School of Science and Engineering
Division of Engineering Programs

Julio J. González

New Paltz
STATE UNIVERSITY OF NEW YORK

EGE416 Control Theory - Spring 2021

COURSE DESIGN PROJECT

Due Date: Thursday, May 6, 2021

Your Names: Christopher Maggio and Caroline Kucher

Rules for submitting design project

- Groups of two students maximum
- 1 submission per group.
- Create a PDF file with **<u>this cover sheet</u>** and your work
- Upload your file to Blackboard → Course Content → Project → Design Project before:
- Deadline: 5 PM on Thursday, May 6, 2021.

Abstract: This project deals with the design of a controller to control the altitude of a quadrotor drone. Design specifications are in terms of a desired altitude step response.

Table of Contents

Introduction

The objective of this project is to control the altitude of a quadrotor drone using a closed-loop phase-lead controller and to observe the output using a step function as the input.

In this control system, the input is a DC current and the visible feedback information is the drone's altitude. The drone is powered by four propellers, each driven by a DC motor. The assumption is made that all propellers are constantly synchronized. The flight dynamics is shown in Figure 1 below, where y represents the vertical displacement from a reference altitude. F(t) represents the total lift force exerted by the four propellers, W represents the total weight of the drone, and $b_F \cdot y'$represents the frictional force the drone experiences as it moves vertically through air.

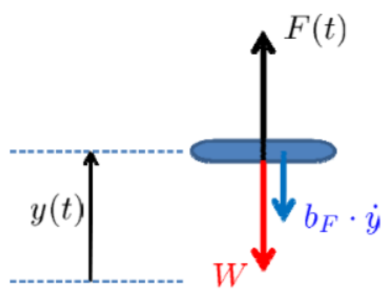


Figure 1. Forces Acting on Drone Dynamics. [1]

This is modeled by Equation 1,

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

where M is the mass of the drone, $b_F$is the constant friction coefficient, and g is the gravitational constant.

The following image, Figure 2, illustrates the propulsion dynamics of the propeller.

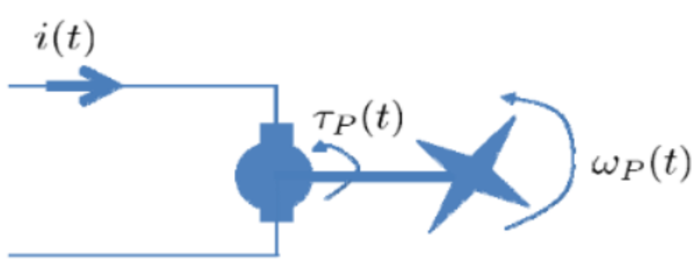


Figure 2. Propulsion Model. [1]

The motor torque, $\tau_p(t)$, is proportional to the current, $i(t)$, as described in Equation 2.

$$\tau_p(t) = K_T * i(t) \quad (2)$$

The propeller has a moment of inertia, J_p, and is affected by the frictional force. Equation 3 describes how the angular velocity, $\omega_p(t)$, can be mathematically modeled,

$$J_p \frac{d\omega_p(t)}{dt} = \tau_P(t) - b_p \omega_p(t) \quad (3)$$

where b_p is the friction coefficient of the propeller. The assumption is made that the total lift thrust force is proportional to the angular velocity of the propellers, as described in Equation 4.

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

Attached below in Table 1 are all the required constant values to design the phase-lead controller.

TABLE 1
VALUE OF CONSTANTS USED [1]

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			

Theory

A root locus is a plot of the location of the root of the closed-loop transfer function for increasing values of the controller parameter, K. These roots define the damping factors and oscillation frequency of the closed-loop system. In other words, the controller parameter, K, controls the damping factors and oscillation frequency of the system. [2]

To plot a root locus, the characteristic equation of the closed-loop transfer function needs to be in evan's format which is defined in Equation 5,

$$K \frac{\sum_{i=1}^{m} (s - z_i)}{\sum_{i=1}^{n} (s - p_i)} = -1 \quad (5)$$

for $n > m$. Equation 5 splits into two conditions that must be met in order for the value of s to be a root. These two conditions are called the magnitude condition and the angle condition. They are defined as follows in Equation 6 and Equation 7, respectively,

$$\frac{\sum_{i=1}^{m}|s-z_i|}{\sum_{i=1}^{n}|s-p_i|} = K \quad (6)$$

$$\sum_{i=1}^{n} < (s - p_i) - \sum_{i=1}^{m} < (s - z_i) = \pi \quad (7)$$

Any value of s that satisfies these two conditions is a root of the closed loop transfer function. [3]

The root locus begins at a pole and ends at a zero and is located on the left side of an odd number of singularities on the s-plane, where singularities are defined as poles and zeros. Some poles will find a zero at an asymptote. There are n-m asymptotes, where n is the number of poles and m is the number of zeros. The asymptotes on the root locus are defined by θ, the angle from the real axis on the s-plane, and the centroid, σ_c, which is where the asymptote intersects the real axis.θ and σ_c are defined in Equation 8 and Equation 9, respectively.

$$\theta = \frac{\pi(2l+1)}{(n-m)},\ l = 0, 1, 2... \quad (8)$$

$$\sigma_c = \frac{\sum_{i=1}^{n} p_i - \sum_{i=1}^{m} z_i}{n-m} \quad (9)\ [3]$$

A block diagram of a phase-lead controller is shown in the following image; where R(s) is the reference or desired output, E(s) is the error, C(s) is the controller, U(s) is the input, P(s) is the plant, Y(s) is the output.

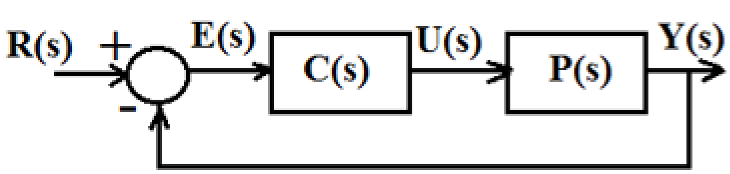


Figure 3. Block Diagram of a Phase-Lead Controller. [4]

Equation 10 defines the controller as the following:

$$C(s) = K_c \frac{aTs+1}{Ts+1} \quad (10)$$

The controller in evan's format is described in Equation 11:

$$C(s) = (aK_c)\frac{s+(\frac{1}{aT})}{s+(\frac{1}{T})} \quad (11)$$

The zero of this controller transfer function is, $z_c = -\frac{1}{aT}$ and the pole is, $p_c = -\frac{1}{T}$. Due to the "a" term in the denominator of z_c, the zero is closer to the imaginary axis than the pole, as illustrated in Figure 4.

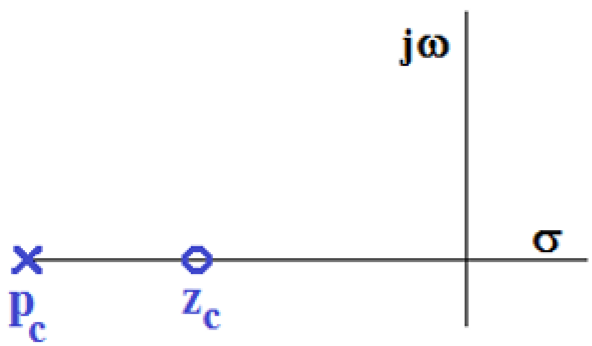


Figure 4. Pole and Zero Location of Controller in Evan's Format. [4]

When considering the plant, P(s), and the controller, C(s), Equation 12 represents the appropriate format to describe the angle condition:

$$< (s - p_c) - < (s - z_c) + \sum_{i=1}^{n-1} < (s - p_i) - \sum_{i=1}^{m-1} < (s - z_i) = \pi \quad (12)$$

The controller contributes a negative angle, $-\gamma$, to the angle condition which is due to the fact that the zero is closer to the imaginary axis than the pole. This is illustrated in Figure 5 below.

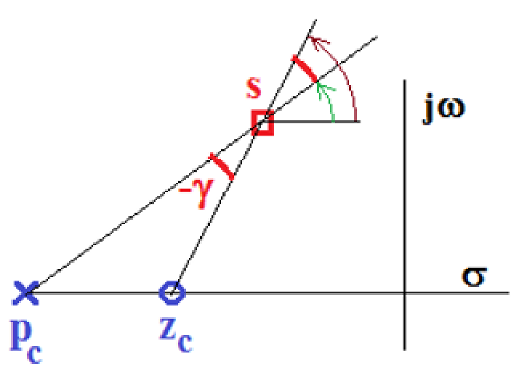


Figure 5. Root and Pole Location of Controller on the S-Plane. [4]

Equation 13 represents a simplified version of the angle condition that includes the factor of γ.

$$\sum_{i=1}^{n-1} < (s - p_i) \; - \; \sum_{i=1}^{m-1} < (s - z_i) \; = \; \pi + \gamma \; (13)$$

To design for a phase-lead controller, the root location is chosen based off of the desired overshoot, OS, and transient time, t_t; where $s = -\alpha + j\beta$ and $s = -\alpha - j\beta$, $t_t = \frac{5}{\alpha}$, and $OS = e^{-\frac{\alpha}{\beta}\pi}$. A proportional controller, $C(s) = K_c$, can be used to find the excess angle γ contributed by the plant. Next, the pole-zero cancellation method is employed to find the values of z_c and p_c that provide for an angle of $-\gamma$ to satisfy the angle condition. Once these values are found, a and T from Equation 11 can be calculated. The root locus can be drawn and can verify that the roots of the closed-loop transfer function lies in this plot. Finally, K_c can be found using the magnitude condition. [4]

Calculations

The following images display the required calculations for designing a phase-lead controller. In Task 1, the model given by Equation 1 through Equation 4 was linearized by offsetting all the variables to cancel the weight of the drone, as shown in Figure 6 below.

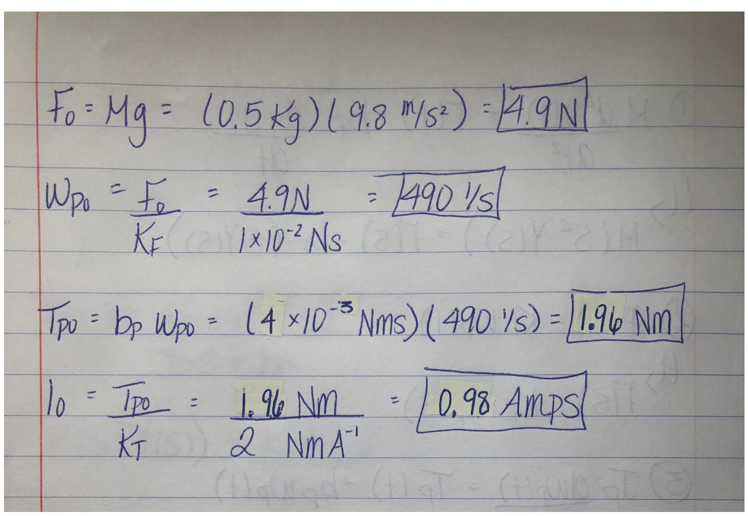
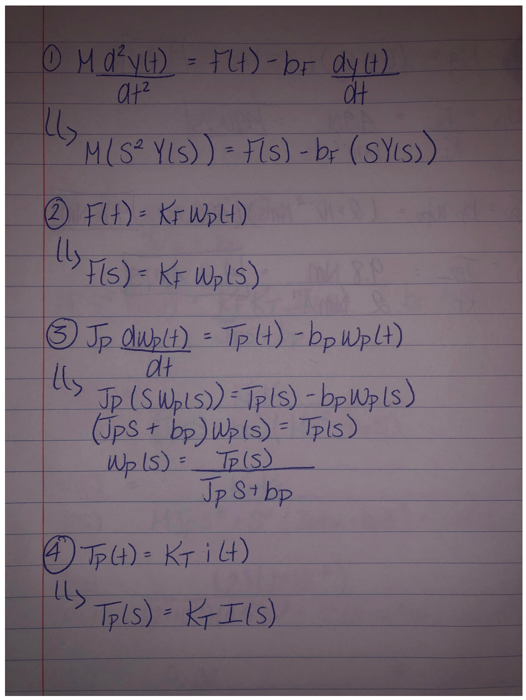


Figure 6. Offset Calculations.

In Task 2, as shown below in Figure 7 and Figure 8, the plant transfer function, P(s), was computed from the DC current, $i(t)$, to the altitude, $y(t)$.



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

↳ $M(S^2Y(S)) = F(S) - b_F(SY(S))$

② $F(t) = K_F W_p(t)$

↳ $F(S) = K_F W_p(S)$

③ $J_p \frac{dW_p(t)}{dt} = T_p(t) - b_p W_p(t)$

↳ $J_p(S W_p(S)) = T_p(S) - b_p W_p(S)$

$(J_p S + b_p) W_p(S) = T_p(S)$

$W_p(S) = \frac{T_p(S)}{J_p S + b_p}$

④ $T_p(t) = K_T i(t)$

↳ $T_p(S) = K_T I(S)$

Figure 7. Laplace Transformations.

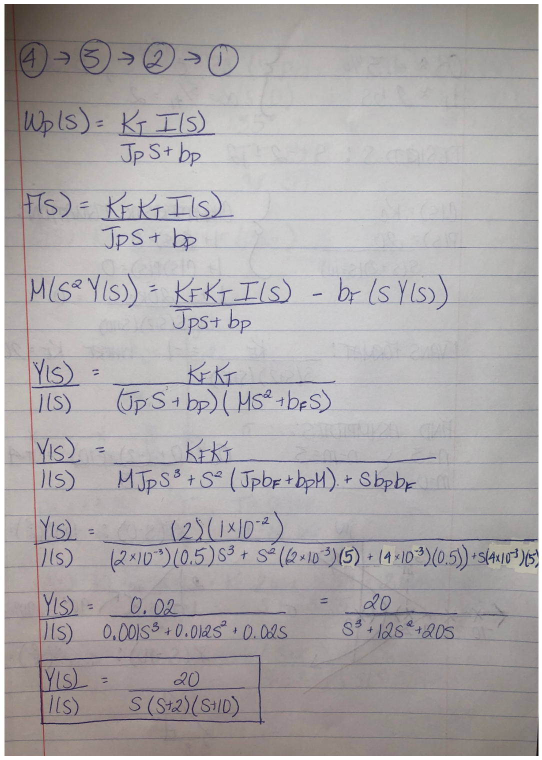


Figure 8. Calculations for the Plant Transfer Function.

In Task 3, the root locus technique and pole-zero cancellation method were used to design a phase-lead controller to meet the design specifications. The design work is shown in Figure 9, Figure 10, and Figure 11.

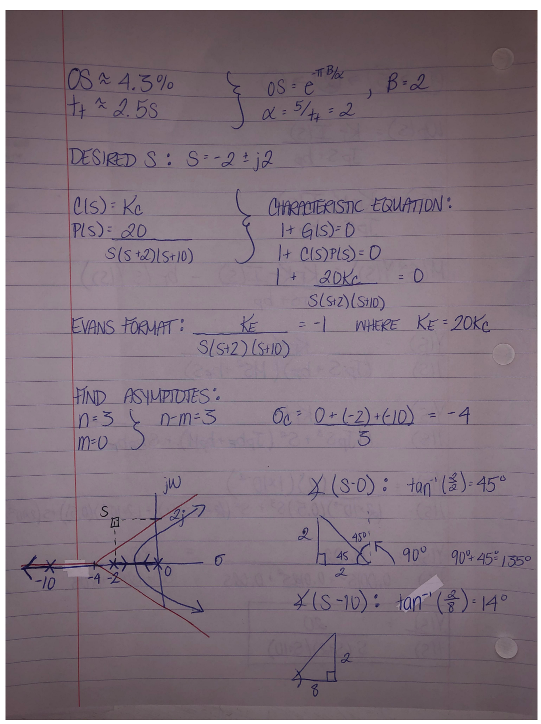
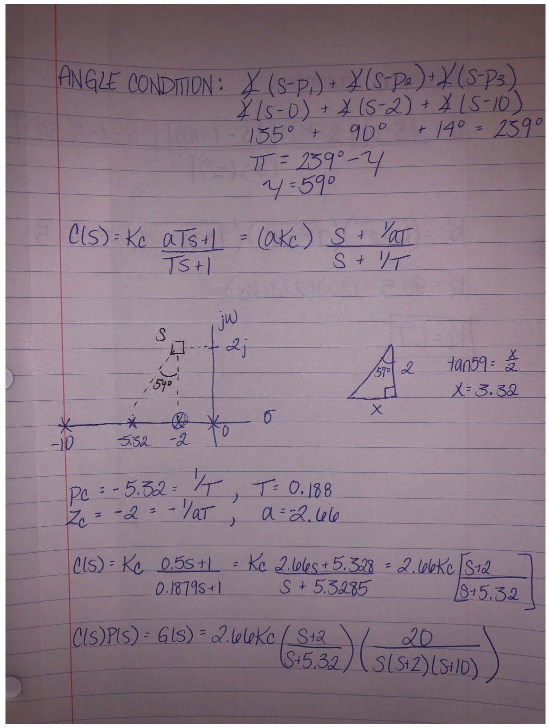


Figure 9. Calculations for Creating a Root Locus Plot.



ANGLE CONDITION: $\measuredangle(S-P_1) + \measuredangle(S-P_2) + \measuredangle(S-P_3)$

$\measuredangle(S-0) + \measuredangle(S-2) + \measuredangle(S-10)$

$135^\circ + 90^\circ + 14^\circ = 239^\circ$

$\pi = 239^\circ - \gamma$

$\gamma = 59^\circ$

$C(S) = K_c \dfrac{aTS+1}{TS+1} = (aK_c)\dfrac{S + 1/aT}{S + 1/T}$

$\tan 59 = \frac{X}{2}$

$X = 3.32$

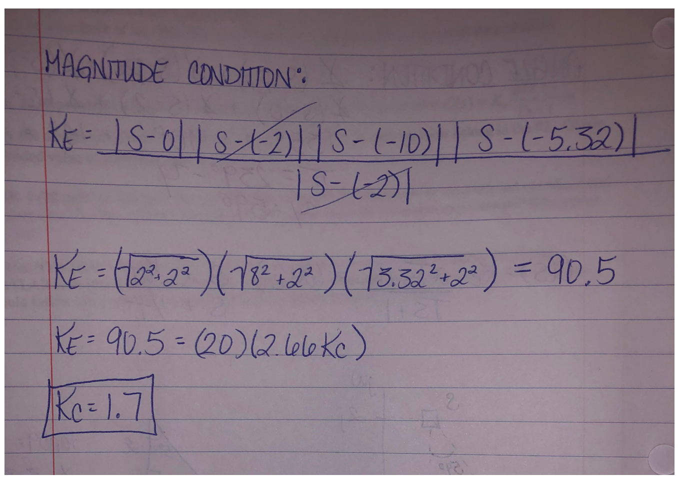
$P_c = -5.32 = 1/T, \quad T = 0.188$

$Z_c = -2 = -1/aT, \quad a = 2.66$

$C(S) = K_c \dfrac{0.5S+1}{0.1879S+1} = K_c \dfrac{2.66S + 5.328}{S + 5.3285} = 2.66K_c\left[\dfrac{S+2}{S+5.32}\right]$

$C(S)P(S) = G(S) = 2.66K_c\left(\dfrac{S+2}{S+5.32}\right)\left(\dfrac{20}{S(S+2)(S+10)}\right)$

Figure 10. Calculations for Satisfying the Angle Condition.



MAGNITUDE CONDITION:

$K_E = \dfrac{|S-0|\,\cancel{|S-(-2)|}\,|S-(-10)|\,|S-(-5.32)|}{\cancel{|S-(-2)|}}$

$K_E = \left(\sqrt{2^2+2^2}\right)\left(\sqrt{8^2+2^2}\right)\left(\sqrt{3.32^2+2^2}\right) = 90.5$

$K_E = 90.5 = (20)(2.66K_c)$

$\boxed{K_c = 1.7}$

Figure 11. Calculation for Controller Gain.

Results

Two types of simulations were performed in MATLAB to verify the design of the phase-lead controller: the first simulation confirmed that the roots $s = -2 + j2$ and $s = -2 + j2$ exist on the root locus plot, and the second simulation plotted the system step response, verifying that a 4.3% OS and a transient time of approximately 2.5 seconds were obtained. Attached below in Figure 12 is the MATLAB code written to obtain the appropriate root locus plot. The controller parameters - α, T, and K_C - computed in the **Calculations** section were inputted into the MATLAB code, as shown below.

```
% Enter plant tf
P = zpk([],[0,-2,-10],20);
% Enter controller tf
T = .18766799928921;
a = 2.6642794823504;
nc = [a*T 1];
dc = [T 1];
C1 =  tf(nc,dc);
% Root Locus
Kcmax = 1.699;
K = 0:Kcmax/999:Kcmax;
G1 = C1*P;
figure(2);
rlocus(G1,K);
[Kc, r] = rlocfind(G1)
```

Figure 12. MATLAB Code for Plotting the Root Locus Diagram.

As shown in Figure 13, the following root locus plot is obtained when K_C is equal to 1.69. The red cross-marks on the root locus plot represent the desired root locations ($s = -2 + j2$ and $s = -2 + j2$). These values are verified in Figure 14 in the MATLAB command window. It's important to note that in Figure 13 and Figure 14 there's an additional root that exists on the real axis at -11.3298. The presence of a third root will not influence the transient time; however, it will affect the OS - the altitude of the drone. The reason the transient time is not impacted is because the third root has a larger damping factor than the other two roots. In other words, the additional root's influence on the system will vanish after a very short period of time. Quantitatively speaking, by employing the equation $t_t = \frac{5}{\alpha}$, where α is defined as 11.3298 seconds, it's found that after .44 seconds, the effects of the third root are gone, so the transient time will be defined by the real part of the roots S_1 and S_2. When it comes to OS, the process occurs at the beginning of operation, so in contrast to the transient time, the third root will influence the system, producing a slight shift in the OS.

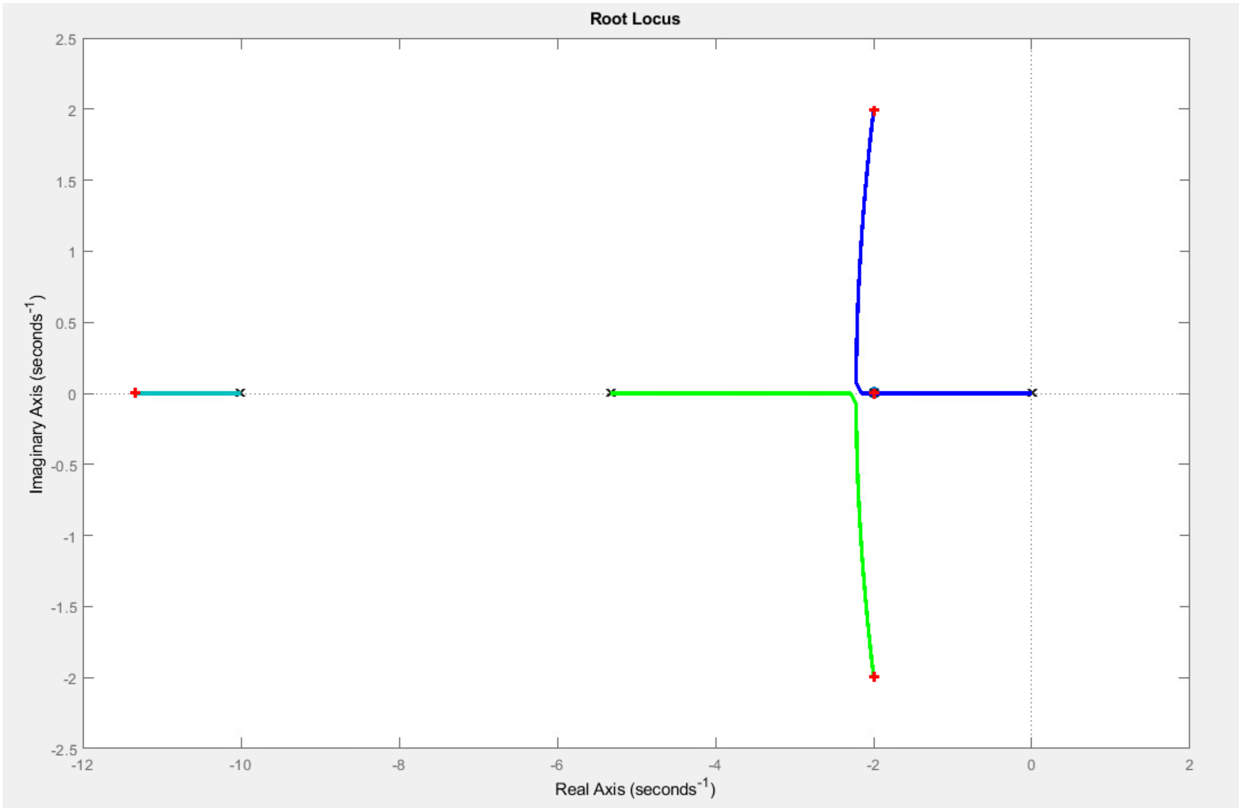


Figure 13. MATLAB Simulated Root Locus Plot when $K_C = 1.69$.

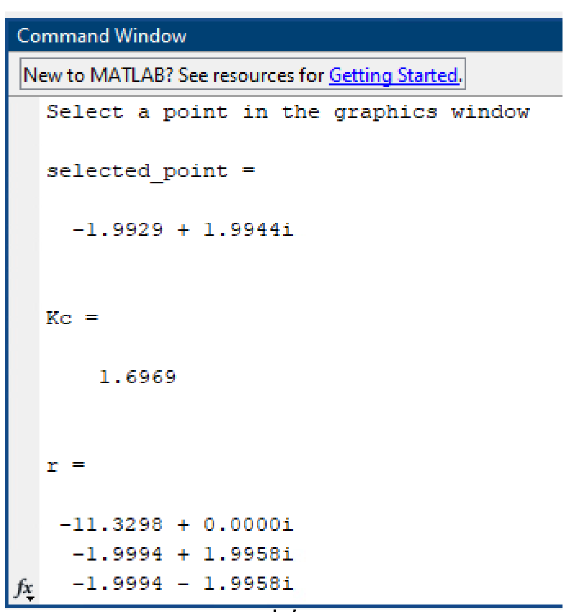
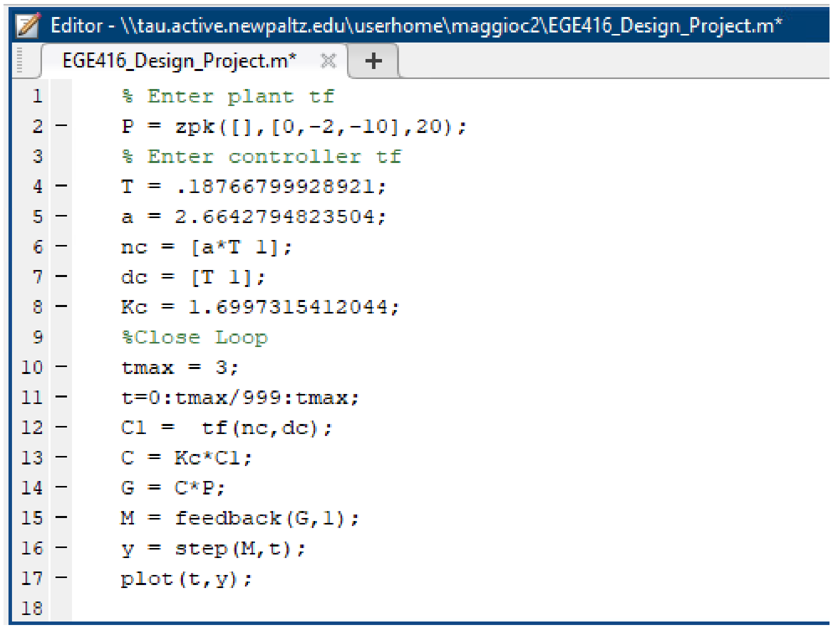


Figure 14. MATLAB Command Window Verifying the Desired Location on the Root Locus Plot.

Attached below in Figure 15 is the MATLAB code written to obtain the simulated plot of the system step response. Like Figure 12, the values obtained for the controller parameters were inputted into the MATLAB code.




```
% Enter plant tf
P = zpk([],[0,-2,-10],20);
% Enter controller tf
T = .18766799928921;
a = 2.6642794823504;
nc = [a*T 1];
dc = [T 1];
Kc = 1.6997315412044;
%Close Loop
tmax = 3;
t=0:tmax/999:tmax;
C1 =  tf(nc,dc);
C = Kc*C1;
G = C*P;
M = feedback(G,1);
y = step(M,t);
plot(t,y);
```

Figure 15. MATLAB Code for Plotting the System Step Response.

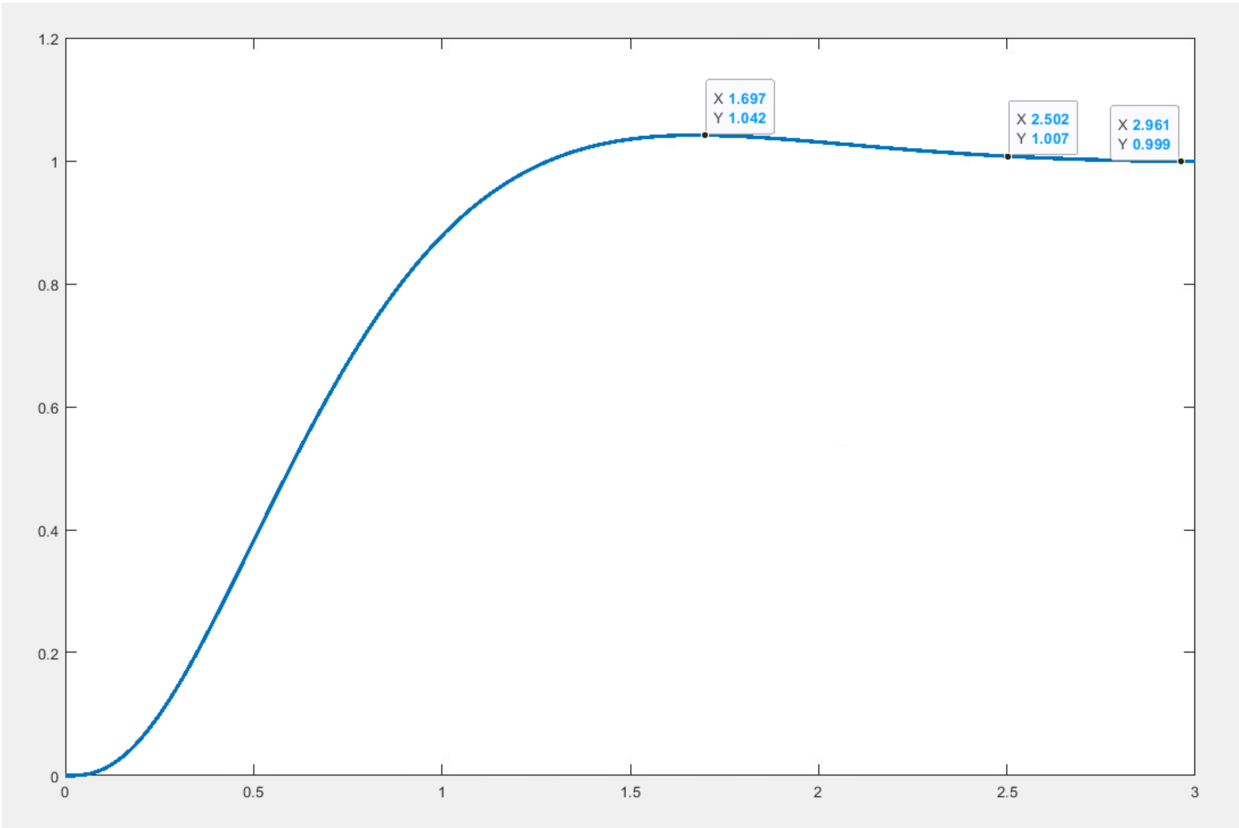


Figure 16. MATLAB Simulation of the System Step Response.

As shown in Figure 16, the simulated step response of the system produced a transient time of approximately 2.5 seconds and an OS of 4.3%, satisfying the design criteria.

Conclusion

A phase-lead controller was successfully designed to meet the design specifications of an OS of 4.3% and a transient time of approximately 2.5 seconds. In addition to this, experience in creating root locus plots, designing closed-loop control systems, and coding in MATLAB was gained.

References

[1] J. Gonzalez, “Course Design Project”, [PDF]. Accessed on: April 8, 2021.

[2] J. Gonzalez, "Root Locus Concept", [*PowerPoint* slides]. Accessed on: April 8, 2021.

[3] J. Gonzalez, "Root Locus Rules", [*PowerPoint* slides]. Accessed on: April 8, 2021.

[4] J. Gonzalez, "Phase-Lead Controller", [*PowerPoint* slides]. Accessed on: April 8, 2021.