

Task 1

In order to simplify calculations for the remaining portions of the analysis, offset values for the current, propeller torque, propeller angular velocity, and thrust force were calculated in order to cancel the effects of gravitational acceleration. In other words, any value calculated for these variables must have the offset values added to them. Using algebra, these values were calculated as follows:

$$\mathrm{Mg} = (9.8\ \tfrac{m}{s^2})(0.5\ \mathrm{Kg}) = 4.9\ \mathrm{N}$$

$$0 = F(t) - (4.9\ \mathrm{N}) \Rightarrow \mathbf{F_0(t) = 4.9\ N\ (Offset\ lift\ thrust\ force)}$$

$$4.9\ \mathrm{N} = (1 \times 10^{-2}\ \mathrm{Ns})(w_p(t)) \Rightarrow \mathbf{w_{p0}(t) = 490}\ \tfrac{rad}{sec}\ \mathbf{(Offset\ angular\ velocity)}$$

$$0 = \tau_p(t) - (4 \times 10^{-3}\ \mathrm{Nms})(490\ \tfrac{rad}{sec}) \Rightarrow \boldsymbol{\tau_{p0}(t)} \mathbf{= 1.96\ Nm\ (Offset\ propeller\ torque)}$$

$$1.96 = (2\ \mathrm{NmA^{-1}})(i(t)) \Rightarrow \mathbf{i_o(t) = 0.98\ A\ (Offset\ current)}$$

With these calculations in mind, equation 1 can be rewritten as:

$$(5)\quad M\frac{d^2y(t)}{dt^2} = F(t) - b_F\frac{dy(t)}{dt}$$

Task 2

In order to find the plant transfer function (TF) of the drone's control system, equations 2-5 will be taken from the time domain into the s domain. Then, algebra will be used to rewrite the vertical displacement in terms of the input current.

$$(2s)\quad \boldsymbol{\tau}_p(t) = K_\tau i(t) \Rightarrow \boldsymbol{\tau}_p(s) = K_\tau I(s)$$

$$(3s)\quad J_p(t)\frac{dw(t)}{dt} = \boldsymbol{\tau}_p(t) - b_p w_p(t) \Rightarrow (J_P(s))(sW_p(s)) = \boldsymbol{\tau}_p(s) - b_p W_p(s)$$

$$(4s)\quad F(t) = K_F w_p(t) \Rightarrow F(s) = K_F W_p(s)$$

$$(5s)\quad M\frac{d^2y(t)}{dt^2} = F(t) - b_F\frac{dy(t)}{dt} \Rightarrow M(s^2Y(s)) = F(s) - b_f(sY(s))$$

$$M(s^2Y(s)) = F(s) - b_f(sY(s)) \Rightarrow F(s) = M(s^2Y(s)) + b_f(sY(s))$$

$$M(s^2Y(s)) + b_f(sY(s)) = K_F W_p(s)$$

$$W_p(s) = \frac{Y(s)(Ms^2 + sbf)}{Kf}$$

$$J_p(s)\left(\frac{Y(s)(Ms^2 + sbf)}{Kf}\right) = \boldsymbol{\tau}_p(s) - b_p\left(\frac{Y(s)(Ms^2 + sbf)}{Kf}\right)$$

$$\boldsymbol{\tau}_p(s) = (b_p + J_p s)\left(\frac{Y(s)(Ms^2 + sbf)}{Kf}\right)$$

$$I(s) = \left(\frac{1}{K\tau}\right)\left((b_p + J_p s)\left(\frac{Y(s)(Ms^2 + sbf)}{Kf}\right)\right)$$

$$\frac{Y(s)}{I(s)} = \frac{K\tau KF}{(bp + Jps)(Ms^2 + sbf)} = \frac{0.02}{((4\ x\ 10^{-3}) + (2\ x\ 10^{-3})s)(0.5s^2 + 5s)}$$

Denominator = D(s) = $0.002s^2 + 0.02s + 0.001s^3 + 0.01s^2 = 0.001s^3 + 0.012s^2 + 0.02s$
$= 0.001(s^3 + 12s^2 + 20s)$

$$\frac{Y(s)}{I(s)} = \mathbf{P(s)} = \frac{20}{s(s+10)(s+2)}$$

Task 3

After determining the plant transfer function, we designed a controller targeting an overshoot of approximately 4.3% and a settling time of about 2.5 seconds. In practical terms, this means that when the drone ascends to a specified altitude, it may momentarily exceed the target by 4.3% before stabilizing. The drone will then settle at the desired altitude within roughly 2.5 seconds. To find the roots of the controller, we performed the following calculations:

$$\text{Transient} = \frac{5}{\alpha} = 2.5 \text{ seconds} \Rightarrow \alpha = 2$$

$$\text{Overshoot (OS)} = e^{\frac{-\alpha\pi}{\beta}} = 0.043 = e^{\frac{-2\pi}{\beta}}$$

$$\ln(0.043) = \frac{-2\pi}{\beta} \Rightarrow \beta \cong 2$$

$$\textbf{Roots} = -\alpha \pm j\beta = -2 \pm j2$$

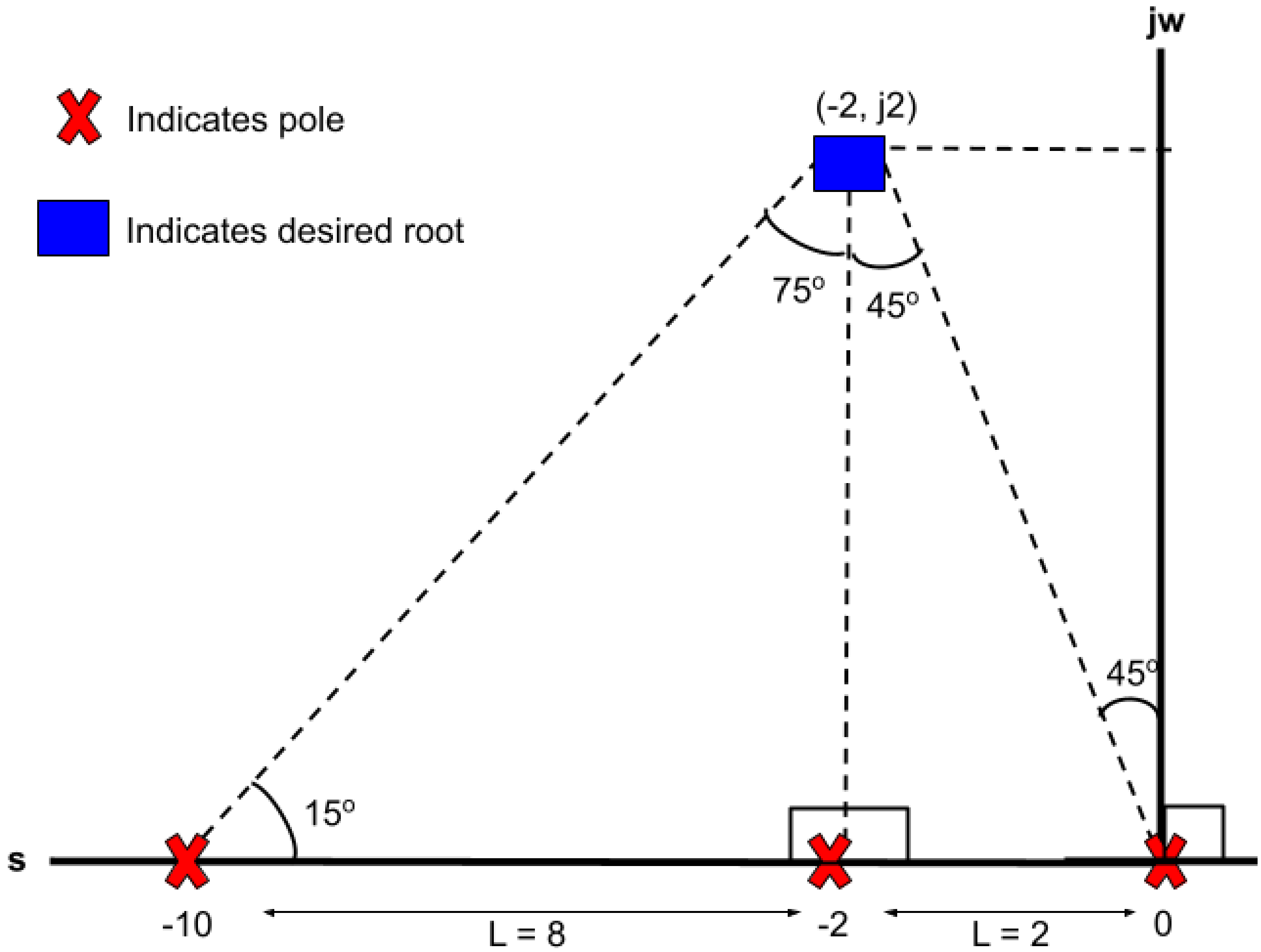

Figure 1: Angle relationship between the plant poles and the desired controller roots

As can be seen in Figure 1, the angle condition for the current system is not met because the sum of the three angles with respect to the desired roots is:

$$135^{o} + 90^{o} + 14^{o} = 180^{o} + 59^{o}$$

The sum of the angles from the existing plant poles to the desired closed-loop pole location is 59° away from satisfying the angle condition. Therefore, a controller is needed to shift the poles in such a way as to satisfy the angle condition. For this scenario, a phase lead controller of form $C(s) = K_c\frac{aTs+1}{Ts+1}$ will be used.

In order to design the phase lead controller, the pole-zero cancellation method was equipped over the bisector method because it allows for a more stable system with regards to the overshoot. The pole we specifically chose to cancel was the one at s = -2. This is because the pole at s = -10 only contributes 15º, so an angle of 59º cannot be removed. The pole at s = 0 was not removed because it is at the intersection between the left half and the right half of the s plane, which means a slight miscalculation could lead to massive instability. Using simple trigonometry, we chose to introduce the new pole at s = -5.328. The calculations as well as Figure 2 can be seen below.

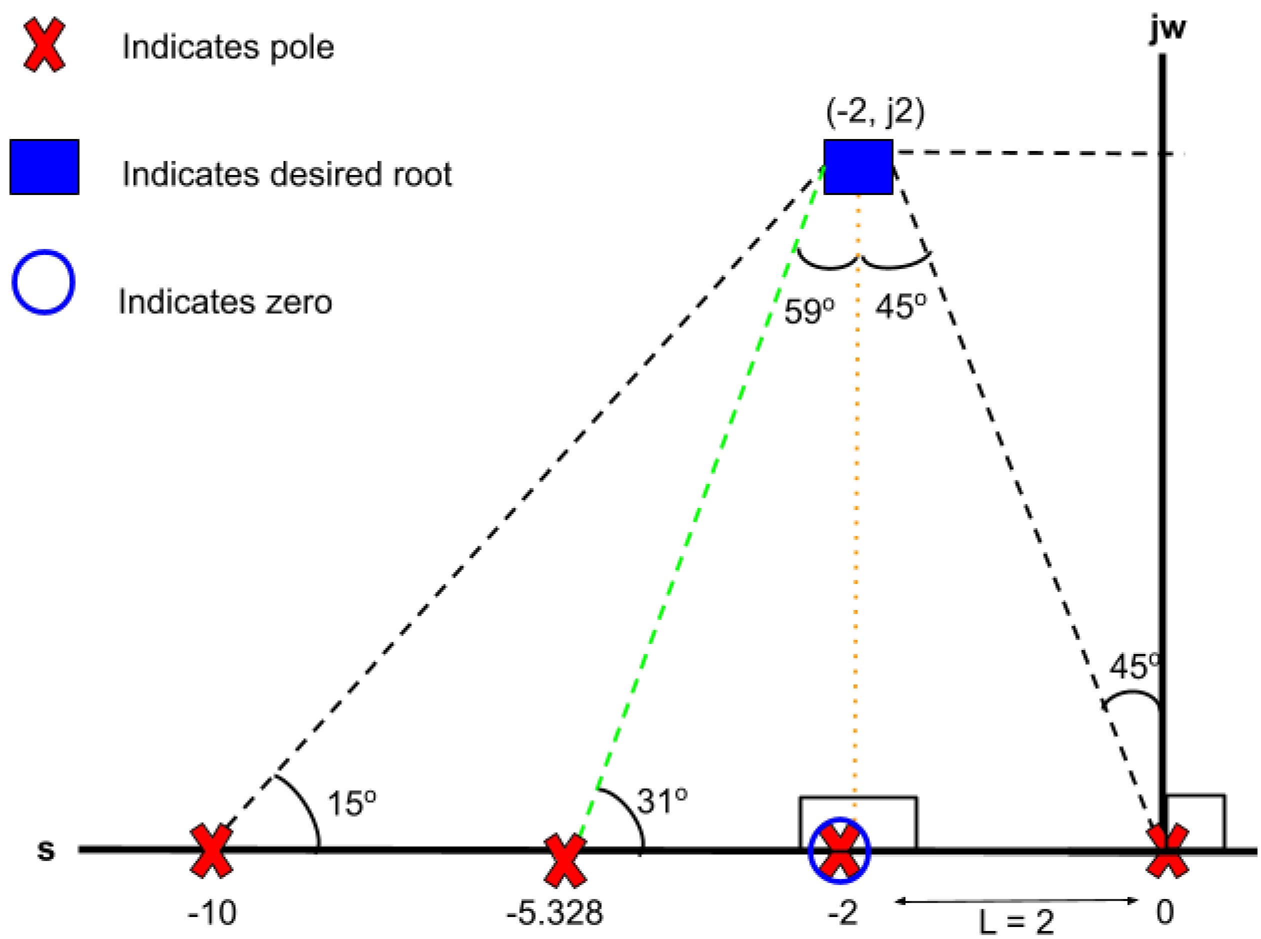

Figure 2: Angle relationship between new poles and included zero

- $90^{\circ} - 59^{\circ} = 31^{\circ}$ => Need a pole who has an angle of 31° with respect to the desired root

- If a line is drawn from the desired root to the horizontal point in such a way that it makes a 31° angle, then it will be obvious that it makes a right triangle with the orange line. It can immediately be seen that the length of the orange line is 2, therefore we can solve for the length of the triangle's base

$$\tan(31^{\circ}) = \frac{2}{x} \Rightarrow x = 3.328$$

$$\text{New pole} = -1*(3.328 + 2) = -5.328$$

If the "skeleton" of the controller gain equation is rewritten, then:

$$C(s) = K_c \frac{aTs + 1}{Ts + 1} = aK_c \frac{s + (1/aT)}{s + (1/T)}$$

It becomes apparent from Figure 2 that $\frac{1}{aT}$ must equal 2 and $\frac{1}{T}$ must equal 5.328. Substitution of known values gives:

$$\frac{1}{T} = 5.328 \Rightarrow \mathbf{T = 0.1876}$$

$$\frac{1}{a(0.1876)} = 2 \Rightarrow \mathbf{a = 2.665}$$

Using the characteristic equation for stability, the controller gain can be found:

$$G(s) = C(s)P(s) = -1 \Rightarrow \left(\frac{20}{s(s+10)(s+2)}\right)(2.665*K_c)\left(\frac{s + (1/aT)}{s + (1/T)}\right) = -1$$

Setting $K_E = 20*2.665*K_c$ yields:

$$K_E\left(\frac{20}{s(s+10)(s+2)}\right)\left(\frac{s + (1/aT)}{s + (1/T)}\right) = -1$$

Using the magnitude condition of all closed loop control systems, and the pythagorean theorem, K_E turns out to be:

$$K_E = (\sqrt{2^2 + 2^2})*(\sqrt{3.328^2 + 2^2})*(\sqrt{8^2 + 2^2}) = 90.56$$

Solving for K_c yields:

$$90.56 = 20*2.665*K_c \Rightarrow \mathbf{K_c = 1.7}$$

This means that in order To meet the overshoot and transient requirements while ensuring stability, the controller should be designed with a gain of 1.7.

In order to verify our design for the controller, we generated a root locus in MATLAB, which can be seen in Figure 3 below:

```
>> T = 0.1876;
a = 2.665;
s = tf('s');
Gp = 20/(s*(s+10)*(s+2));
Gc = (1.7*a)*((s + (1/(a*T)))/(s+(1/T)));
G = Gp*Gc;

s_d = -2 + (1*i*2);
s_di = -2 - (1*2*i);

rlocus(G);

hold on;

plot(real(s_d), imag(s_d), 'r*', 'MarkerSize', 3, 'LineWidth', 2);
plot(real(s_di), imag(s_di), 'r*', 'MarkerSize', 3, 'LineWidth', 2);

hold off;
>>
```

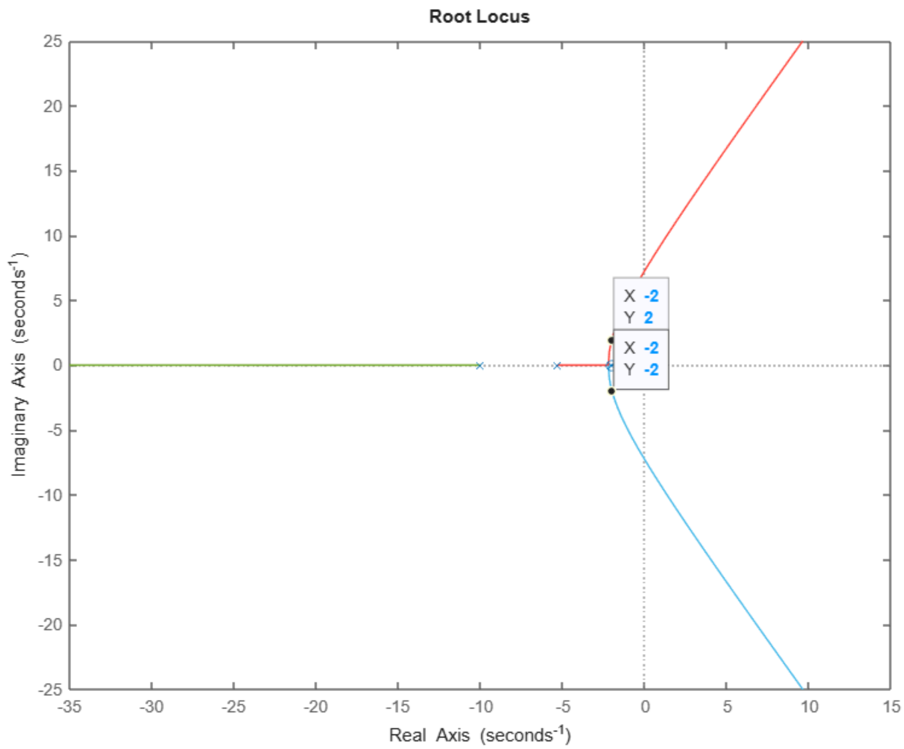

Figure 3: Root locus of the system with the designed controller

This root locus confirms that the pole-zero cancellation we performed was effective in modifying the system dynamics. By strategically placing a zero to cancel one of the system's poles, we simplified the open-loop transfer function, which in turn altered the shape of the root locus. As a result, the root locus now passes through the desired closed-loop pole locations, indicating that those roots are achievable for a suitable range of gain values.

Task 4

To verify that a gain of 1.7 gives an overshoot of 4.3% and a transient of 2.5 seconds, a simulation was run in MATLAB to see how the system would react when a step function was applied. The results can be seen in Figures 4a and 4b below:

```
>> T = 0.1876;
a = 2.665;

s = tf('s');
Gp = 20/(s*(s+10)*(s+2));
Gc = (1.7*a)*((s + (1/(a*T)))/(s+(1/T)));
G = Gp*Gc;

%s_d = -2+(1*i*2);
%s_di = -2-(1*i*2);

sys_clos = feedback(G,1);

figure;
step(sys_clos,5);

grid on;
>>
```

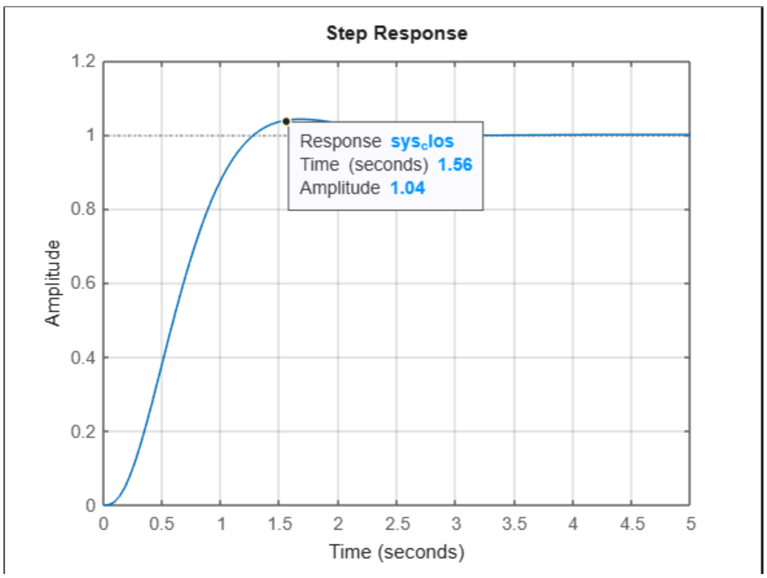

Figure 4a: Overshoot of the system with a gain of 1.7

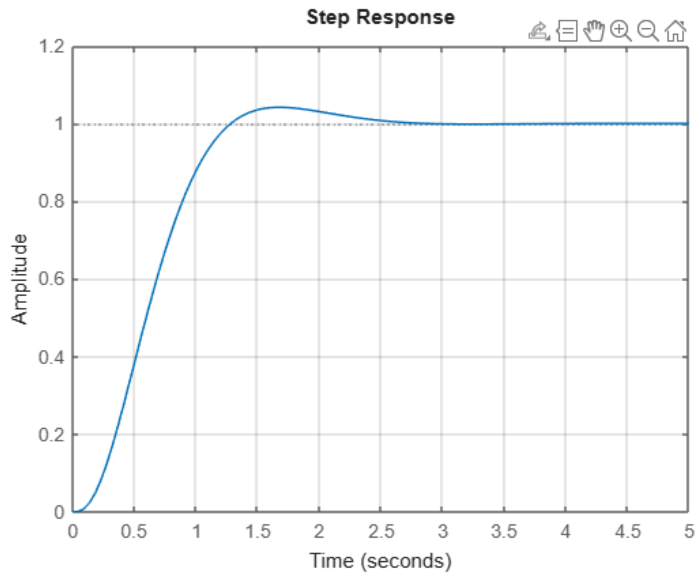

Figure 4b: Transient of the system with a gain of 1.7

As can be seen, the overshoot is slightly lower than 4.3% (about 4%), but that is acceptable. In addition, the transient is nearly a perfect 2.5 seconds. Therefore, the system can be deemed as successful. This means that if a drone were to change its altitude, it would initially miss the desired height by 4%, but regulate itself down in roughly 2.5 seconds.

Conclusion

In conclusion, a phase lead controller for a drone was successfully designed using pole zero cancellation method. The system has a gain of 1.7, a transient of 2.5 seconds, and an overshoot of 4%.