RYERSON UNIVERSITY

Faculty of Engineering, Architecture and Science

Department of Electrical & Computer Engineering
Program: Biomedical Engineering

Course Number	**BME 639**
Course Title	Control Systems and Bio-robotics
Semester/Year	Winter 2020
Instructor	Dr. Saba Sedghizadeh

Lab/Tutorial Report NO.	2

Report Title	Transfer Function Modeling of Physical Systems and Control Introduction to Lead/Lag Compensator

Submission Date	March 27, 2020
Due Date	March 27, 2020

Name	Student ID	Signature*
Kushal Shah	xxxx43903	K.S.
[redacted]	[redacted]	[redacted]

(Note: remove the first 4 digits from your student ID)

**By signing above you attest that you have contributed to this submission and confirm that all work you have contributed to this submission is your own work. Any suspicion of copying or plagiarism in this work <u>will result in an investigation of Academic Misconduct and may result in a "0" on the work,</u> an "F" in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at:* http://www.ryerson.ca/senate/policies/pol60.pdf.

Part 1:

A1:

A1) $J_{eq}\omega_l'(t) + B_{eq}\omega_l(t) = A_m V_m(t)$

$\rightarrow J_{eq}y'(t) + B_{eq}y(t) = A_m x(t)$

let $\left.\begin{aligned} V_m(t) &= x(t) \\ \omega_l(t) &= y(t) \\ \Rightarrow \omega_l'(t) &= y'(t) \end{aligned}\right\} \mathcal{L} \begin{cases} V_m(s) = X(s) \\ \Omega(s) = Y(s) \\ \Omega'(s) = sY(s) \end{cases}$

$\mathcal{L}$aplace

$$J_{eq}sY(s) + B_{eq}Y(s) = A_m X(s)$$

$$Y(s)\left[J_{eq}s + B_{eq}\right] = A_m X(s)$$

$$\frac{Y(s)}{X(s)} = \frac{A_m}{J_{eq}s + B_{eq}}$$

$$\therefore \frac{\Omega(s)}{V_m(s)} = \frac{A_m}{J_{eq}s + B_{eq}}$$

Table 1. Characteristics of the speed-to-voltage transfer function.

	$\frac{\Omega(s)}{V(s)}$	System Order	System Type	DC Gain (k)	Time Constant (τ)
Voltage-to-speed	$\frac{Am}{Jeqs + Beq}$	First order	Type 0	$\frac{Am}{Beq}$	$\frac{Jeq}{Beq}$

A2:

A2) Same process as above

from lecture slides

$$\therefore \frac{\Omega(s)}{V_m(s)} = \frac{A_m}{J_{eq}s + B_{eq}}, \text{ where } \Omega(s) = s\Theta(s)$$

$$\Rightarrow \frac{\Theta(s)}{V_m(s)} = \frac{A_m}{J_{eq}s^2 + B_{eq}s}$$

Table 2. Characteristics of the position-to-voltage transfer function.

	$\frac{\Theta(s)}{V(s)}$	System Order	System Type	Poles	Zeros
Voltage-to-position	Am / $Jeqs^2$+Beqs	Second order	Type 1	$\frac{Jeq}{Beq}$, 0	2 at infinity

A3:

A3) $J_{eq} = \eta_g k_g^2 J_m + J_e$

$= (0.90)(14\times5)^2(461\times10^{-7}) + (1.03\times10^{-4})$

$= 2.136\times10^{-3}$

$$B_{eq} = \frac{\eta_g k_g^2 \eta_m k_t k_m + B_{eq1} R_m}{R_m}$$

$B_{eq1} = \eta_g k_g^2 B_m + B_e$

$= 0.015$

$$= \frac{(0.90)(14\times5)^2(0.69)(7.68\times10^{-3})(7.68\times10^{-3}) + (0.015)(2.6)}{2.6}$$

$= 0.084$

$$A_m = \frac{\eta_g k_g \eta_m k_t}{R_m}$$

$$= \frac{(0.90)(14\times5)(0.69)(7.68\times10^{-3})}{2.6}$$

$= 0.1284$

Dc gain $= k = \frac{A_m}{B_{eq}} = \frac{0.1284}{0.084} = 1.53$

$\tau = \frac{J_{eq}}{B_{eq}} = \frac{2.136\times10^{-3}}{0.084} = 0.025$

A4:

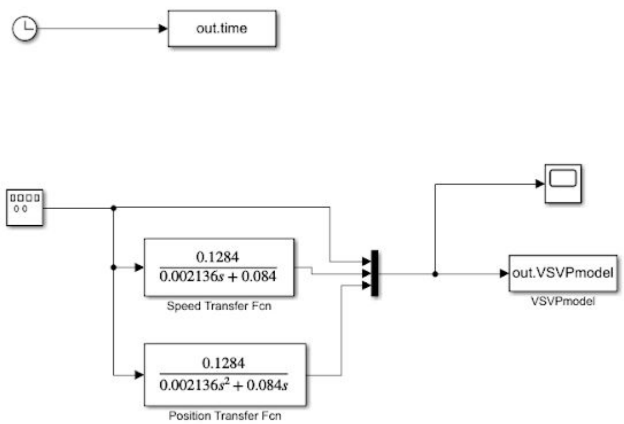

Figure 1. Block diagram for voltage-to-speed and voltage-to-position transfer functions.

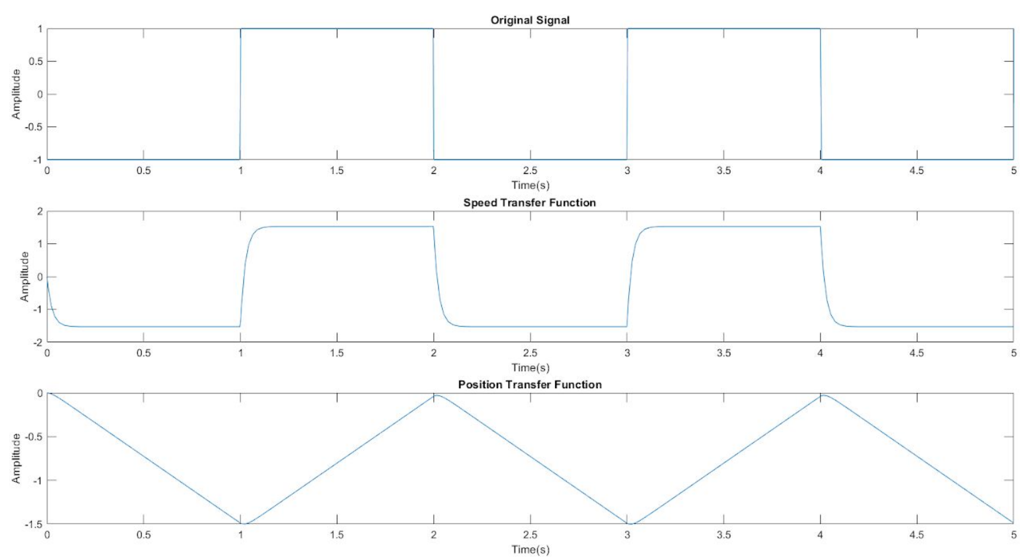

Figure 2. Plot of input signal and output signals of the speed-to-time and position-to-time block diagram in Figure 1. Input signal is a square wave with amplitude of 1 and frequency of 0.5Hz.

Quickly looking at Figure 2, we can see the characteristics of the plots. Basic calculus tells us that velocity, in this case speed, is the derivative of position. Therefore, when the position signal has a negative slope the speed signal should be at a constant negative value, and when the position signal has a positive slope the speed signal will be at a constant positive value. Knowing this, we can say that our plots do follow this characteristic. With the speed signal being at 1 when the position signal has a slope of 1 and the speed signal has a value of -1 when the position signal has a slope of -1.

B1:

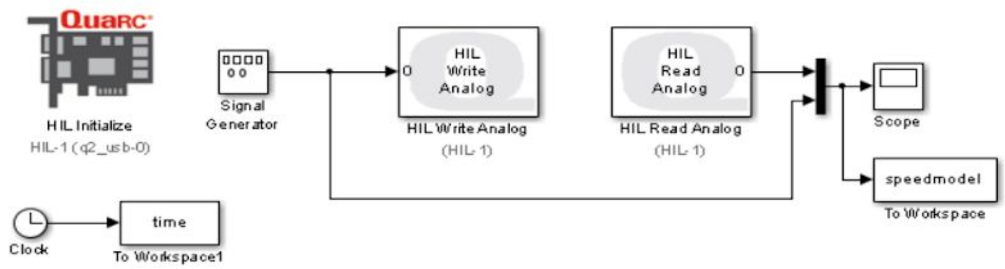

Figure 3. Block diagram of speed model using QUARC and external motor.

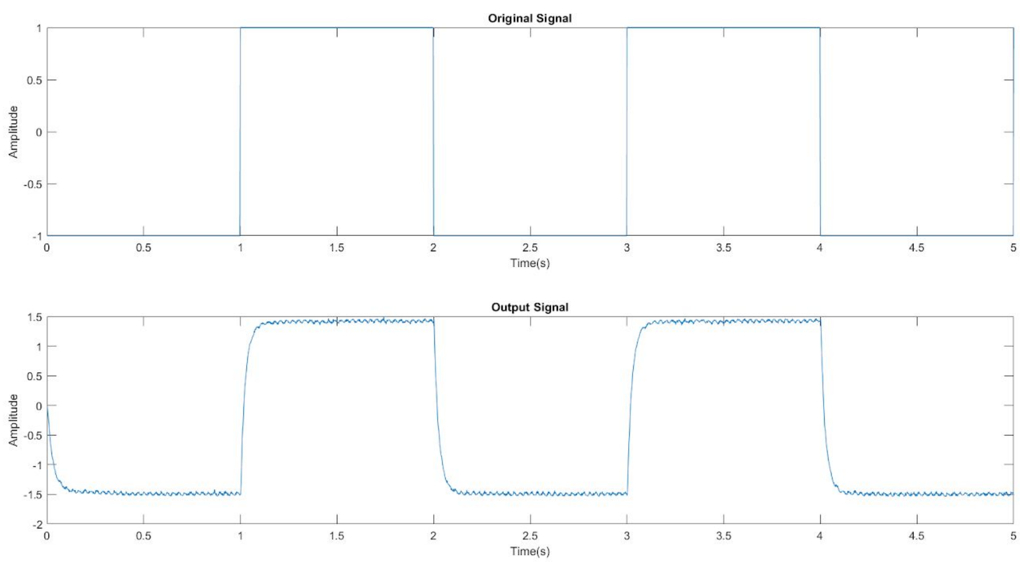

Figure 4. Plot of input and output signal of the system in Figure 3. Input signal is a square wave with amplitude of 1 and frequency of 0.5Hz.

B2:

B2) $\frac{\Omega(s)}{V_m(s)} = \frac{A_m}{J_{eq}s + B_{eq}}$

$\boxed{= \frac{0.1284}{(2.136\times10^{-3})s + 0.084}}$ voltage to speed transfer function

$k = \lim_{s\to 0} s\,G(s)\frac{1}{s} = \lim_{s\to 0} G(s) = \lim_{s\to 0} \frac{0.1284}{(2.136\times10^{-3})s + 0.084} = \frac{0.1284}{0.084} \Rightarrow \boxed{k = 1.529}$

unit step function

In a first order system: $G(s) = \frac{k}{\tau s + 1}$ OR $s = -\frac{1}{\tau}$

$\therefore \frac{0.1284}{(2.136\times10^{-3})s + 0.084} \Rightarrow \frac{1.529}{0.025s + 1} \Rightarrow \boxed{\tau = 0.025}$

B3:

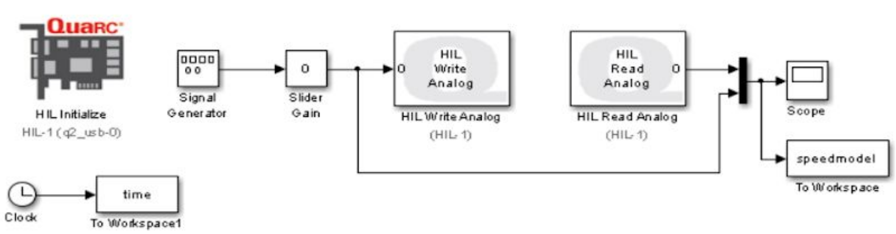

Figure 5. Block diagram of speed model using QUARC and external motor. Contains a slider gain module.

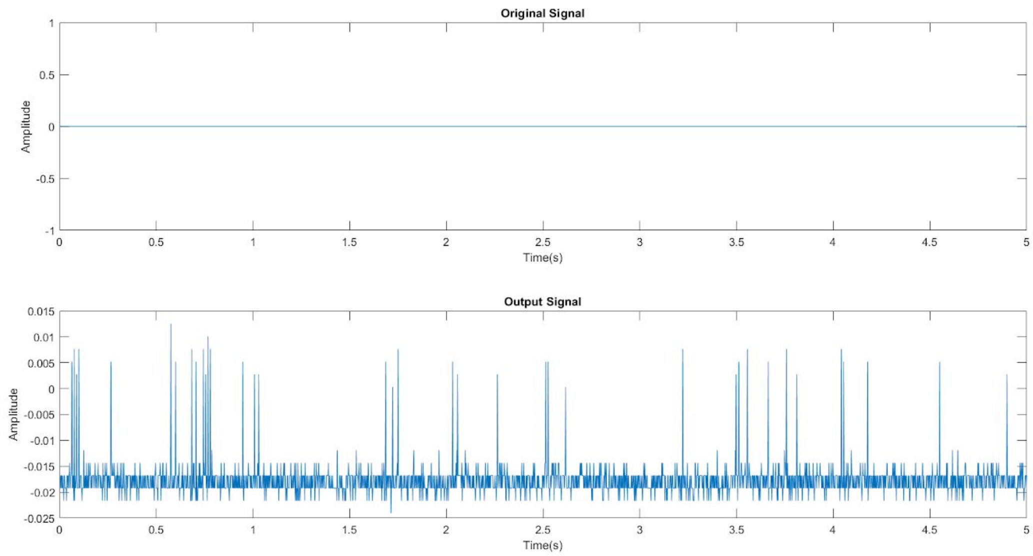

Figure 6. Plot of the input and output signal of the system in Figure 5. Input signal is a square wave with amplitude of 1 and frequency of 0.5Hz. Slider gain is 0.

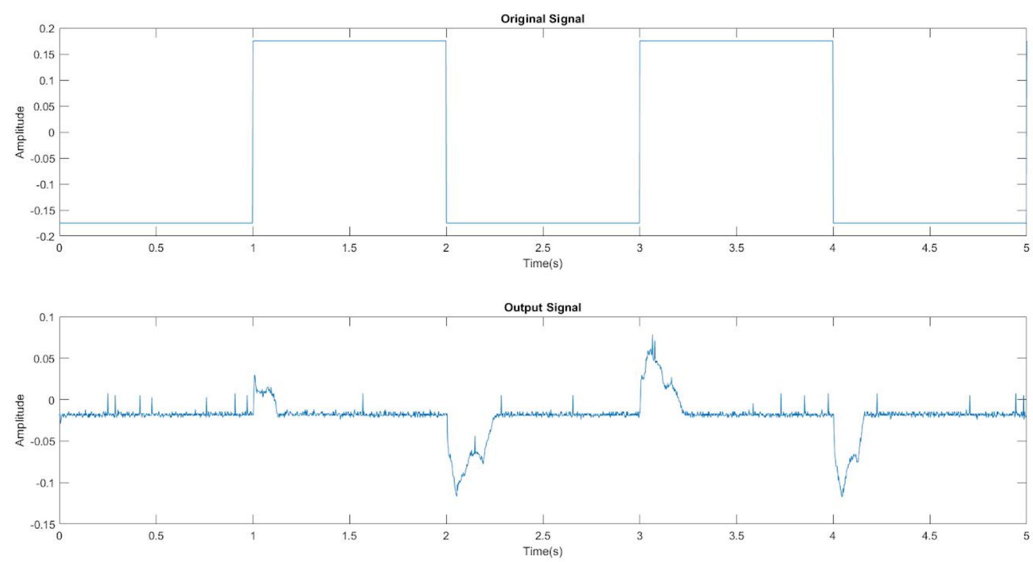

Figure 7. Plot of the input and output signal of the system in Figure 5. Input signal is a square wave with amplitude of 1 and frequency of 0.5Hz. Slider gain is 0.1751.

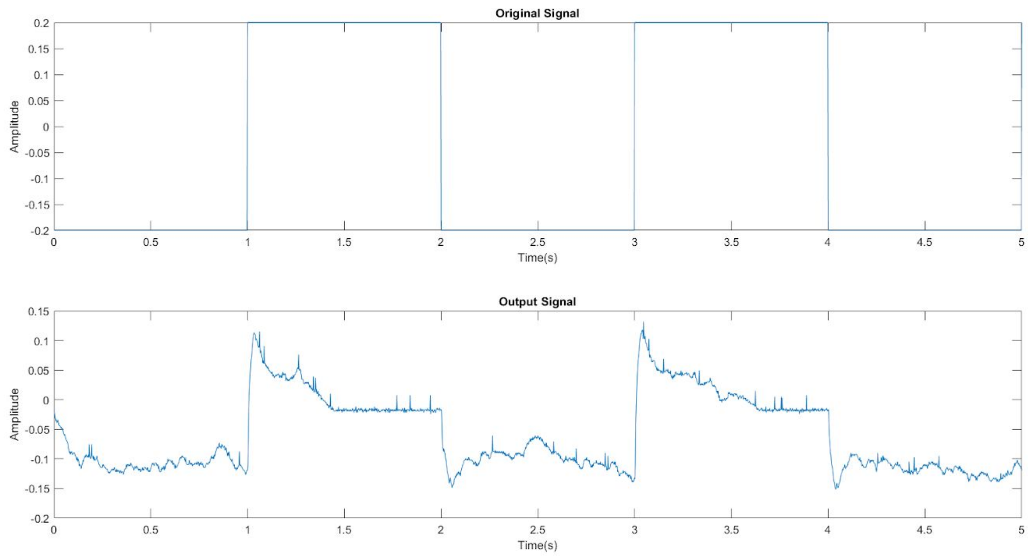

Figure 8. Plot of the input and output signal of the system in Figure 5. Input signal is a square wave with amplitude of 1 and frequency of 0.5Hz. Slider gain is 0.2.

In B3 we were to find the gain at which the motor would start to work/move. To test this we would change the gain of the gain slider and then, with our eyes, see when the slightest bit of movement started. We began at a gain of zero, which as seen in Figure 6 there was no input signal and the output was just noise. Next the slider was moved to about 0.1751 which did have an input signal and the output signal was pretty flat with the occasional dip which means we were close to the gain value which would start to move the motor. We lastly moved the gain slider to about 0.2 which finally made the motor move. In the output plot we can see that we are getting square-wave-like waveforms which correspond to the times of the input signal. Therefore, there is a dead-zone range between a gain of 0-0.2. This is very important to know for a control circuit because it tells us which voltages can be used to operate the motor. It gives us the minimum voltage in this case. Therefore, when making a control circuit for this motor, the voltage it sends to the motor has to be around 0.2V.

C1:

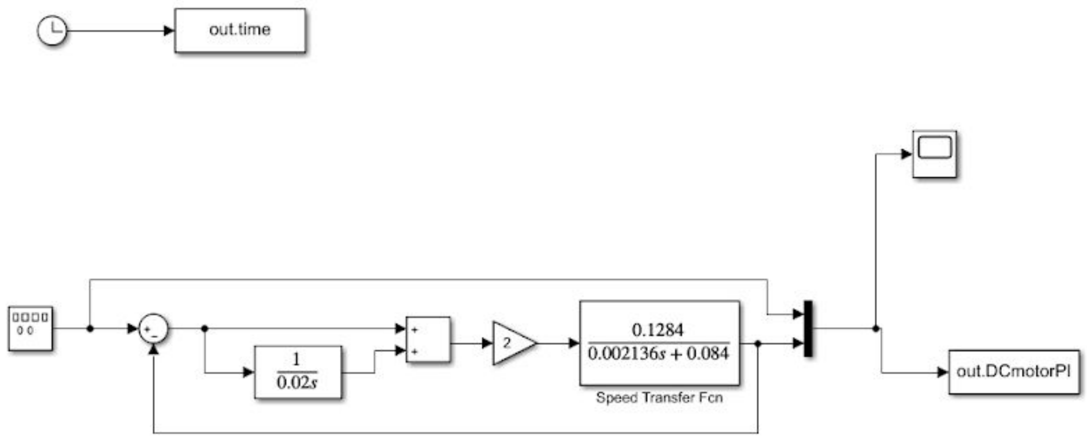

Figure 9. Block diagram of voltage-to-speed system with PI controller. The PI controller has a Ti = 0.02 and kp =2.

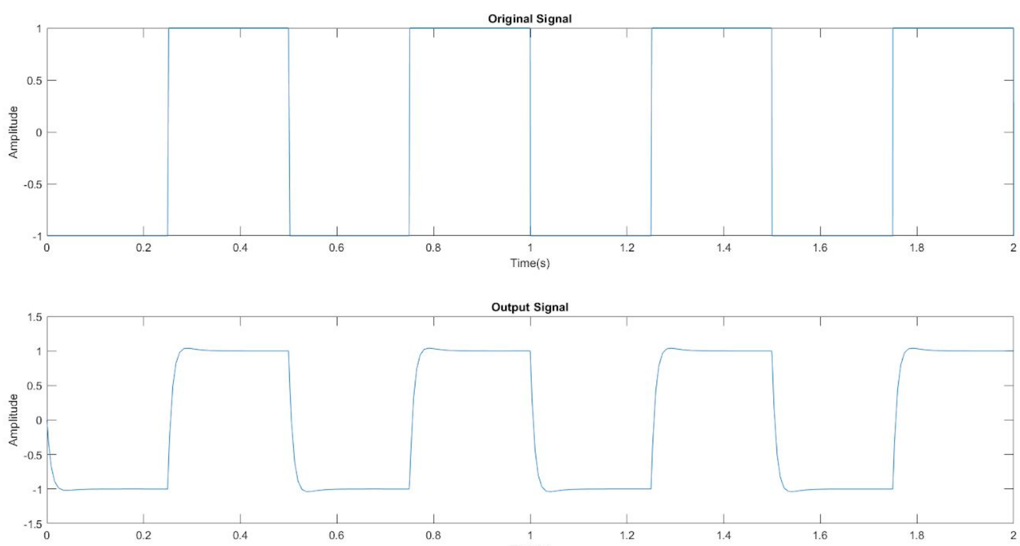

Figure 10. Plot of input and output signals for the system in Figure 9. Input signal is a square wave with amplitude of 1 and frequency of 2Hz.

C2:

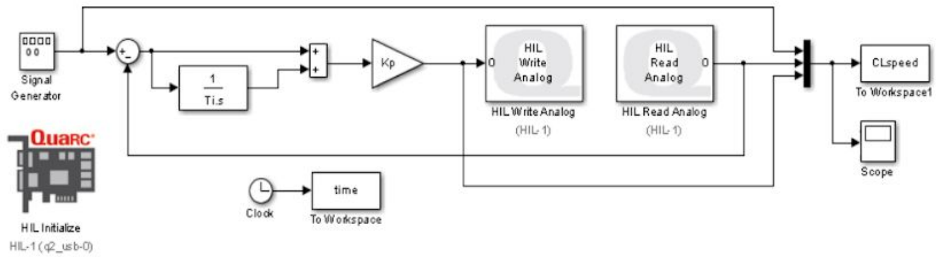

Figure 11. Block diagram of speed model with QUARC, an external motor, and PI controller. The PI controller has a Ti = 0.02 and kp =2.

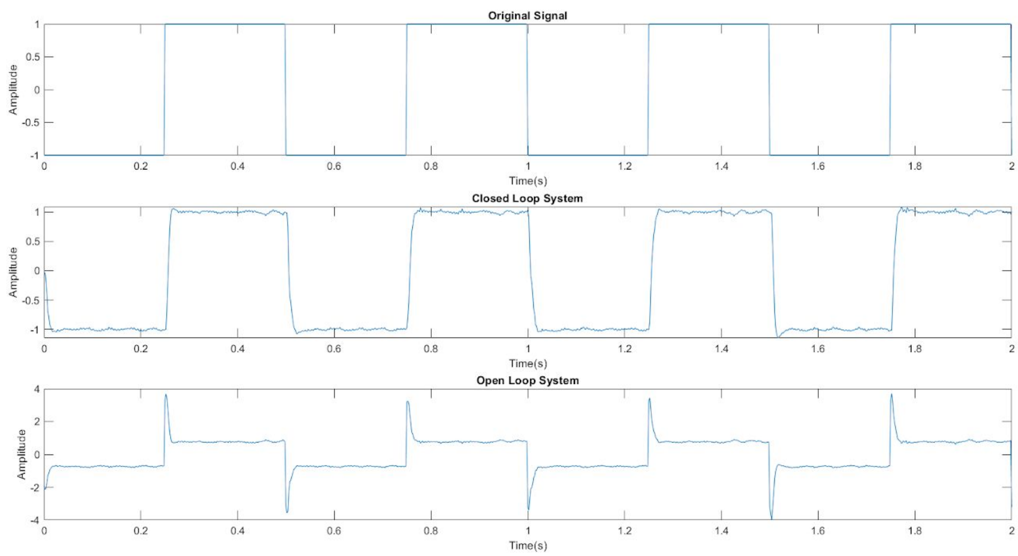

Figure 12. Plot of the input and output signals of the system in Figure 11. The input signal is a square wave with amplitude of 1 and frequency of 2Hz.

To compare the plots in C1&2 we will compare the output signal plot of C1 with the closed loop output in C2. Firstly, we can see that the signal lengths are roughly the same. Both being just longer than 0.2 seconds. Secondly, They both output signals which look similar to square waveforms. Lastly, both have amplitudes which have a max at around 1. They do not exactly hit an amplitude of 1 but they come very close. We can see that the plot in C2 seems to fluctuate at the top and bottom of the square waveform which is due to the practical nature of a physical motor. It doesn't stay 100% consistent all the time, it has very small fluctuations.

D1:

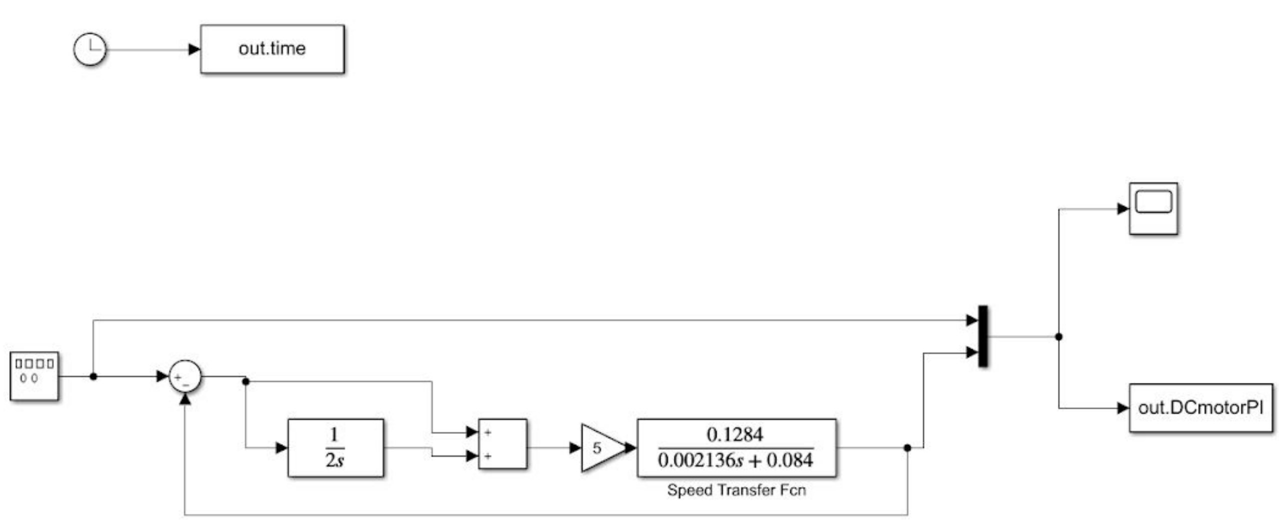

Figure 13. Block diagram of position-to-speed system with PI controller. The PI controller has a Ti = 0.02 and kp =2.

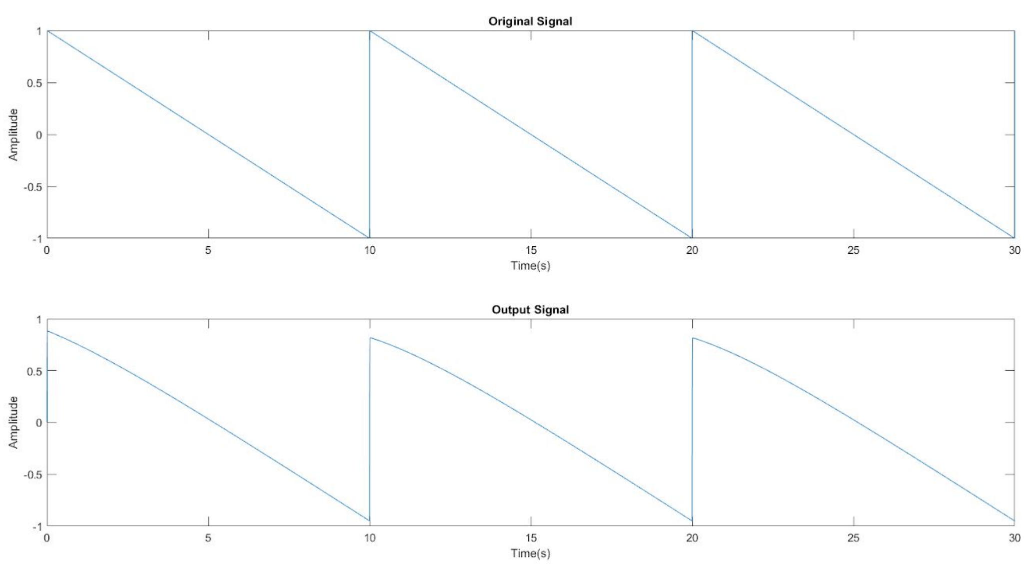

Figure 14. Plot of the input and output signals of the system in Figure 11. The input signal is a sawtooth wave with amplitude of 1 and frequency of 0.1Hz.

D2:

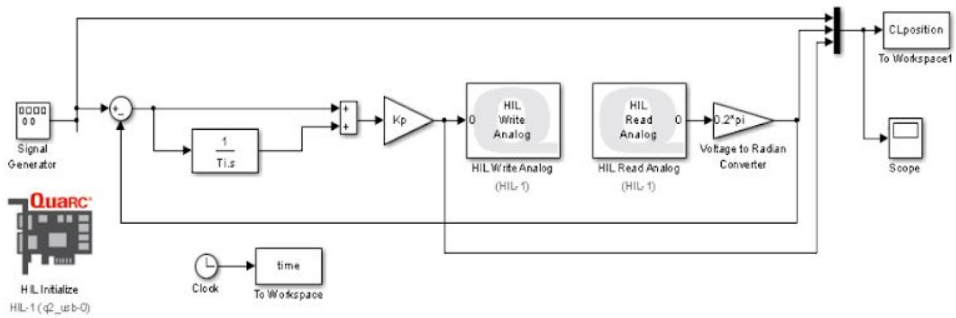

Figure 15. Block diagram of position model with QUARC, an external motor, and PI controller. The PI controller has a Ti = 2 and kp =5.

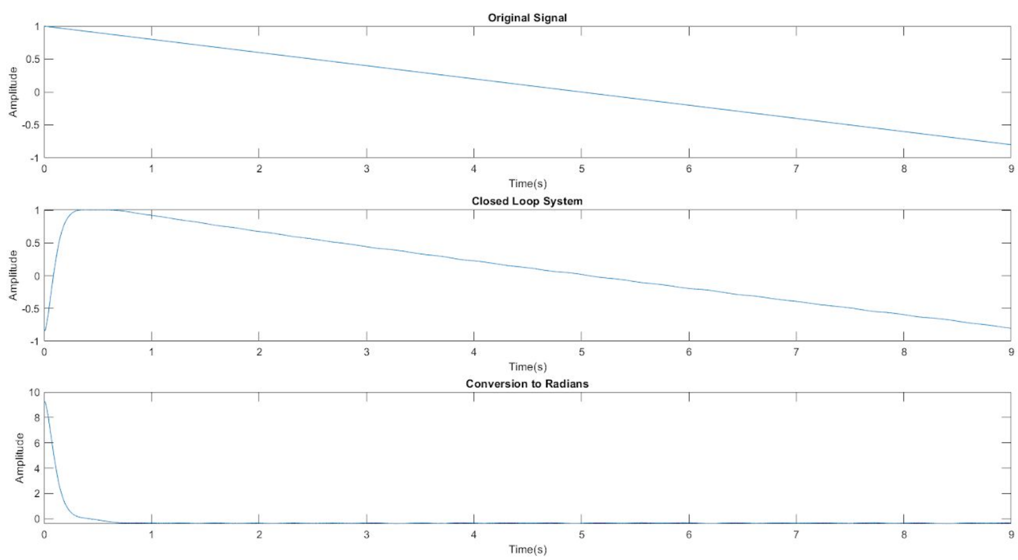

Figure 16. Plot of input and output signals from the system in Figure 15. The input signal is a sawtooth wave with amplitude of 1 and frequency of 0.1Hz. 9 seconds of data had to be taken because the data would reset at 10 seconds or more.

To compare the plots in D1&2 we have to take into account that there was a problem in D2 where we could not retrieve the full 30 second data set. So we will only consider roughly 10 seconds, which will be enough to compare. We will also use the closed loop plot in D2 to compare to the output signal in D1. Firstly, we can see that the lengths of both signals are about 10 seconds. With the D2 plot being 9 seconds but that is due to restrictions, if we interpolate it would probably be around 10 seconds in length. Secondly, the D1 plot has a max value at around 0.82 and the D2 plot has a max value above 1. These values are similar but still quite different. Of course this can be due to different practical motor characteristics which may oppose the motion. In other words the motor isn't 100% efficient therefore there will be discrepancies. Lastly, both waveforms look like right-angled triangles with the D2 plot having a slightly rounded top corner.

Part 2:

A1.

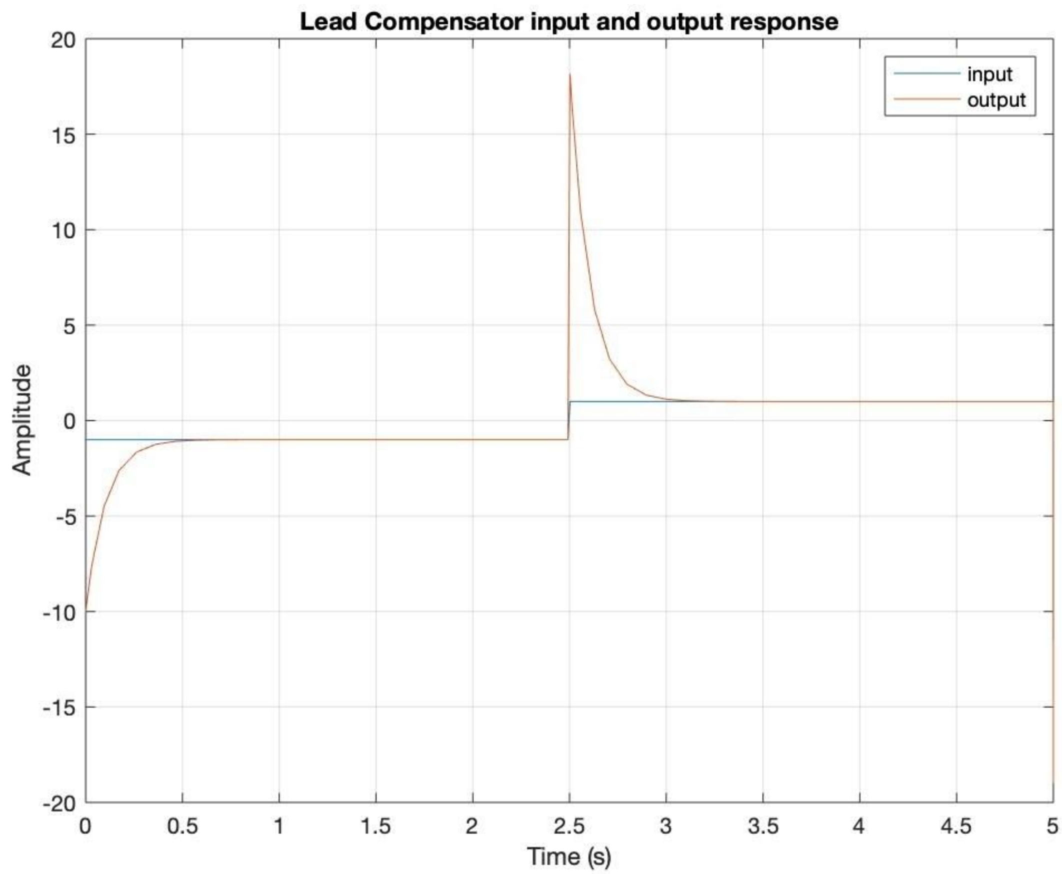

Figure 17: Input and output of the lead compensator

The above graph is of s+1 / 0.1s+1. The blue graph is the output of the lead compensator which eventually approaches the steady state error. It is evident from the graph that high frequency results in a high gain and a low gain results in a gain of 1 with the unit step function.
Time constant = 0.1 seconds
DC gain = 1
High frequency gain = 18

The calculated high frequency gain is DC gain / tau which is 10 however the output system has a high frequency of 18. This discrepancy could be because of the noise or other interferences by the signal generator.

A2.

Zero = -1
Pole = -10

```
>> num = [1 1];
>> den = [0.1 1];
>> sys1 = tf(num,den)

sys1 =

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

Continuous-time transfer function.

>> figure;
>> bode(sys1)
>>
```

Figure 18: Code for the bode plot of the lead compensator.

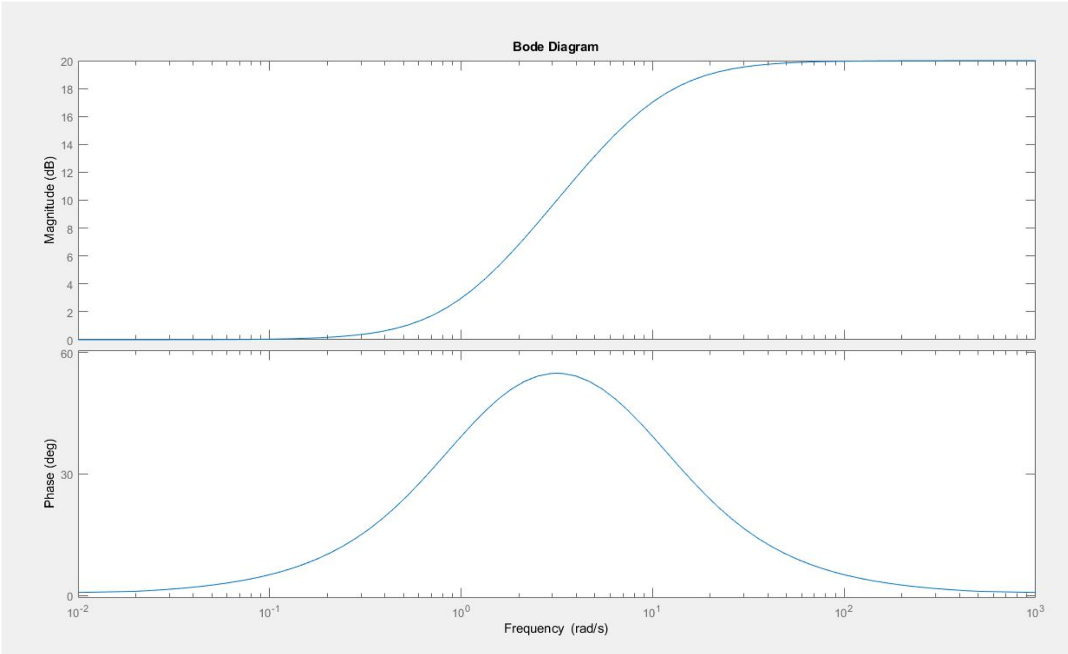

Figure 19: Bode diagram of the lead compensator

The minimum phase is roughly 0 degrees, while the maximum phase is almost 55 degrees.

B.1.:

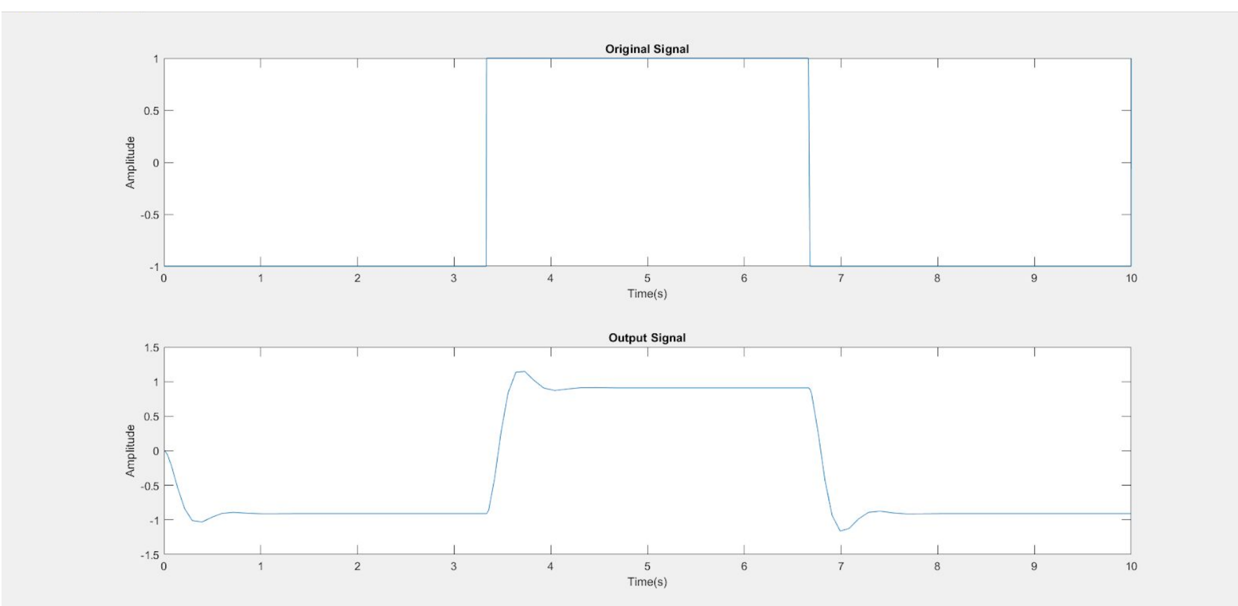

Figure 20: Input and output of the system

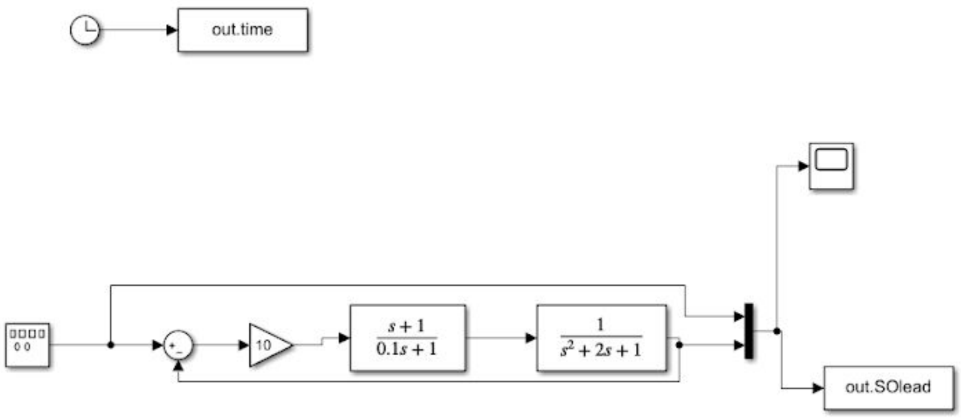

Figure 21: Simulink model of lead compensator with second order system

B.2:

```
Command Window
>> num = [10 10];
den = [0.1 1.2 12.1 11];
sysB2 = tf(num,den)

sysB2 =

              10 s + 10
  -------------------------------
  0.1 s^3 + 1.2 s^2 + 12.1 s + 11

Continuous-time transfer function.

>> figure; stepplot(sysB2)
fx >>
```

Figure 22: Code for the closed loop transfer function of the system Y(s) / R(s).

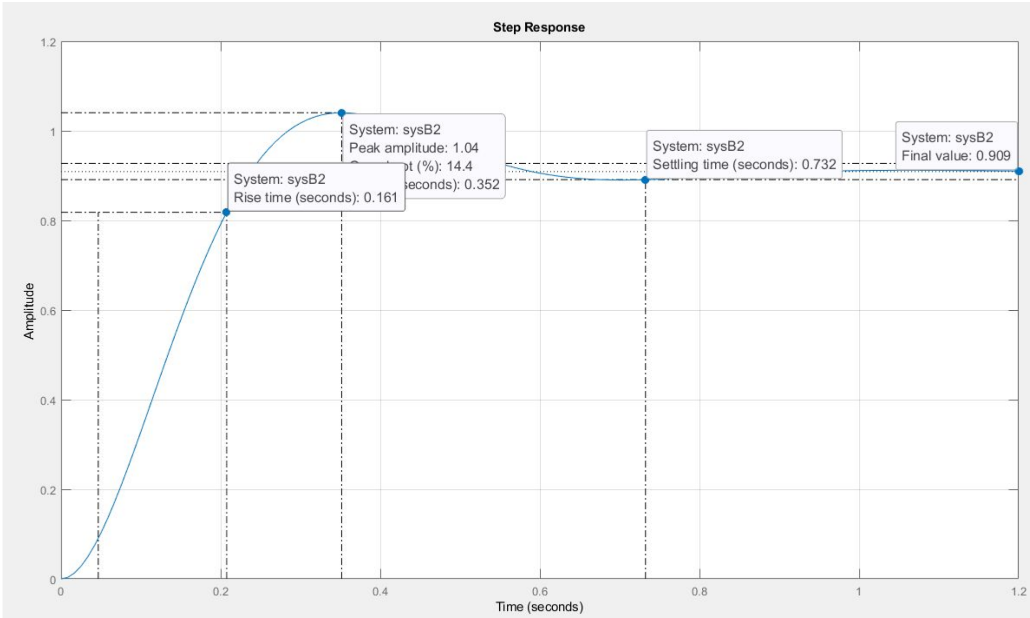

Figure 23: Closed loop transfer function with the lead compensator and K = 10

Rise time, tr = 0.161 seconds
Percentage of the maximum overshoot O.S.% = 14.4
Settling time, ts = 0.732 seconds
Steady state error, ess = 1- 0.909 = 0.091

B.3.:

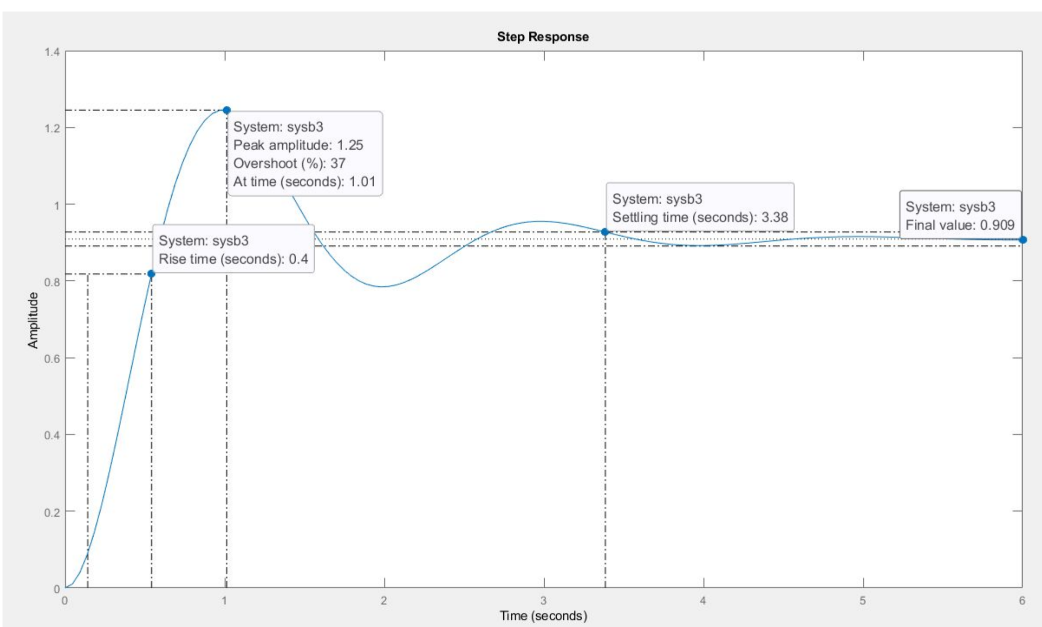

Figure 24: Closed loop transfer function without the lead compensator and K = 10

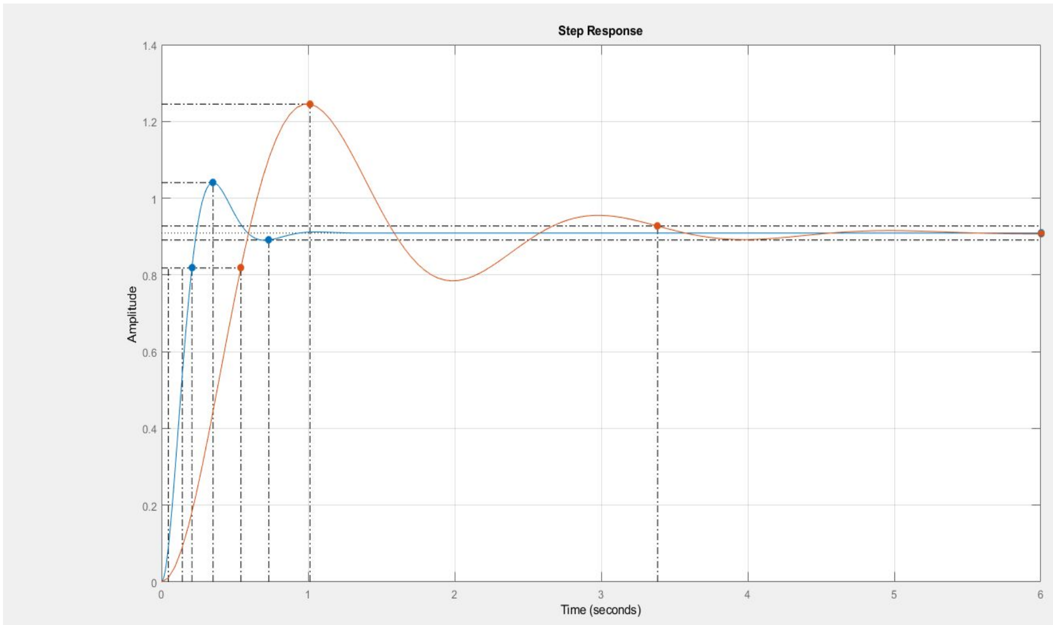

Figure 25: Plot of the closed loop transfer function with K=10, with (blue line) and without (orange line) the lead compensator.

It is evident from the points on the plots that adding a lead compensator decreases the percent overshoot, decreases the rise time, settling time, and doesn't change the steady state error. The ess, is the same because the DC gain is 1 and thus adding or removing the lead compensator does not affect the steady state error. This behaviour of lead compensator increases the speed and stability of the system response.

	Rise time (in seconds)	%Max. O.S.	Settling time (in seconds)	Steady state error
K = 10 with lead	0.161	14.4	0.732	0.091
K = 10	0.4	37	3.38	0.091

B.4.:

Command Window

```
>> num = [10 10];
den = [0.1 1.2 2.1 1];
sysb4 = tf(num,den)

sysb4 =

             10 s + 10
  -----------------------------
  0.1 s^3 + 1.2 s^2 + 2.1 s + 1

Continuous-time transfer function.

>> num = [10];
>> den = [1 2 1];
>> sysb42 = tf(num,den)

sysb42 =

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

Continuous-time transfer function.

>> figure; bode(sysb4, sysb42)
>>
```

Figure 26: Code of the bode plot with and with the lead compensator and with only k = 10

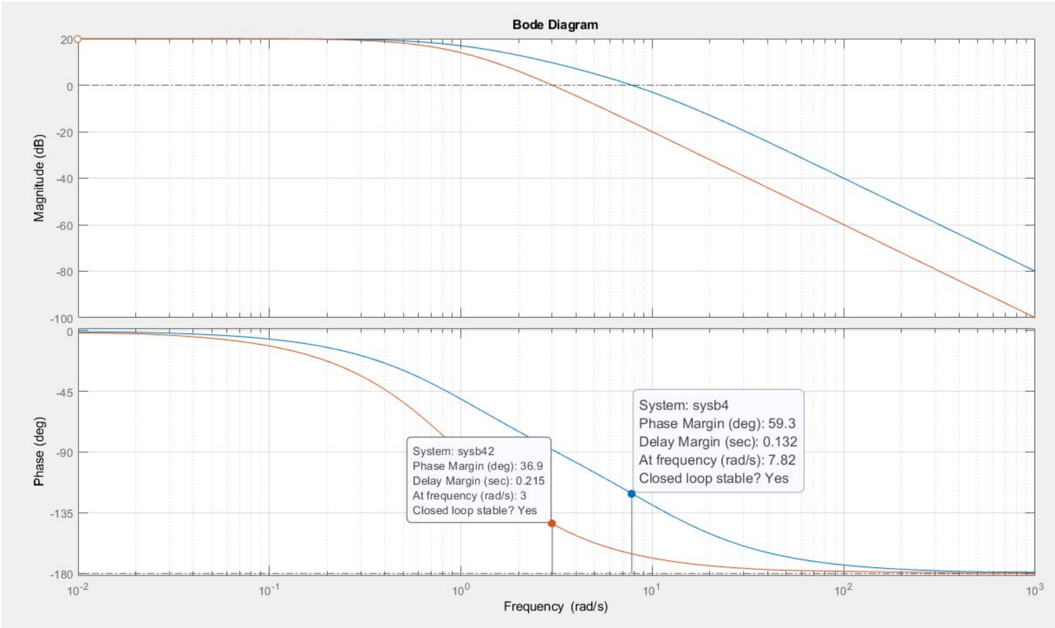

Figure 27: Plot of the open loop transfer function with the lead compensator and with only k = 10

In the above plot, the blue graph is with the lead compensator and orange graph is without the lead compensator with K = 10. It is evident that adding the lead compensator affects the gain crossover frequency. It shifts the gain crossover frequency to higher frequencies/ The lead compensator increases the stability of the system and enhances the transient response. The gain margin is always infinity so that is not a concern, and the phase margin does not cross -180 degrees. The lead compensator increases the phase margin to 59.3 degrees from 36.9 degrees. In general, a phase margin of 60 degrees gives a good graph so this bode plot is acceptable.

C

C.1.:

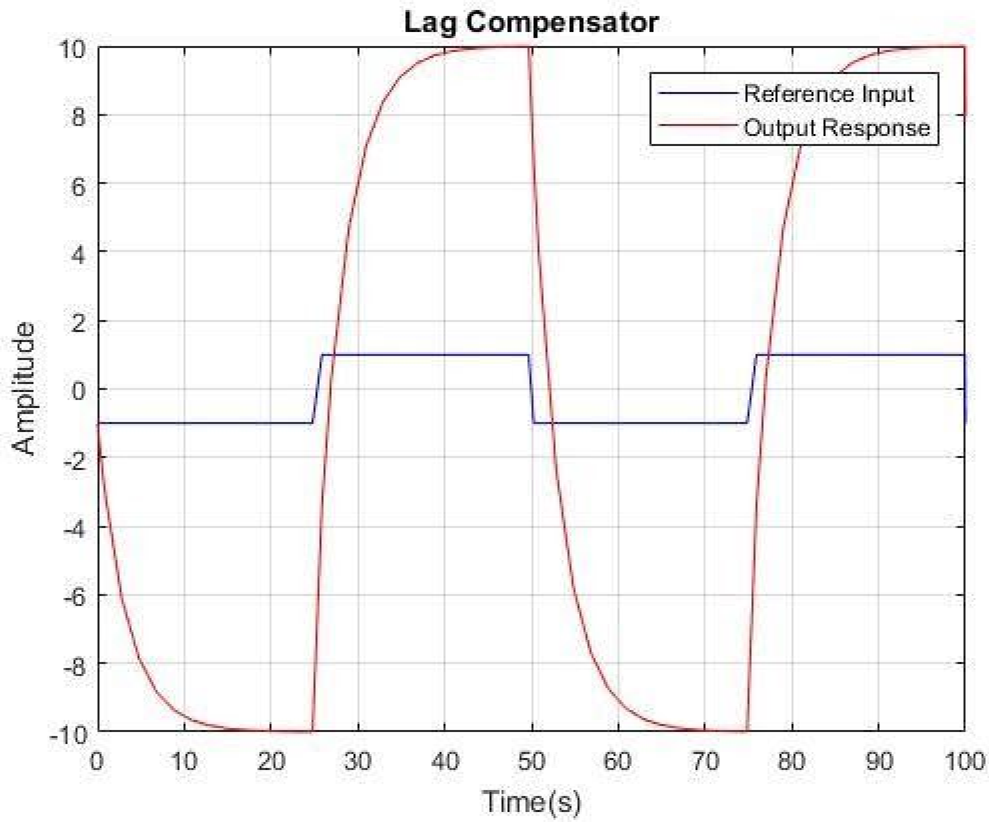

Figure 28: Input and output of the lag compensator to the square wave.

Time constant tau = 3.34s
DC gain: 10
High frequency gain: 1

The behaviour of this system is the total opposite of the lead compensator. The theoretical high frequency does not match with the output of the system due to discrepancies in the signal generator. The high frequency of the output signal is ten times the input signal.

C.2:

Pole of the lag compensator is -0.3 while the zero is -3.

```
Command Window
>> num = [1 3];
>> den = [1 0.3];
>> sysc1 = tf(num, den)

sysc1 =

    s + 3
  -------
  s + 0.3

Continuous-time transfer function.

>> figure; bode(sysc1)
fx >> |
```

Figure 29: Code for the bode diagram of the lag compensator.

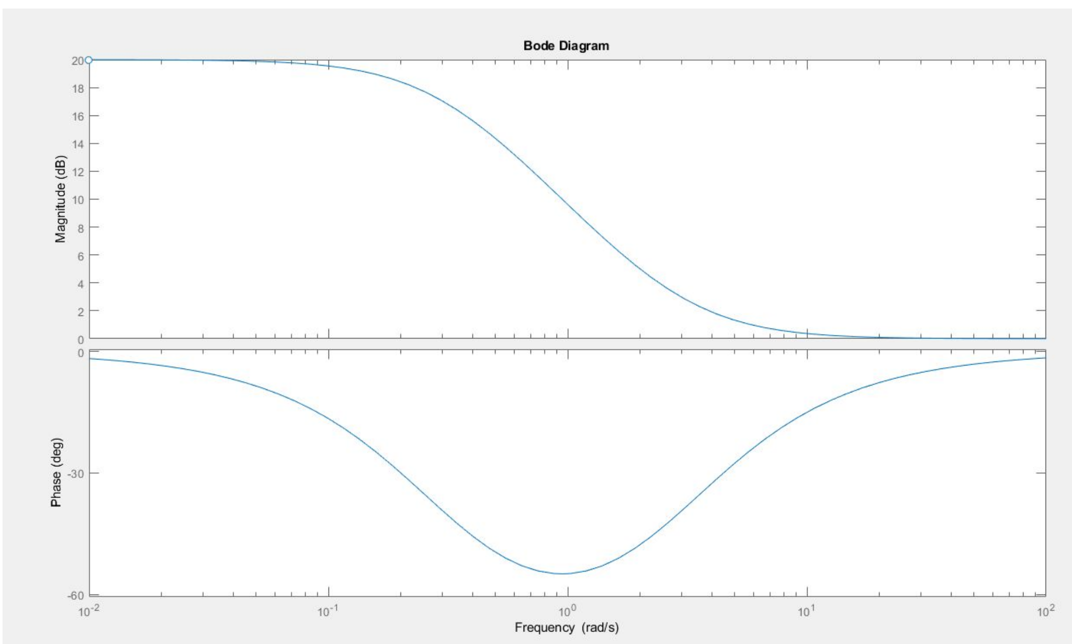

Figure 30: Bode plot of the lag compensator.

The step function above has a high gain at low frequencies and low gain at high frequencies. The minimum phase is -55 degrees and the maximum phase is 0 degrees, which is obtained from the above graph.

D

D.1:

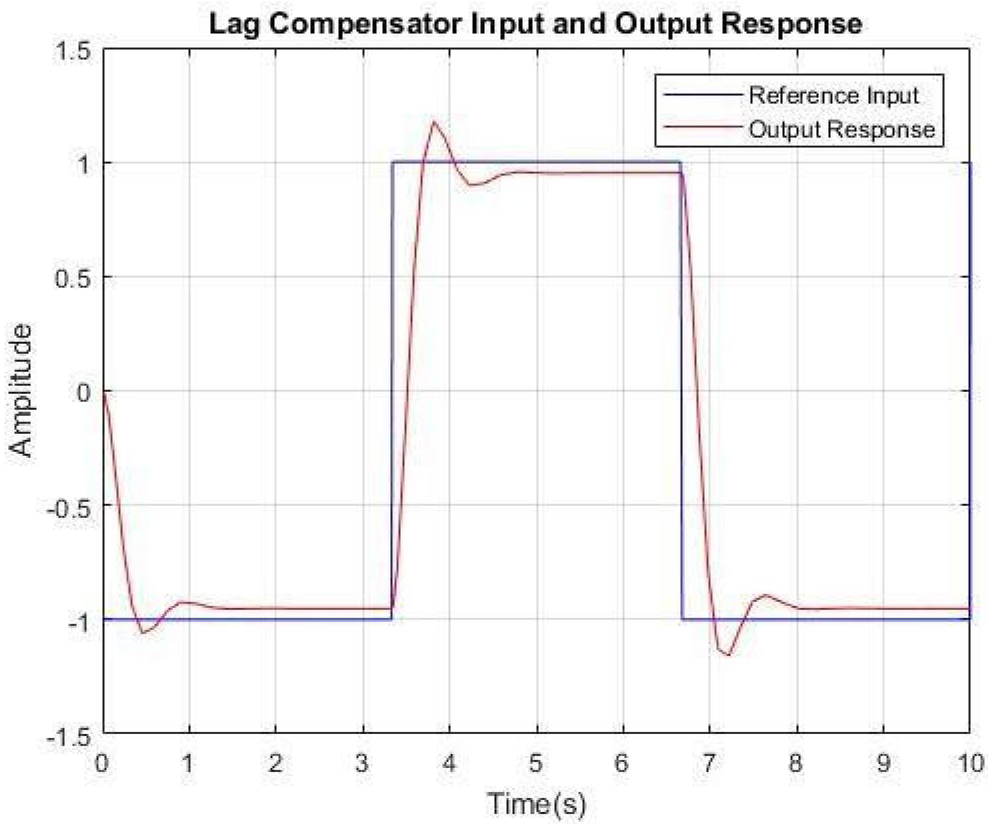

Figure 31:: Plot of input and output signal to the lag compensator.

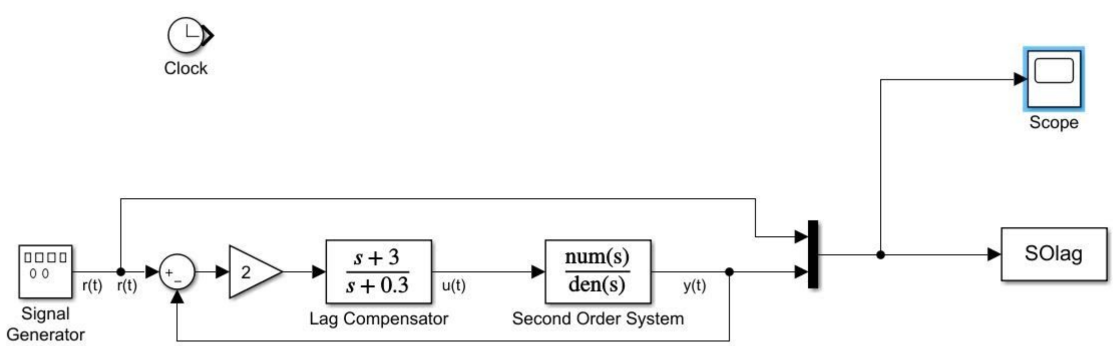

Figure 32:: Simulink model of the lag compensator.

D.2.:

```
Command Window
>> num = [50 150];
>> den = [1 10.3 78  157.5];
>> sysd2 = tf(num, den)

sysd2 =

          50 s + 150
  ----------------------------
  s^3 + 10.3 s^2 + 78 s + 157.5

Continuous-time transfer function.

>> figure; stepplot(sysd2)
fx >> 
```

Figure 33: Code of the closed loop transfer function

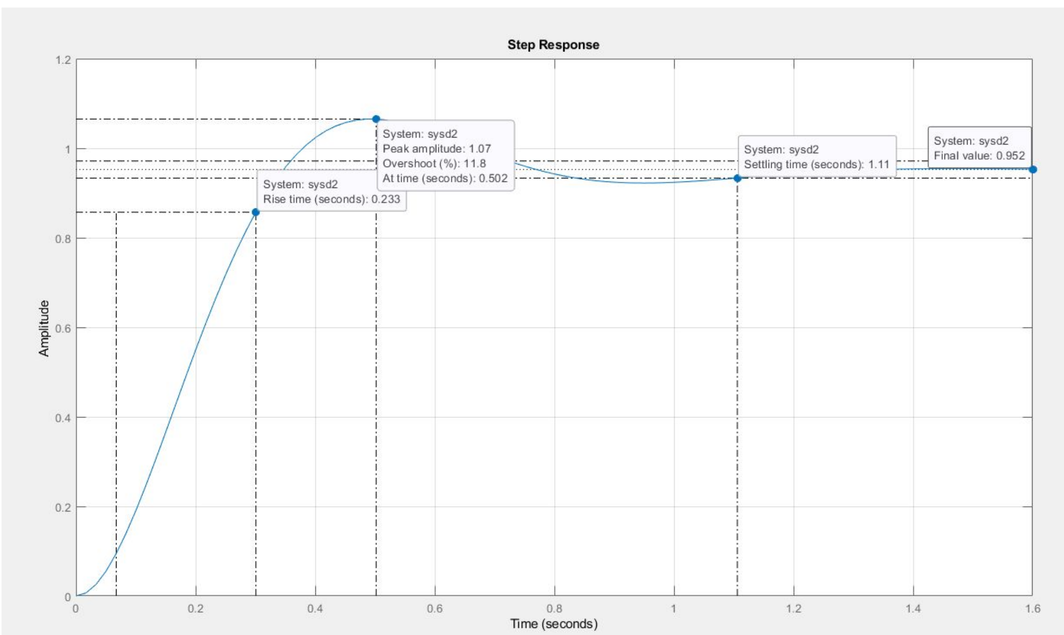

Figure 34: Plot of the closed loop transfer function of the system

Rise time = 0.233 seconds
%O.S.: 11.8
Settling time = 1.11 seconds
Steady state error = 0.048

D.3:

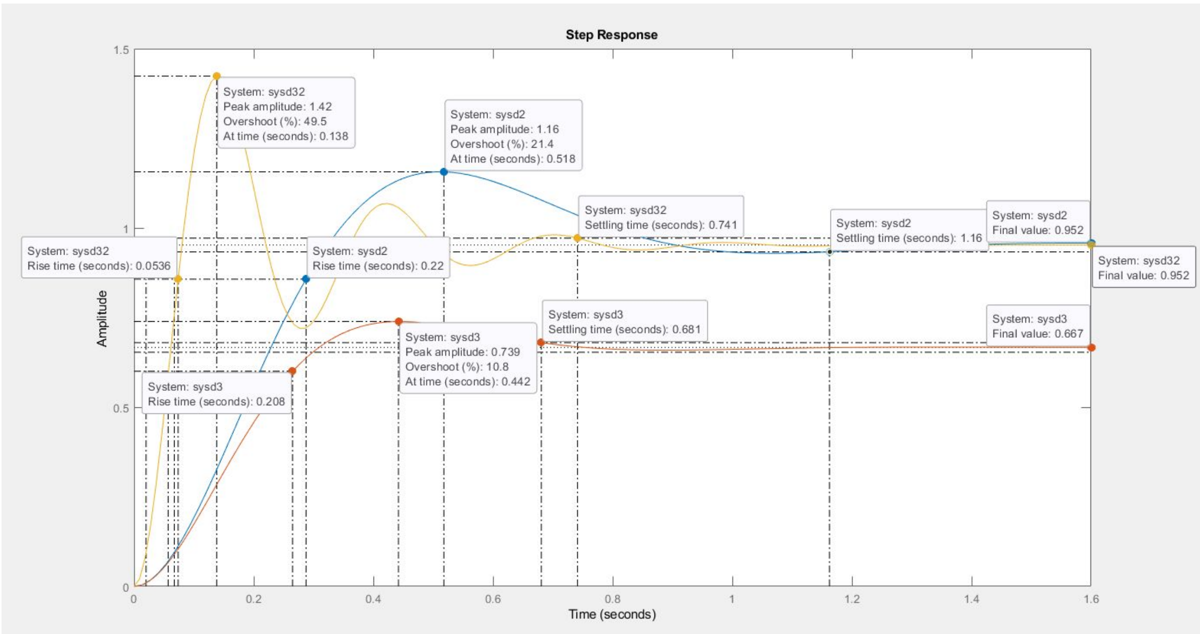

Figure 35:: Three plots with K = 20 (yellow), systems with lag compensator (blue), and systems with K = 2 (orange).

For the system with lag compensator, transient wise its behaviour is similar to K = 2, as well as a similar rise time. However, it is different for K = 20, since it has higher oscillations which affects the rise time. The percent overshoot for the lag compensator and K = 2 is similar, while the K = 20 system and lag compensator have similar steady error. The lag compensator enhances the steady state error. The system with K = 2 is good for transient response and the system with K = 20 is ideal for steady state error. On the other hand, the lag compensator has a good behaviour for both the characteristics i.e. the transient response and steady state error.

	Rise time	%Max. Overshoot	Settling time	Steady state error
K = 2 with lag	0.22	21.4	1.16	0.048
K = 2	0.208	10.8	0.681	0.333
K = 20	0.0536	49.5	0.741	0.048

D.4:

Command Window

```
>> num = [50 150];
den = [1 10.3 28 7.5];
sysd4 = tf(num, den)

sysd4 =

          50 s + 150
  ---------------------------
  s^3 + 10.3 s^2 + 28 s + 7.5

Continuous-time transfer function.

>> num = [50];
>> den = [1 10 25];
>> sysd42 = tf(num, den)

sysd42 =

        50
  ---------------
  s^2 + 10 s + 25

Continuous-time transfer function.

>> figure; bode(sysd4, sysd42)
>>
```

Figure 36: Code for the bode plot of the lag compensator and with only K = 2

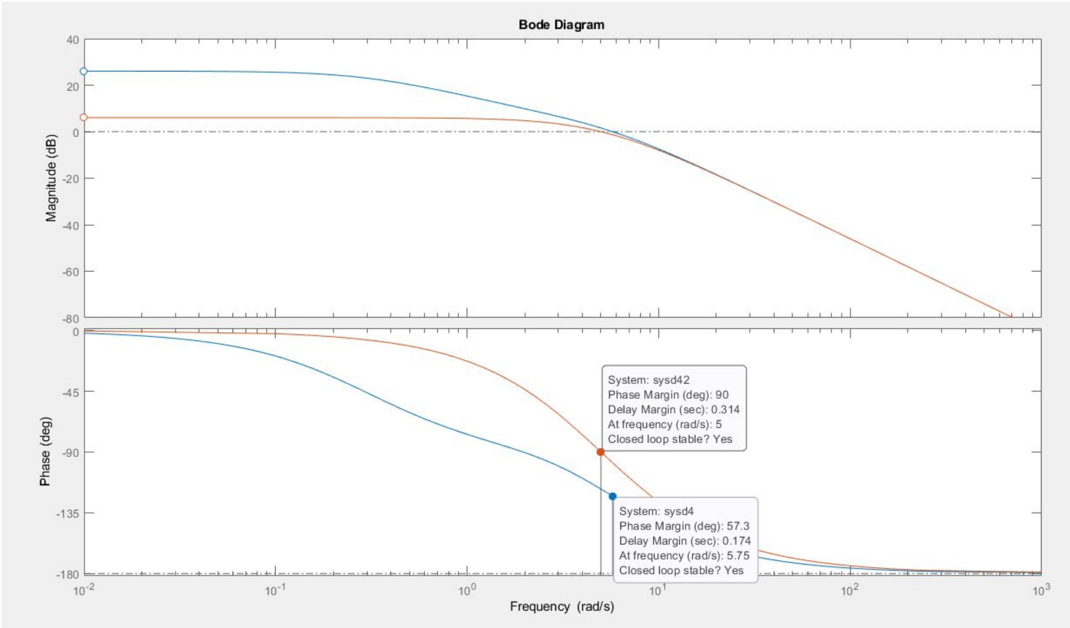

Figure 37: Bode plot of the lag compensator and with only K = 2 with gain and phase margin.

The above bode plots include the lag compensator (blue) and K = 2 (orange). Talking about the magnitude plot, it has a high gain for low frequencies and at the high frequencies, the behaviour is the same for both systems. For the phase plot, the lag compensator has seen a decreased phase margin (PM) of 57.3 degrees from 90 degrees. As mentioned before, a PM of 60 degrees has a good %O.S. for a system. Furthermore, a good % max O.S. ensures good stability of the system. Thus, the lag compensator affects the stability of a closed loop system.

Appendix:

PART 1:

```
%% A4

tA4 = out.VSVPmodel.Time;
iA4 = out.VSVPmodel.data(:,1);
speed = out.VSVPmodel.data(:,2);
position = out.VSVPmodel.data(:,3);

figure(1);
%Original signal
subplot(311)
plot(tA4,iA4)
xlabel('Time(s)')
ylabel('Amplitude')
title('Original Signal')
subplot(312)
%Speed Transfer Function
plot(tA4,speed)
xlabel('Time(s)')
ylabel('Amplitude')
title('Speed Transfer Function')
%Position Transfer Function
subplot(313);
plot(tA4,position);
xlabel('Time(s)')
ylabel('Amplitude')
title('Position Transfer Function')

%% B1

tB1 = time;
inputB1 = speedmodel(:,2);
outputB1 = speedmodel(:,1);

figure(2)
%Original signal
subplot(211)
plot(tB1,inputB1)
```

```
xlabel('Time(s)')
ylabel('Amplitude')
title('Original Signal')
%output signal
subplot(212)
plot(tB1,outputB1)
xlabel('Time(s)')
ylabel('Amplitude')
title('Output Signal')

%% B3 0 gain

tB3 = time;
inputB3 = speedmodel(:,2);
outputB3 = speedmodel(:,1);

figure(2)
%Original signal
subplot(211)
plot(tB3,inputB3)
xlabel('Time(s)')
ylabel('Amplitude')
title('Original Signal')
%output signal
subplot(212)
plot(tB3,outputB3)
xlabel('Time(s)')
ylabel('Amplitude')
title('Output Signal')

%% B3 0.1751

tB3 = time;
inputB3 = speedmodel(:,2);
outputB3 = speedmodel(:,1);

figure(2)
%Original signal
subplot(211)
plot(tB3,inputB3)
xlabel('Time(s)')
ylabel('Amplitude')
title('Original Signal')
```

```
%output signal
subplot(212)
plot(tB3,outputB3)
xlabel('Time(s)')
ylabel('Amplitude')
title('Output Signal')

%% B3 0.2 gain

tB3 = time;
inputB3 = speedmodel(:,2);
outputB3 = speedmodel(:,1);

figure(2)
%Original signal
subplot(211)
plot(tB3,inputB3)
xlabel('Time(s)')
ylabel('Amplitude')
title('Original Signal')
%output signal
subplot(212)
plot(tB3,outputB3)
xlabel('Time(s)')
ylabel('Amplitude')
title('Output Signal')
%% C1

tC1 = out.DCmotorPI.time;
iC1 = out.DCmotorPI.data(:,1);
oC1 = out.DCmotorPI.data(:,2);

figure(2)
%Original signal
subplot(211)
plot(tC1,iC1)
xlabel('Time(s)')
ylabel('Amplitude')
title('Original Signal')
%output signal
subplot(212)
plot(tC1,oC1)
xlabel('Time(s)')
```

```
ylabel('Amplitude')
title('Output Signal')

%% D1

tD1 = out.DCmotorPI.time;
iD1 = out.DCmotorPI.data(:,1);
oD1 = out.DCmotorPI.data(:,2);

figure(3)
%Original signal
subplot(211)
plot(tD1,iD1)
xlabel('Time(s)')
ylabel('Amplitude')
title('Original Signal')
%output signal
subplot(212)
plot(tD1,oD1)
xlabel('Time(s)')
ylabel('Amplitude')
title('Output Signal')

%% C2

tC2 = time;
iC2 = CLspeed(:,1);
closedLoopC2 = CLspeed(:,2);
openD2 = CLspeed(:,3);

figure(4);
%Original signal
subplot(311)
plot(tC2,iC2)
xlabel('Time(s)')
ylabel('Amplitude')
title('Original Signal')
subplot(312)
%Closed Loop System
plot(tC2,closedLoopC2)
xlabel('Time(s)')
ylabel('Amplitude')
title('Closed Loop System')
```

```
%Open Loop System
subplot(313);
plot(tC2,openD2);
xlabel('Time(s)')
ylabel('Amplitude')
title('Open Loop System')
%% D2

tD2 = time;
iD2 = CLposition(:,1);
closedLoopD2 = CLposition(:,2);
convertD2 = CLposition(:,3);

figure(5);
%Original signal
subplot(311)
plot(tD2,iD2)
xlabel('Time(s)')
ylabel('Amplitude')
title('Original Signal')
subplot(312)
%Closed Loop System
plot(tD2,closedLoopD2)
xlabel('Time(s)')
ylabel('Amplitude')
title('Closed Loop System')
%Conversion to Radians
subplot(313);
plot(tD2,convertD2);
xlabel('Time(s)')
ylabel('Amplitude')
title('Conversion to Radians')


%% PART 2

%%B1

tB1 = out.SOlead.time;
iB1 = out.SOlead.data(:,1);
oB1 = out.SOlead.data(:,2);
```

```
figure(3)
%Original signal
subplot(211)
plot(tB1,iB1)
xlabel('Time(s)')
ylabel('Amplitude')
title('Original Signal')
%output signal
subplot(212)
plot(tB1,oB1)
xlabel('Time(s)')
ylabel('Amplitude')
title('Output Signal')
```

Personal Summaries:

Kushal:
500843903

In the first part, we applied the control system principles to a servo motor system. Given two transfer functions, relations, and other parameters we produced two transfer functions for voltage to speed and voltage to position.By adding Quarc target blocks, we sent digital to analogue control signals to the motor and return analogue-to-digital information about the motor specifically tachometer data. We first analysed the output of the system to predict the transfer function then changed the input voltages. We then experimented with the voltage to speed transfer function followed by the position to speed transfer function. Through these experiments, we concluded that the systems may not function to perfection as anticipated due to the errors in the components and signal ignorance. In the second part, we explored how the lead and lag compensators affected the output signal. We designed the given circuits in Simulink and set various parameters such as amplitude and frequency in signal generated, followed by the plots in MATLAB. This process was repeated for several instances where there were different k values, different orders, lead, and lag compensators. The bode plots and step plots were constructed eventually with the code written in the command window, with “tf” function for step plots, written in the command window. We right clicked on the plots and activated the different time domain specifications which labelled the exact specifications for us. These values were then filled in the table and used to analyse the characteristics and behaviours of lead and lag compensators. For lag compensators we noticed it worked effectively to reduce the steady state errors than the proportional types. We observed how lag compensator reduces the phase margin to almost 60 degrees which is the ideal case of the % max overshoot of the system and how the % max OS ensures the stability of the system. For lead compensators, it decreases the percent overshoot, decreases the rise time, settling time, and doesn’t change the steady state error. The lead compensator increases the stability of the system and enhances the transient response. To conclude, a lag compensator could be added to improve the steady state error without changing system stability and a lead compensator can be added to improve system stability without changing the steady state error as shown above.

Peter Zastawny

500813230

This lab was split into two parts. In the first part we looked at modeling the transfer function of a DC servo motor. To create the transfer function, we used the equations and parameters given in the lab manual. We had to Laplace Transform the main function to put it into the frequency domain so we could use it in Simulink. The function given would give us the speed-to-voltage transfer function. We know that speed is the derivative of position, therefore in the frequency domain to make the position-to-voltage transfer function we multiply the denominator by s. This gave us the two transfer functions which we would use throughout the rest of part 1. We would then compare the speed-to-voltage transfer function to the actual servo motor. When compared, they produced very similar results. We next looked at PI controllers and modelled them with the calculated transfer functions, and then tested them with the servo motors. Again, we saw very positive results, having the results be very similar. In the second part of the lab we studied the lead and lag compensators. We did this by exploring their time domain and frequency domain characteristics. Compensators are used to enhance the response of the systems. We used bode plots to help us find these characteristics. Bode plots allow us to look at the frequency response of a system. The magnitude plots help us see the magnitude of the system and the phase plots allow us to see the phase shifts of the system. The main characteristics we were looking at were the rise times, % max overshoot, settling times, and the steady-state errors. Firstly, we looked at the lead compensator which we know enhances transient response and it makes the system more stable. As we can see in the bode plots, this is due to the lead compensator increasing the phase margin. Secondly, we looked at the lag compensator which we know decreases the steady-state error without touching the transient response of the system. As we can see in the bode diagram, the lag compensator increases the gain and decreases the phase margin which in turn will decrease the steady-state error. To also note we had a phase margin of around 60 degrees which we know is ideal for steady-state error.

BME639: Control Systems Lab 2 Grading Sheet

Part 1A: DC Servo Motor Modeling Using First-Principles /12

Part 1B: DC Servo Motor Modeling Using Experiments /8

Part 1C: DC Servo Motor Speed Control /7

Part 1D: DC Servo Motor Position Control /7

1. **Part 2A: Time Response of a Lead Compensator /9**

2. **Part 2B: Lead Compensator and Second-Order Systems /11**

3. **Part 2C: Time Response of a Lag Compensator /9**

4. **Part 2D: Lag Compensator and Second-Order Systems /12**

1. **General Formatting: Clarity, writing style, grammar, spelling, layout of the report 2. /10**

Total Mark for the Collaborative Part of the Report /85

Partner 1 (name): Partner 2 (name):

Summary Mark /15 Summary Mark /15

Interview Mark Pass / Fail Interview Mark Pass / Fail

TOTAL: /100 TOTAL: /100

Before submitting your report, your TA asks questions about your report. If there is no consistency between your oral answer and your report, you will lose 50% of your total mark. For this part, Pass or fail will be circled next to your name accordingly.