MTE 360 Automatic Control Systems: Lab 4

Frequency Domain Loop Shaping Control of Cascaded Systems

In this laboratory, you will use the flexible drive system you identified in Lab 3 to design a set of controllers to stabilize and accurately track position commands. Your controller designs will overcome the performance and stability limitations due to the inherent mechanical flexibility by using a cascade of filters and compensators: The controller to be designed will 1. attenuate the mechanical resonance effect with a notch filter, which is a filter that can reduce the system gain at a specific frequency; 2. compensate for the phase lag in the drive response using a double lead compensator (i.e. cascade of two lead compensators to inject phase at the same frequency); and 3. achieve high tracking accuracy by canceling out the anticipated inertial and viscous friction force disturbances with feedforward control. You will verify the performance of your controller design in both experiments and simulations and conduct frequency domain stability analysis of closed-loop system.

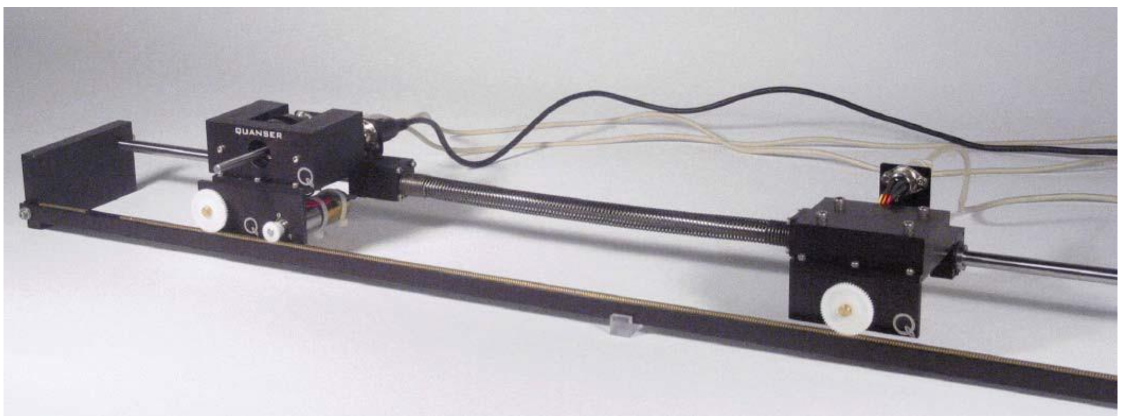

Figure 1: Flexible drive system

This is a significantly intensive project compared to your earlier laboratories. The students must keep up to date with the material covered in class to be able to complete this lab. Every group is responsible for analyzing their measured data and cannot just copy and paste the plots generated by the MATLAB scripts and Simulink models already posted on LEARN. Every group must work independently, communication between groups is not allowed.

Report format and submission: (please check laboratory general rules and report guideline first)
Every group will submit one report, as an electronic submission on LEARN
Paper-copy submissions are not required.

1 Introduction and Preliminaries

1.1 Control Design Objectives:

After having developed a model which captures the significant dynamics of the flexible drive system (in Lab 3), we will now proceed to design a tracking controller for the 2nd mass.
This will be carried out mostly using the frequency domain techniques you are covering in class and you are strongly recommended to review frequency domain analysis and controller design materials.
You will be using the "sisotool" of MATLAB, which helps to interactively design controllers using Bode plots and root locus designs tools by placing poles and zeros and modifying the loop gain, and immediately visualizing the outcome. The following specifications have been provided for the controller to be designed.

Specifications:

1. Frequency of 0dB magnitude (Crossover frequency (ω_c)) of 25 [rad/sec]
2. Gain margin (GM) of at least 2 in absolute value (or 6 dB)
3. Phase margin (PM) of at least +30°
4. Zero (minimal) steady-state error to step disturbances
5. Minimal tracking error in controlling the position of the 2nd cart (i.e. minimized $|e| = |x_r - x_2|$)

1.2 Gain Selection:

First, we will ascertain the maximum control gain K that could be used if only that gain (P-control) was applied with negative feedback from the 2nd mass. The 2nd mass position transfer function (i.e. $G_2(s)$ in Equation 1 from Lab 3) is composed of a transfer function object, then we start up the SISO Tool that results in the window shown in Figure 2.

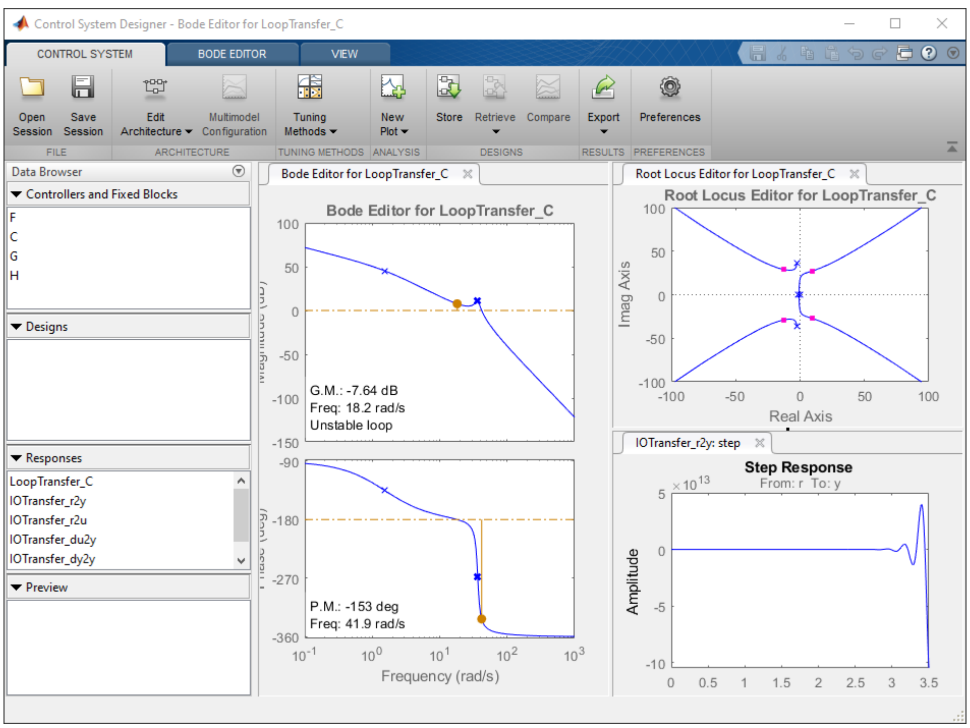

Figure 2: Original state of the loop transfer function in SISO Tool

A few changes should be made at this point to configure the tool. We select *Preferences* and change the magnitude to absolute with log scale. We can show the scale grid on the Bode plot by having a right click on the Bode editor and a checked *Grid* option. Also, we can slide the bar for the *Data Browser* and right window to increase the size of the Bode plot.

By observing the step response, we can see that the system is unstable. This can also be viewed on the root locus plot through the location of the pink squares. The root locus plot shows how the closed-loop poles of the system change as the loop gain changes from zero to $+\infty$.

To recover stability, the loop gain of the system needs to be reduced, as seen in Lab 3. One can click on the magnitude Bode plot and drag it down to reduce the loop gain. The phase margin (PM on the phase plot) will change based on the location of the crossover frequency ω_c (at which $|G_2(j\omega_c)| = 1$). Both the Root Locus and Step Response plots will change in response to lowering the loop gain.

By trial and error, we can find the maximum proportional gain which brings the closed loop system to the verge of instability (i.e. PM=0°). It can be verified analytically that this controller gain does indeed cause the loop transfer function to have a gain of 1 at the frequency where the phase shift is equal to -180°. The value of the proportional gain can be observed by right-clicking on C in the left window and selecting Open Selection, as shown in Figure 3.

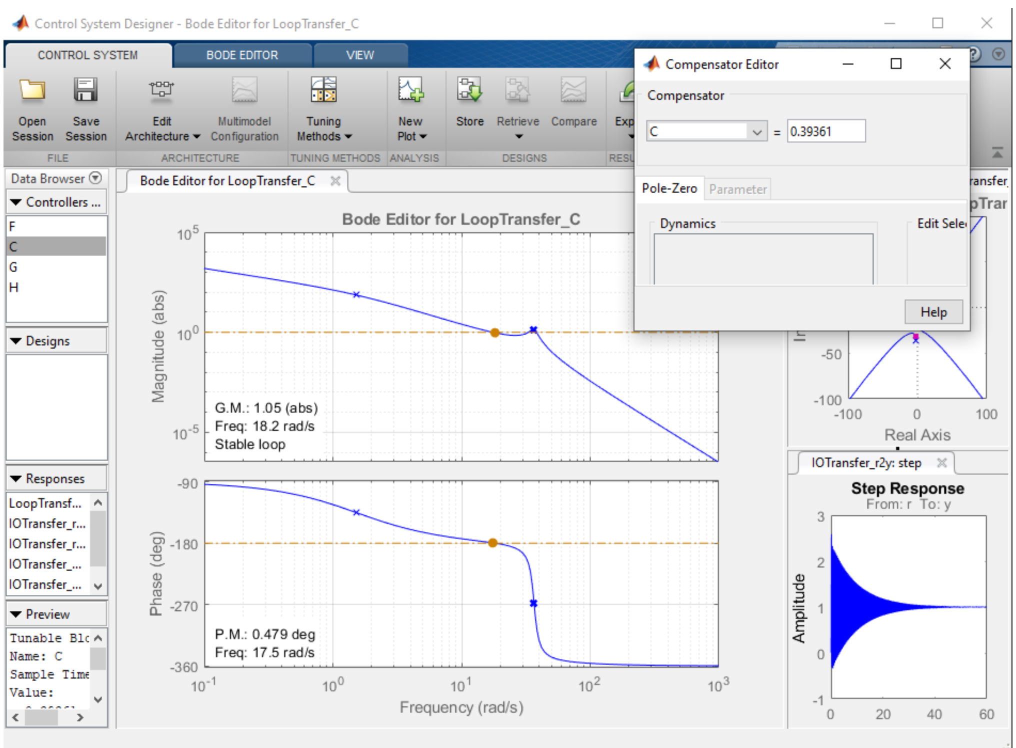

Figure 3: Critical gain for the system

Similarly, we can select a different control gain that will yield a PM = 30° as shown in Figure 4.

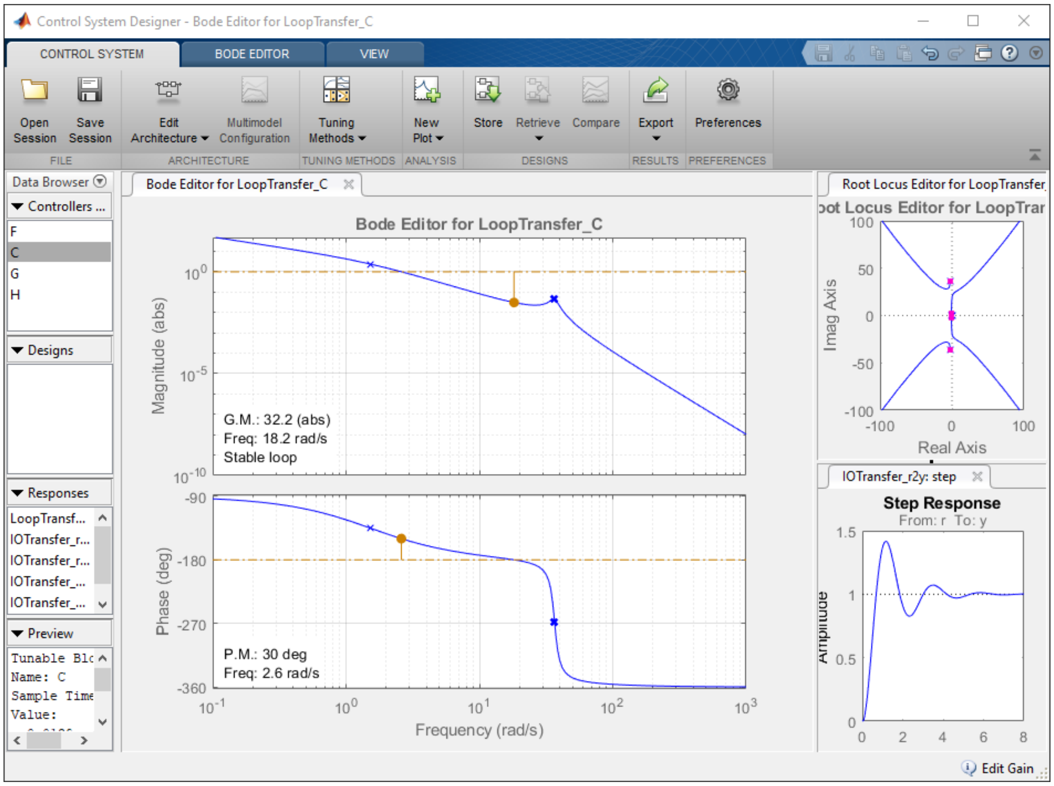

Figure 4: Phase margin of 30°

In Parts 1.3 and 1.4, you will see the most important steps in stabilizing the drive dynamics. Looking carefully at the Bode plot for $G_2(s)$ in Figure 4, one can notice two major shortcomings that need to be addressed:

1. The resonance has a jeopardizing effect on the stability. It makes the gain transition around the crossover frequency irregular, thus reducing the stability margin. Hence, the resonance needs to be compensated. For this purpose, a notch filter will be designed.

2. There is a very large phase lag in the motion of the 2nd mass $x_2(s)$ in relation to the actuation delivered to the first mass $u(s)$. The phase shift at the desired crossover frequency $(25[\text{rad/s}])$ is much lower than -180°. Hence lead compensation will need to be used to recover phase around this frequency. We will see that one lead filter will not be sufficient (due to the maximum phase a lead controller can inject). Hence, we will use a double lead filter (i.e. two identical lead filters cascaded after one another).

1.3 Notch Filter Design:

NOTE: Notch filtering (i.e. "phase-compensation",) works only in cases where the oscillatory dynamics (i.e. natural frequencies and damping ratios) are known with reasonable accuracy and do not change considerably over time. Notch filter vibration attenuation is not a robust technique. It is likely to fail if the vibratory dynamics are not accurately identified. In some cases, this may even lead to instability! Hence, this technique is used with caution in the engineering practice!

The objective of designing a notch filter is to attenuate the resonance caused by poorly damped poles. This is achieved by canceling their effect out with zeros placed at the same natural frequency and damping location and placing new poles with the same natural frequency but higher damping values (preferably $\zeta \geq 0.707$). The notch filter also needs to have a unit dc gain. The natural frequency ω_n and damping ratio ζ of the oscillatory poles $(p_{3,4})$ were identified in Lab 3.

A notch filter in the form

$$G_{notch}(s) = \frac{s^2+2\zeta\omega_n s+\omega_n^2}{s^2+2\omega_n s+\omega_n^2} \quad \text{(e.g., denominator poles have } \zeta = 1.0) \qquad (1)$$

will satisfy the above objectives. Note that the poles in the denominator have a damping ratio of 1. The frequency response of this notch filter is shown in Figure 5.

In the SISO Tool, we can incorporate the notch filter poles and zeros as part of your controller by opening the *Compensator Editor* by right-clicking on *C* in the left window and selecting *Open Selection*. From here, Right-Click in the empty space under *Dynamics* then *Add Pole or Zero* and adding *Complex Pole* or *Complex Zero* or *Notch*. You might find it easier to directly use Complex Poles and Complex Zeros rather than selecting Notch. Using the damping/natural frequency format will make it easier to directly enter values of ω_n and ζ. Notice that the SISO Tool builds the compensator in the Bode form such that the trailing gain represents the compensator's (i.e. controller's) DC gain:

$$C(s) = K\frac{\frac{1}{\omega_n^2}s^2 + \frac{2\zeta}{\omega_n}s + 1}{(1+\frac{s}{\omega_n})(1+\frac{s}{\omega_n})} \qquad (2)$$

Then, we need to readjust the gain to yield 30° phase margin. A notch filter can effectively damp resonance by attenuating the specific frequency at which resonance occurs, while allowing other frequencies to pass through unaltered.

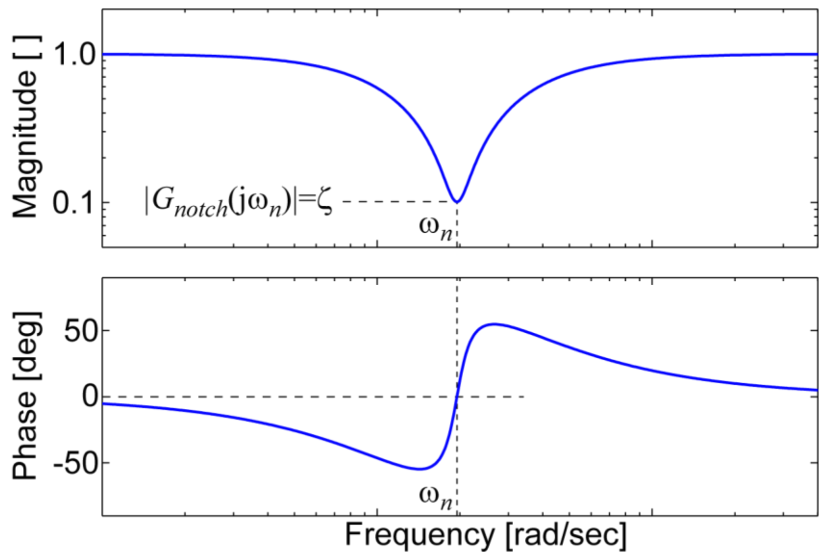

Figure 5: Frequency Response of notch filter

1.4 Lead Filter Design:

Following attenuation of the resonance with the notch filter, you will recover the phase shift required at the desired crossover frequency using two lead filters. A lead filter is realized by placing the zero at a lower frequency than the pole:

$$G_{lead}(s) = \frac{\alpha Ts + 1}{Ts + 1} \qquad (3)$$

$$\text{where: } 1 < \alpha$$

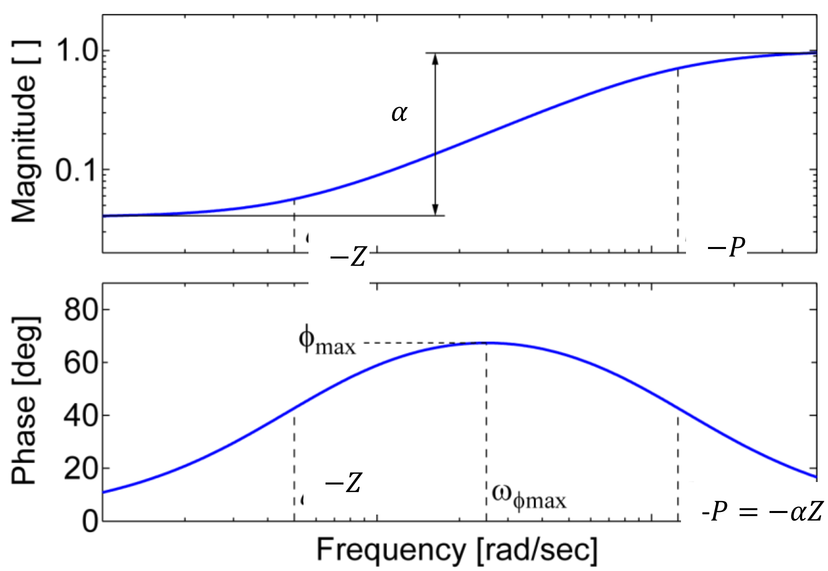

Figure 6: Frequency response of lead filter

The frequency response of this filter is shown in Figure 6. Defining the lead ratio between the pole and the zero as: $\alpha = \text{pole}/\text{zero}$, (where $\alpha > 1$), it can be verified that the maximum phase lead of this filter will be:

$$\phi_{max} = \sin^{-1}\left(\frac{\alpha - 1}{\alpha + 1}\right) = 90° - 2\tan^{-1}\left(\sqrt{\frac{1}{\alpha}}\right) \tag{4}$$

This will occur at the frequency of:

$$\omega_{max} = 1/T\sqrt{\alpha} \tag{5}$$

Hence, if we know how much phase lead we need, the zero/pole frequency ratio α can be found as:

$$\alpha = \frac{1 \mp \sin\phi_{max}}{1 \pm \sin\phi_{max}} \tag{6}$$

The next step is to examine the loop transfer function obtained for $G_2(s)$ and $G_{notch}(s)$ in the SISO Tool, and read the phase at the desired crossover freq. $\omega_c = 25\,[\mathrm{rad/sec}]$. We can also analytically calculate the phase by evaluating: $\angle\{G_2(j\omega_c)G_{notch}(j\omega_c)\}$. In order to achieve 30° PM at $25[\mathrm{rad/s}]$, we need to have a phase of $-150°$ or higher. It is good to demand another 10° when sizing the filter parameters. Hence, our desired phase at $25[\mathrm{rad/s}]$ is $-140°$.

Based on this discussion, we can determine how much phase contribution (ϕ_{max}) each lead filter needs to bring into the loop. Also, the necessary lead ratio (α) for one lead filter can be determined using Equation (6). The filter zero and pole frequencies are selected such that the maximum phase lead occurs at the desired crossover frequency using Equation (5). We can choose a slightly larger lead ratio, but this will result in a higher frequency for the pole, and therefore greater noise amplification proportional to the square of α, since we will be using a double lead filter. Therefore, it is a good idea to ensure the lead (α) ratio remains less than or equal to 25.

The lead filter poles and zeros are incorporated into the compensator in the SISO Tool. Since this is a double lead filter, each zero and pole should be entered twice. The Zero/Pole Location format is used instead of specifying the damping ratio and natural frequency. We need to readjust the gain again to yield 30° PM.

The designed compensator with the SISO Tool has the following form

$$C(s) = \underbrace{K}_{\substack{\text{DC Gain}\\\text{(adjustable)}}} \underbrace{\left[\frac{s + \frac{1}{\alpha T}}{s + \frac{1}{T}}\right]^2}_{\substack{\text{Double Lead}\\\text{Poles \& Zeros}}} \underbrace{\left[\frac{\frac{1}{\omega_n{}^2}s^2 + \frac{2\zeta}{\omega_n}s + 1}{(1 + \frac{s}{\omega_n})(1 + \frac{s}{\omega_n})}\right]}_{\text{Notch Filter}} \tag{7}$$

Considering $Z = -\frac{1}{\alpha T}$ and $P = -\frac{1}{T}$, this controller can be implemented in a slightly different format for Simulink simulations, as shown in Figure 7:

$$C(s) = K_p \underbrace{\left(\frac{s - Z}{s - P}\right)^2}_{G_{lead}(s)} \underbrace{\left(\frac{s^2 + 2\zeta\omega_n s + \omega_n{}^2}{s^2 + 2\omega_n s + \omega_n{}^2}\right)}_{G_{notch}(s)} \tag{8}$$

The exact value of K_p to maintain a crossover frequency of $\omega_c = 25[\mathrm{rad/s}]$ can be calculated using the exact analytical expressions for $G_2(s)$, $G_{notch}(s)$, and $G_{lead}(s)$ instead instead of using the value adjusted for K in the SISO Tool as follows

(9)

$$K_p = \frac{1}{|G_2(j\omega_c)| \cdot |G_{notch}(j\omega_c)| \cdot |G_{lead}(j\omega_c)|^2} \quad , \text{ where } \omega_c = 25 \text{ [rad/sec]}$$

This designed controller will be tested in simulation and on the experimental setup in Section 2.2. The simulation should look like the one in Figure 7.

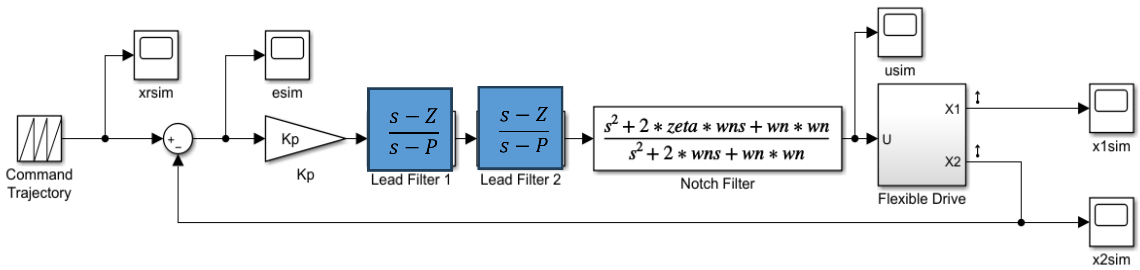

Figure 7: Simulation model for control with notch and double lead compensation.

1.5 Feedforward Controller Design:

As a final improvement to our control system, we will cancel out the anticipated inertial and viscous friction forces in feedforward, which will help further achieve design objective #3 (minimal tracking error). By making use of the commanded velocity and acceleration profiles and the known mass and viscous damping values, this step becomes straightforward to implement and is an effective way to further improve the tracking performance of servo systems when the feedback gain, hence the crossover frequency, cannot be increased anymore due to limitations caused by sensor noise, actuator saturation, or un-modeled dynamics interfering with the loop stability.

Feedforward control assumes that the dynamics being cancelled out are well known and do not change over time. In spite of its effectiveness, it is not a robust compensation technique and should be used with caution. **You should never try to cancel out poles or zeros with unstable or poorly damped characteristics, as this will lead to either highly oscillatory behavior, or instability!**

Feedforward action is incorporated into the controller design as shown in Figure 8. The commanded velocity and acceleration are estimated by taking smooth derivatives of the reference position values given in Figure 9 Figure 13. The smoothing pole for the derivative blocks is assigned to have a frequency of $q = 1000[\text{rad/s}]$. We can use a transfer function to implement the derivative blocks. The mass and viscous friction values m_1, m_2, b_1, b_2 were already estimated in Lab 3. The feedforward control signal is injected right before the notch filter, so that discontinuities caused by sharp transitions in the acceleration profile have minimal influence in exciting the structural vibrations of the drive.

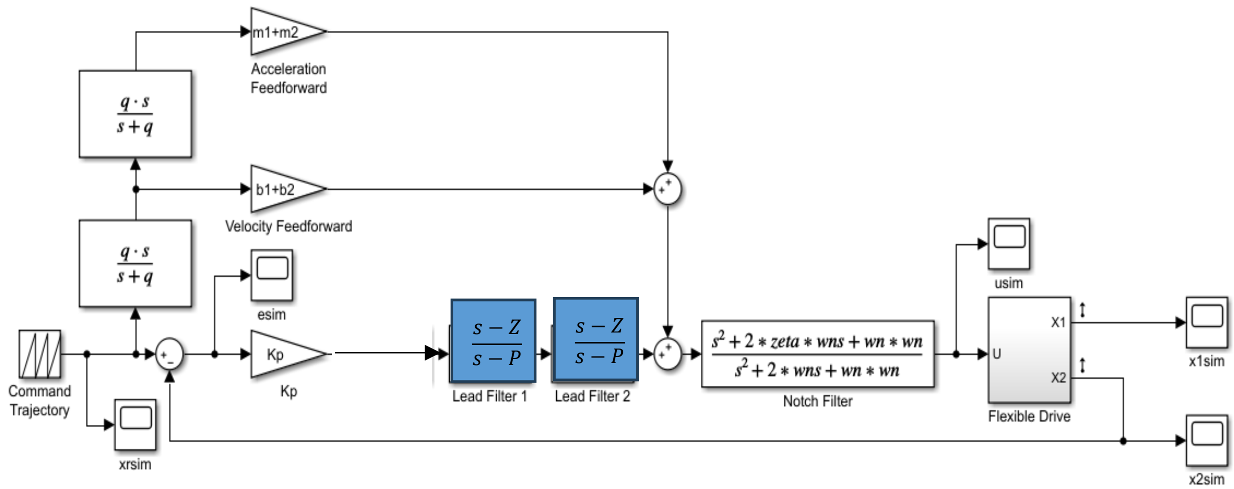

Figure 8: Simulation model for notch, double lead, and feed-forward compensation.

The feedforward controller, if designed to have stable poles, will not affect the stability of the closed loop system. Hence, earlier calculations regarding the crossover frequency and stability margins will still hold as the loop transfer function will be the same. The feedforward controller will only modify the command response, in order to improve the tracking. <u>It is important not to feed discontinuous motion commands to a control system with feedforward action. This will result in amplification of the command discontinuities, resulting in highly rough motion. That is why we use the smooth trajectory shown in Figure 13 instead of a step input or a square wave input.</u>

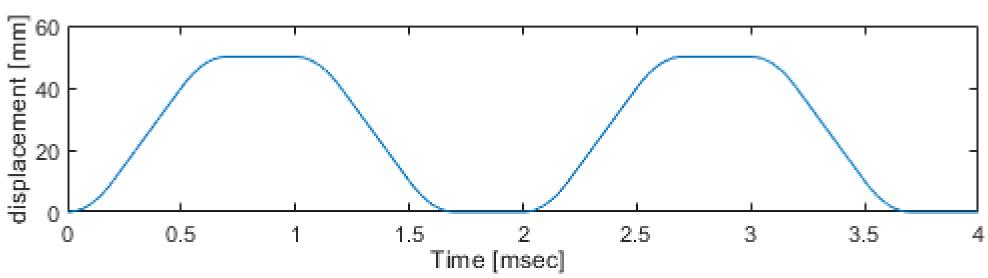

Figure 9: Smooth Trajectory.

2 In-Lab

2.1 Designing the controller using SISO tool:

Please review Part 3.1 of this document to collect everything required for the report while completing the design. Follow the steps below to design the controller as discussed in Part 1

1. Construct the 2nd mass $G_2(s)$ from Equation 1 in Lab 3 as a transfer function object and start up the SISO Tool by entering the identified parameters from Lab 3 in *SISO_Analysis.m* and running it.
2. By trial and error, find the maximum proportional gain that can be used, which brings the closed loop system to the verge of instability (i.e. PM=0°). Record a screen capture similar to Figure 3.
3. Select a different control gain that will yield a PM = 30°. Record the value of this gain. Record a screen capture similar to Figure 4.
4. Incorporate the notch filter poles and zeros as part of your controller as discussed in part 1.3. Readjust the gain to yield 30° phase margin and provide another screen capture of the SISO tool with Bode plot and command step response plots clearly visible.
5. Incorporate the lead filter poles and zeros into your compensator in the SISO Tool as discussed in part 1.4. Readjust the gain to yield 30° PM and take a screen capture if these specifications are met, otherwise iterate on the controller design to achieve these specifications.
6. Determine the equivalent value of the proportional gain K_p which will bring the loop transfer function crossover to $25[\mathrm{rad/s}]$.
7. Once the controller specifications for crossover frequency (#1), gain margin (#2), phase margin (#3), and zero (minimal) steady state error (#4) are met, capture, and present the Bode plot, as well as the command and step responses.

In the next sections, you will implement the designed controller on simulation and on the experimental.

2.2 Simulation & Experimental Tracking: Notch and Double Lead Compensation:

Use *flexibleDrive_NotchLead.slx* to test the designed controller in Parts 1.3 and 1.4 in simulation (see Figure 7) before implementing it on the experimental setup. Make sure to update the controller gain K_p, and the compensator parameters. Then, implement the designed controller on the experimental setup, as shown in Figure 10.

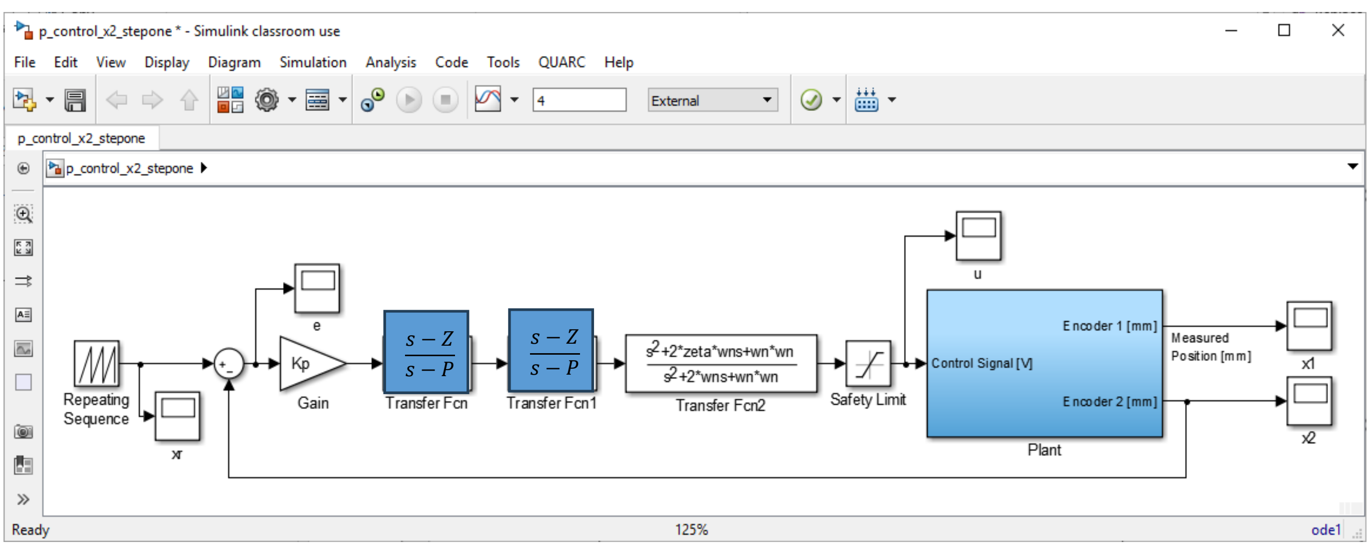

Figure 10: Implementation of notch and double lead filters

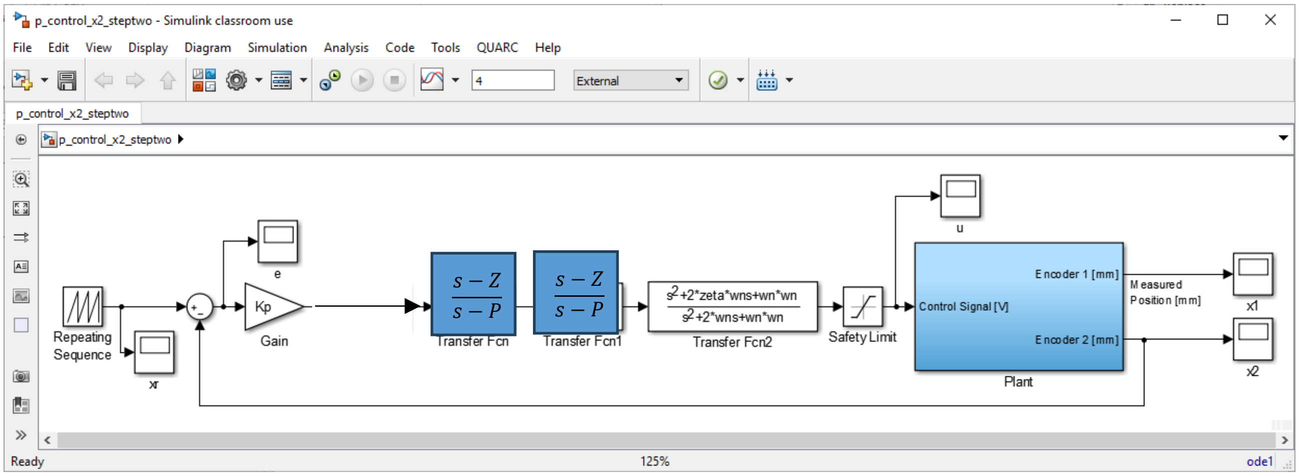

Figure 11: Implementation of notch, and double lead filters

It is your responsibility to prepare and keep track of your real-time files, experimental data, and MATLAB scripts. Run the tracking experiment and capture data for 4[sec] (two forward and backward motions), as you did in Part 2.2.

2.3 Simulation & Experimental Tracking: Notch, Double Lead, and Feedforward Compensation:

Use *flexibleDrive_compensated_feedforward.slx* to test the proper operation of the feedforward controller you designed in part 1.5 on the simulation block (see Figure 12) before trying it out on the experimental setup. You may preserve all other control components you had designed earlier. Then, implement your final controller design on the experimental setup and run the tracking experiment once again.

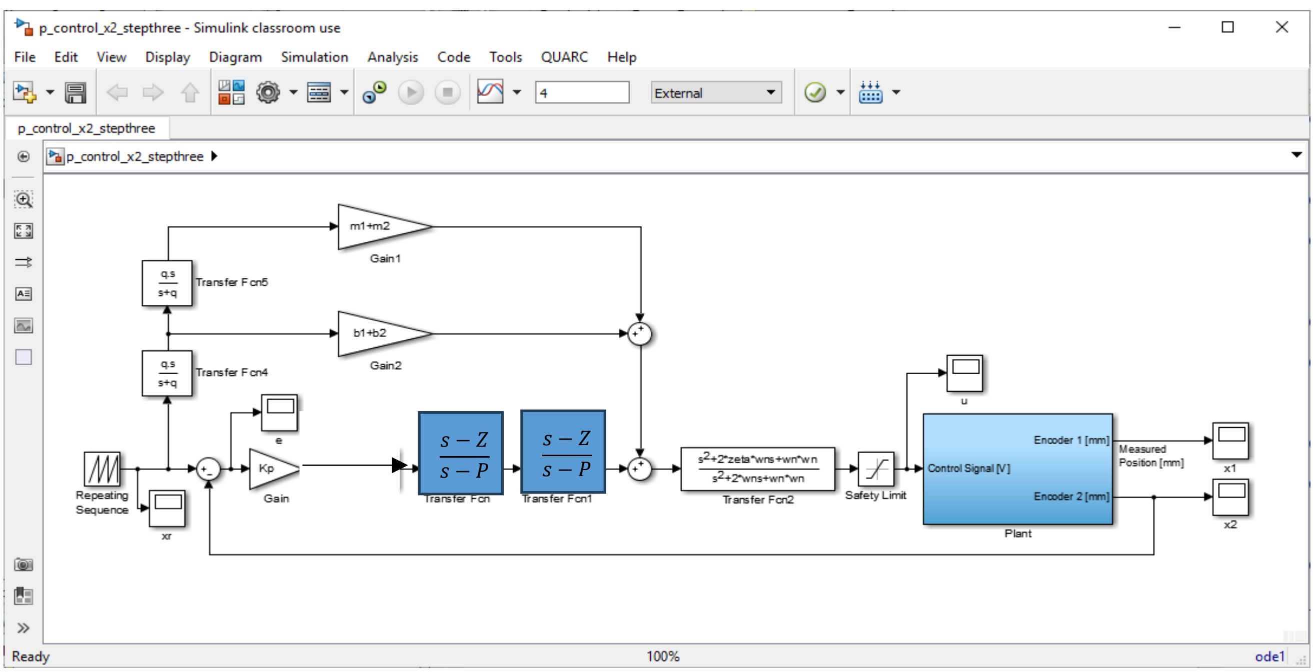

Figure 12: Implementation of notch, double leadfilters and feedforward controller

Collect commanded and measured position and control signal data for two forward and backward movements with the smooth trajectory (i.e. for 4[sec]), as you did in Part 2.3.

3 Post-Lab

3.1 Controller Design Details:

Present the screen captures and observations as described in the controller design section, Part 2.1.

- Screen capture of original system at the critical gain and the value of the critical gain
- Screen capture of system with PM = 30° and the gain value for PM = 30°
- Crossover frequency value, whether spec, #1 is satisfied, intermediate conclusion regarding using only P-control, and which controller design specifications are met with only P-control.
- Screen capture of SISO tool with the notch filter added, the values used for ω_n and ζ should be reported.
- Comparison of the notch filter controller and P-control, including crossover frequency, GM, command step rise time, and overshoot. What is the main contribution of the notch filter?
- For the lead filters, report the values of ϕ_{max}, α, T, Z, P
- Provide a screen capture showing that specifications #1, #2, and #3 have been met.
- What is the new crossover frequency after adding the lead filters?
- Report the value of K_p that was used in the lab with the double lead and notch filters.
- Report the features of this design including, Gain Margin (GM), Phase Margin (PM), command rise time, command overshoot, disturbance max value, and disturbance steady state.
- Report the values of m_1, m_2, b_1, b_2 used in the feedforward design.

3.2 Simulation & Experimental Results: Notch and Double Lead Compensation:

From your experiment in Part 2.2, record the simulated position and control signal response. Overlay the experimental and simulated tracking results on top of each other using different line styles or preferably colors. Present your results in the following format:

1) Cart 1 position: x_1 versus t (on the top)
2) Commanded trajectory and cart 2 position: x_r and x_2 versus t (2nd from the top)
3) Cart 2 tracking error: $e = x_r - x_2$ versus t (2nd from the bottom)
4) Control signal: u versus t (at the bottom).

Comment on your observations in terms of the effectiveness of this controller in achieving good tracking accuracy. Also comment on the similarities and/or discrepancies you notice between the simulation and experimental results.

3.3 Simulation Result: Notch, Double Lead, and Feedforward Compensation:

Repeat the instructions of Part 3.2 with the notch and lead filters added using the results recoded from the experiment in Part 2.3.

3.5 Final Comments:

Compare the tracking results you obtained with P-control (from Lab 3) to those you have achieved by developing a notch and double lead filter (part 2.2), and a feedforward controller (Part 2.3). State your comments on what you have achieved during this Lab.