ME3534 - Lab 5 - Advanced Compensator Design and Validation

Name: Zachary Rohloff

Date: 4/23/26

Lab Bench #: 3

This LiveScript documents the results of the pre-lab, in-lab, and post-lab processing for ME3534 Lab 5. A PDF version of this document will serve as your Lab 5 report. In order to receive credit for your Lab 5 submission, you must convert the LiveScript document to a PDF and upload both to Canvas as well as all three .mat files collected from your experimental testing.

In this lab, you will have the opportunity to apply material you learned in ME3534 to evaluate more complicated but very common feedback compensator designs.

Task 1: Load the motor Model from Lab 3 (Pre-Lab)

[2 pts] Load and **display** the continuous-time motor model from Lab 3. You will need to display the LTI object in the LiveScript in order to receive full credit. To do this, simply type the name of your model (e.g. **`Pmotor`**) into a LiveScript code block, and when the LiveScript is executed, Matlab will display the s-domain model in the LiveScript. Do NOT use the **`disp()`** function.

NOTE: If your motor model from Lab 3 is not correct or is not accurate enough, you will not be able to complete Lab 5. It is important to fix any errors you may have from the Lab 3 s-domain motor model **before** starting this lab.

```
Pmotor =

            0.05
  -------------------------
  0.0003229 s^2 + 0.01046 s

Continuous-time transfer function.
Model Properties
```

Task 2a: PI Compensator Design - Simulation Study (Pre-Lab)

For this design, you will evaluate the performance of a Proportional + Integral (PI) feedback compensator to meet all of the performance requirements. From Lab 4, we saw that a proportional gain of KP=5.0 gave better performance (i.e. higher bandwidth) than KP=1.0, and it did not have as "noisy" of a closed-loop response as for KP=10. You will use **KP=5.0** for this task. You will need to use a guess-and-check approach to determine the integral control gain.

You are welcome to use the PIDTUNE function to give you a reasonable starting point for both the KP and KI gains; however, the result from PIDTUNE will NOT be sufficient for this task, so you will need to make an adjustment to KI. This is an iterative design approach where you will look at the time-domain and frequency-domain results, then iterate on the gain selection.

Mandatory Design Requirements (for the Closed-Loop Simulation):

1. $-1.65V \leq V_{\text{fbk}} \leq +1.65V$
2. $-1.65V \leq V_{\text{cmd}} \leq +1.65V$
3. $\text{NIAE} < 2.0$
4. zero steady-state error for steps, ramps, and dwells (visual inspection only!)
5. $5\,\text{Hz} \leq 3\text{dB Bandwidth} \leq 6\,\text{Hz}$
6. $\text{Gain Margin} \geq 40\,\text{dB}$
7. $\text{Phase Margin} \geq 30^{\circ}$

[4 pts] Define and display your continuous-time controller design. Save your PI controller LTI object to a .mat file for the experimental evaluation.

```
C_PI =

  5 s + 16.8
  ----------
      s

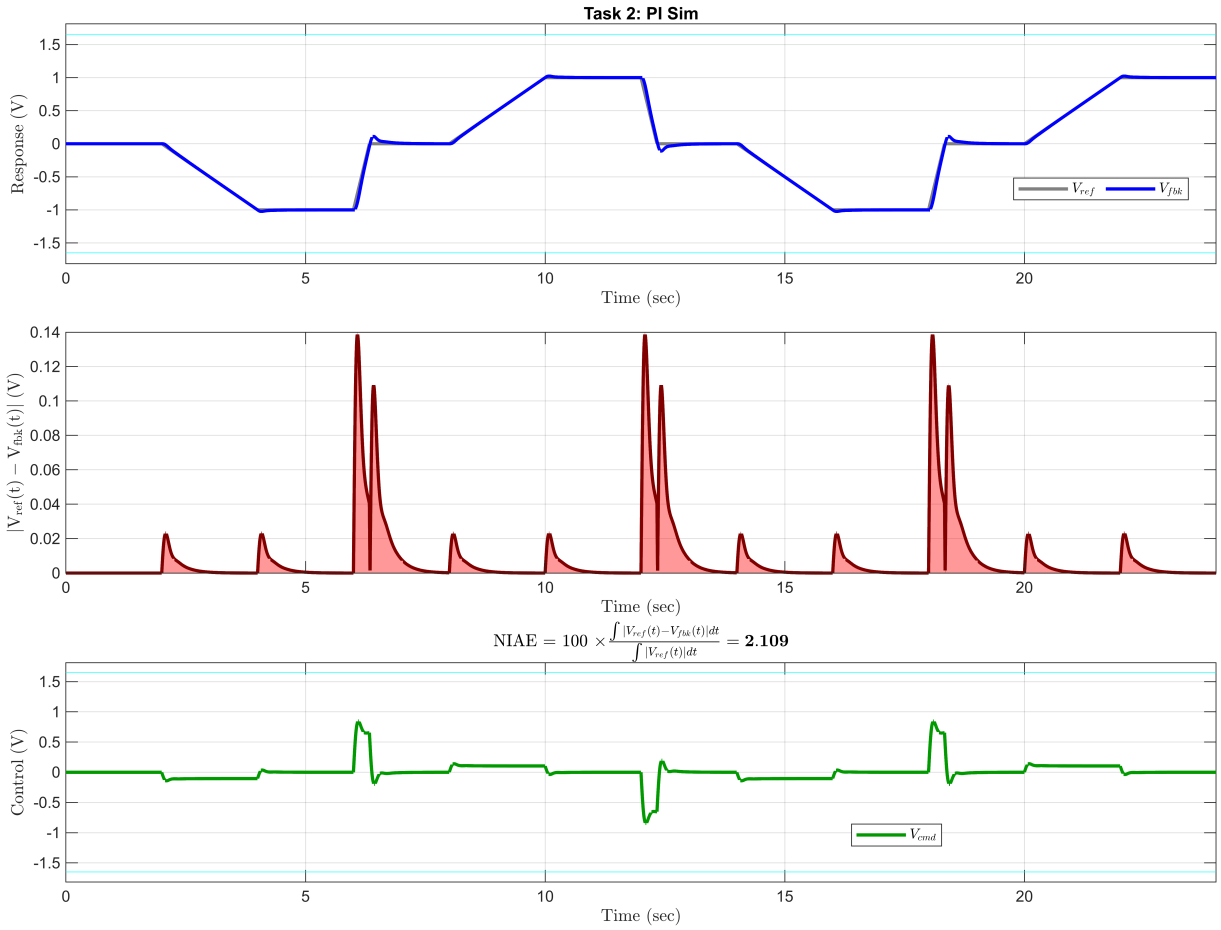
Continuous-time transfer function.
Model Properties
```

[2 pts] Compute the frequency of the compensator zero (roots of numerator) with units of Hz

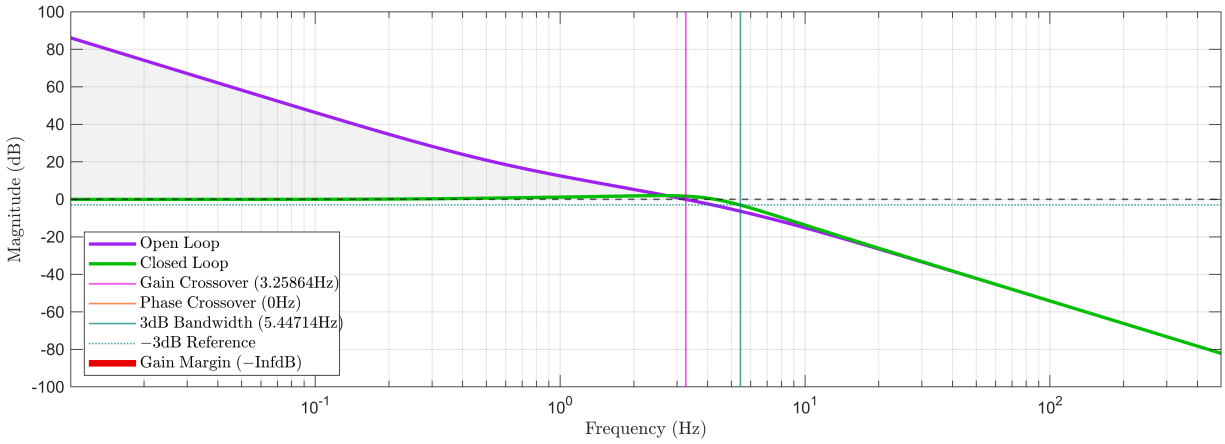
```
PI Zero Frequency: 0.5348 Hz
```

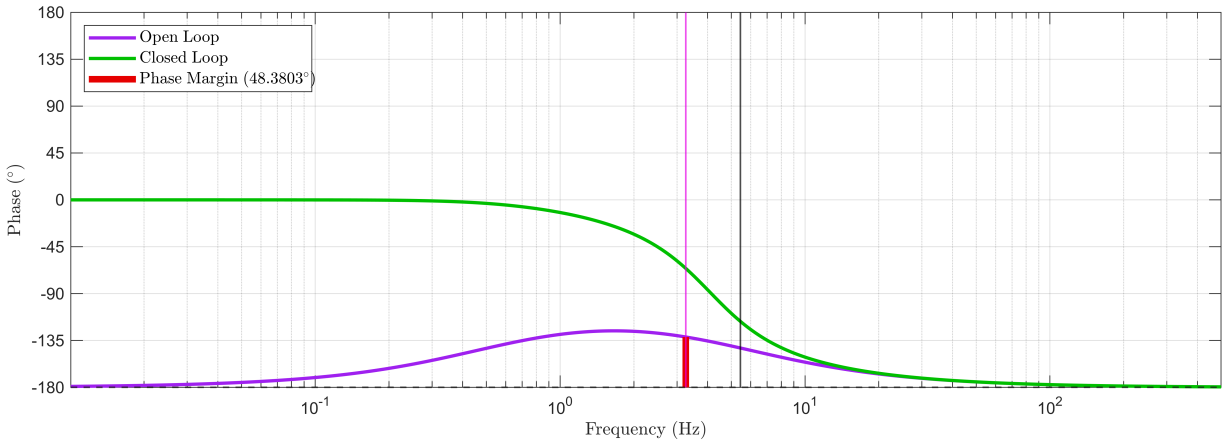
[3 pts] Construct closed-loop models for simulating the time response as well as the control signal, then simulate the closed-loop time responses of the simulated feedback signal (`fbk_sim`) as well as the simulated control signal (`cmd_sim`). You will need to load the reference profile from `lab5_reference.mat`. Note: This is exactly the same operations you did in Lab 4 but using the PI controller designed above.

[3 pts] Generate the time-domain (`iae_analysis.p`) and frequency domain (`freq_domai_analysis.p`) performance metrics and use the plots to answer the questions below.



$$\text{NIAE} = 100 \times \frac{\int |V_{ref}(t) - V_{fbk}(t)| dt}{\int |V_{ref}(t)| dt} = \mathbf{2.109}$$





Does your compensator design result in zero steady-state error for both steps (dwells), and ramps?

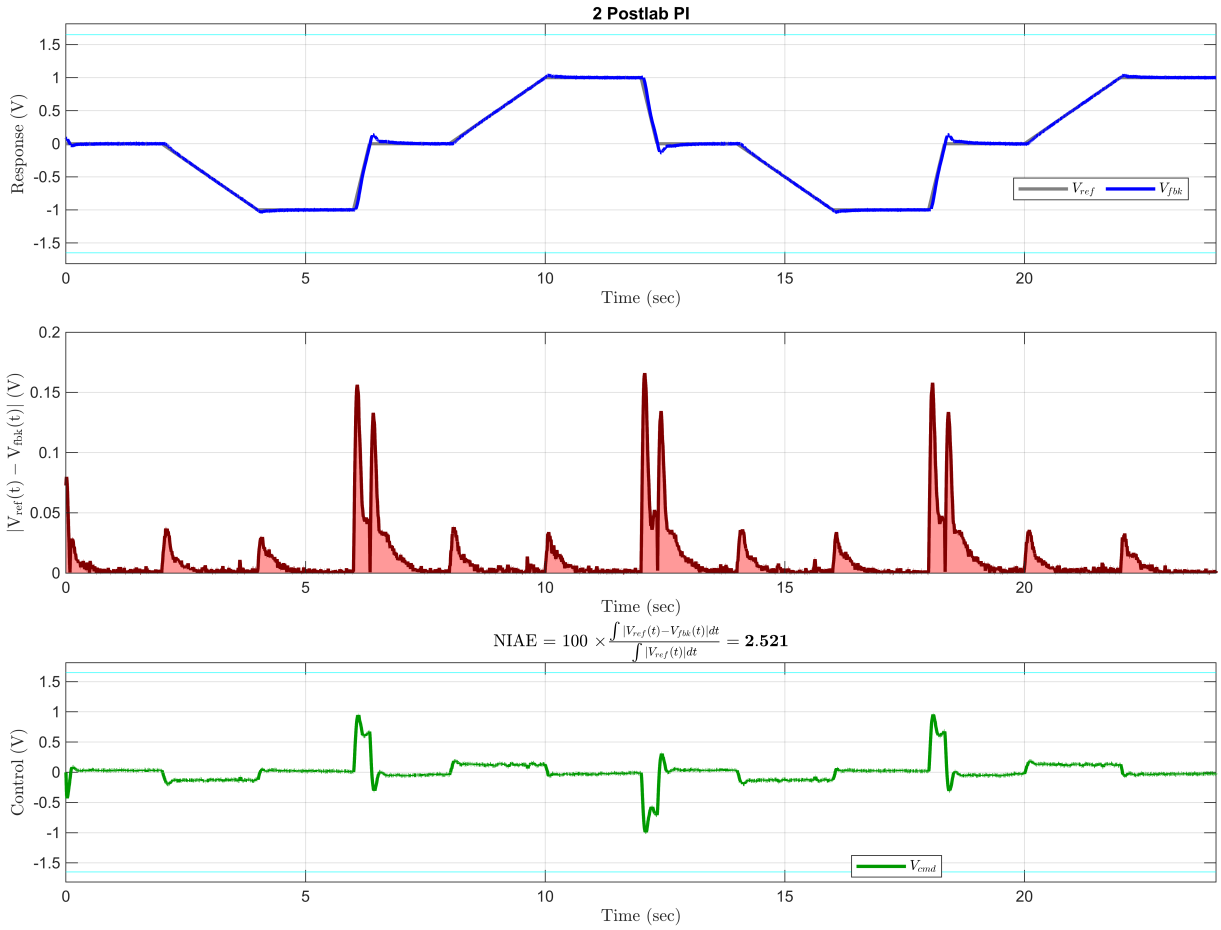
Does your compensator design meet all seven design requirements? (It is required to before you can test in the lab!)

Use the graphics cursor in the time-domain Response plot above to determine the maximum overshoot during the step inputs?

Does the closed-loop frequency response have any magnitude amplification (i.e. gain > 0dB)? If so, what is the approximate frequency range where the magnitude is increased?

Task 2b: PI Compensator - Experimental Validation (Post-Lab)

[2 pts] Load the .mat file containing your experimental data for the PI compensator design, then call the **`iae_analysis`** function to generate the time-domain analysis plots.



[2 pts] Answer the following questions based on the experimental iae_analysis plots compared to the simulated iae_analysis plots

Does the response, error, and command time-domain signal match the general shape and amplitude between the experiment and simulation?

Yes it looks to me almost as if the original graph got transposed up from the origin by a base level of error, there is also much more noise in the error graph. The error graph also clearly has a higher peak as it has been re-scaled by matlab's auto scaling.

Responce looks identical to me but there is likley a small difference.

Since the experimental data has additional noise on it, we would expect the experimental NIAE to be larger than the simulated NIAE. What is the percent increase in the experimental NIAE?

```
PI Experimental NIAE percent increase: 19.53%
```

Task 3a: PID Compensator Design - Simulation Study (Pre-Lab)

For this design, you will evaluate the performance of a Proportional + Integral + Derivative (PID) feedback compensator to meet all of the performance requirements.

If you use PIDTUNE, beware that not only do you need to use the correct PID compensator (see lab5_instructions), but you also need to convert the LTI object to a conventional transfer function using the TF() function in Matlab. The me3534_lab_gui will not accept PID LTI objects generated by PIDTUNE unless you convert them to conventional TF objects first. Also note that PIDTUNE will not give you a working solution without a lot of extra modification on your part. It is probably easier to just tune the PID gains by hand.

The preferred design method is to use a guess-and-check approach to determine all three control gains. Here are some guidelines to get you started. Begin with the following gains:

- use the same KP you used for Task 2a (you may want to make a small adjustment to this later)
- use half of the same KI you used for Task 2a
- begin adjusting KD from 0 to no more than 0.1 and check the responses

Do NOT choose KD>0.1 for this design. Remember that deritivative gain makes the loop very "sensitive" to noise, so you don't need much to achieve the goal of reducing the overshoot that you saw in Task 2.

As described in the lab5_instructions.pdf document, you will also need to implement a "proper" version of the PID controller if you want to use LSIM to simulate Vcmd! The lab5_instructions.pdf document shows several examples of how to make a PID compensator "proper". You are strongly recommended to design a 1st order low-pass filter with a break frequency of 60 Hz and cascade this result with your PID compensator LTI object. Don't forget that poles and zeros in transfer functions are required to have units of $\left(\frac{\text{rad}}{s}\right)$ and not Hz!

Mandatory Design Requirements (for the Closed-Loop Simulation):

1. $-1.65V \leq V_{\text{fbk}} \leq +1.65V$
2. $-1.65V \leq V_{\text{cmd}} \leq +1.65V$
3. $\text{NIAE} < 4.0$ (Note that this is larger than the requirement for Task 2)
4. zero steady-state error for steps, ramps, and dwells (visual inspection only!)
5. $5\,\text{Hz} \leq 3\text{dB Bandwidth} \leq 6\,\text{Hz}$
6. $\text{Gain Margin} \geq 40\,\text{dB}$
7. $\text{Phase Margin} \geq 50^{\circ}$

[4 pts] Define and display your continuous-time controller design. Save your PID controller LTI object to a .mat file for the experimental evaluation.

```
C_PID =

  18.85 s^2 + 1885 s + 3167
  -------------------------
         s^2 + 377 s

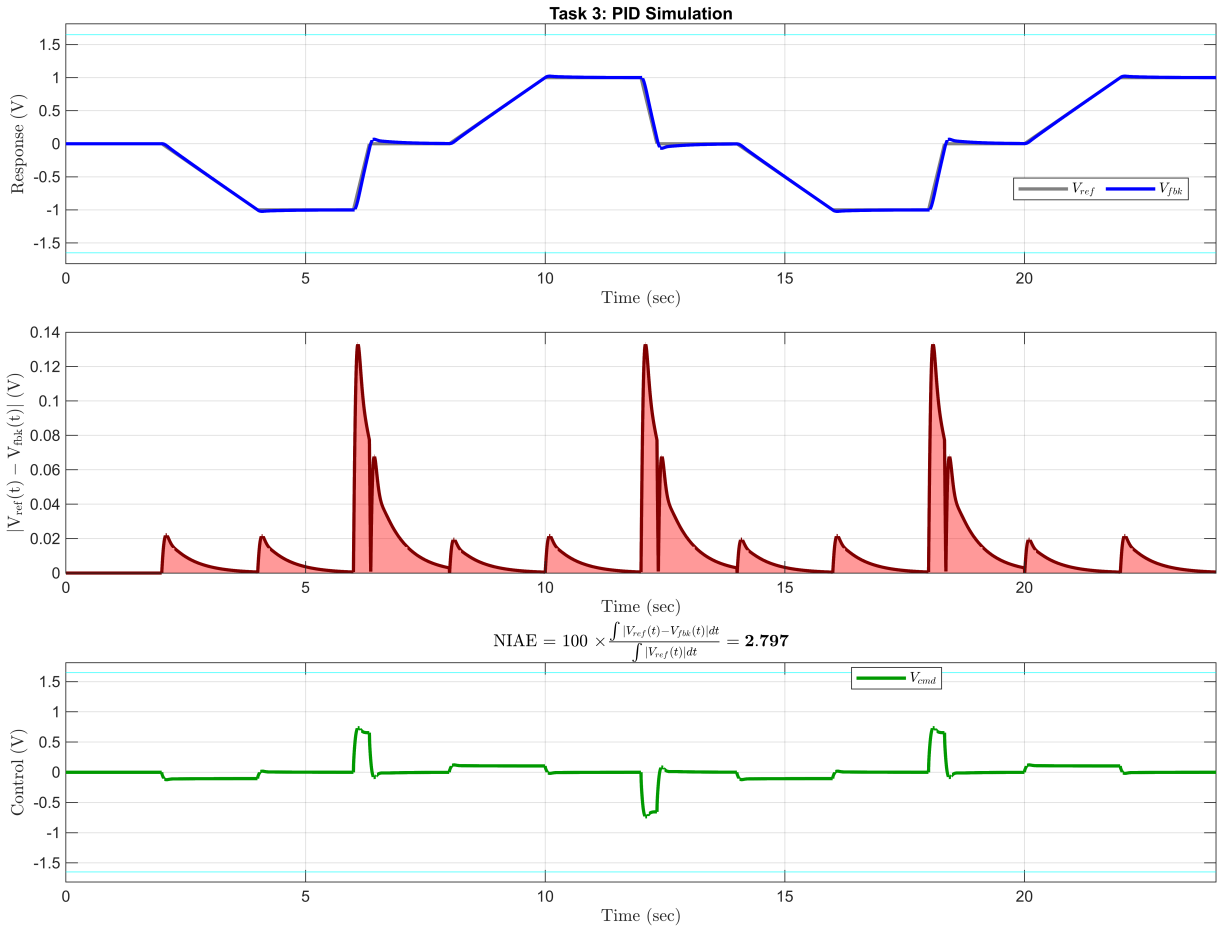
Continuous-time transfer function.
Model Properties
```

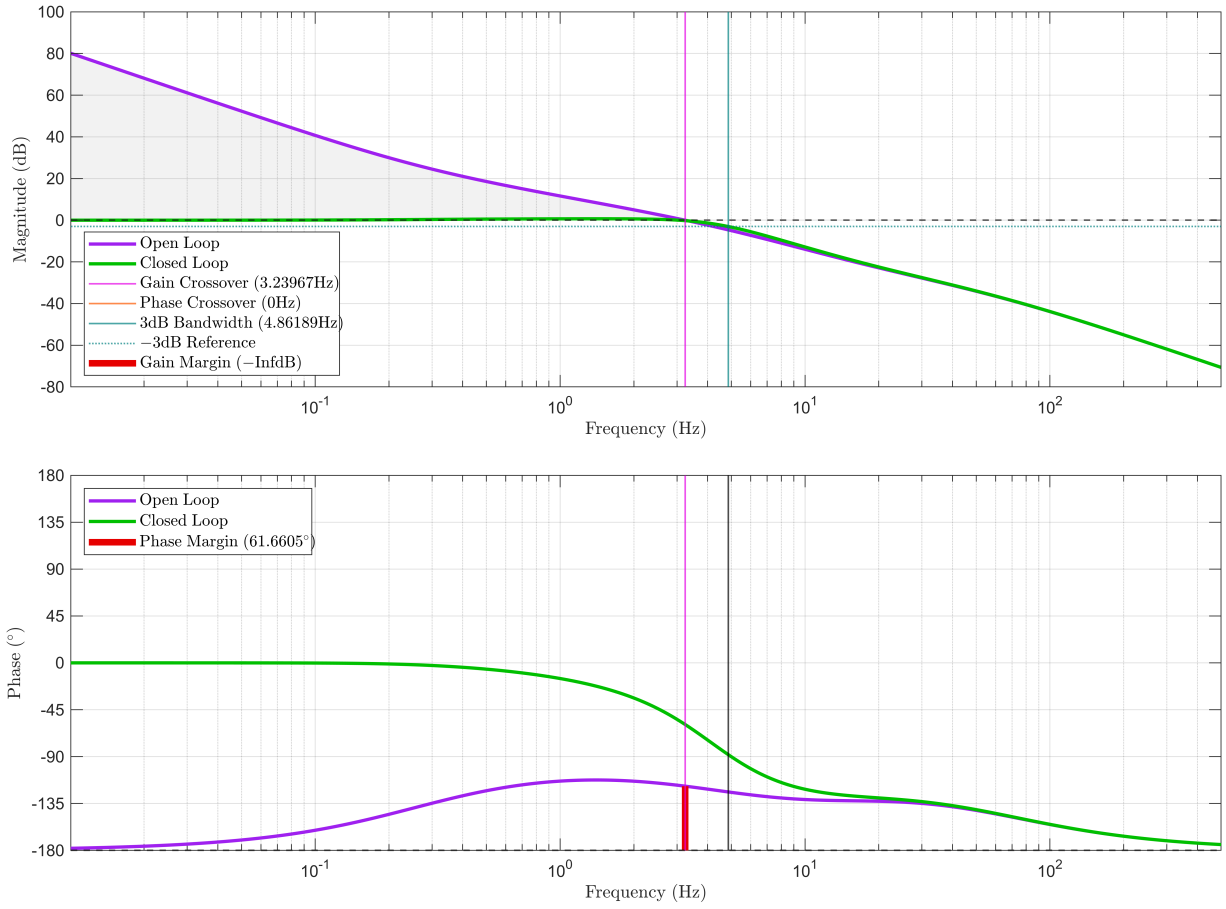
[2 pts] Compute the frequencies of the compensator zeros (roots of numerator) with units of Hz

```
   15.6434644316928
  0.272029877496752
```

[3 pts] Construct closed-loop models for simulating the time response as well as the control signal, then simulate the closed-loop time responses of the simulated feedback signal (fbk_sim) as well as the simulated control signal (cmd_sim). You will need to load the reference profile from reference.mat. Note: This is exactly the same operations you did in Lab 4 but using the PID controller designed above.

[3 pts] Generate the time-domain (`iae_analysis.p`) and frequency domain (`freq_domai_analysis.p`) performance metrics and use the plots to answer the questions below.





Does your compensator design result in zero steady-state error for both steps (dwells), and ramps?

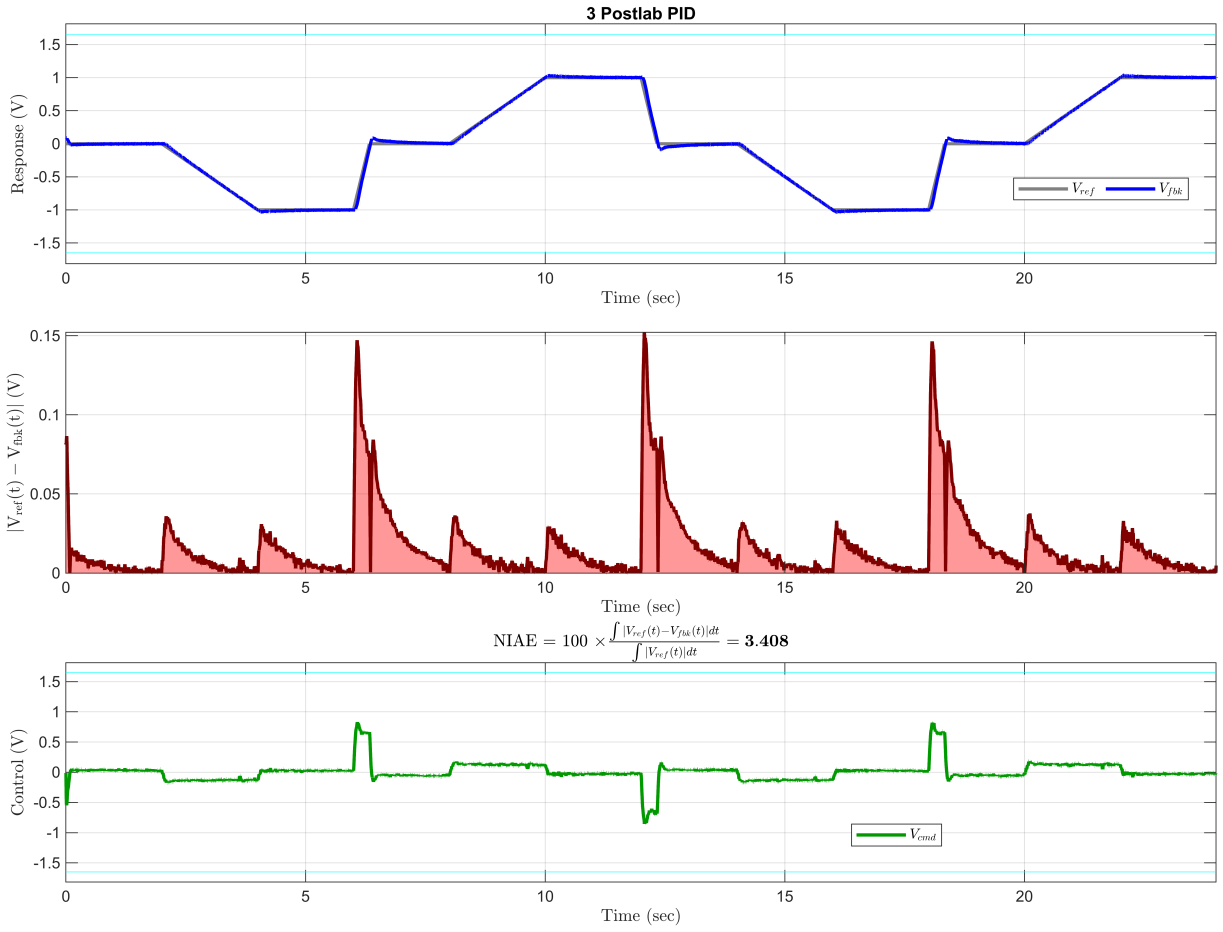
Does your compensator design meet all seven design requirements? (It is required to before you can test in the lab!)

Use the graphics cursor in the time-domain Response plot above to determine the maximum overshoot during the step inputs. Did the overshoot reduce or increase relative to the previous design?

Does the closed-loop frequency response have any magnitude amplification (i.e. gain > 0dB)? If so, what is the approximate frequency range where the magnitude is increased?

Task 3b: PID Compensator - Experimental Validation (Post-Lab)

[2 pts] Load the .mat file containing your experimental data for the PID compensator design, then call the **`iae_analysis`** function to generate the time-domain analysis plots.



[2 pts] Answer the following questions based on the experimental iae_analysis plots compared to the simulated iae_analysis plots

Does the response, error, and command time-domain signal match the general shape and amplitude between the experiment and simulation?

Yes it looks to me almost as if the original graph got transposed up from the origin by a base level of error, there is also much more noise in the error graph.

Responce looks identical to me but there is likley a small difference.

Since the experimental data has additional noise on it, we would expect the experimental NIAE to be larger than the simulated NIAE. What is the percent increase in the experimental NIAE?

```
PID Experimental NIAE percent increase: 21.83%
```

Task 4a: Lead Compensator Design - Simulation Study (Pre-Lab)

For this design, you will evaluate the performance of a LEAD compensator to meet all of the performance requirements.

When designing a LEAD compensator, the primary objective is to increase the phase angle of the loop transfer function over a specific frequency range. Increasing the loop phase angle will enable you to push out the phase crossover to a higher frequency which will allow a higher closed-loop bandwidth. The phase angle of a LEAD compensator will begin to increase at the frequency associated with the zero (numerator root), then the phase will return back to zero starting at the frequency associated with the pole (denominator root). For this task, we would like the phase to begin increasing at 0.6 Hz, and we would like the phase to begin returning back to nominal at 1 Hz. This defines the zero and pole of the lead compensator; however, you are responsible for tuning the overall compensator gain using a guess-and-check approach to meet the requirements below.

Mandatory Design Requirements (for the Closed-Loop Simulation):

1. $-1.65V \leq V_{\text{fbk}} \leq +1.65V$
2. $-1.65V \leq V_{\text{cmd}} \leq +1.65V$
3. $\text{NIAE} < 6.0$ (Note that this is larger than the requirement for Task 3)
4. zero steady-state error for steps and dwells (visual inspection only!)
5. $5\,\text{Hz} \leq 3\text{dB Bandwidth} \leq 6\,\text{Hz}$
6. $\text{Gain Margin} \geq 40\,\text{dB}$
7. $\text{Phase Margin} \geq 50^{\circ}$

[4 pts] Define and display your continuous-time controller design. Save your LEAD controller LTI object to a .mat file for the experimental evaluation.

```
C_Lead =

  5 s + 18.85
  -----------
   s + 6.283

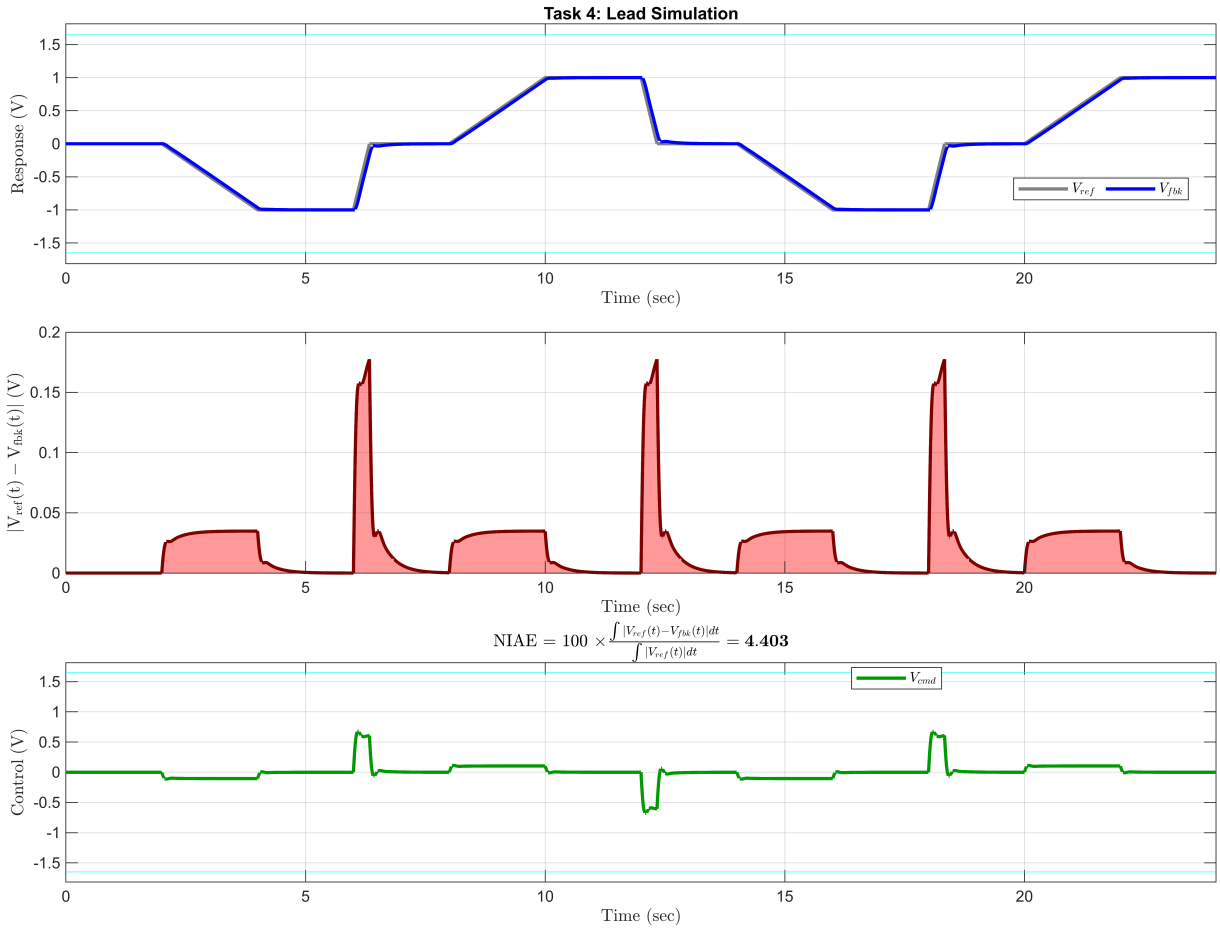
Continuous-time transfer function.
Model Properties
```

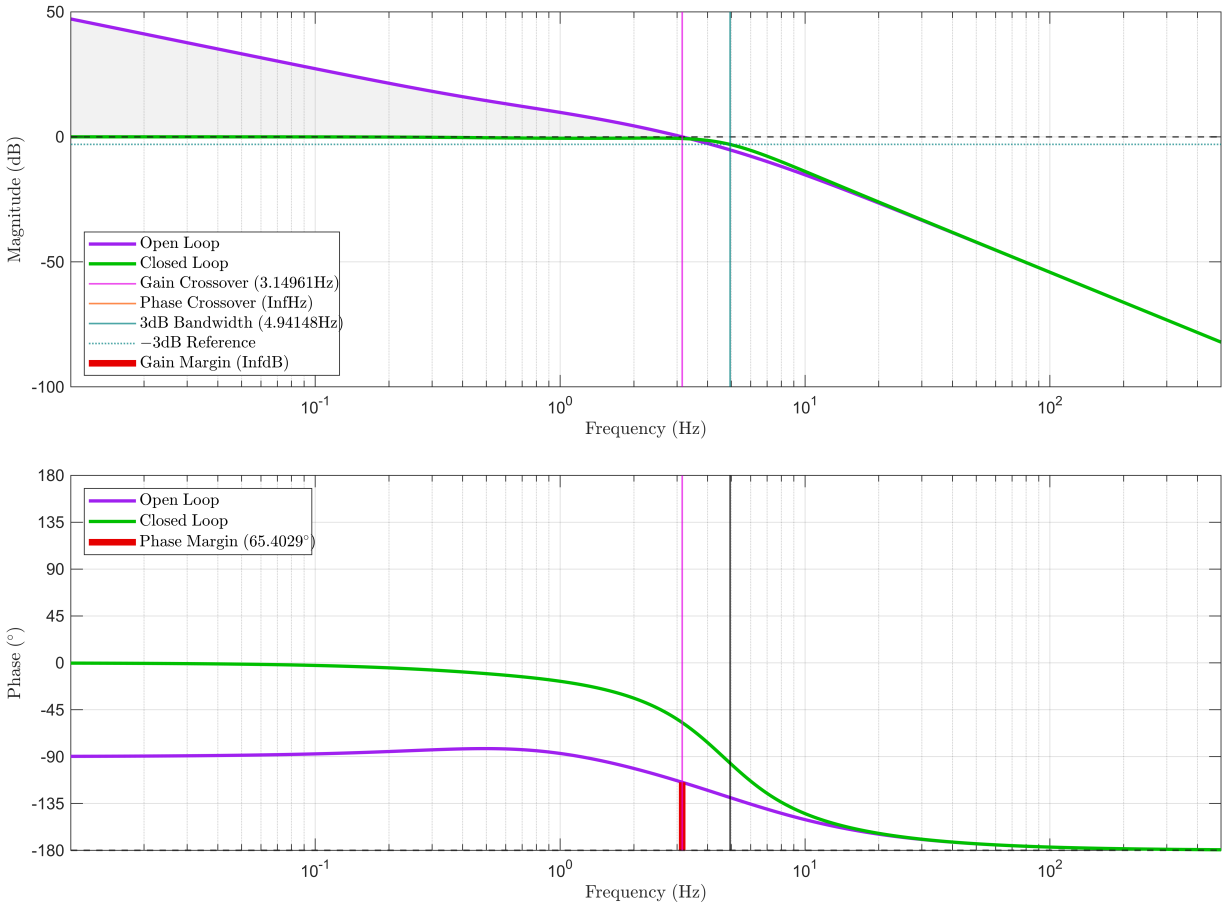
[2 pts] Compute the frequencies of the compensator zero (roots of numerator) and pole (roots of denominator) with units of Hz

```
Lead Zero: 0.6000 Hz, Lead Pole: 1.0000 Hz
```

[3 pts] Construct closed-loop models for simulating the time response as well as the control signal, then simulate the closed-loop time responses of the simulated feedback signal (fbk_sim) as well as the simulated control signal (cmd_sim). You will need to load the reference profile from reference.mat. Note: This is exactly the same operations you did in Lab 4 but using the LEAD controller designed above.

[3 pts] Generate the time-domain (`iae_analysis.p`) and frequency domain (`freq_domai_analysis.p`) performance metrics and use the plots to answer the questions below.


Task 4: Lead Simulation
Response (V)
Time (sec)
V_{ref}
V_{fbk}
$|V_{ref}(t) - V_{fbk}(t)|$ (V)
Time (sec)
$\text{NIAE} = 100 \times \frac{\int |V_{ref}(t) - V_{fbk}(t)| dt}{\int |V_{ref}(t)| dt} = \mathbf{4.403}$
Control (V)
V_{cmd}
Time (sec)



Does your compensator design result in zero steady-state error for steps and dwells?

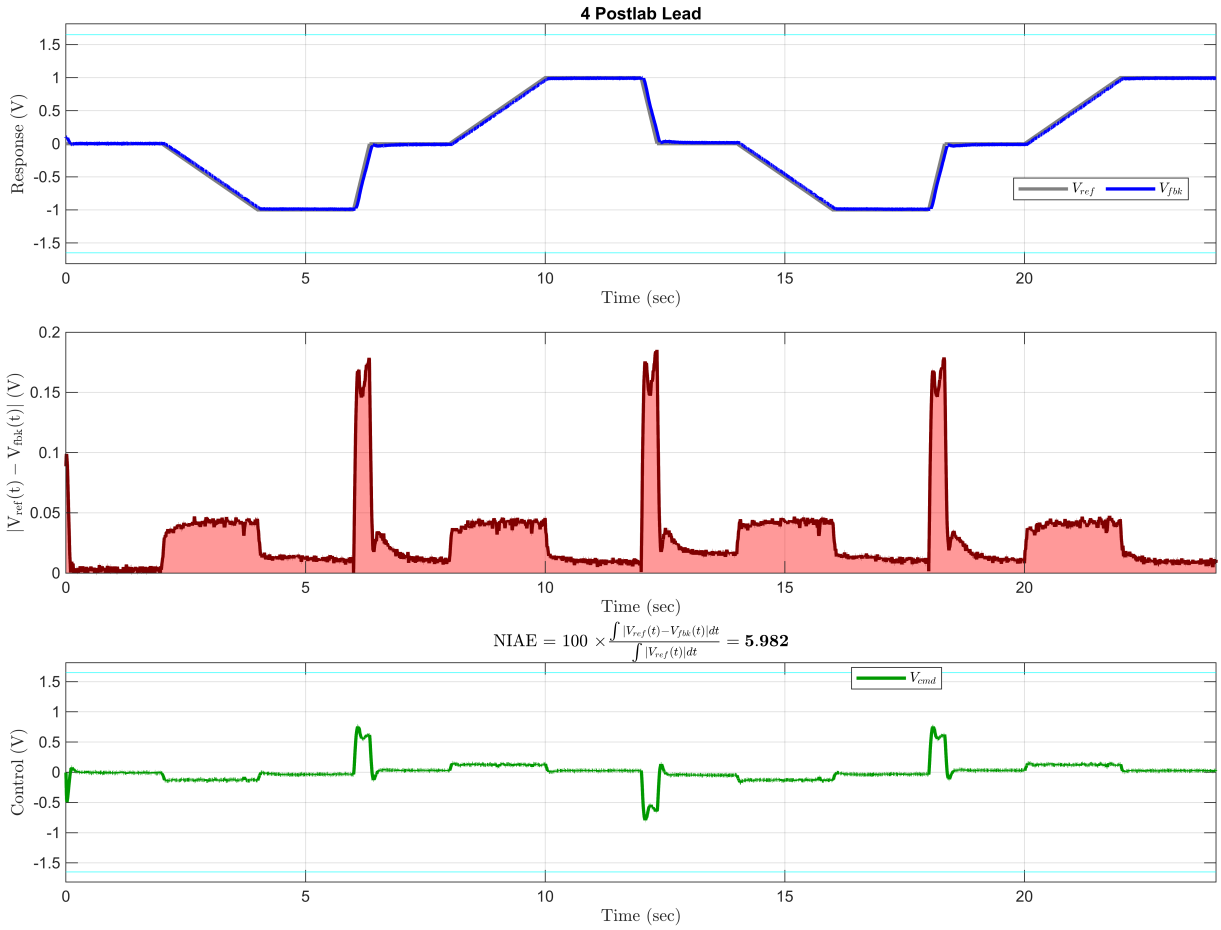
Does your compensator design meet all seven design requirements? (It is required to before you can test in the lab!)

Does the lead compensator result in any overshoot at any point in the response?

Does the closed-loop frequency response have any magnitude amplification (i.e. gain > 0dB)? If so, what is the approximate frequency range where the magnitude is increased?

Task 4b: Lead Compensator - Experimental Validation (Post-Lab)

[2 pts] Load the .mat file containing your experimental data for the LEAD compensator design, then call the **`iae_analysis`** function to generate the time-domain analysis plots.



[2 pts] Answer the following questions based on the experimental iae_analysis plots compared to the simulated iae_analysis plots

Does the response, error, and command time-domain signal match the general shape and amplitude between the experiment and simulation?

Yes it looks to me almost as if the original graph got transposed up from the origin by a base level of error, there is also much more noise in the error graph.

Responce looks identical to me but there is likley a small difference.

Since the experimental data has additional noise on it, we would expect the experimental NIAE to be larger than the simulated NIAE. What is the percent increase in the experimental NIAE?

```
Lead Experimental NIAE percent increase: 35.84%
```