Lab 3 – Position Control

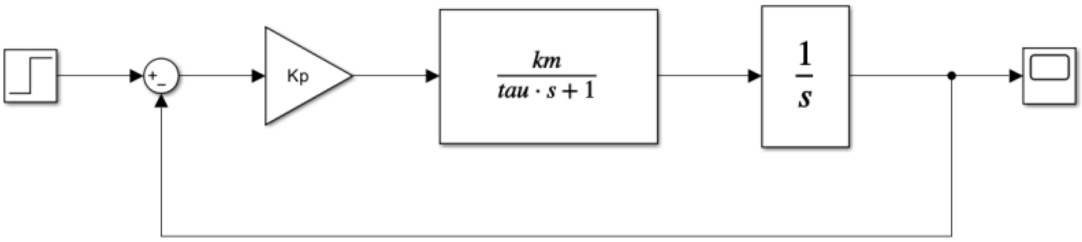
Submitters:

Or Shaul

Saray Sokolovsky

2. Pre Lab

2.1 Second order system identification



a. Given the system above, assuming the plant is a DC motor using its first order approximation, write what the blocks represent and the units of the input and output signals:
The input is a step function. In the open loop we can see K_p- This component yields a control action that is directly proportional to the instantaneous error. The magnitude of this response is governed by the proportional gain.
The integrator $\frac{1}{s}$ is used because we want the output to be the angle of the rotor (rad). So, you integrate ω to get θ. The input signal has units of voltage (V), and the output signal is the angle of the rotor and has units of radian (rad).

b. Given the system above, what is the close loop transfer function:
The OL transfer is:

$$L(s) = k_p\left(\frac{k_m}{s(\tau s + 1)}\right)$$

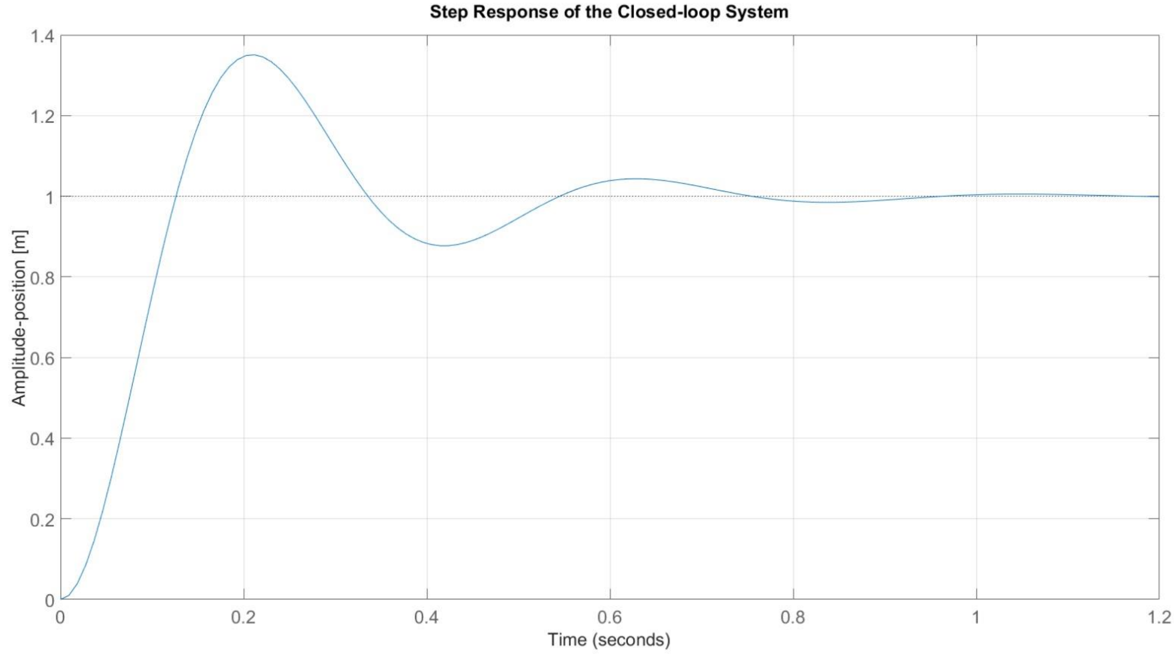
Therefore, the closed loop –

$$T(s) = \frac{L(s)}{1 + L(s)} = \frac{k_p k_m}{\tau s^2 + s + k_p k_m}$$

c. Compare the transfer function derived in b. To the standard second order transfer function of the form, Find representations of ξ and ω_n as function of τ, k_m and k_p:
By comparing the close loop transfer function we got in the previous part we get:

$$\omega_n = \sqrt{\frac{k_p k_m}{\tau}}, \xi = \frac{1}{2\tau\omega_n} = \frac{1}{2\sqrt{k_m k_p \tau}}, K = \frac{k_p k_m}{\tau}$$

d. Using MATLAB, plot the step response of the following first order system $G(s) = \frac{2.5}{0.1s+1}$ using the system represented above with gain of $k_p = 10$:



Step info:

```
Command Window
>> Q2_1d
Step Response Characteristics:
         RiseTime: 0.0852
    TransientTime: 0.7072
     SettlingTime: 0.7072
      SettlingMin: 0.8772
      SettlingMax: 1.3507
        Overshoot: 35.0670
       Undershoot: 0
             Peak: 1.3507
         PeakTime: 0.2118

Damping Ratio (ζ): 0.31642
fx >>
```

e. Based on the peak time and over-shoot performance metrices for a standard second order system. Find the values of ξ, ω_n:

From the formulas we got in part 1.3: $OS = 100e^{-\frac{\pi\xi}{\sqrt{1-\xi^2}}}, T_p = \frac{\pi}{\omega_d} = \frac{\pi}{\omega_n\sqrt{1-\xi^2}}$

From the MATLAB we got OS=35.0670 , $T_p = 0.2118\, sec$ and $\xi = 0.316$

Lets calculate ω_n:

$$T_p = \frac{\pi}{\omega_n\sqrt{1-\xi^2}} \rightarrow \omega_n = \frac{\pi}{T_p\sqrt{1-\xi^2}} = \frac{\pi}{0.2118\sqrt{1-0.316^2}} = 15.63$$

f. Based on the equations derived in C verify the transfer function of the DC motor in the form of $G(s) = \frac{k_m}{\tau s+1}$

Using the equations derived in part C we get:

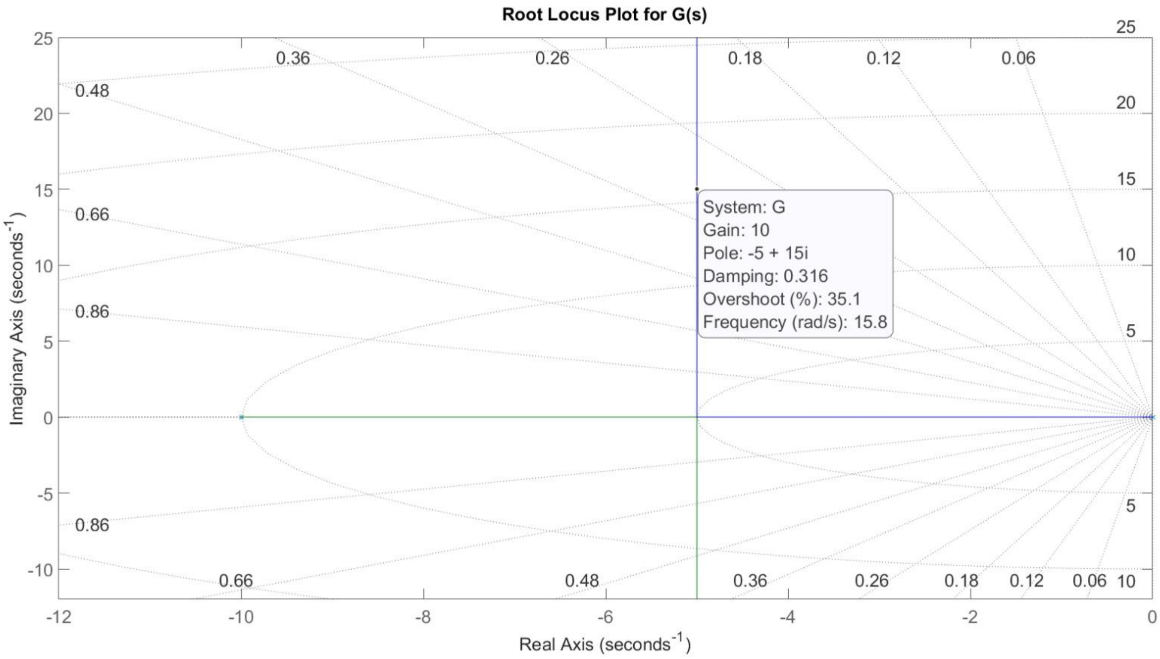
$\omega_n = \sqrt{\frac{k_p k_m}{\tau}} \rightarrow k_m = \frac{\omega_n^2 \tau}{k_p} = 15.63^2 \cdot \frac{0.1}{10} = 2.44$

Error: $100 \cdot \left(1 - \frac{2.44}{2.5}\right) = 2.4\%$

$\xi = \frac{1}{2\tau\omega_n} \rightarrow \tau = \frac{1}{2\xi\omega_n} = \frac{1}{2\cdot 0.316 \cdot 15.63} = 0.1$ (like the theory)

So, the transfer function - $G(s) = \frac{2.44}{0.1s+1}$

g. Using root locus plot for the DC motor transfer function G verify that the over-shoot, ζ and ωn for Kp=10 is the same as measured and calculated in d,e (up to a small rounding error). Explain how root locus can be beneficial in designing a proportional controller.



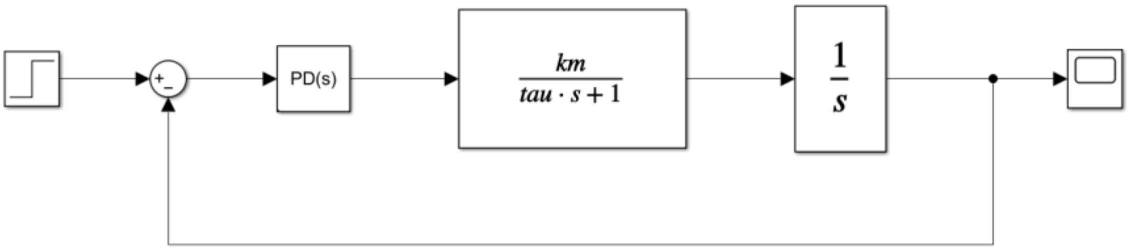
from the plot we can see that the gain is $K_p = 10, \xi = 0.316, OS = 35.1\%, freq\ \omega_n = 15.8 \left(\frac{rad}{sec}\right)$

All those parameters match exactly to the parameters we got previously.

Root locus can be beneficial in designing a proportional controller:

Designing a proportional controller can be difficult. RL plot, gives up a visual tool. We can easily see on the plot where the system is stable and what are the step response parameters (OS, ξ, ω_n) for every gain. This method helps us finding the optimal proportional controller with ease.

2.2 PD controller



a. Given the system above, assuming the plant is a DC motor using its first order approximation $G(s) = \frac{k_m}{\tau s+1}$, and $PD(s) = k_p + k_d s$ What would be the close loop transfer function $T_1(s)$:
The OL : $L(s) = (k_p + k_d s)\frac{k_m}{\tau s+1}$
The closed loop:

$$T_1(s) = \frac{L(s)}{1+L(s)} = \frac{(k_p + k_d s)\frac{k_m}{\tau s+1}}{1+(k_p + k_d s)\frac{k_m}{\tau s+1}} = \frac{k_m k_d s + k_m k_p}{\tau s^2 + (1+k_m k_d)s + k_m k_p}$$

b. Compare the poles of the transfer function $T_1(s)$ to the poles of the standard second order system represented as $T_2(s) = \frac{K}{s^2 2\xi\omega_n + \omega_n^2}$And find representation of ξ and $\omega_n$as a function of k_m, τ, k_p, k_d:
The poles of $T_2(s)$ are $-\xi\omega_n + \omega_n\sqrt{\xi^2 - 1}, -\xi\omega_n - \omega_n\sqrt{\xi^2-1}$
The poles of $T_1(s)$ are:

$$-\frac{1+k_m k_d}{2\tau} \pm \sqrt{\frac{k_m k_p}{\tau}}\sqrt{\frac{(1+k_m k_d)^2}{4\tau k_m k_p} - 1}$$

On comparing, we get

$$\xi = \frac{1+k_m k_d}{2\sqrt{\tau k_m k_p}}, \omega_n = \sqrt{\frac{k_m k_p}{\tau}}$$

As a sanity check, we indeed see that the product of ξ, ω_ngives us the first term of the poles.

c. Find the values of ξ, ω_n which will result in a step response for the standard second order system $T_2(s) = \frac{K}{s^2 2\xi\omega_n + \omega_n^2}$ with the following criteria: $OS = 25\%, t_P = 0.2(sec)$:
Using the formula of overshoot given in the pdf, we have the following equation:

$$0.25 = e^{-\frac{\pi\xi}{\sqrt{1-\xi^2}}}$$

However, I will use a simplified version of the above formula (From Practical feedback systems course):

$$0.25 = 1 - \frac{\xi}{0.6} \to \xi = 0.45$$

$$T_p = \frac{pi}{\omega_d}$$

Substituting the values, we get $\omega_d = 15.7\frac{rad}{sec}$. also, $\omega_d = \omega_n\sqrt{1-\xi^2}$

So $\omega_n = 17.58\frac{rad}{sec}$

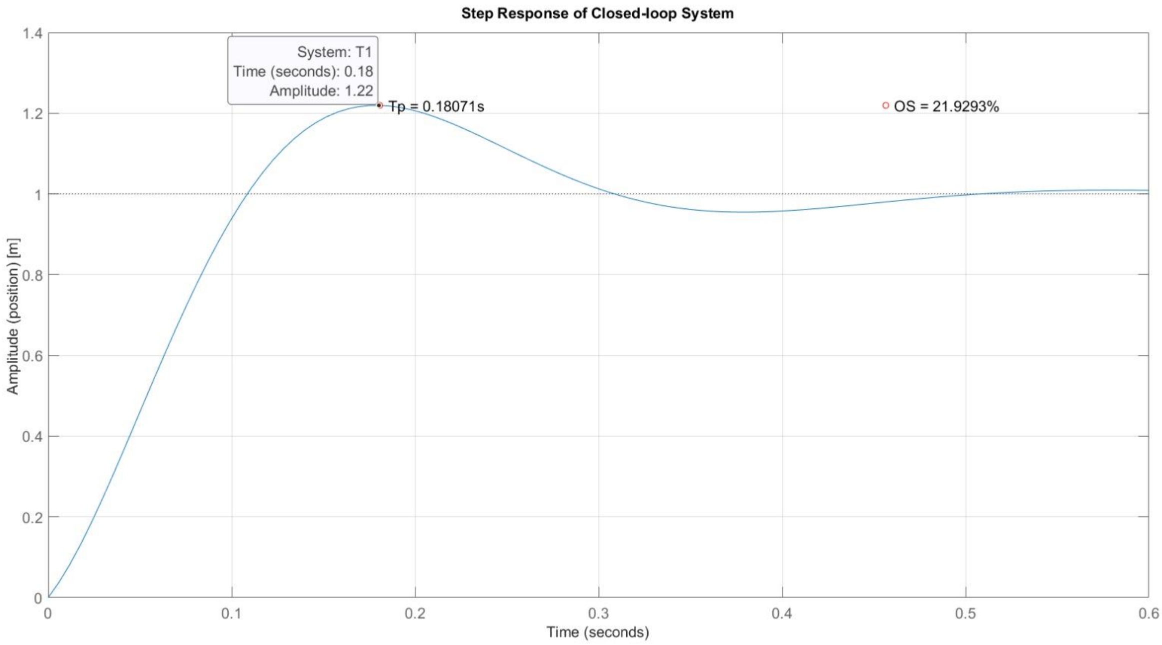
d. Based on the equations found in b. And assuming the motor has a transfer function of $G(s) = \frac{2.5}{0.1s+1}$ find the values for k_p, k_d:

Using the equations found in b and using the values of ξ, ω_n we found in d , we get the following two equations:

$$\omega_n = 17.58 = \sqrt{\frac{2.5k_p}{0.1}}, \xi = 0.45 = \frac{1+2.5k_d}{2\sqrt{0.1\cdot k_p}}$$

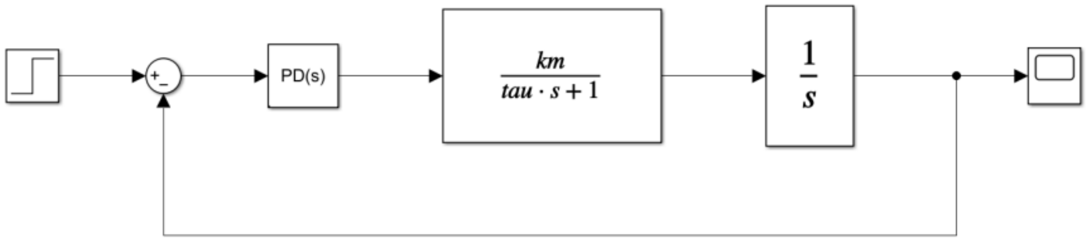
$$K_p = 12.36, k_d = 0.232$$

e. Simulate the system at a. (in matlab or simulink). using the found controller gains and dc motor transfer function from d, Plot the step response of the system and mark os and t_p:

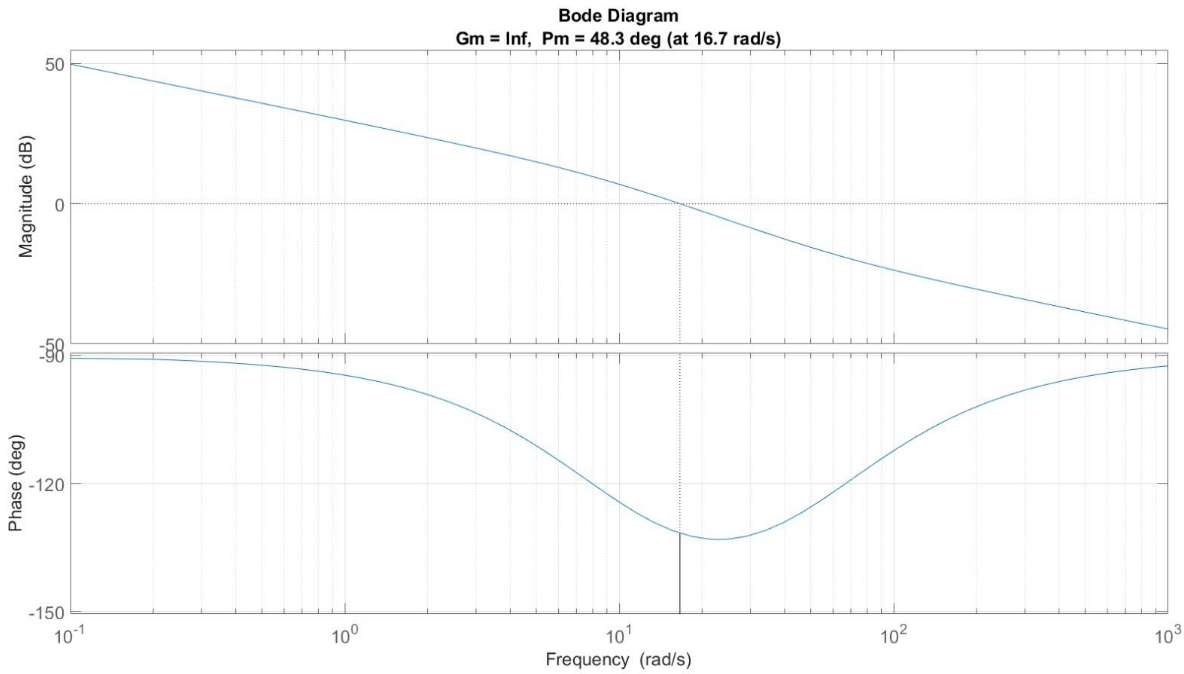


We can see that the $OS = 21.9293\%, T_p = 0.18071$. There is a slight difference due to the accurency of the numbers.

2.3 Lead-Lag compensator



a. Given the system above, and using the values from the previous section, plot in matlab the open loop bode of the system $PD(s) \cdot P(s), P(s) = \frac{k_m}{s(\tau s+1)}$, and evaluate the ω_c – crossover frequency and phase margin of the system:



The Crossover frequency is $\omega_c = 16.7 \frac{rad}{sec}$

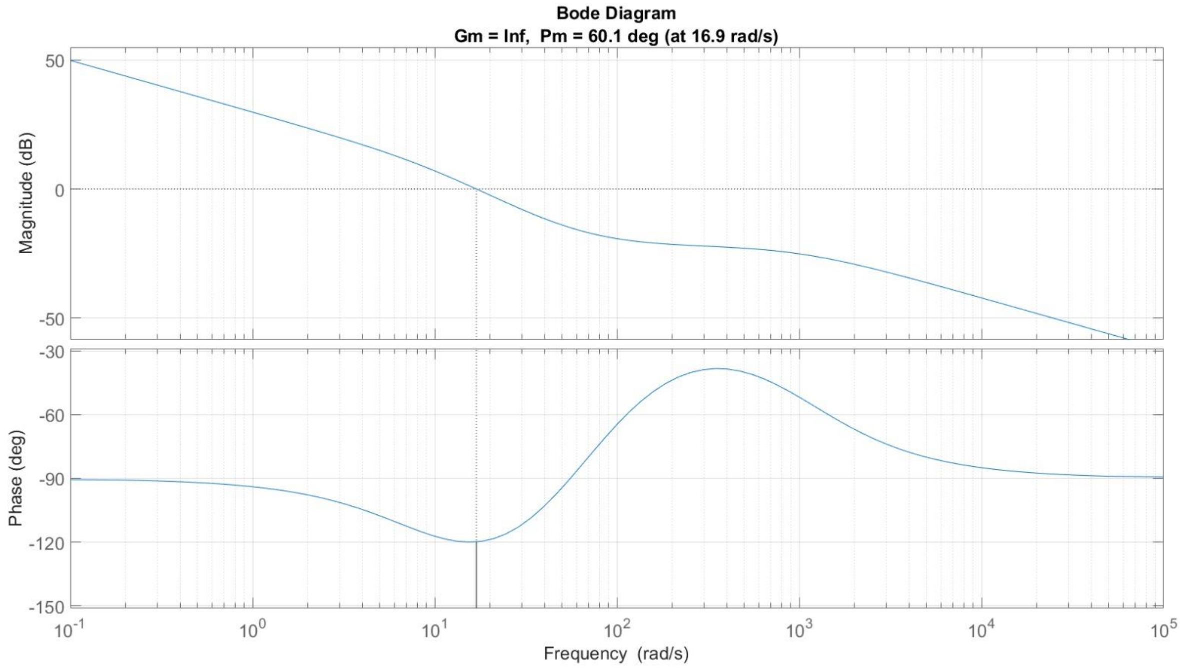
b. Design a lead lag compensator to fix the Phase margin to 60° . Add to the bode plot in a. the compesated bode and mark the new ωc, Phase margin. Plot the step responses from both of the systems on the same graph. Mark the OS%, Tp. Of both. (use the graphical method or any other bmethod of your choice):

From the OL bode plot we got: $PM = 48.3\ deg,\ \ 16.7 \frac{rad}{sec}$.

To obtain $PM = 60\ deg$, we need to add 60-48.3=11.7 deg. So, we took the parameters (to get the PM we want we need lead Compensator) $a = 15, u = 0.0135 \rightarrow \tau = \frac{u}{a} = \frac{0.0135}{15} = 0.0001$

$$C(s) = \frac{a\tau s + 1}{\tau s + 1} = \frac{0.0135s + 1}{0.0001s + 1}$$

The Bode plot obtained in MATLAB for the open loop with the compensator:

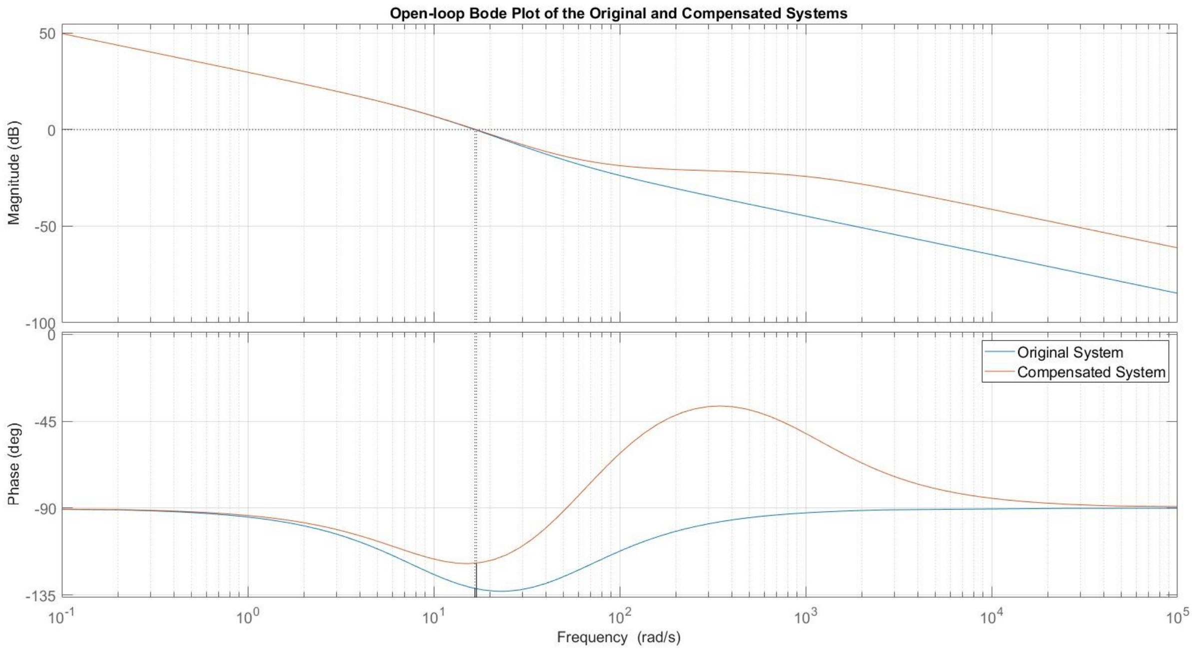


We can see that the PM = deg.

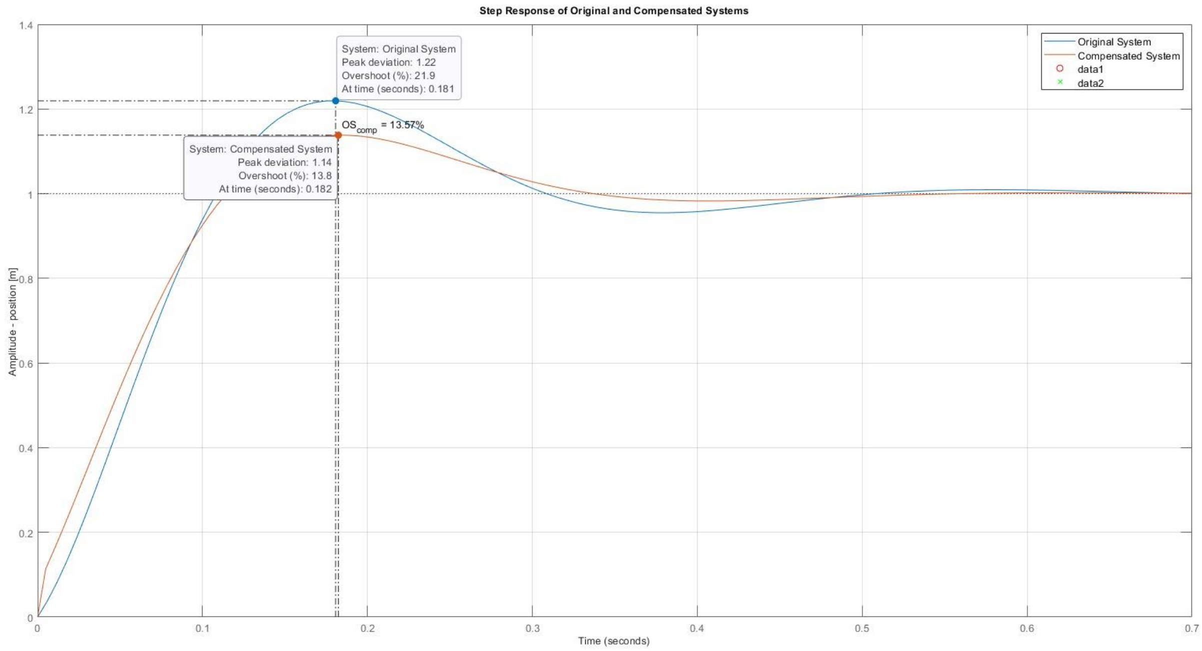
$$error = 100\left(1 - \frac{60}{60.1}\right) = 0.166\%$$

$$\omega_c = 16.9\frac{rad}{s}$$

Following is the Bode plot obtained in MATLAB for both the regular open- loop and with the addition of the lead lag compensator:



Following is the step response obtained in MATLAB for both the regular open loop and with the addition of the lead lag compensator:



$$OL: OS = 21.9\%, T_p = 0.181(s)$$
$$OL_{with\ lead}: OS = 13.8\%, T_p = 0.182(s)$$

3. In Lab

3.1 Introduction

Measuring both velocity and position control is essential for a wide range of applications. While the previous lab experiment delved into the nuances of velocity control and the fundamentals of a DC motor, this experiment pivots to emphasize position control. Position control, vital for accurate manoeuvring and precise positioning of mechanisms, demands its own unique approach. This experiment outlines objectives ranging from system identification using step and frequency responses to the design and actual implementation of control strategies. In this experiment, we learn how to tune second order transfer functions to obtain the desired specifications. We also estimate the transfer function of the DC motor using two different methods:

1. Using closed loop step response
2. Using frequency response

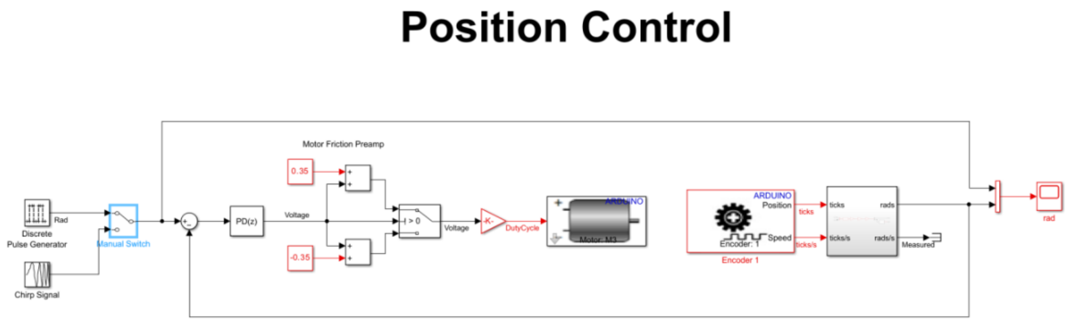
3.2 Revisiting Coulomb Friction

To find the exact value of the minimal voltage required to start the motor motion, we entered a low voltage input until there is a movement and symmetrical in both direction of motion.

We got that the minimal voltage is:

$$V_{min} = 0.52\ (V)$$

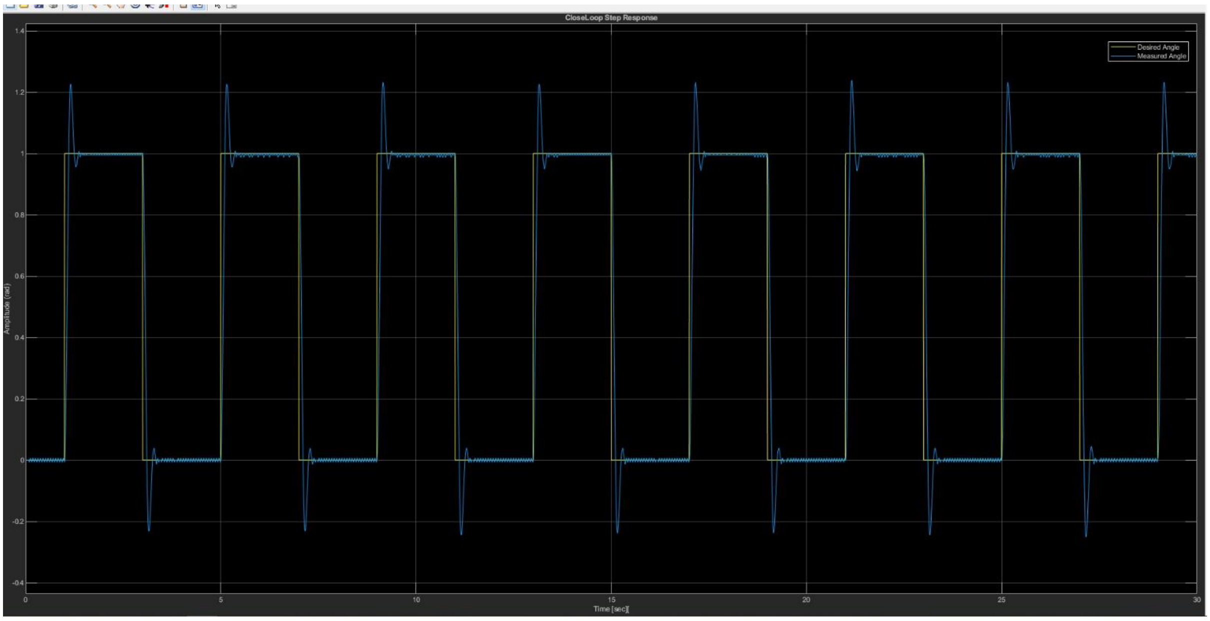
3.3 Step response modelling



The Motor Friction Preamp gains was updated with the value :

$$V_{min} = 0.52\ (V)$$

We adjust the PD controller to proportional only with $K_p = 10$ and update stop time to 30 seconds:



a. Using the closed-loop step response, estimating the DC motor's transfer function following the method previously described and practiced in the pre-lab section:

From the Matlab we got the $\xi = 0.442$

```
Damping Ratio (xi): 0.44273
```

From the plot we got - $OS = 21.2\%$,$T_p = 0.16$ (sec)

Using the formulas from the pre lab we can calculate:

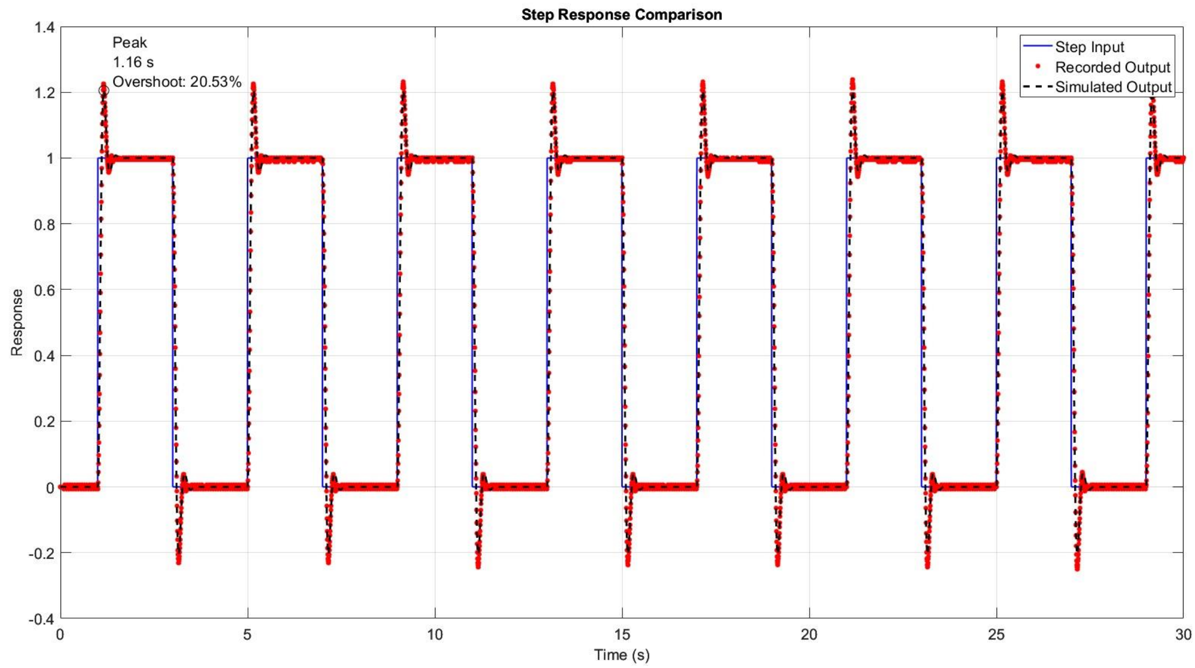
$$\omega_n = \frac{\pi}{T_p\sqrt{1-\xi^2}} = \frac{\pi}{0.16\sqrt{1-0.442^2}} = 21.89\left(\frac{rad}{sec}\right)$$

$$\tau = \frac{1}{2\xi\omega_n} = \frac{1}{2\cdot 0.422\cdot 21.89} = 0.0516$$

$$k_m = \frac{\omega_n^2\tau}{k_p} = 21.89^2\cdot\frac{0.0516}{10} = 2.472$$

$$G(s) = \frac{k_m}{\tau s+1} = \frac{2.47}{0.05s+1}$$

b. Create a plot in Matlab which contains the response from the recordings and as a simulation using the lsim function in Matlab for the identified transfer function. Mark peak time and over-shoot values for both signals:



From the plot we can see that there is some difference between the overshoot value of the actual and theoretical response. The actual response's overshoot is lower and the peak time is bigger from the theoretical response. But we can still say that the results show great similarity.

3.4 Frequency response modelling

Using the same "HexaMotor_Position_Control.slx" Simulink model. We Moved the switch to a chirp input.

a. Utilize the closed-loop chirp response to estimate the second order system's transfer function using the 'tfest' function in MATLAB. Then, estimate the DC motor's transfer function based on the approach we previously outlined and practiced in the pre-lab section.

From the MATLAB:

```
Command Window

>> Q3_4a

T =
  From input "u1" to output "y1":
          325.2
  ---------------------
  s^2 + 14.15 s + 351.4

Continuous-time identified transfer function.

Parameterization:
   Number of poles: 2   Number of zeros: 0
   Number of free coefficients: 3
   Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using TFEST on time domain data "data".
Fit to estimation data: 90.63%
FPE: 0.004475, MSE: 0.004468

Model Properties
fx >>
```

The second order system's transfer function:

$$T(s) = \frac{325.2}{s^2 + 14.15s + 351.4}$$

By compering the denominator coefficients:

$$s^2 + 2\xi\omega_n s + \omega_n^2 = s^2 + 14.15s + 351.4$$

From this we get that:

$$\omega_n = \sqrt{351.4} = 18.745$$

$$\xi = \frac{14.15}{2\omega_n} = \frac{14.15}{2 \cdot 18.745} = 0.3774$$

Using the parameters we can calculate:

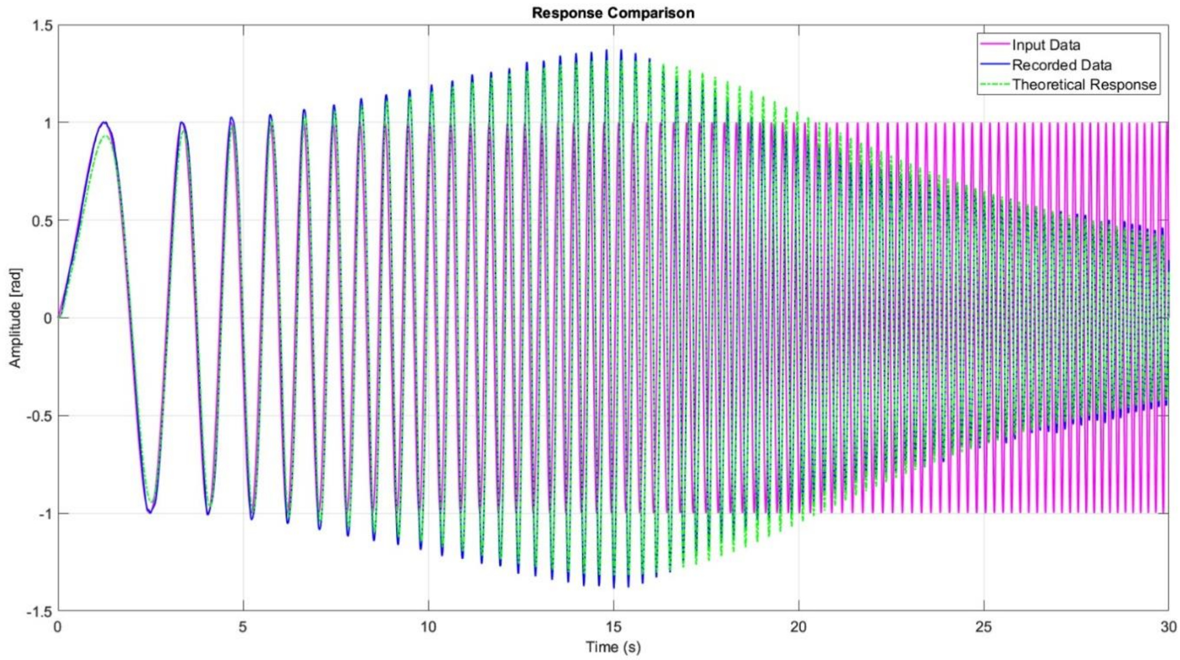
$$\xi = \frac{1}{2\tau\omega_n} \to \tau = \frac{1}{2\xi\omega_n} = \frac{1}{2 \cdot 0.3774 \cdot 18.745} = 0.0716$$

$$\omega_n = \sqrt{\frac{k_p k_m}{\tau}} \to k_m = \frac{\omega_n^2 \tau}{k_p} = 18.745^2 \cdot \frac{0.0716}{10} = 2.515$$

The DC motor transfer function:

$$G(s) = \frac{k_m}{\tau s + 1} = \frac{2.515}{0.0716s + 1}$$

b. Create a plot in Matlab which contains the response from the recordings and as a simulation using the lsim function in Matlab for the identified transfer function.



From the plot we can notice that apart from slight change in amplitude, mainly until 15 seconds, the actual response behaves just like the theoretical response.

3.5 PD controller

The transfer function obtained from the frequency response is

$$G(s) = \frac{2.515}{0.0716s + 1}$$

We use the method defined in the prelab to find k_p and k_d. From the overshoot and peak time, we calculate the values of ξ and ω_n. We got $\omega_n = 23.45 \frac{rad}{sec}, \xi = 0.45$

The formulas used:

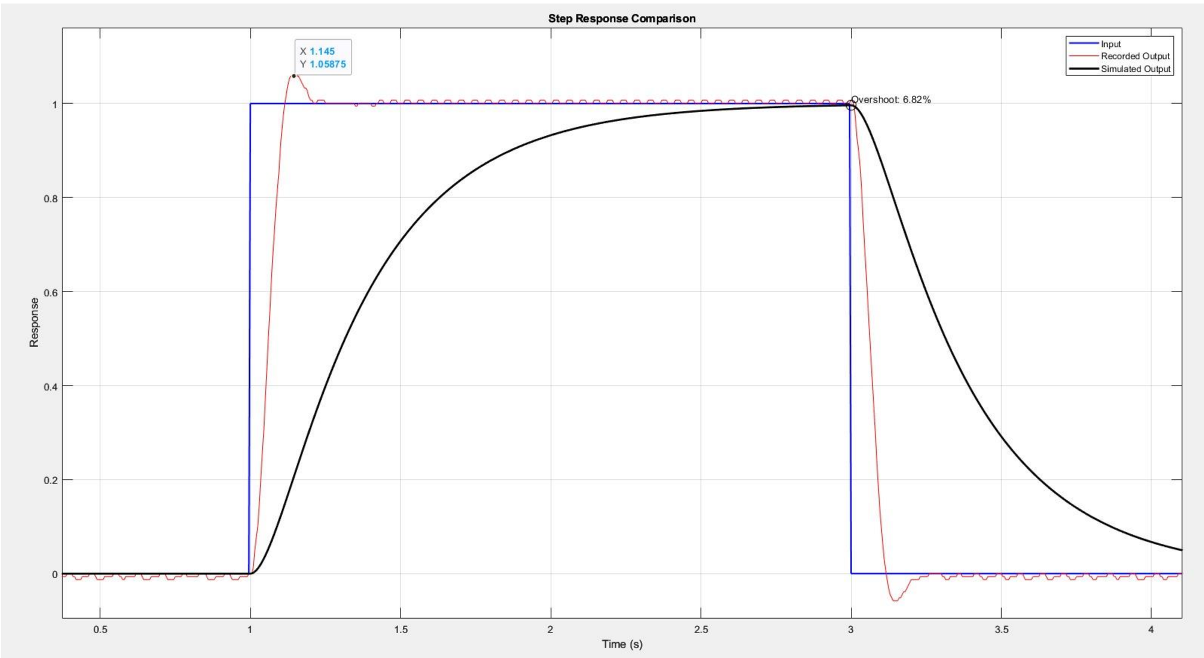
$$0.25 = 1 - \frac{\xi}{0.6} \rightarrow \xi = 0.45$$

$$T_s = \frac{\pi}{\omega_d}$$

$$\omega_d = \omega_n \sqrt{1 - \xi^2}$$

$$\omega_n = \sqrt{\frac{k_m k_p}{\tau}}, \xi = \frac{1 + k_m k_d}{2\sqrt{\tau k_m \cdot k_p}}$$

$$k_m = 2.515, \tau = 0.0716 \; so \; we \; are \; getting - \quad K_p = 15.655, k_d = 0.203$$



From the marked overshoot value and peak time, we indeed see that the os% is ~6% and the peak time is 0.145s

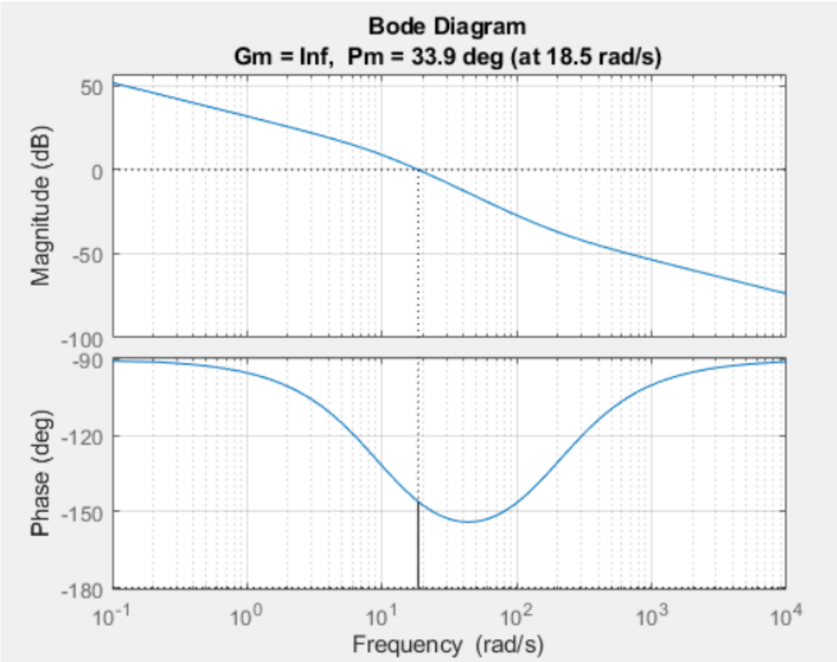
3.6 Lead-Lag Compensator

a. Present your calculations and illustrate the variations in the system's Bode plot:

First, we find the phase and gain margins of the open loop:

$$G(s) = \frac{2.5}{0.1s + 1}, C(s) = 15.655 + 0.203s$$

We got the following bode plot:



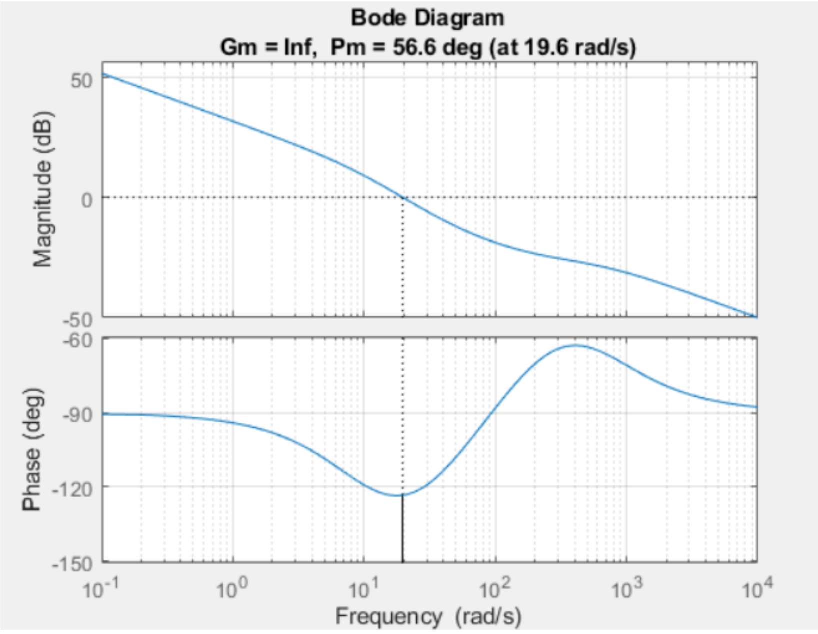
We see that the phase margin needs to be increased by 26.1 (deg). Since the phase needs to increase, we need a lead-lag network. From the plots given in the pdf, we choose:

$$u = , a =, \tau =$$

(The value of τ is obtained after you select a desirable u you divide u by ω_{co}, in our case $\omega_{c0} = 18.5\frac{rad}{sec}$. selecting u such that only the phase is affected. The magnitude should not change!) We get the following lead-lag compensation network-

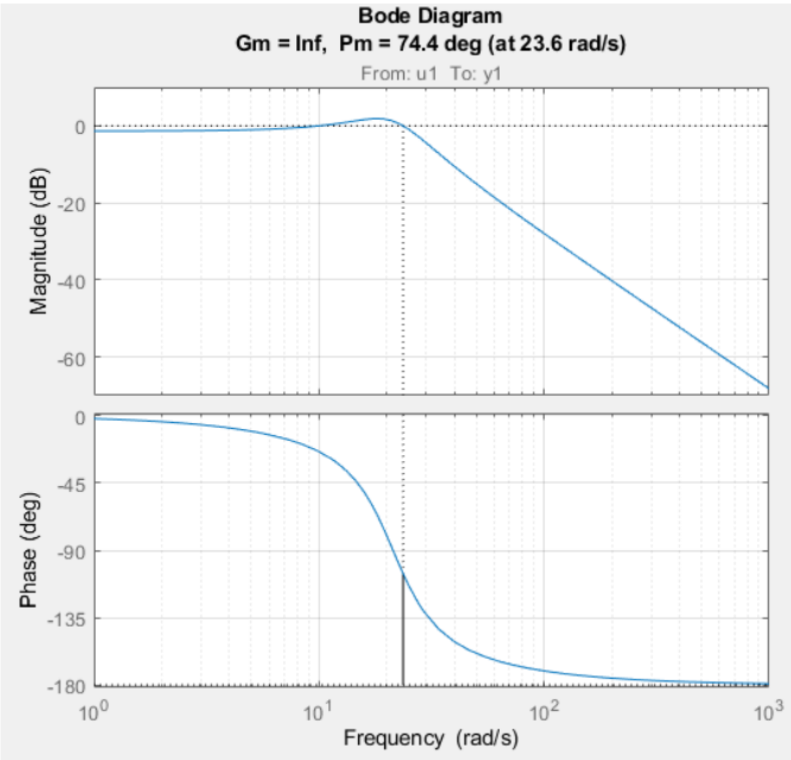
$$C(s) = \frac{0.0243s + 1}{0.00162s + 1}$$

After adding the compensation network, we get the following bode plot-

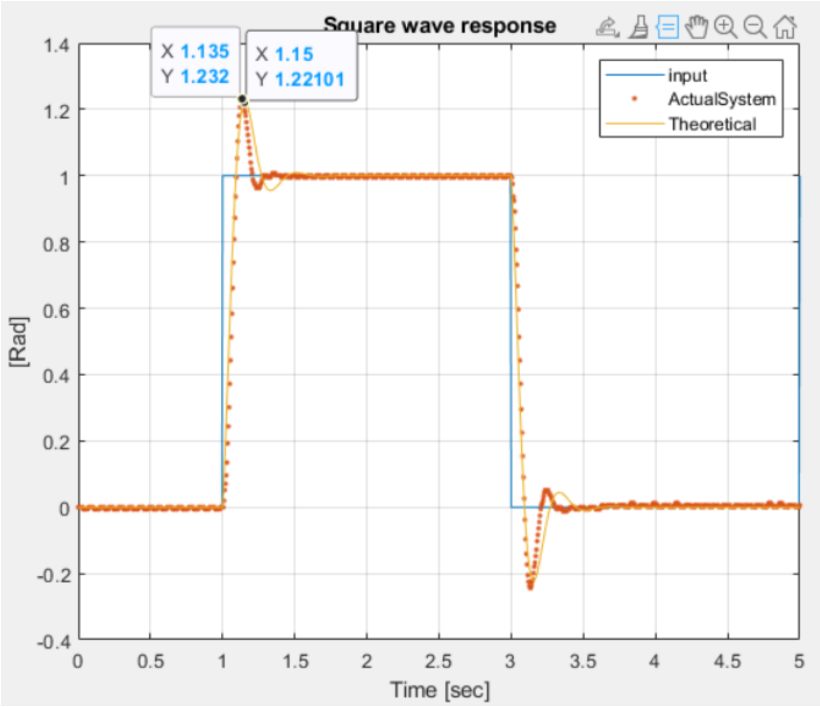


The new phase margin is very close to what is the desired phase margin (60 deg).

b. Bode plot we got from MATLAB with LeadLag Compensator (PM=74.4 [deg]):



Square wave response we got from MATLAB:



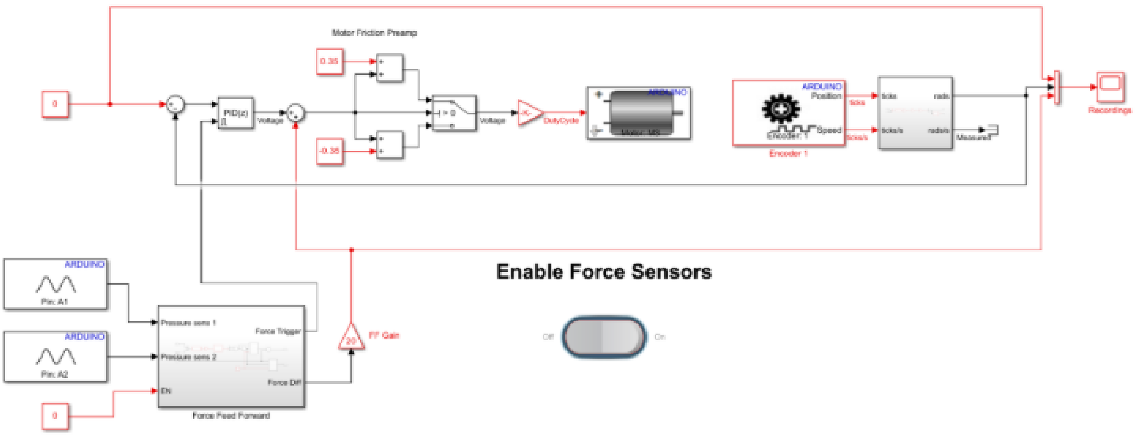
We see that the plot regarding the phase margin is incorrect. At the time of the lab, we were not able to figure out the error. However, at home, we realized that we have an error in the code. We have multiplied the transfer function by instead of dividing it by. If this error would be ignored, we have the plot which corresponds to the phase margin being around 60 (see two plots above)

The code that we should have taken –

```
% Generate plots from recordings and simulated transfer function
s=tf('s');
Kp = 15.714;
Kd = 0.0817;
C = Kp+Kd*s;
G = 2.5/(0.1*s+1)/s; % update the transfer function values
L = (0.0202*s+1)/(0.005*s+1);
T =feedback(L*C*G,1);
figure(1)
plot(ScopeData.time,ScopeData.signals.values(:,1))
hold on
plot(ScopeData.time,ScopeData.signals.values(:,2),'.');
[y,t] = lsim(T,ScopeData.signals.values(:,1),ScopeData.time);
plot(t,y);
grid on
title('Square wave response')
```

3.7 Force FeedForward example - compliant motor control

Using the "HexaMotor_Force_FF.slx" Simulink model, we ensured the switch is set to the Pulse Generator input.



Next, we replaced the rotating mass with the force sensor rod and set the gain to 20. We observed that the force sensor rod detects any attempt to move it and applies an opposing force to return to its initial position, which is set to 0 in our Simulink model.

4. Problems during the experiment

The primary issue encountered during the experiment involved the lead-lag compensator. We have included MATLAB plots to demonstrate that, had this error not occurred, the results would have aligned with expectations.

5. Conclusion

We successfully performed the position control experiment and gained a clear understanding of how to design lead-lag or lag-lead networks to adjust the phase margin. We estimated the transfer function from both the step response and the chirp response, finding them to be quite accurate.