Vaughn College of Aeronautics and Technology

Control Systems-ELE350L

Professor Bustamante

Lab #3

By: Alyssa Mitchell and Omomhene Eimunjeze
Room of Experiment: W148
Date:2/15/19

Objective
The objective of this laboratory is to be familiar with the use of MATLAB and SIMULINK to find the responses of linear time-invariant systems. Also, the purpose of this laboratory is to be familiar how to run a program using Laplace Transforms and to understand step responses, impulse responses, transfer functions and systems pulse response.

Equipment/Materials
- MATLAB Software R2014a
- Simulink

Results

QUESTION 1
Consider the system whose differential equation is as follows:

$y\,(iv) + 2y\,''' + 6y\,'' + 8y\,' + 4.8y = r\,'' + 9r\,' + 6r.$

i) Find Transfer function.

ii) Find and plot the roots (zeros, poles) of the transfer function.

iii) Obtain the coefficients of the transfer function.

iv) Obtain the natural response, is the system stable? Explain.

v) Obtain the ramp response, is the system stable? Explain.

$$G(s) = \frac{Y(s)}{r(s)} = \frac{s^2 + 9s + 6}{s^4 + 2s^3 + 6s^2 + 8s + 4.8}$$

i)

```
G= tf([1 9 6], [1 2 6 8 4.8]);
[p,z] = pzmap(G)
grid on
```

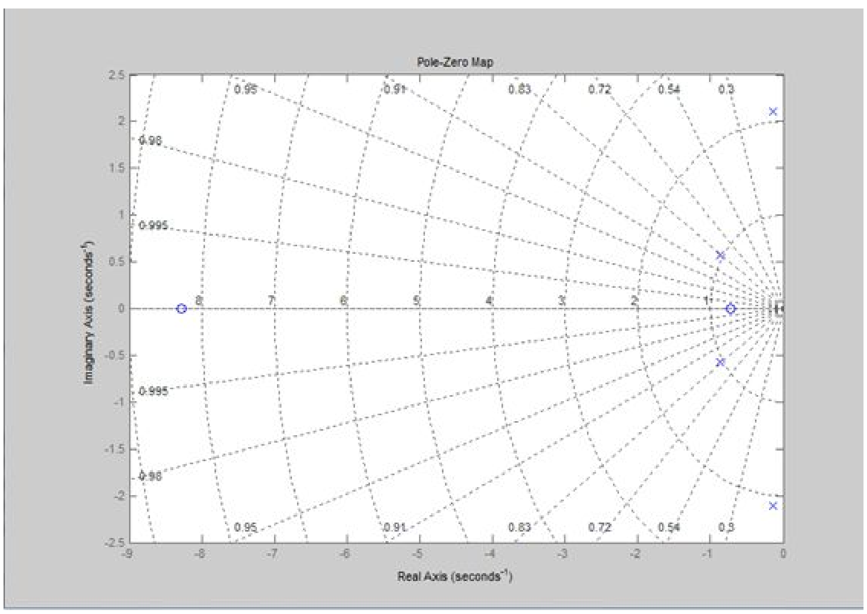
ii)

Figure 1: Code to obtain poles and zeros and plot the results

The poles and zeros are plotted with two axes (real and imaginary) using pzmap(G) in image 1.

```
p =

  -0.1349 + 2.1066i
  -0.1349 - 2.1066i
  -0.8651 + 0.5735i
  -0.8651 - 0.5735i


z =

   -8.2749
   -0.7251
```

The values shown above for p and z are the poles and zeros, respectively. The poles are real (as shown on the left side) and are also imaginary (as shown on the right side).



Graph 1: A plot of the poles (imaginary values labeled as “x”) and zeros values (“o”)

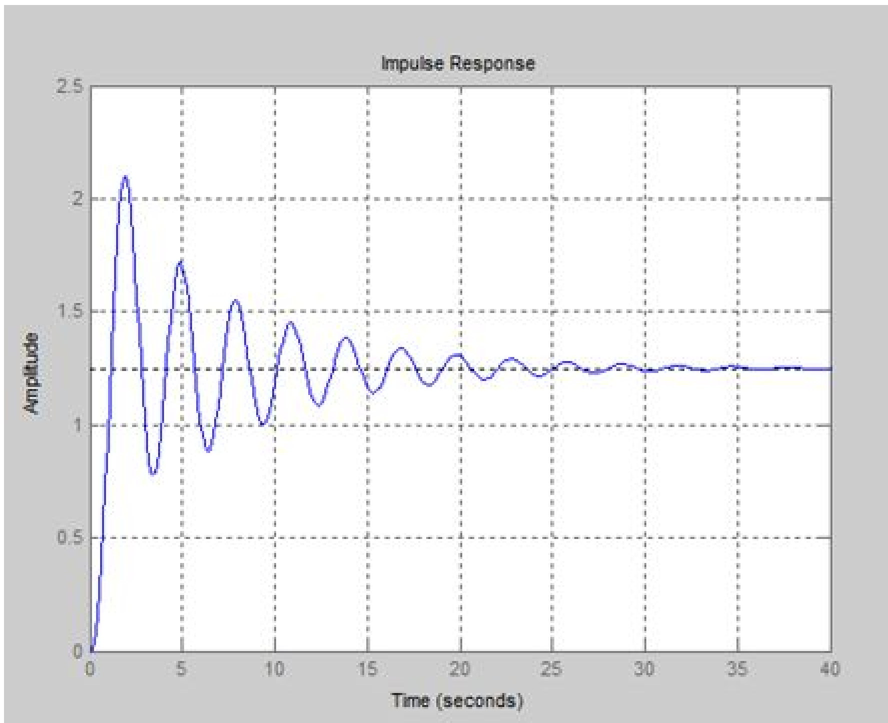
The transfer function is shown through graph 1.

iii) Numerator coefficients: [1 9 6] → Denominator coefficients: [1 2 6 8 4.8] iiii)

```
clear all
G= tf([1 9 6], [1 2 6 8 4.8 0]);
impulse(G)
grid on
```

Figure 2: Code utilized to generate the impulse response of the transfer function by using

impulse(G)



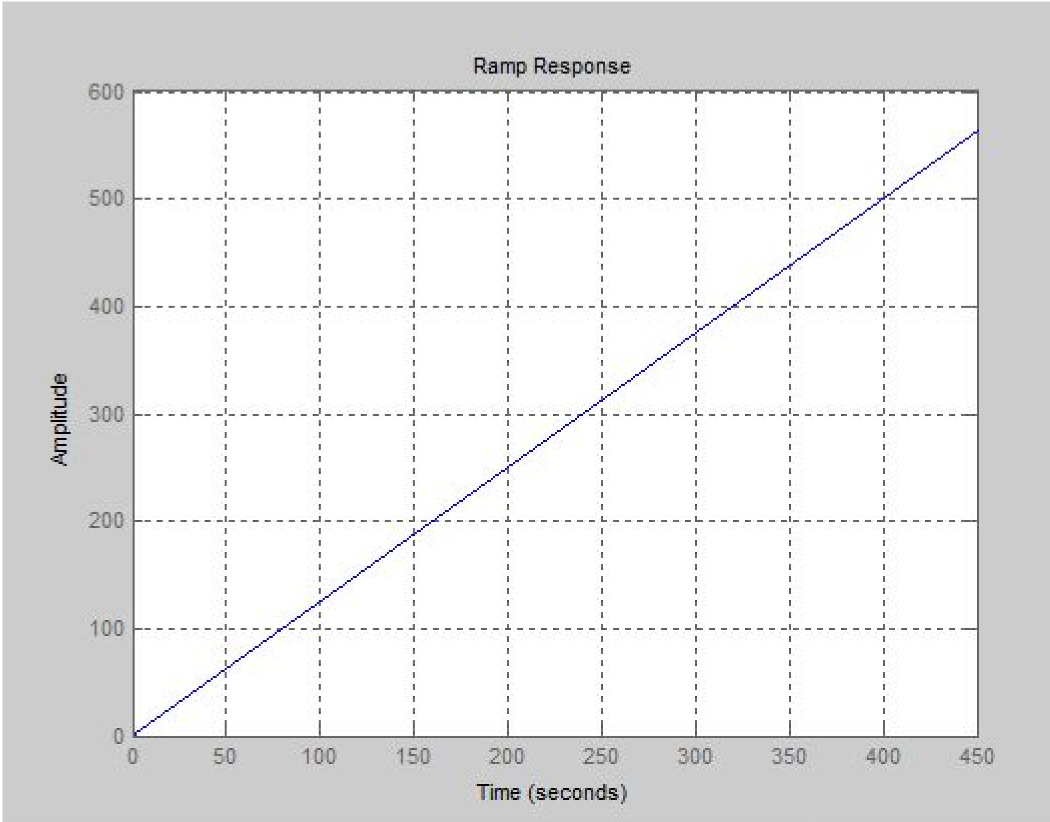
Graph 2: Results generated for the impulse response that stabilizes eventually as it goes to zero

v)

```
clear all
G= tf([1 9 6], [1 2 6 8 4.8 0]);
step(G)
grid on
title('Ramp Response')
```

Figure 3: Using the step function to obtain the ramp function

The transfer function is multiplied by 1/s because its laplace transform is 1/s^2



Graph 3: The unstable ramp response is shown, it is unstable because it does not settle at a value after time, it continues on a straight horizontal path.

QUESTION 2

Consider the system whose differential equation is as follows.

$$y''' - 9y'' + 27y' + 21y = 66r' + 44r$$

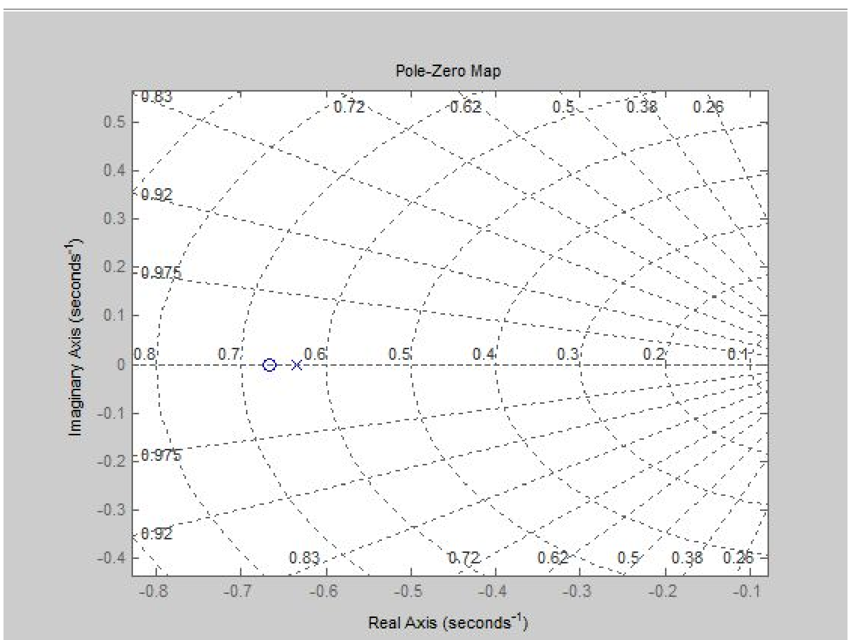
i) Find the Transfer function.
ii) Find and plot the roots (zeros, poles) of the transfer function.
iii) Obtain the coefficients of the transfer function.
iv) Find the transient and steady state for the step and impulse response. What can you conclude about the results.

i)----------> Transfer Function

```
G =

        66 s + 44
  ----------------------
  s^3 - 9 s^2 + 27 s + 21
```

ii) Poles(p) and Zeros(z) Values

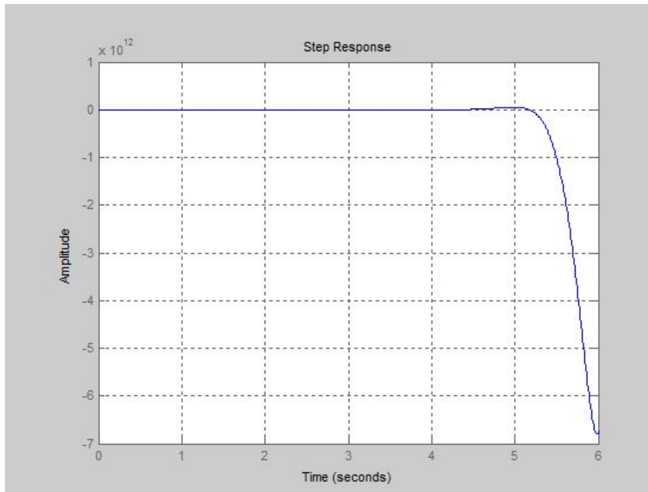
```
p =

   4.8171 + 3.1473i
   4.8171 - 3.1473i
  -0.6342 + 0.0000i


z =

   -0.6667
```



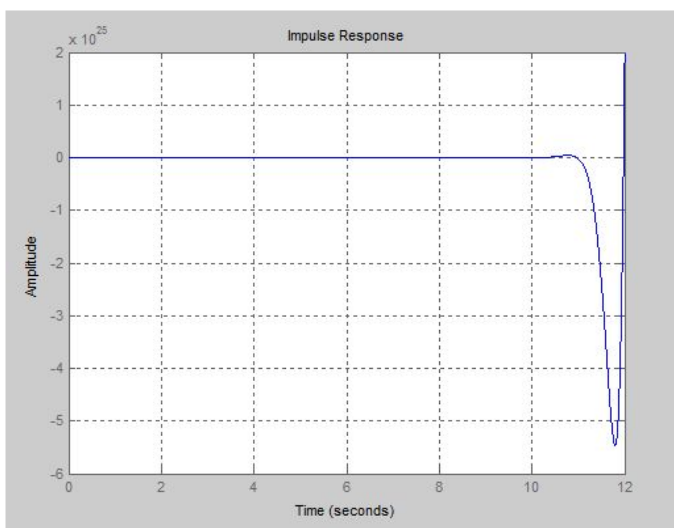
Graph 4: The Poles and Zeros represented on the map

iii) Using the code below to find the coefficients of the transfer function

```
clear all
G = tf([66 44],[1 -9 27 21])
pzmap(G)
grid on
```



Graph 5: The step response of the system



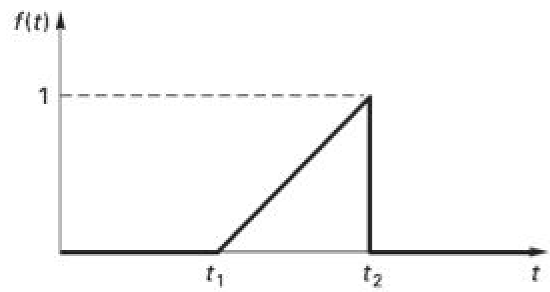
Graph 6: The impulse response of the system

QUESTION 3

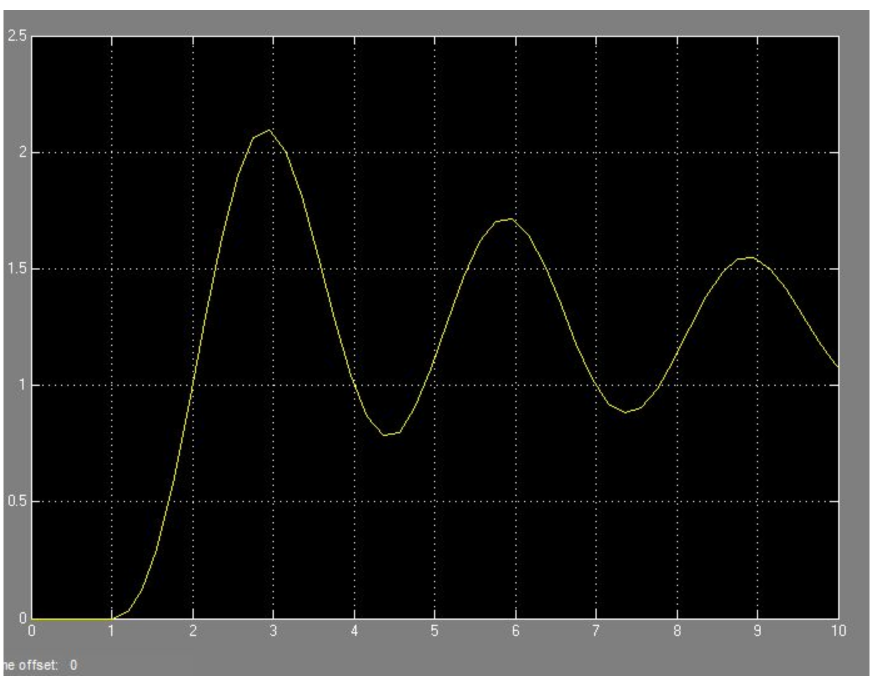
A) Find f(t) and plot both f(t) and F(s) functions on the same plot. Describe the time function graph. What can you conclude about the time and frequency domain.

$$F(s) = \frac{e^{-t_1 s} - e^{-t_2 s}}{s} \qquad t_2 > t_1$$

B) The f(t) is represented by a triangular waveform shown below:



Find the Laplace transform.



Graph 7: Result of the transfer function after it was simulated in simulink using ramp

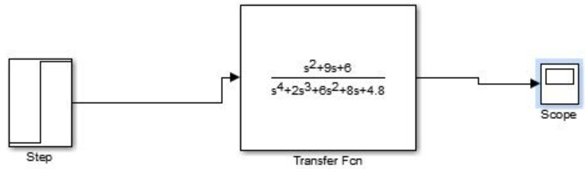
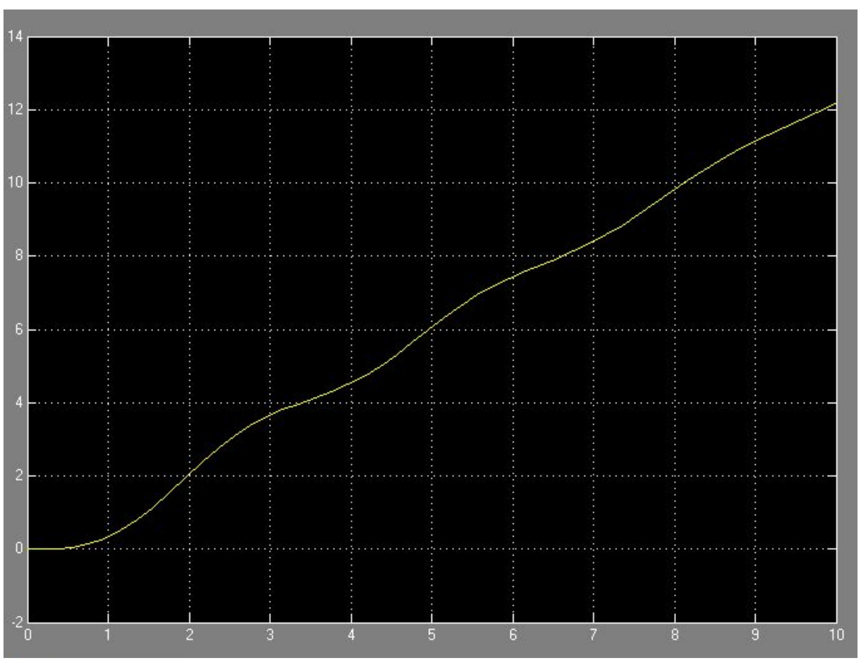


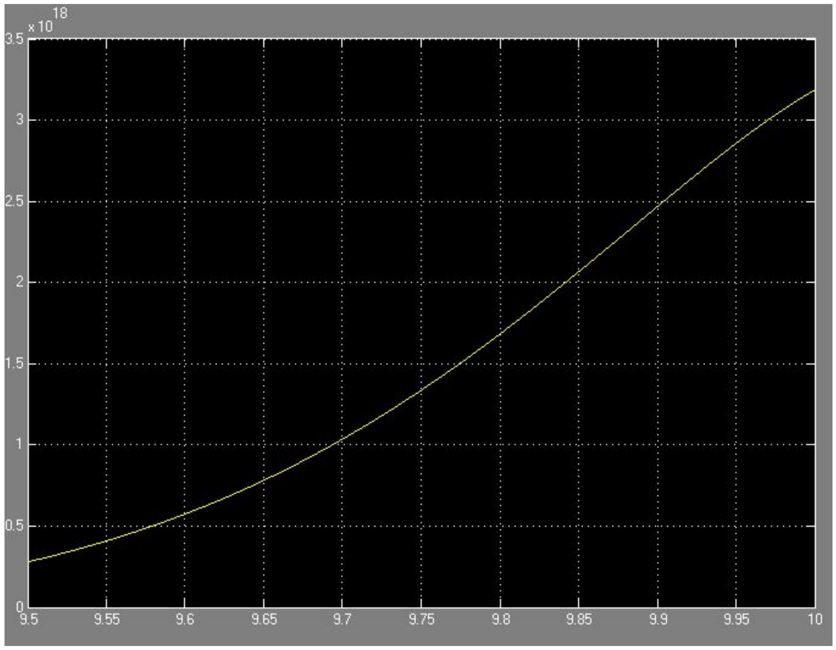
Figure 4: The simulation of the transfer function using simulink.

QUESTION 4

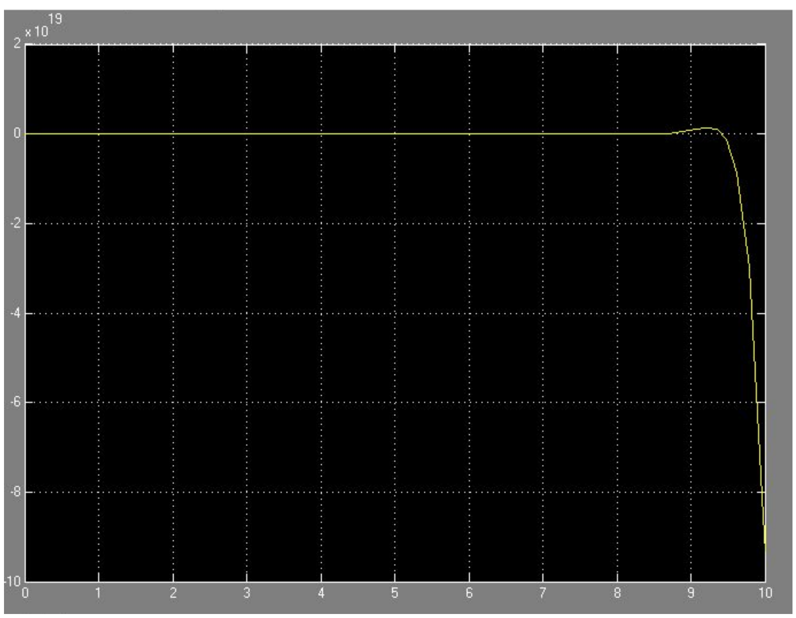
Use Simulink to do Q1, Q2 and Q3 make a comparison of these results with the one obtained using MATLAB code. Also make sure to create both system using Simulink, time domain and frequency domain. Do you know the difference? Can you explain?



Graph 8: Q1 Result of the transfer function after it was simulated in simulink using ramp



Graph 9: Q2 Result of the transfer function after it was simulated in simulink using ramp



Graph 10:Q3 Result of the transfer function after it was simulated in simulink using ramp

Each of the graphs differ because of the differences in stability.

QUESTION 5

For each find the inverse Laplace and the original differential equation. Obtain the natural response for each. Explain your answers using cases base on the poles (Real, imaginary).

(a) $F(s) = \dfrac{5}{s(s+1)(s+2)}$

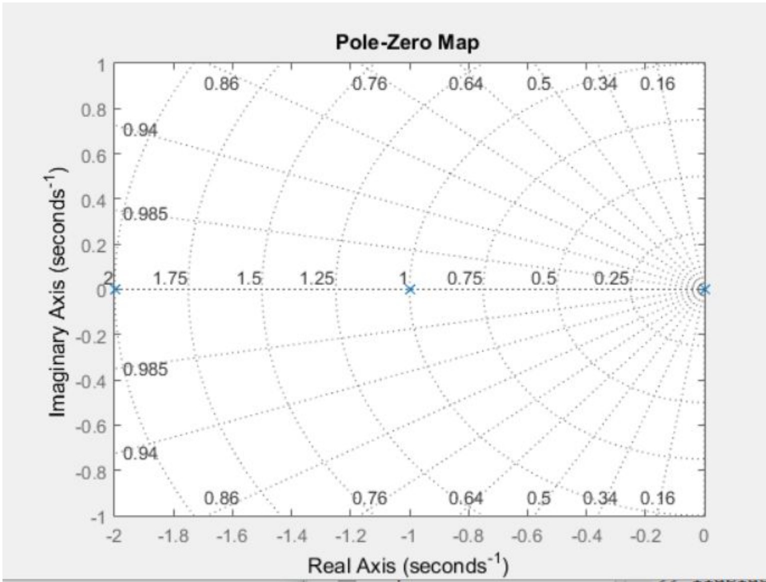
no5.m | no6.m

```matlab
syms f s t
f(s) = 5 / (s * (s + 1) *(s + 5));
%inverse laplace
disp('The inverse laplace is:')
g = ilaplace(f, s, t);
pretty(g)
%original equation
disp('The original differential equation is:')
pretty(int(g))
%cases based on poles
a = [5];
b = [1 3 2 0];
[r, p, k] = residue(a, b)
```

Command Window

```
>> no5
The inverse laplace is:
exp(-5 t)    5 exp(-t)
--------- - --------- + 1
    4           4

The original differential equation is:
    5 exp(-t)    exp(-5 t)
t + --------- - ---------
        4           20


r =

    2.5000
   -5.0000
    2.5000


p =

    -2
    -1
     0


k =
```

Code for Inverse Laplace Exercise 5a



Graph of Poles and Zeros of Exercise 5a

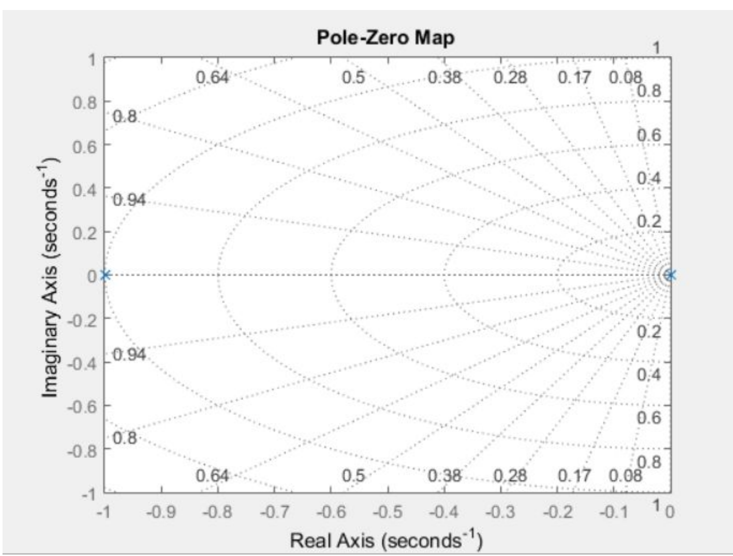
B.

(b) $F(s) = \dfrac{1}{s^2(s+1)}$

```
syms f s t
f(s) = 1 / (s^2 * (s + 1));
%inverse laplace
disp('The inverse laplace is:')
g = ilaplace(f, s, t);
pretty(g)
%original equation
disp('The original differential equation is:')
pretty(int(g))
%cases based on poles
a = [1];
b = [1 1 0 0];
[r, p, k] = residue(a, b)
```

```
The inverse laplace is:
t + exp(-t) - 1

The original differential equation is:
 2
t
-- - exp(-t) - t
 2


r =

     1
    -1
     1


p =

    -1
     0
     0


k =
```

Code for Inverse Laplace Exercise 5b



Graph of Poles and Zeros of Exercise 5b

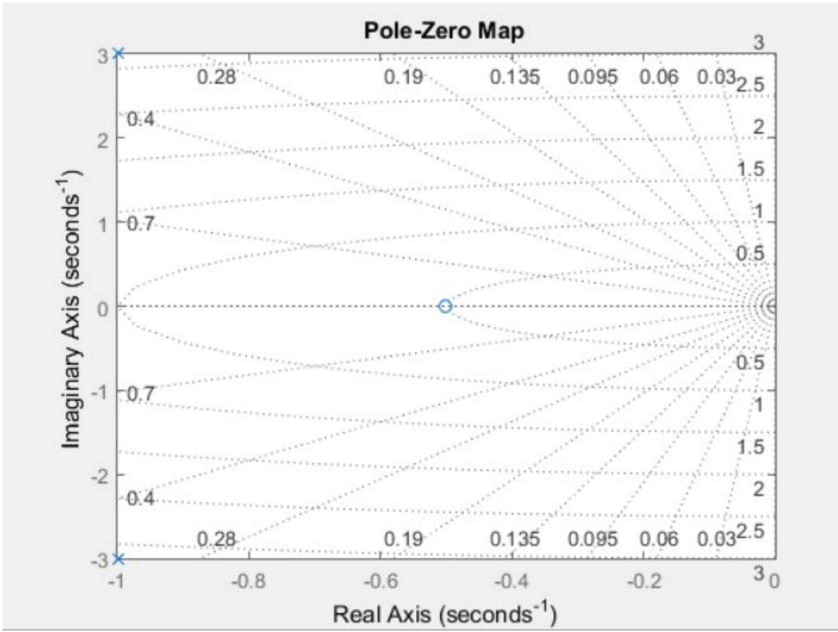
C.

(c) $F(s) = \dfrac{2s + 1}{s^2 + 2s + 10}$

```
syms f s t
f(s) = (2*s + 1) / (s^2 + 2*s + 10);
%inverse laplace
disp('The inverse laplace is:')
g = ilaplace(f, s, t);
pretty(g)
%original equation
disp('The original differential equation is:')
pretty(int(g))
%cases based on poles
a = [2 1];
b = [1 2 10];
[r, p, k] = residue(a, b)
```

```
The inverse laplace is:
        /            sin(3 t) \
exp(-t) | cos(3 t) - -------- | 2
        \               6     /

The original differential equation is:
  exp(-t) (cos(3 t) 3 - sin(3 t) 19)
- ----------------------------------
                  30


r =

   1.0000 + 0.1667i
   1.0000 - 0.1667i


p =

  -1.0000 + 3.0000i
  -1.0000 - 3.0000i


k =

     []
```

Code for Inverse Laplace Exercise 5c



Graph of Poles and Zeros of Exercise 5c

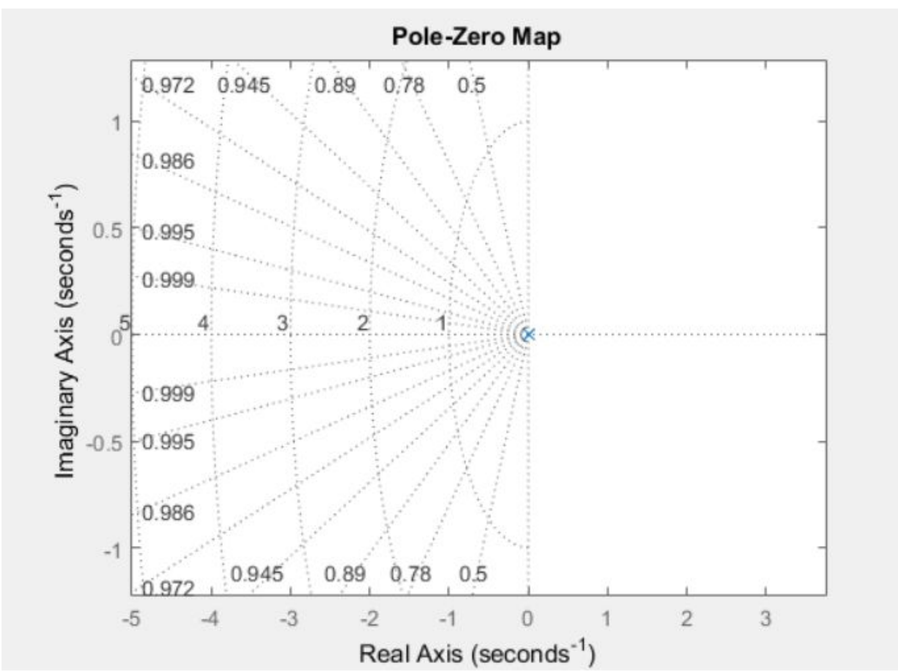
D.

(d) $F(s) = \dfrac{s - 30}{s(s^2 + 4s + 29)}$

```
syms f s t
f(s) = (s - 30) / (s(s^2 + 4*s + 29));
%inverse laplace
disp('The inverse laplace is:')
g = ilaplace(f, s, t);
pretty(g)
%original equation
disp('The original differential equation is:')
pretty(int(g))
%cases based on poles
a = [1 -30];
b = [1 4 29 0];
[r, p, k] = residue(a, b)
```

Code for Inverse Laplace Exercise 5d

```
The inverse laplace is:
          /            sin(5 t) 89 \
exp(-2 t) | cos(5 t) + ----------- | 30
          \                150     /      30
--------------------------------------- - --
                  29                      29

The original differential equation is:
sin(5 t) exp(-2 t) 572   cos(5 t) exp(-2 t) 149   30 t
---------------------- - ---------------------- - ----
         4205                     841              29


r =

   0.5172 - 0.3069i
   0.5172 + 0.3069i
  -1.0345 + 0.0000i


p =

  -2.0000 + 5.0000i
  -2.0000 - 5.0000i
   0.0000 + 0.0000i


k =

     []
```



Graph of Poles and Zeros of Exercise 5d

Question 6

Given that $f(t) = 4e^{-2(t-3)}$

(a) Find $\mathcal{L}[df(t)/dt]$ by differentiating $f(t)$ and then using the Laplace

no6.m

```
syms t s
f = 4 * exp(-2*(t-3));
g = diff(f)
F = laplace(g, t, s)

pretty(F)
```

Command Window

```
>> no6

g =

-8*exp(6 - 2*t)


F =

-(8*exp(6))/(s + 2)

  8 exp(6)
- --------
    s + 2
```

Code for Exercise 6a

(c) Repeat (a) and (b), for the case that $f(t) = 4e^{-2(t-3)}u(t-3)$.

Editor - C:\Users\Daedalus\Desktop\no6.m

no6.m

```
syms t s
f(t) = 4 * (exp(-2*(t-3))) * heaviside(t - 3);
g = diff(f)
F = laplace(g, t, s)
pretty(F)
```

Command Window

```
>> no6

g(t) =

4*dirac(t - 3)*exp(6 - 2*t) - 8*heaviside(t - 3)*exp(6 - 2*t)


F =

4*exp(-3*s) - (8*exp(-3*s))/(s + 2)

                 exp(-3 s) 8
exp(-3 s) 4 - -----------
                    s + 2
```

Code for Exercise 6c

Part 6.1

Find the Laplace transforms of

(a) $f_1(t) = 5e^{-2(t-1)}$

```
syms t s
f(t) = 5 * (exp(-2*(t-1)));
g = diff(f)
F = laplace(g, t, s)
pretty(F)
```

Command Window

```
>> no6

g(t) =

-10*exp(2 - 2*t)


F =

-(10*exp(2))/(s + 2)

  10 exp(2)
- ---------
    s + 2
```

Code for Exercise 6.1a

(b) $f_2(t) = 5e^{-2(t-1)}u(t-1)$

```
syms t s
f(t) = 5 * (exp(-2*(t-1))) * heaviside(t - 1);
g = diff(f)
F = laplace(g, t, s)
pretty(F)
```

Command Window

```
>> no6

g(t) =

5*dirac(t - 1)*exp(2 - 2*t) - 10*heaviside(t - 1)*exp(2 - 2*t)


F =

5*exp(-s) - (10*exp(-s))/(s + 2)

             10 exp(-s)
5 exp(-s) - ----------
                s + 2
```

Code for Exercise 6.1b

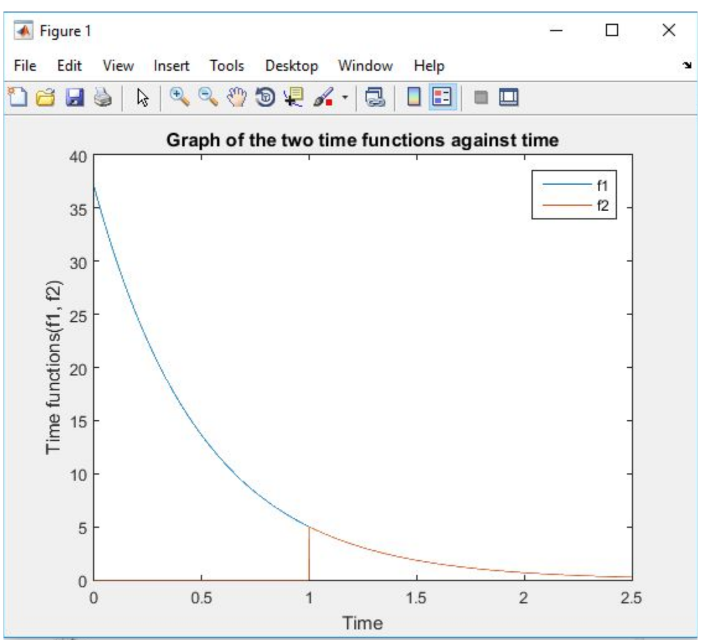
(c) Sketch the two time functions.

```
syms t s
f1(t) = 5 * (exp(-2*(t-1)));
g1 = diff(f1);
F1 = laplace(g1, t, s);
pretty(F1);

f2(t) = 5 * (exp(-2*(t-1))) * heaviside(t-1);
g2 = diff(f2);
F2 = laplace(g2, t, s);
pretty(F2);

t = 0:0.001:2.5;
plot(t, f1(t), t, f2(t))
title('Graph of the two time functions against time')
ylabel('Time functions(f1, f2)')
xlabel('Time')
```

Command Window

```
  10 exp(2)
- ---------
    s + 2

             10 exp(-s)
5 exp(-s) - ----------
                s + 2
```

Code for Exercise 6.1c



Graph of exercise 6.1c showing the two time functions, f1 and f2 graphed against time

(d) Why are the two transforms different?

The two transforms are different, since the function f1(t) has no delay and the function f2(t) is delayed by one second

Conclusion

Overall, in this laboratory we have learned how to use MATLAB and Simulink to find the responses of linear time-invariant systems by using Laplace transforms. We also learned how to develop a transfer function from Simulink to find the system impulse response, step response and system responses of a general input. We learnt to plot transient responses using the Simulink software package and present and reduce the errors that we found from the systems. In this laboratory, we also got familiar with the system and most of the functions in the Simulink library browser which we used them to represent the systems. I was able to simulate the response of the system by a unit step input. Overall, the laboratory was very useful to understand the Simulink and MATLAB. Finally, all tasks were performed and successfully completed.

Vaughn College of Aeronautics and Technology

Control Systems-ELE350L

Professor Bustamante

Lab #4

By: Alyssa Mitchell and Omomhene Eimunjeze
Room of Experiment: W148
Date:2/13/19

Objective

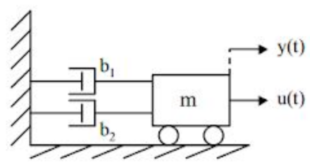
The purpose of this lab was to show how transfer functions can be entered and solved within the MATLAB software. The transfer function of a system can be found through hand calculations. However, by using the MATLAB software, the transfer function of a system can be obtained by utilizing the appropriate function in the code, such as sys=parallel(), or sys=series().

Equipment/Materials

- MATLAB Software
- Simulink

Results

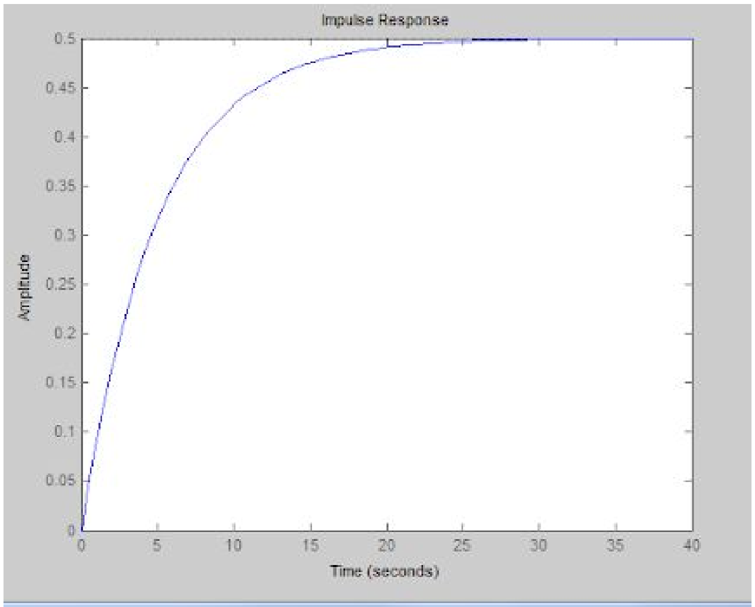
1. Find the transfer function between the input force u(t) and the output displacement y(t) for the system shown below.



where b1 and b2 are the frictional coefficients. For b1 = 0.5 N-s/m, b2 = 1.5 N-s/m, m = 10 kg and u(t) is a unit-impulse function, what is the response y(t)? Check and plot the response with MATLAB.

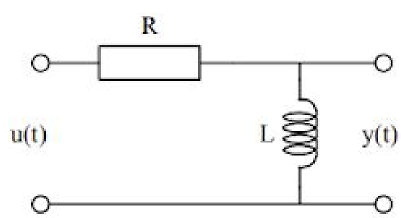
```
num = [0 0 1];
den = [10 2 0];
SYS = tf(num, den)
impulse(SYS)
```

```
SYS =

       1
  ------------
  10 s^2 + 2 s

Continuous-time transfer function.
```



2.

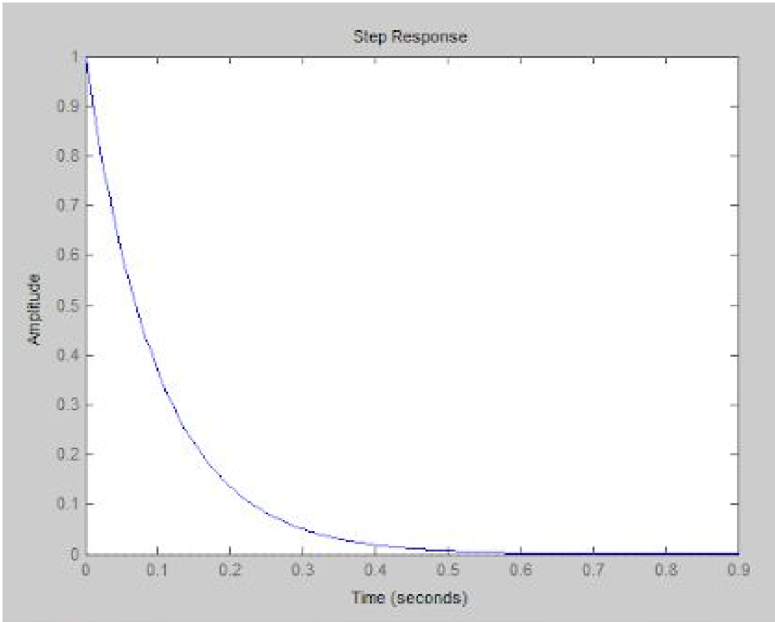
For the following circuit, find the transfer function between the output voltage across the inductor y(t), and the input voltage u(t).



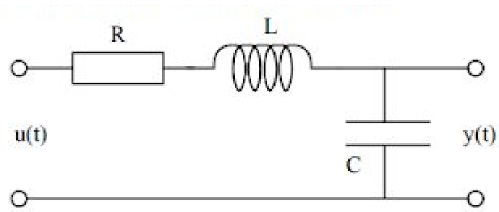
For R = 1 Ω, L = 0.1 H, and u(t) is a unit-step function, what is the response y(t)? Check and plot the result using MATLAB.

```
num = [0.1 0];
den = [0.1 1];
SYS = tf(num, den)
step(SYS)
```

```
SYS =

     0.1 s
  ---------
  0.1 s + 1

Continuous-time transfer function.
```



3. Find the transfer function of the electrical circuit shown below.



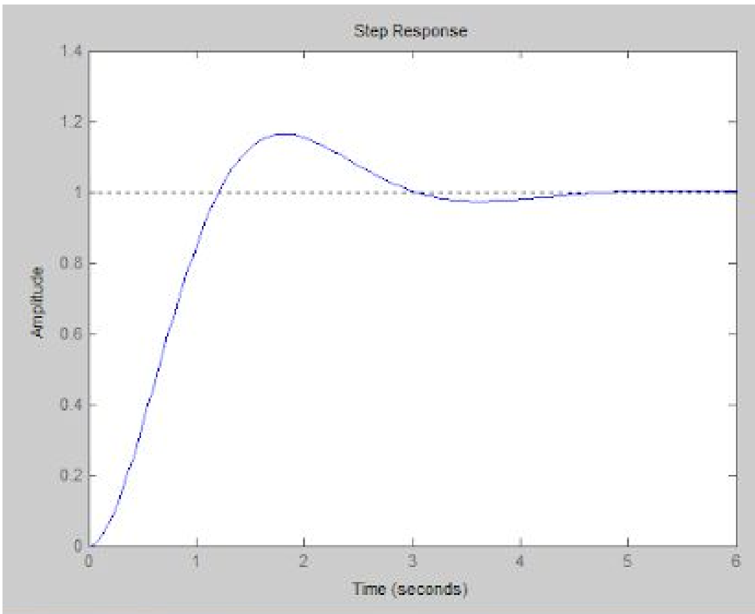
For R = 1 Ω, L = 0.5 H, C = 0.5 F, and a unit step input u(t) with zero initial conditions, compute y(t). Sketch the time function y(t) and plot it with MATLAB.

Transfer Function

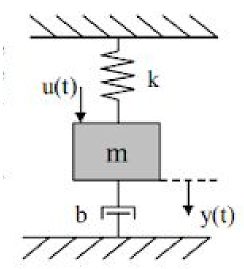
$$G(s) = \frac{1}{LCs^2 + RCs + 1}$$

```
num = [0 0 1];
den = [0.25 0.5 1];
SYS = tf(num, den)
step(SYS)
```

```
SYS =

            1
  --------------------
  0.25 s^2 + 0.5 s + 1

Continuous-time transfer function.
```



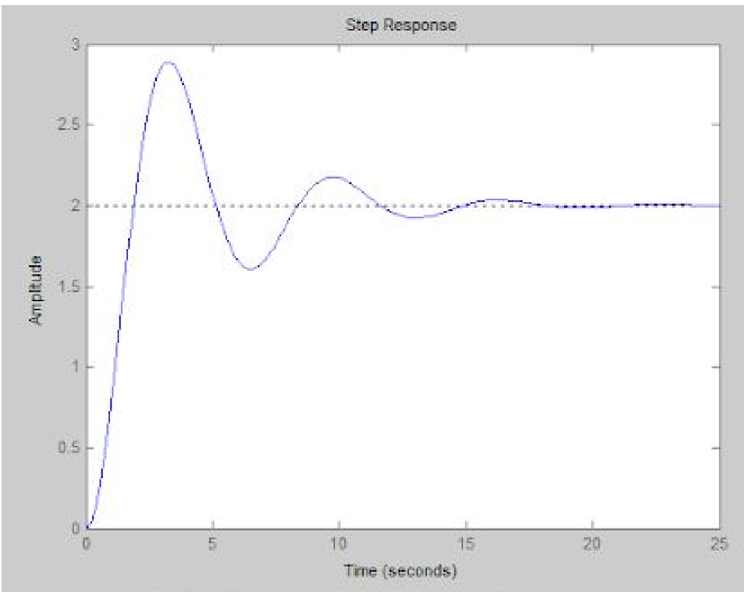
4. In the mechanical system shown in the figure below, m is the mass, k is the spring constant, b is the friction constant, u(t) is the external applied force and y(t) is the corresponding displacement. Find the transfer function of this system. For m = 1 kg, k = 1 kg/s2 , b = 0.5 kg/s, and a step input u(t) = 2 N, compute the response y(t) and plot it with MATLAB.



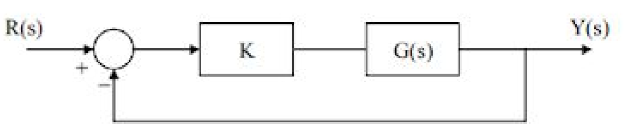
$$G(s) = \frac{1}{ms^2 + bs + k}$$

```
num = [0 0 1];
den = [1 0.5 1];
SYS = tf(num, den)
step(2*SYS)
```

```
SYS =

         1
  ---------------
  s^2 + 0.5 s + 1

Continuous-time transfer function.
```



5. Write down the transfer function Y(s)/R(s) of the following block diagram.



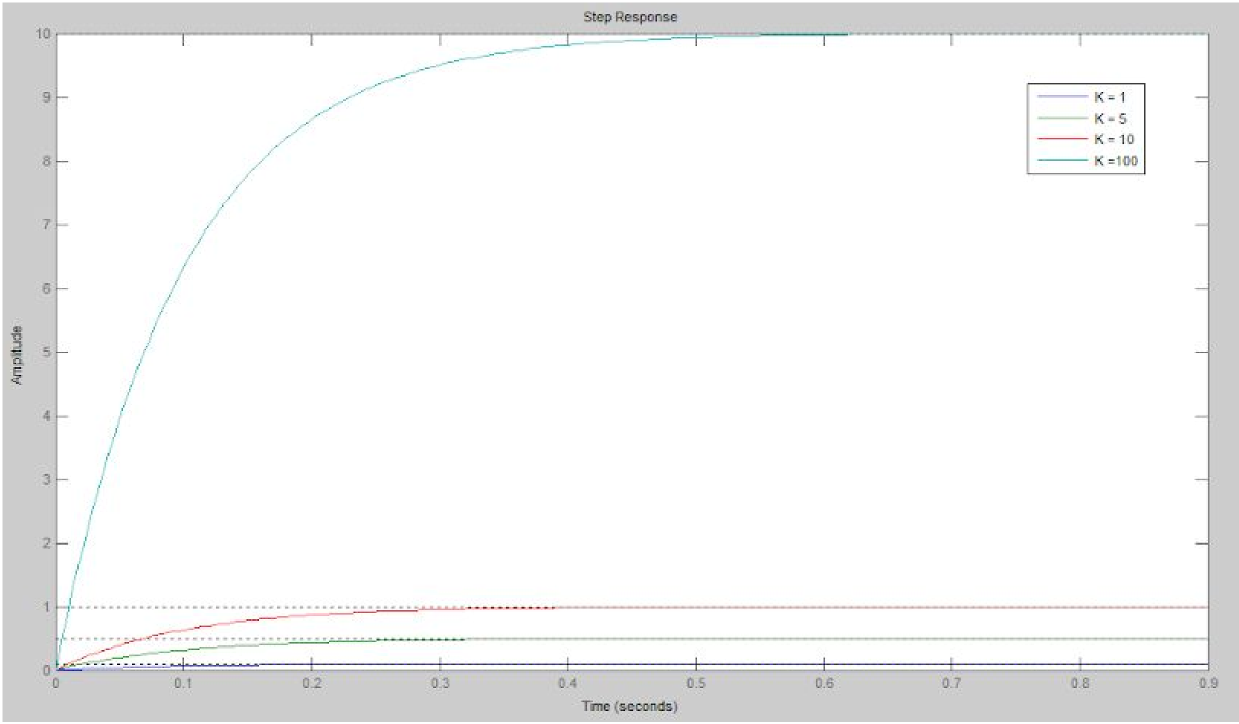
a) For G(s) = 1/(s + 10) and K = 10, determine the closed loop transfer function with MATLAB.

```
K = tf([0 10], [0 1]);
G = tf([0 1], [1, 10]);
SYS = series(K, G)
```

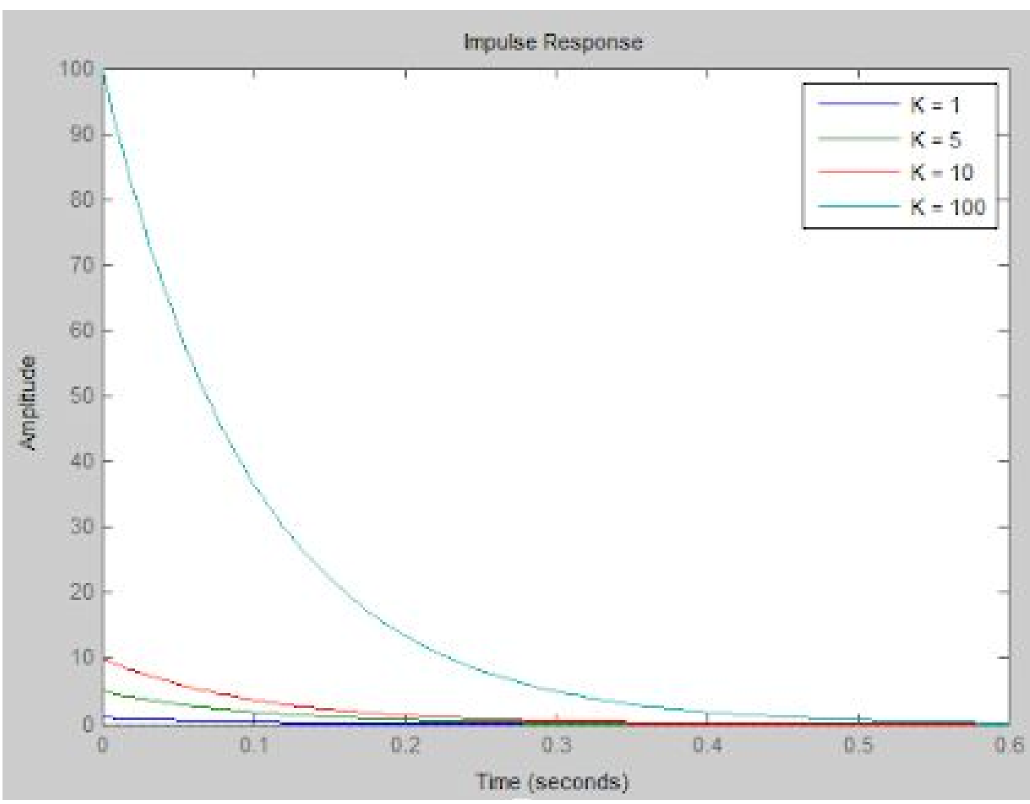
```
SYS =

    10
  ------
  s + 10

Continuous-time transfer function.
```

b) For K = 1, 5, 10, and 100, plot y(t) on the same window for a unit-step input r(t) with MATLAB, respectively. Comment on the results.

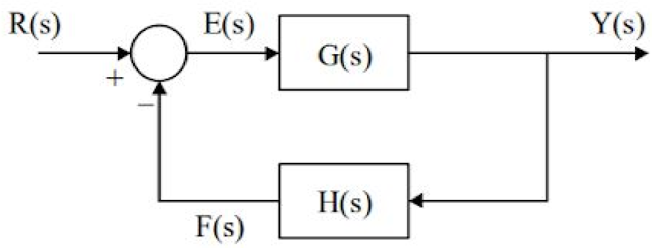
```
SYS =

     1
  ------
  s + 10

Continuous-time transfer function.


SYS1 =

     5
  ------
  s + 10

Continuous-time transfer function.


SYS2 =

    10
  ------
  s + 10

Continuous-time transfer function.


SYS3 =

   100
  ------
  s + 10

Continuous-time transfer function.
```



c) Repeat (b) with a unit-impulse input r(t).



6. Find the closed loop transfer function for the following diagram.



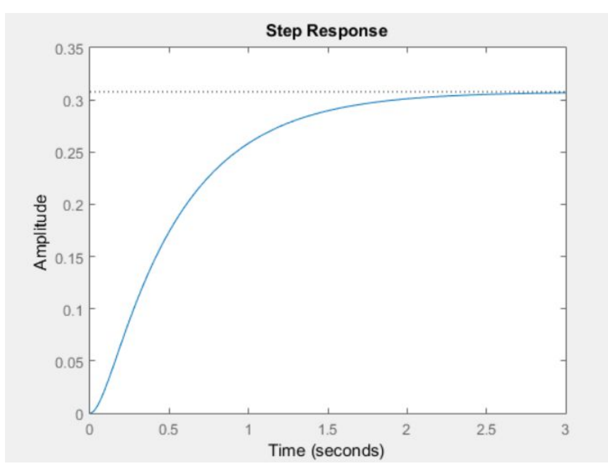
a) For G(s) = 8/(s2 + 7s + 10) and H(s) = s+2, determine the closed loop transfer function with MATLAB.

```
sys =

          8
  ---------------
  s^2 + 15 s + 26

Continuous-time transfer function.
```

Figure 1: The closed-loop transfer function for the diagram in question 6.

The closed loop transfer function is determined using Matlab. Within Matlab, the function "Cloop=feedback()" is used to obtain the transfer function of the diagram shown above in figure1.

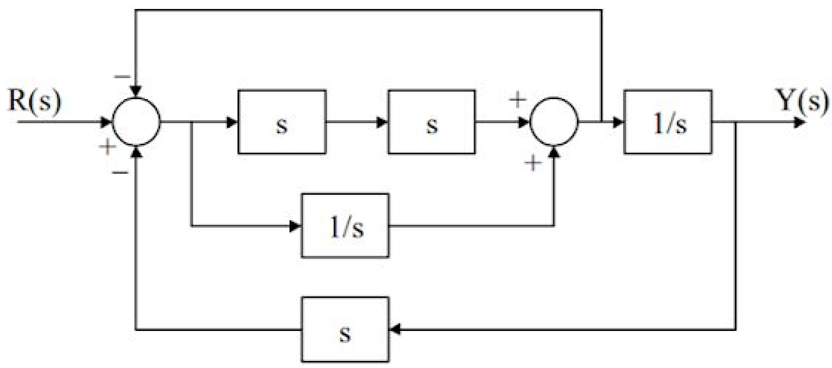
b) Plot y(t) for a unit-step input r(t) with MATLAB.



Graph 1: y(t) plotted for a unit-step input with MATLAB

In order to plot this graph, the step() function and the plot() function were utilized within the code to get the desired outcome.

7. Determine the transfer function of the following diagram. Check your answer with MATLAB.

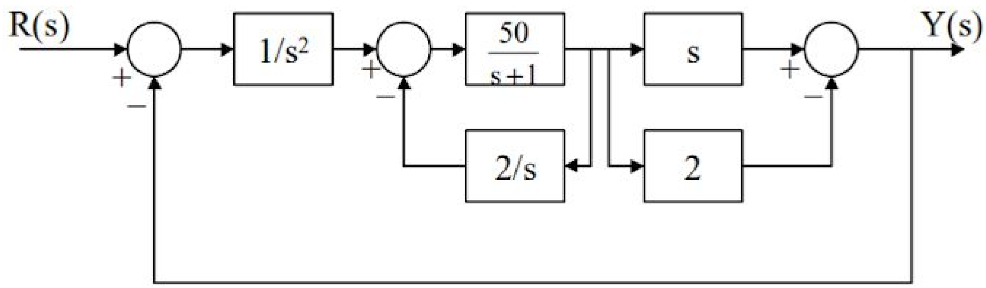



```
sys =

       s^3 + 1
  -----------------
  2 s^4 + s^2 + 2 s

Continuous-time transfer function.
```

```
G3= feedback(G2,H1);
G4= series(G3,H);
sys= feedback(G4,G)
```

The snippet of the code above is used to create the transfer function shown above for the system shown in problem 6. It displays how Matlab simplifies the various blocks within the system into the valid transfer function.

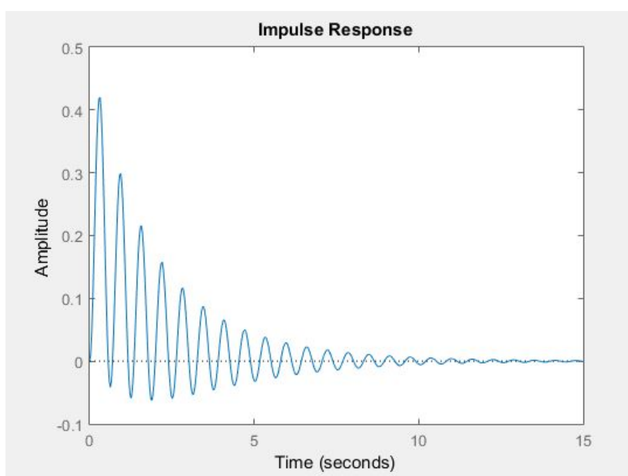
8. Determine the transfer function of the following diagram.



a) Check your result with MATLAB.

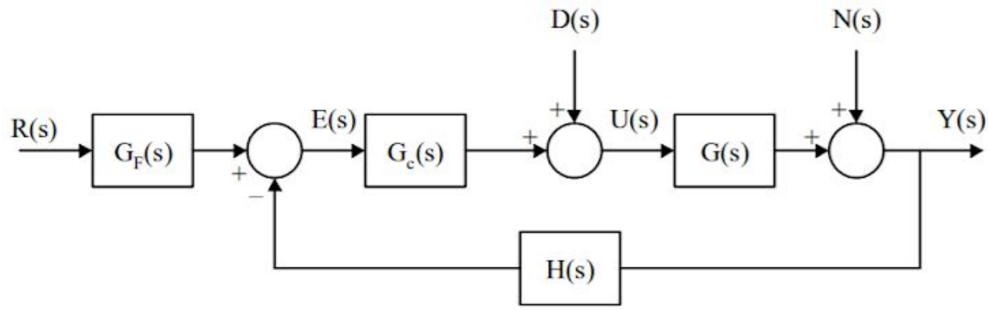
```
              50 s^2
  ----------------------------------
  2 s^5 + 3 s^4 + 201 s^3 + 150 s^2

Continuous-time transfer function.
```

b) Plot y(t) for a unit-impulse input r(t) with MATLAB.



The graph above shows the impulse response of the system, the transfer function is used in conjunction with the impulse() function.

9. Determine the total output Y(s) for the following system.



$$\mathbf{Y(s)} = \frac{R(S)\ G_f(S)G_c(S)G(S) + D(S)G_c(S)G(S) + N(S)}{1 + G_c(S)G(S)H(S)}$$

The equation above is the transfer function calculated through Matlab using the feedback() function as well as accounting for the disturbance within the system.

Conclusion

Therefore, the purpose of this lab was to utilize MATLAB to compute the transfer functions. The transfer function of a system can be found through hand calculations. However, by using the MATLAB software, the transfer function of a system can be obtained by utilizing the appropriate function in the code, such as sys=parallel(), or sys=series(). After finding the transfer functions, it was then easier to display a graphical representation of the system. All procedures were carried out successfully.

Vaughn College of Aeronautics and Technology

Control Systems-ELE350L

Professor Bustamante

Lab #5

By: Alyssa Mitchell and Omomhene Eimunjeze
Room of Experiment: W148
Date:4/3/19

Objective

The purpose of this lab was to show how transfer functions can be entered and solved within the MATLAB software. The transfer function of a system can be found through hand calculations. However, by using the MATLAB software, the transfer function of a system can be obtained by utilizing the appropriate function in the code, such as sys=parallel(), or sys=series().

Equipment/Materials

- MATLAB Software
- Simulink

Results

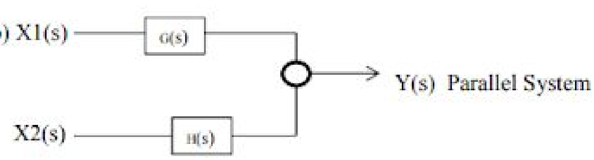
LAB SECTION:

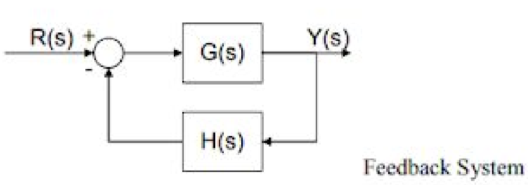
Using the Following Functions $G(S) = \frac{s+3}{s^2+7s+15}$ and $H(s) = \frac{s}{s+4}$ Explain with respect to the natural, damping ratio and forced response questions 1 and 2 for each System (Series, Parallel, Open Loop and Closed Loop (Only for part C)).

Question 1

1. Enter the G(s) and H(s) functions. (Take advantage of using either symbolic tool or entering vector format with Commands like `tf` to generate the transfer function.) Your goal is to find the following first by hand using the skills learned during lecture and then using Matlab/Simulink:

a) 

b) 



c)

(Hint: Use commands like series(tf), parallel(tf) and feedback(tf))

```
disp('Transfer function 1');
G = tf([0 1 3],[1 7 15])

disp('Transfer function 2');
H = tf([0 1 0],[0 1 4])

disp('Cascade system');
cascaded_sys = series(G,H)

disp('Parallel system');
parallel_sys = parallel(G,H)

disp('Feedback system');
feedback_sys = feedback(G,H)
```

Figure 1: Matlab code for finding the transfer function, Cascade, Parallel and Feedbacks system of the G(s) and H(s) functions

```
Transfer function 1

G =

      s + 3
  --------------
  s^2 + 7 s + 15

Continuous-time transfer function.

Transfer function 2

H =

    s
  -----
  s + 4

Continuous-time transfer function.

Cascade system

cascaded_sys =

         s^2 + 3 s
  ------------------------
  s^3 + 11 s^2 + 43 s + 60

Continuous-time transfer function.

Parallel system

parallel_sys =

  s^3 + 8 s^2 + 22 s + 12
  ------------------------
  s^3 + 11 s^2 + 43 s + 60

Continuous-time transfer function.

Feedback system

feedback_sys =

       s^2 + 7 s + 12
  ------------------------
  s^3 + 12 s^2 + 46 s + 60

Continuous-time transfer function.
```

Figure 2: Transfer function, Cascade, Parallel and Feedback systems for the G(s) and H(s) functions

Question 2

2. Use the results from question 1, Test your TF functions (1a and 1b are Open Loop TF, 1c is a Closed Loop TF also find the Open Loop TF for (c) using the impulse and step forced responses. Explain your results obtain above in terms of Percent overshoot %OS, settling time T_s, steady state SS, Natural Frequency ω_n and Damping Ration ξ. Also explain base on your poles and zeros the behavior of your TF. Is there any difference between OpenLoop (OL) and ClosedLoop (CL) TF? Explain. What happens to the poles and Zeros of OL and CL TF?

```
Cascade_sys =

           s^2 + 3 s
  ------------------------
  s^3 + 11 s^2 + 43 s + 60

Continuous-time transfer function.


ans =

        RiseTime: 0
    SettlingTime: 1.2182
     SettlingMin: -7.8933e-04
     SettlingMax: 0.0891
       Overshoot: Inf
      Undershoot: Inf
            Peak: 0.0891
        PeakTime: 0.2368
```

Figure 3: Result of the stepinfo of the Cascade system

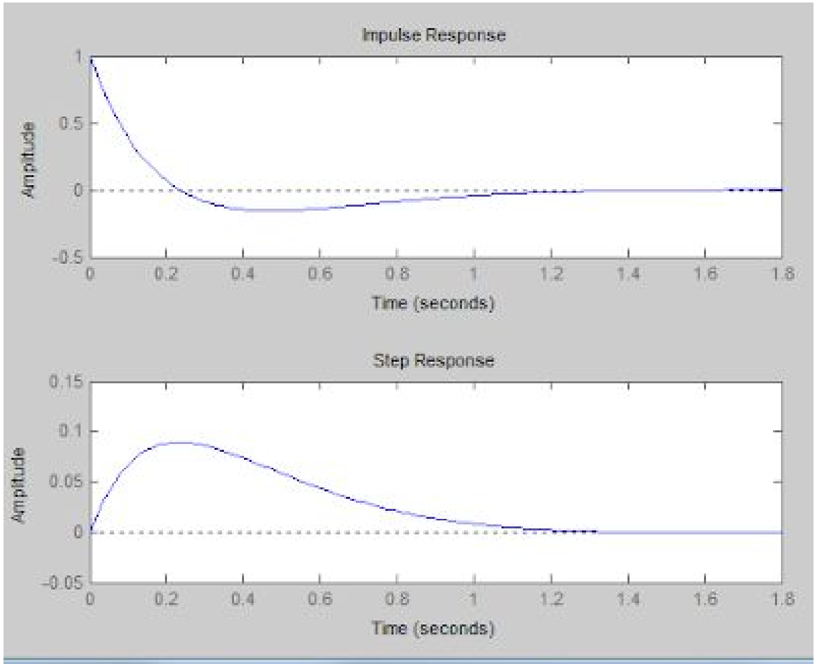


Figure 4: Impulse and step responses for the Cascade system

```
Parallel_sys =

  s^3 + 8 s^2 + 22 s + 12
  -----------------------
  s^3 + 11 s^2 + 43 s + 60

Continuous-time transfer function.


ans =

        RiseTime: 0.6252
    SettlingTime: 1.1182
     SettlingMin: 0.2003
     SettlingMax: 0.2788
       Overshoot: 400.0000
      Undershoot: 0
            Peak: 1
        PeakTime: 0
```

Figure 5: Result of the stepinfo of the Parallel system

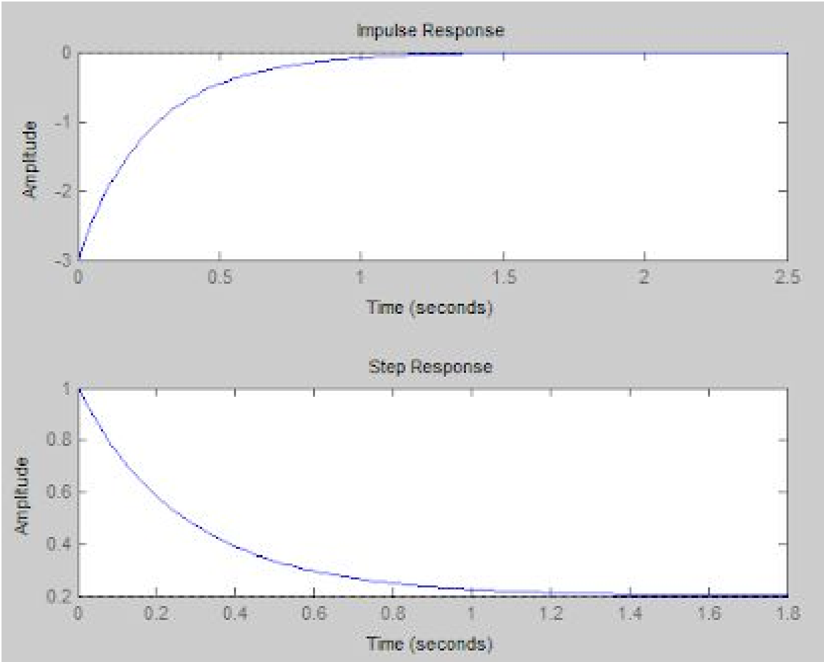


Figure 6: Impulse and step responses for the Parallel system

```
Feedback_sys =

        s^2 + 7 s + 12
  ------------------------
  s^3 + 12 s^2 + 46 s + 60

Continuous-time transfer function.


ans =

        RiseTime: 0.4183
    SettlingTime: 0.6695
     SettlingMin: 0.1807
     SettlingMax: 0.2015
       Overshoot: 0.7500
      Undershoot: 0
            Peak: 0.2015
        PeakTime: 1.1513
```

Figure 7: Result of the stepinfo of the Feedback system

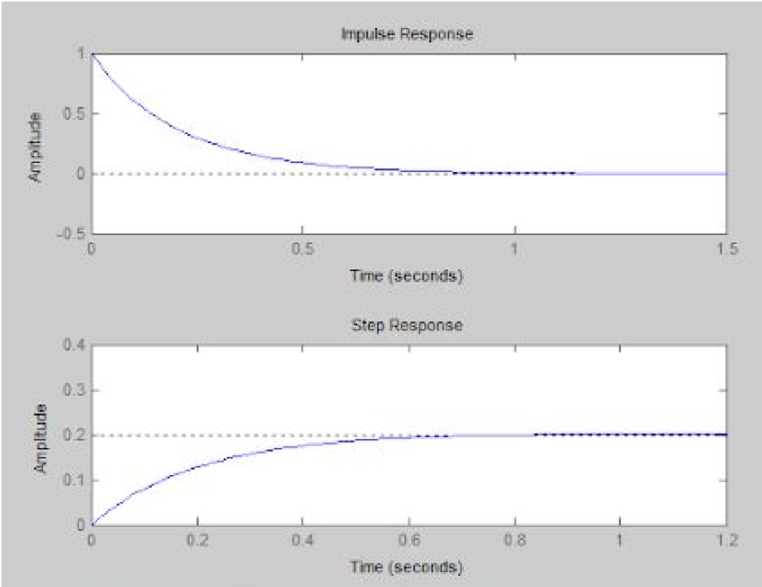


Figure 8: Impulse and step responses for the Feedback system

Question 3

3. As we did in class do only for (c) The following: Kv, Ka and Kp using: The Step, Ramp and Parabolic input functions. Explain your results obtain in terms of Percent overshoot %OS, settling time T_s, steady state SS, Natural Frequency ω_n and Damping Ration ξ.

```
s=tf('s');
G=tf([1 3],[1 7 15])
H=tf([1 0],[1 4])
sys1=feedback(G,H)
step(sys1)
S1 = stepinfo(sys1,'RiseTimeLimits',[0.05,0.95])
Kp=dcgain(sys1)
sgf=tf([1 0],1)*sys1;
sgf=minreal(sgf)
Kv=dcgain(sgf)
ssgf=tf([1 0],1)*sgf;
ssgf=minreal(ssgf)
Ka=dcgain(ssgf)
```

Figure 9: Matlab code for finding Kv Ka and Kp of the feedback system

```
G =

      s + 3
  --------------
  s^2 + 7 s + 15

Continuous-time transfer function.


H =

    s
  -----
  s + 4

Continuous-time transfer function.


sys1 =

       s^2 + 7 s + 12
  ------------------------
  s^3 + 12 s^2 + 46 s + 60

Continuous-time transfer function.


S1 =

        RiseTime: 0.5400
    SettlingTime: 0.6695
     SettlingMin: 0.1902
     SettlingMax: 0.2015
       Overshoot: 0.7500
      Undershoot: 0
            Peak: 0.2015
```

```
          Peak: 0.2015
      PeakTime: 1.1513


Kp =

    0.2000


sgf =

     s^3 + 7 s^2 + 12 s
  ------------------------
  s^3 + 12 s^2 + 46 s + 60

Continuous-time transfer function.


Kv =

     0


ssgf =

    s^4 + 7 s^3 + 12 s^2
  ------------------------
  s^3 + 12 s^2 + 46 s + 60

Continuous-time transfer function.


Ka =

     0
```

Figure 10: Transfer function,Kp, Kv and Ka values for the feedback system

Vaughn College of Aeronautics and Technology

Control Systems-ELE350L

Professor Bustamante

Lab #6

By: Alyssa Mitchell and Omomhene Eimunjeze
Room of Experiment: W148
Date:4/24/19

Objective

The objective of this lab is to utilize Matlab in order to construct a root locus. By obtaining the response of the system, the root locus can then be gathered from this information.

Question 1

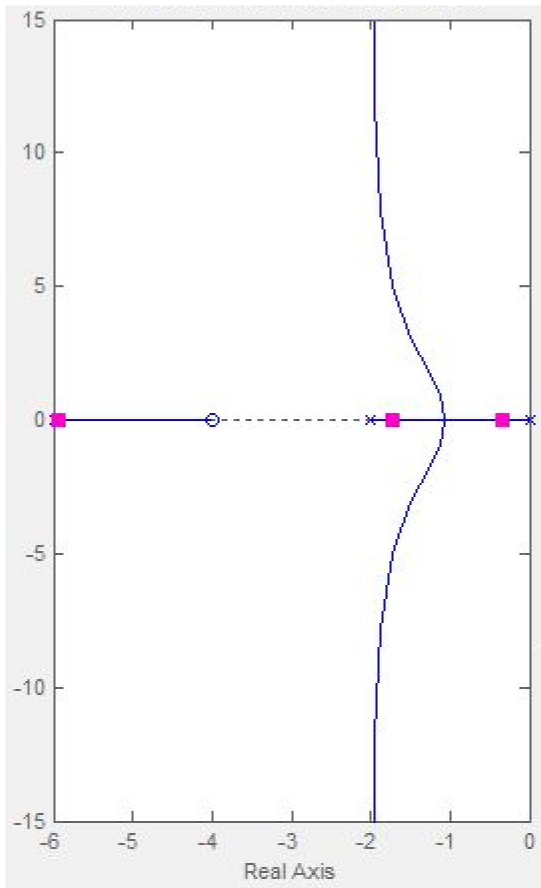


Figure 1: The root locus of the system.

The poles are represented by x's and the zeros are represented by o's. The x's and o's are located on the left-hand side of the graph, indicating that the system is stable. The sisotool() on Matlab is utilized in order to construct this root locus.

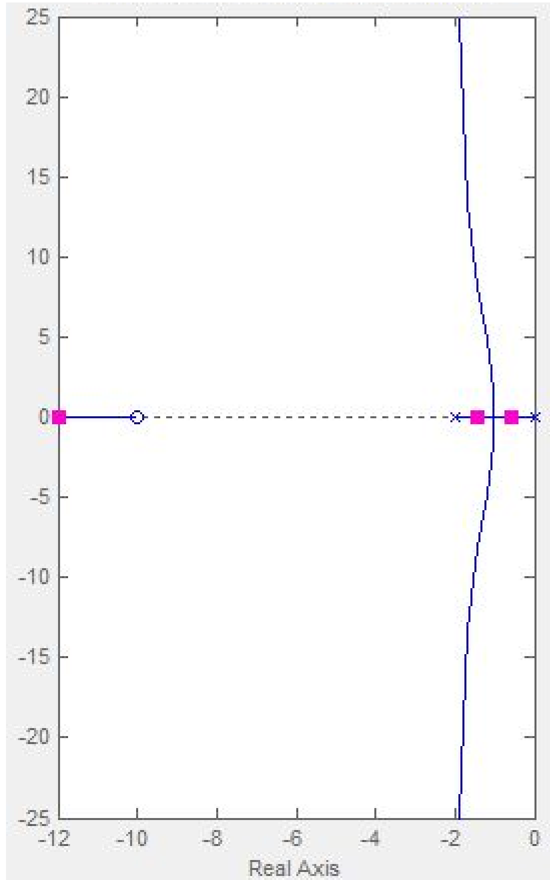


Figure 2: The root locus of the system.

This is another one of the two possibilities for the root locus of a unity negative-feedback system with the open-loop configuration

Question 2

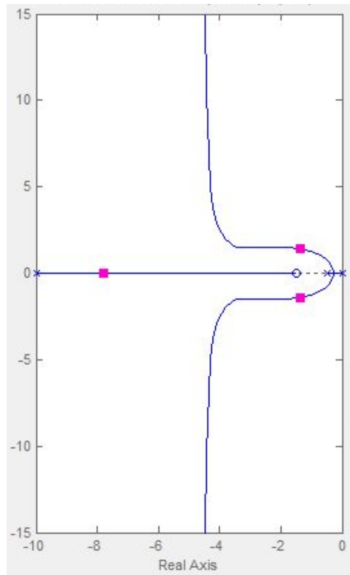
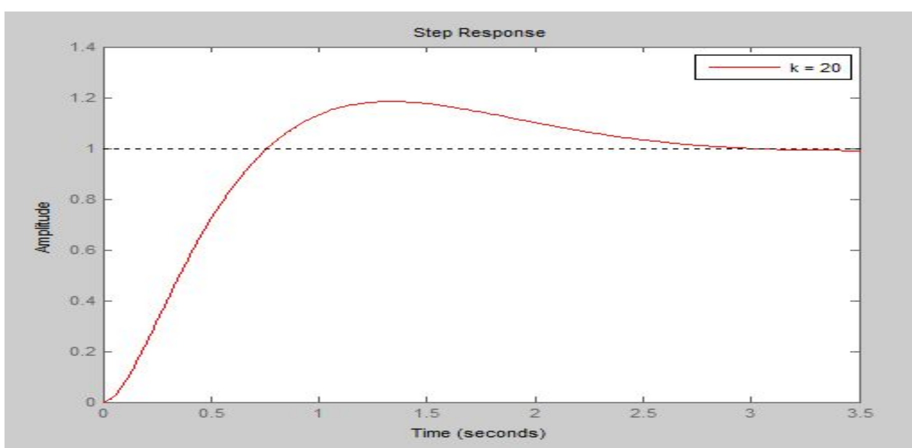


Figure 3: The root locus of the response when gain(k)=20

```
    RiseTime: 0.5314
SettlingTime: 2.6390
 SettlingMin: 0.9010
 SettlingMax: 1.1841
   Overshoot: 18.4064
  Undershoot: 0
        Peak: 1.1841
    PeakTime: 1.3372
```



The step response is shown above as well as the values for important aspects such as rise time, the sisotool() function was utilized for this analysis. Ts= 2.63 seconds.

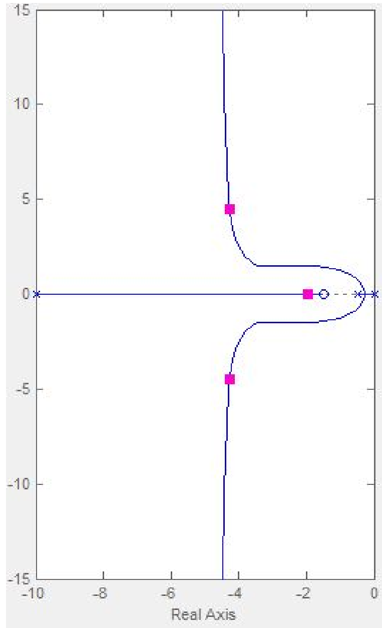
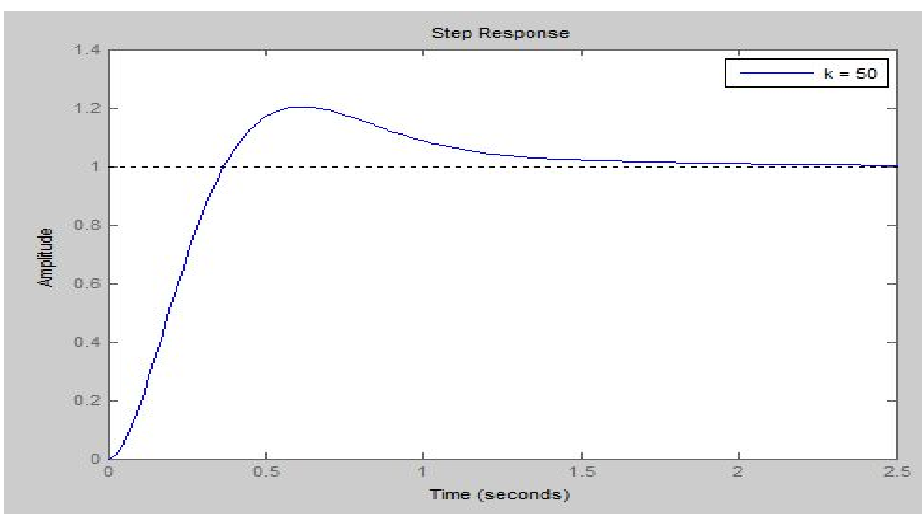


Figure 4: Gain(k)= 50

```
    RiseTime: 0.2522
SettlingTime: 1.5399
 SettlingMin: 0.9031
 SettlingMax: 1.2037
   Overshoot: 20.3710
  Undershoot: 0
        Peak: 1.2037
    PeakTime: 0.6040
```



The step response is shown above as well as the values for important aspects such as rise time, the sisotool() function was utilized for this analysis. Ts= 1.5 seconds.

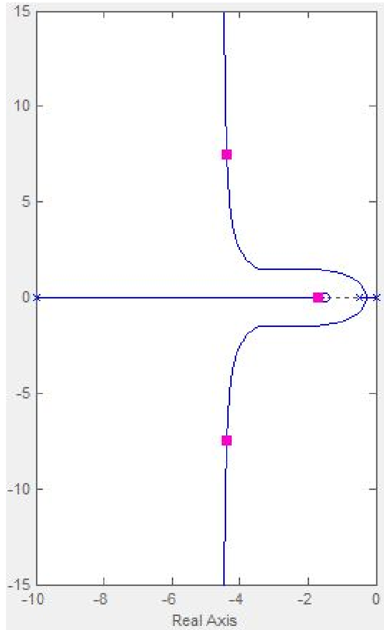
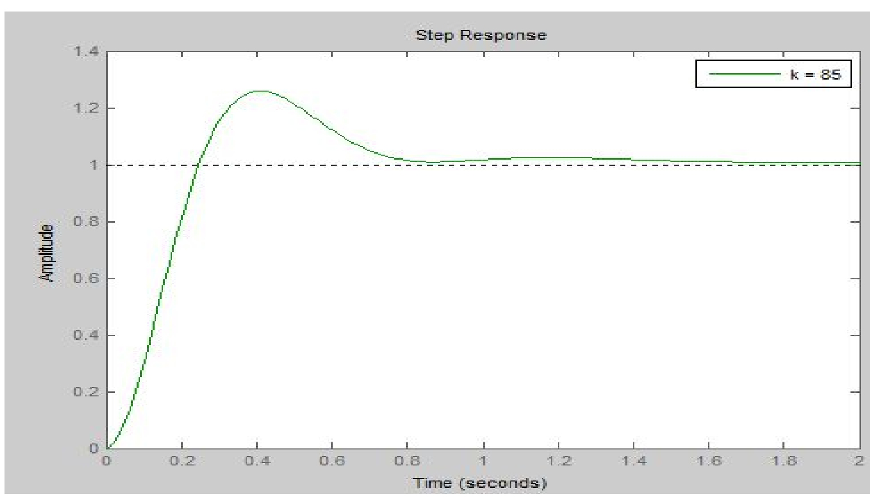


Figure 5: Gain(k)= 85

```
      RiseTime: 0.1695
  SettlingTime: 1.3368
   SettlingMin: 0.9362
   SettlingMax: 1.2593
     Overshoot: 25.9262
    Undershoot: 0
          Peak: 1.2593
      PeakTime: 0.4186
```



The step response is shown above as well as the values for important aspects such as rise time, the sisotool() function was utilized for this analysis. Ts= 1.33 seconds.

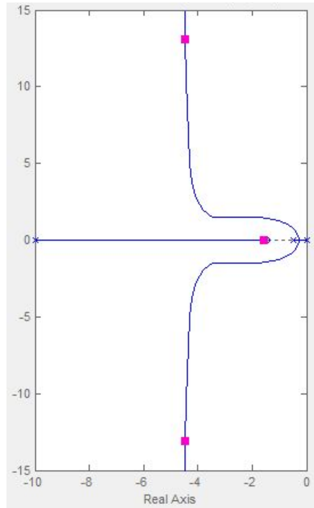
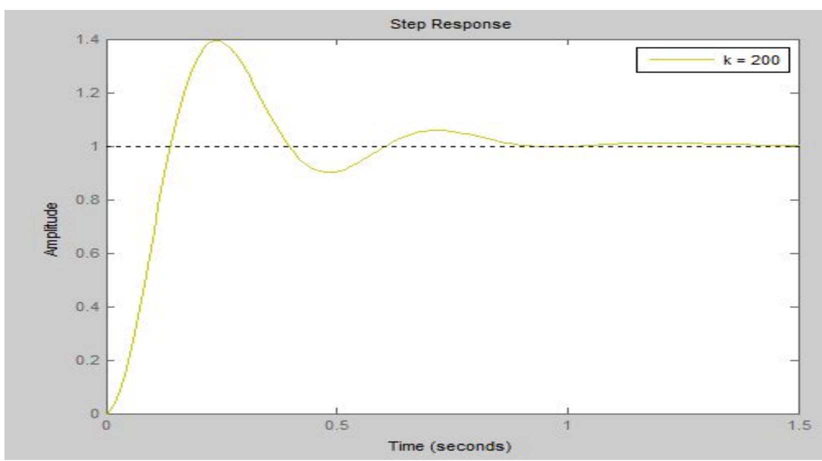


Figure 5: Gain(k)= 200

```
     RiseTime: 0.0943
 SettlingTime: 0.8453
  SettlingMin: 0.9008
  SettlingMax: 1.3938
    Overshoot: 39.3819
   Undershoot: 0
         Peak: 1.3938
     PeakTime: 0.2372
```



The step response is shown above as well as the values for important aspects such as rise time, the sisotool() function was utilized for this analysis. Ts= 0.8453 seconds.

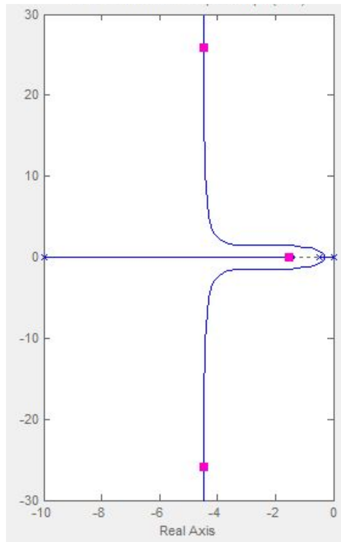
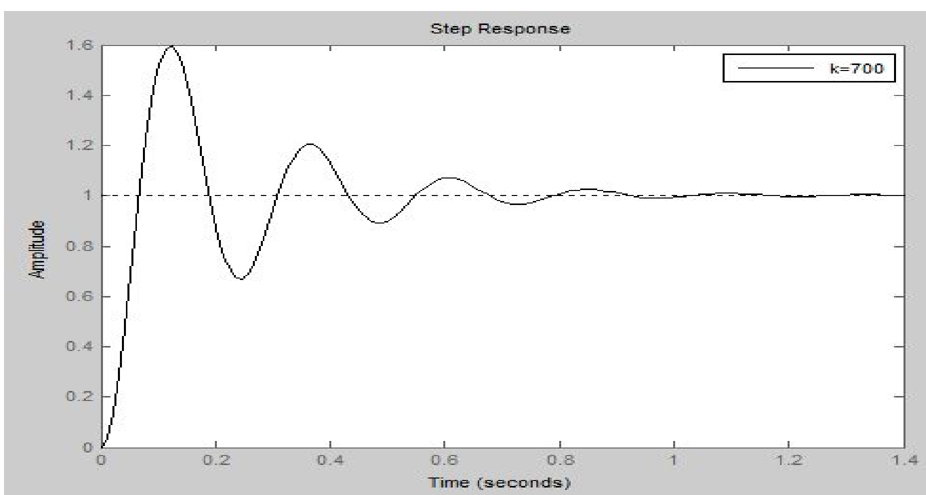


Figure 6: Gain(k)= 700

```
     RiseTime: 0.0448
 SettlingTime: 0.8776
  SettlingMin: 0.6695
  SettlingMax: 1.5973
    Overshoot: 59.7344
   Undershoot: 0
         Peak: 1.5973
     PeakTime: 0.1231
```



The step response is shown above as well as the values for important aspects such as rise time, the sisotool() function was utilized for this analysis. Ts= 0.8776 seconds.

Gain (K), %OS	Settling Time (seconds)
K=20, 18.4%	2.64 seconds
K=50, 20.4%	1.54 seconds
K=85, 25.9%	1.34 seconds
K=200, 39.4%	0.845 seconds
K=700, 59.7%	0.878 seconds

Table 1: Relationship between gain, %os, and settling time.

Question 3

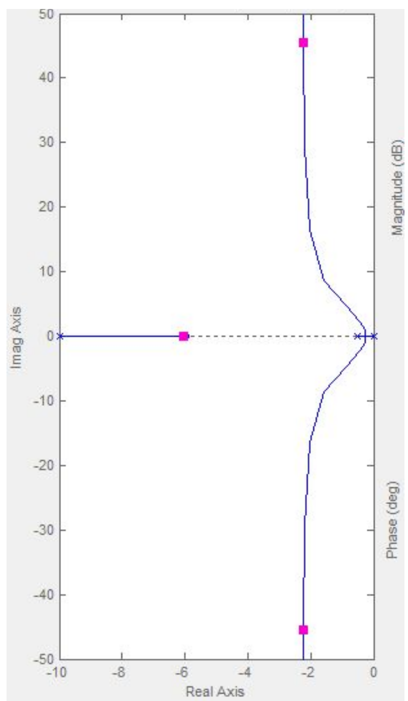


Figure 7: The system as printed by the root locus for the zero at -6.

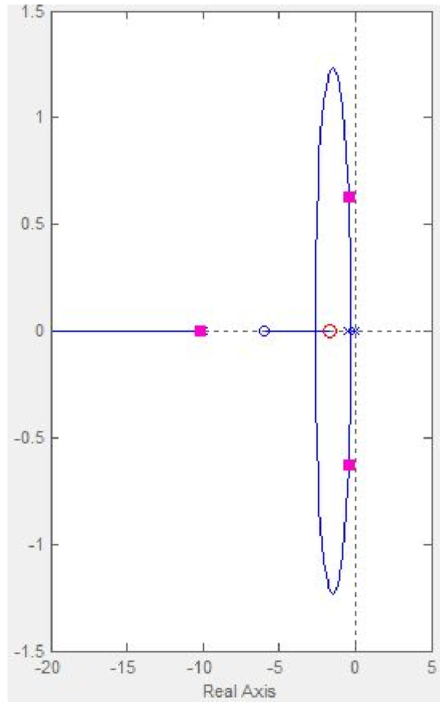


Figure 8: When there are zeros at x=-6 and x=-2.

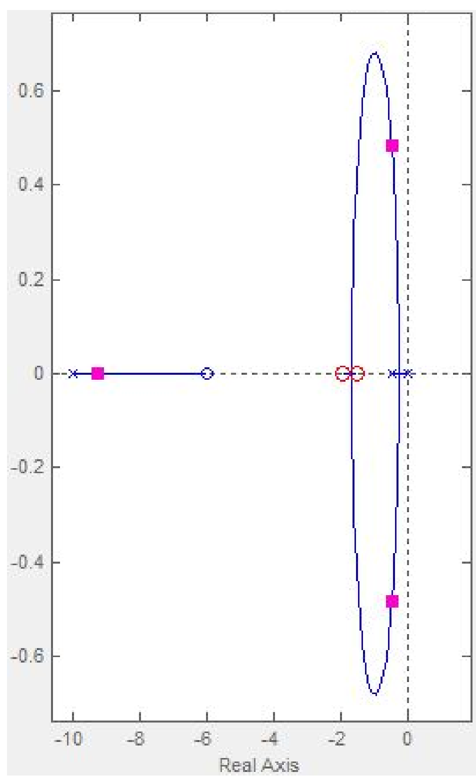


Figure 9: When there are zeros at x=-6 and x=-2, and x=-1.5.

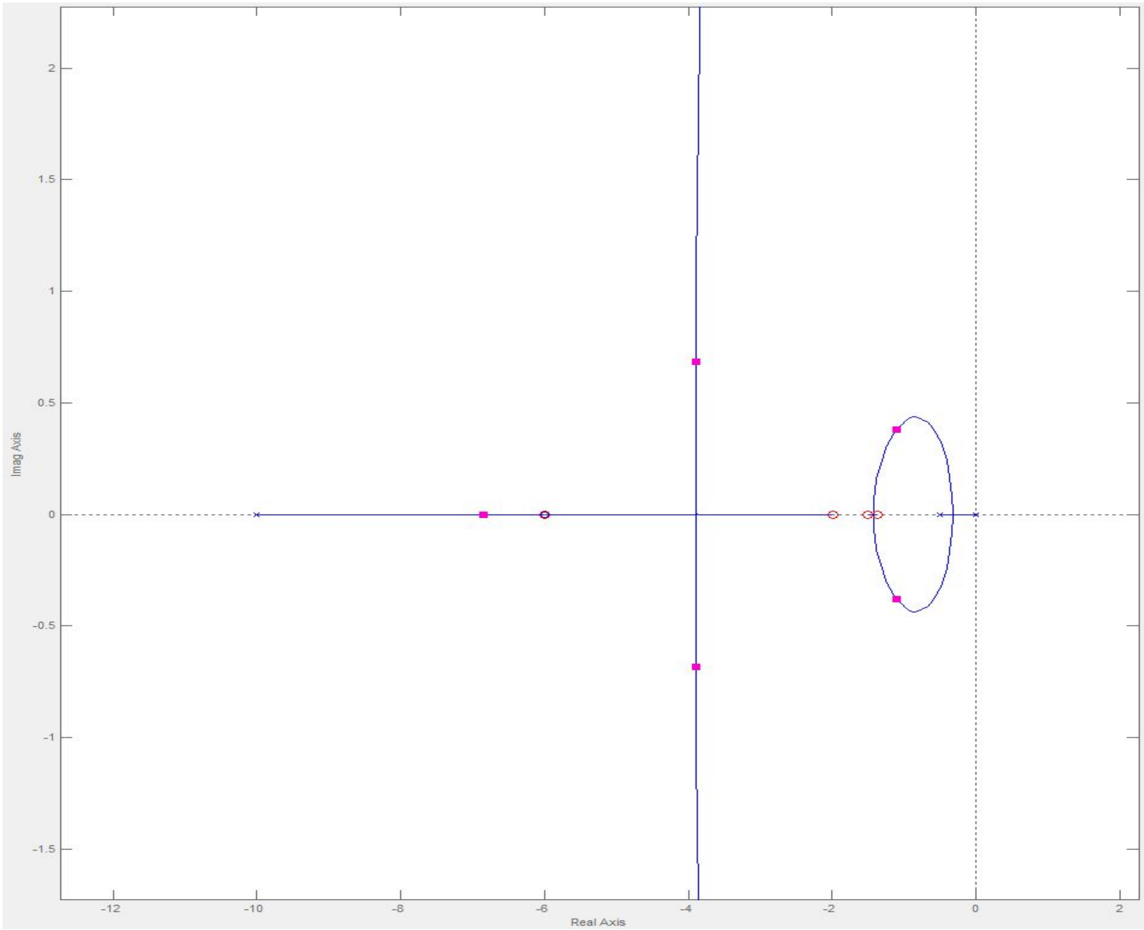


Figure 10: When there are zeros at x=-6 and x=-2, x=-1.5, and x=-1.37.

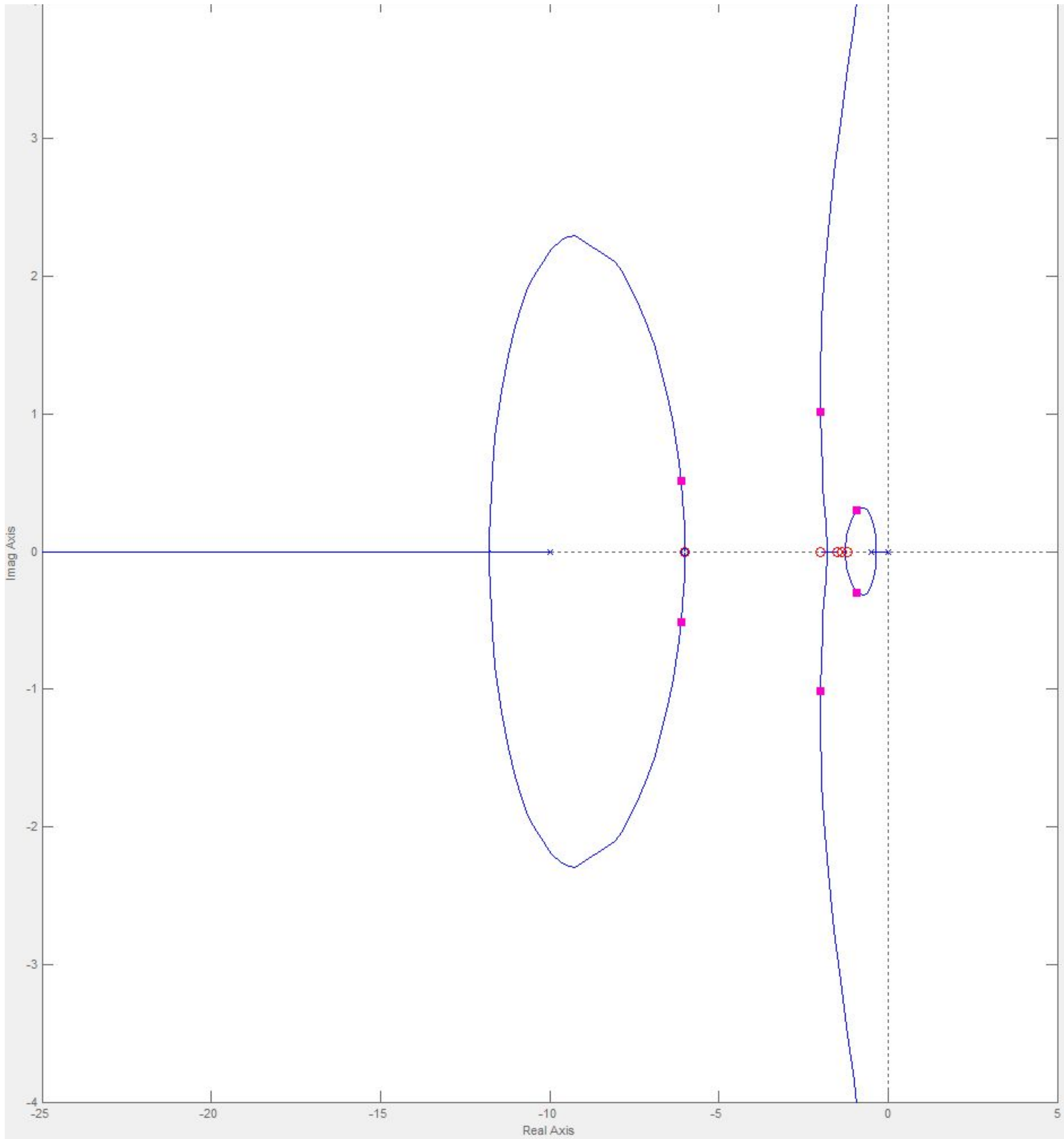


Figure 11: When there are zeros at x=-6 and x=-2, x=-1.5, x=-1.37, and x=-1.2.

Conclusion

Therefore, the purpose of this lab was to use Matlab to solve for the root locus using various gain values. The gain values are represented by k. It was clear that the overall output of the root locus changed after the gain value was changed. The settling time was affected as it decreased as the gain value increased.