ASEN 2803 - Control Arm

Group Members
Daniel Sagasti
Nico Marino
Jared Steffen
Nikolas Welch
Wyatt Stelzig
John Dallin

Note: when attaching figures, make sure to label the axes (including units), add a legend if multiple plots are shown in the same figure and have titles for the plots.

The size of the answer boxes is not intended to limit your response. Feel free to make the boxes bigger if you need to.

SECTION 1 - Objectives

- Design a controller for a second order system
- Observe and characterize the step response of a closed loop control system for a rotary positioner
- Observe the controller performance for a rigid and flexible arm structure
- Apply knowledge of second order systems, vibrations, and control to a real system

SECTION 2 - Experimental Background & Description

Control is a key component on many large and small scale systems. For example, to successfully fly an inherently unstable fighter aircraft, complex control systems for coordination of thrust and control surface adjustments are required. In addition, the very motion of an actuator or control surface such as an aileron requires a smaller and simpler controller to ensure that the desired actuator position is achieved given varying aerodynamic loads. A similar situation occurs on a spacecraft where solar panels are positioned so as to maximize the incident radiation. Clearly the perfor-mance requirements for the two examples are quite different, both in terms of the required accuracy and response time of the system

In this lab we will consider a simple control task that plays an important role in many dynamic systems relevant to aerospace, as well as other engineering disciplines. The objective is to position an arm mounted on a rotary shaft. We will consider two types of loads - a rigid arm mounted to the shaft and a more complex flexible arm mounted to the shaft. In order to effectively position the arm, we need to formulate a dynamic model of both the mechanical and electrical aspects of the system. A control strategy is then developed to achieve a desired response - i.e. we want the arm to move to the desired position quickly, and accurately, without a lot of overshoot or vibration. These control requirements will be defined mathematically in the following sections. The appendices provide a detailed description of the experimental apparatus shown in Figure 1 as well as the derivation for the dynamic model of the rigid and flexible arm system.

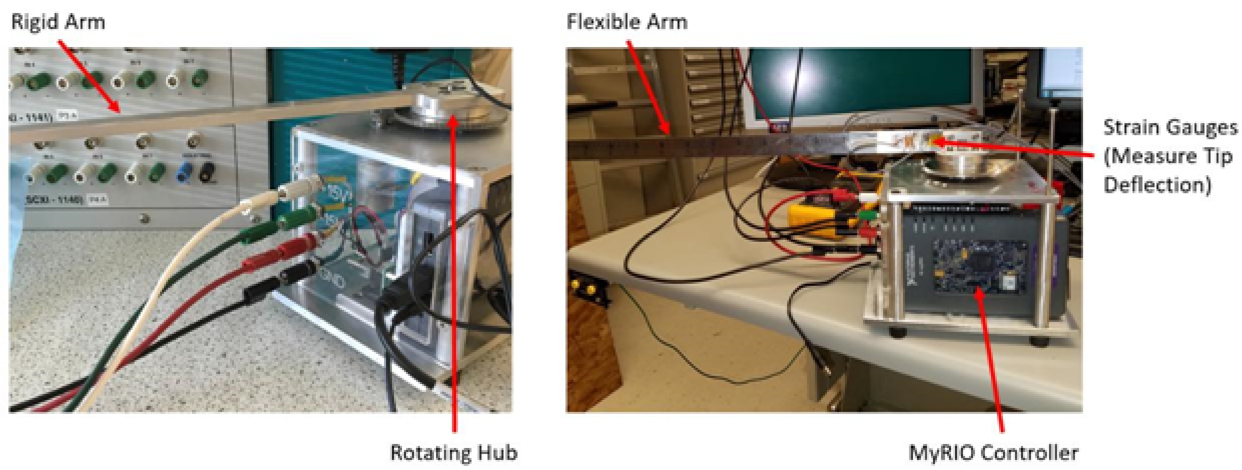


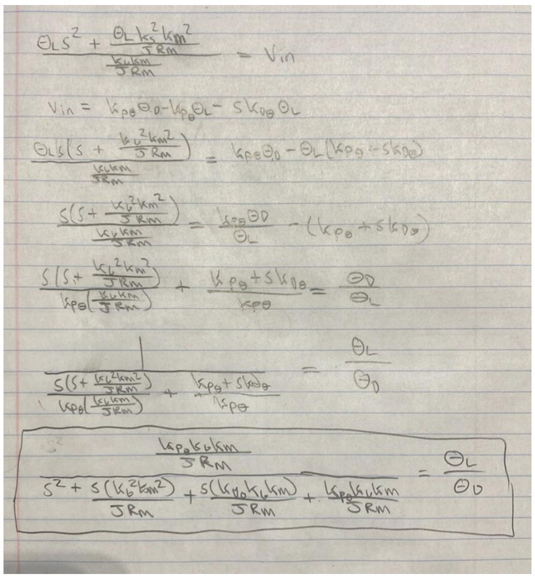
Figure 1: Left: Rigid Arm Module. Right: Flexible Arm Module

A major task is to observe the nominal controller response and see how changes in the control gains affect the perfor-mance. You will simulate the rigid arm in order to predict the response of the systems to specific control gains. The primary objective of this lab assignment is to compare simulated responses to the hardware responses and intelligently modify control gains to achieve specific performance requirements. Very often in control applications the gains predicted using a theoretical approximation will need to be modified based on the response of the real system in order to meet the requirements.

SECTION 3 - Theory & Simulation

Question 3.1

Review the derivations provided in the appendices and derive Eq. 17 and 18 beginning with Eq. 13. (Hint: first solve Eq. 13 for V_{IN}. Substitute the resulting expression into Eq. 16. Finally arrange like terms and divide through to get the closed loop transfer function).



$$\frac{\Theta_L s^2 + \frac{\Theta_L k_G^2 k_m^2}{J R_m}}{\frac{k_G k_m}{J R_m}} = V_{in}$$

$$V_{in} = k_{p\theta}\Theta_D - k_{p\theta}\Theta_L - s k_{D\theta}\Theta_L$$

$$\frac{\Theta_L s\left(s + \frac{k_G^2 k_m^2}{J R_m}\right)}{\frac{k_G k_m}{J R_m}} = k_{p\theta}\Theta_D - \Theta_L(k_{p\theta} + s k_{D\theta})$$

$$\frac{s\left(s + \frac{k_G^2 k_m^2}{J R_m}\right)}{\frac{k_G k_m}{J R_m}} = \frac{k_{p\theta}\Theta_D}{\Theta_L} - (k_{p\theta} + s k_{D\theta})$$

$$\frac{s\left(s + \frac{k_G^2 k_m^2}{J R_m}\right)}{k_{p\theta}\left(\frac{k_G k_m}{J R_m}\right)} + \frac{k_{p\theta} + s k_{D\theta}}{k_{p\theta}} = \frac{\Theta_D}{\Theta_L}$$

$$\frac{1}{\frac{s\left(s + \frac{k_G^2 k_m^2}{J R_m}\right)}{k_{p\theta}\left(\frac{k_G k_m}{J R_m}\right)} + \frac{k_{p\theta} + s k_{D\theta}}{k_{p\theta}}} = \frac{\Theta_L}{\Theta_D}$$

$$\boxed{\frac{\frac{k_{p\theta} k_G k_m}{J R_m}}{s^2 + s\left(\frac{k_G^2 k_m^2}{J R_m}\right) + s\left(\frac{k_{D\theta} k_G k_m}{J R_m}\right) + \frac{k_{p\theta} k_G k_m}{J R_m}} = \frac{\Theta_L}{\Theta_D}}$$

$$\frac{\theta_L}{\theta_D} = \frac{K_{p\theta} K_g K_m}{S^2 + S(K_g^2 K_m^2 + K_{D\theta} K_g K_m) + K_{p\theta} K_g K_m}$$

Using $S''(t) + 2\zeta\omega_n S'(t) + \omega_n^2 S(t)$, we find that

$$\omega_n^2 = \frac{K_{p\theta} K_g K_m}{J R_m}$$ and that

$$\zeta = \frac{K_g^2 K_m^2 + K_{D\theta} K_g K_m}{2\sqrt{K_{p\theta} K_g K_m J R_m}}$$

Question 3.2

Develop a MATLAB simulation for the closed loop behavior of the rigid arm (Eq. 17). Use the physical and electrical parameters given in the spreadsheet to determine the parameters for the equations of motion derived in Question 3.1. The following sample code can help you get started:

```
%% Closed Loop System
num = n1;
den = [d2 d1 d0];
sysTF = tf(num,den);

%% Step Response
[x,t] = step(sysTF);
```

Note: If you want to simulate a more complicated input check out the `lsim` command in MATLAB to find the theoretical response of your system to a specified input u(t). This is useful in comparisons with experimental data where the reference values and time are also recorded on the data file.

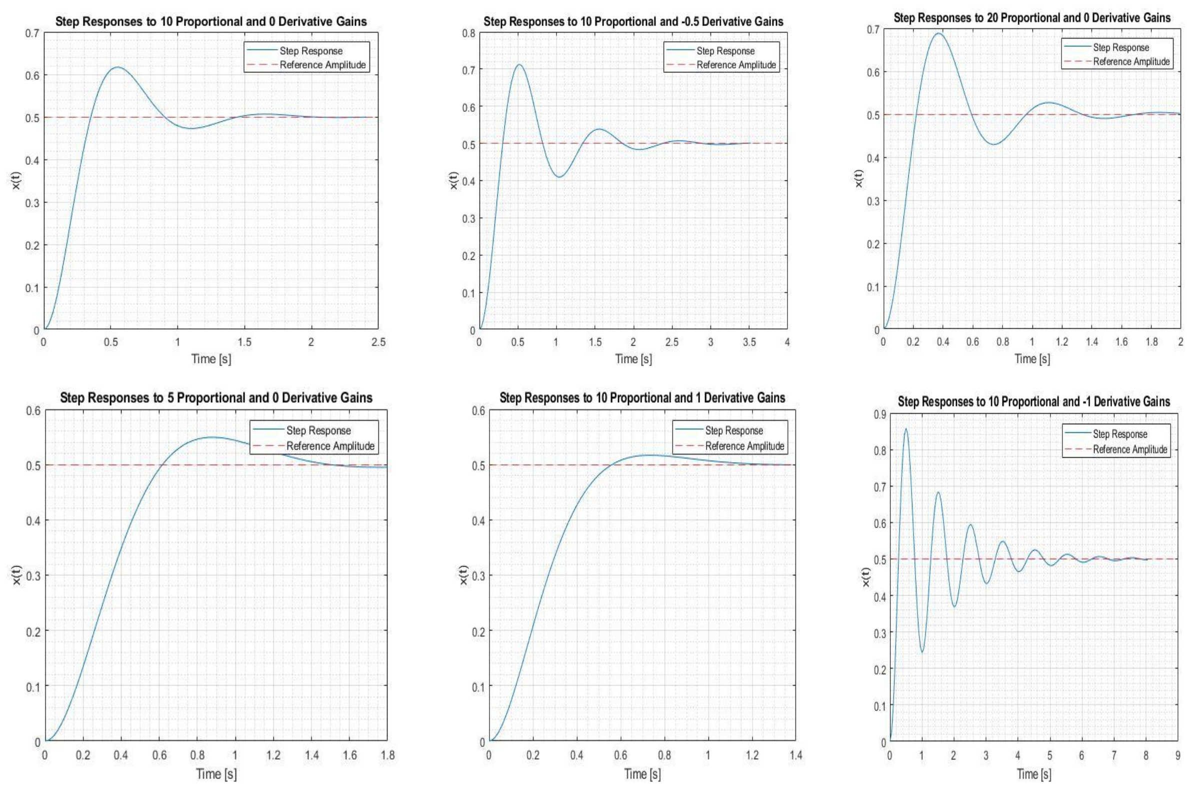
<u>Note</u>: you will need to install the "Control System Toolbox" in MATLAB if you are using your personal laptop.

Question 3.3

Use the simulation developed for Question 3.2 in order to investigate the behavior of the step response of the Rigid Arm system for the gain values provided below to explore the effects of increasing and decreasing the proportional and derivative gains. Set the amplitude of the step response to be **0.5 rad**.

Gains	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6
K1 - Proportional	10	20	5	10	10	10
K3 - Derivative	0	0	0	1	-1	-0.5

Explain the differences and similarities you see between the different sets of gains.



We see that the higher the proportional gain (with the same derivative gain), the more overshoot there is. When positive derivative gains are added, the system converges quicker. If negative derivative gains are added, then the system oscillates for a good bit before converging.

SECTION 4 - Experiment

Question 4.0 - Exploring the Hardware

Using the rigid arm **hardware**, input the gains given in Question 3.3. Set the “Reference Amplitude” to 0 rad. This will make the system want to stay at zero rad.

4.0.a Perturb the system (i.e.: give the rigid arm a little push or move the arm about 45 deg to one side), what do you notice? Does it feel different for different gains? Explain what differences you notice/feel? Does this make sense?

The farther we push it, the harder it is to push. This makes sense as the force required to push it is proportional to the distance away it is.

Question 4.1 - Conceptual Understanding

Using the rigid arm **hardware** explore the effects of increasing and decreasing the proportional and derivative gains with the gain values provided in Question 3.3. Make sure to set the “Reference Amplitude” to 0.5 rad. Compare the behavior of the hardware with **your MATLAB simulation** and with the **provided Matlab/Simulink simulation.**

4.1.a What does a higher/lower **proportional** gain do to the system? What are the similarities and differences between the hardware, your MATLAB simulation and the Simulink simulation?

Higher proportional gain makes it harder to push as we get farther from the reference amplitude, and lower proportional gain makes that force smaller. This is in line with our MATLAB simulation.

4.1.b What does a higher/lower **derivative** gain do to the system? What are the similarities and differences between the hardware, your MATLAB simulation and the Simulink simulation?

Higher derivative gain makes it harder to push as it moves faster, and slower as we slow down. This is also in line with our MATLAB simulation.

4.1.c How can the overshoot be increased/decreased? What are the similarities and differences between the hardware, your MATLAB simulation and the Simulink simulation?

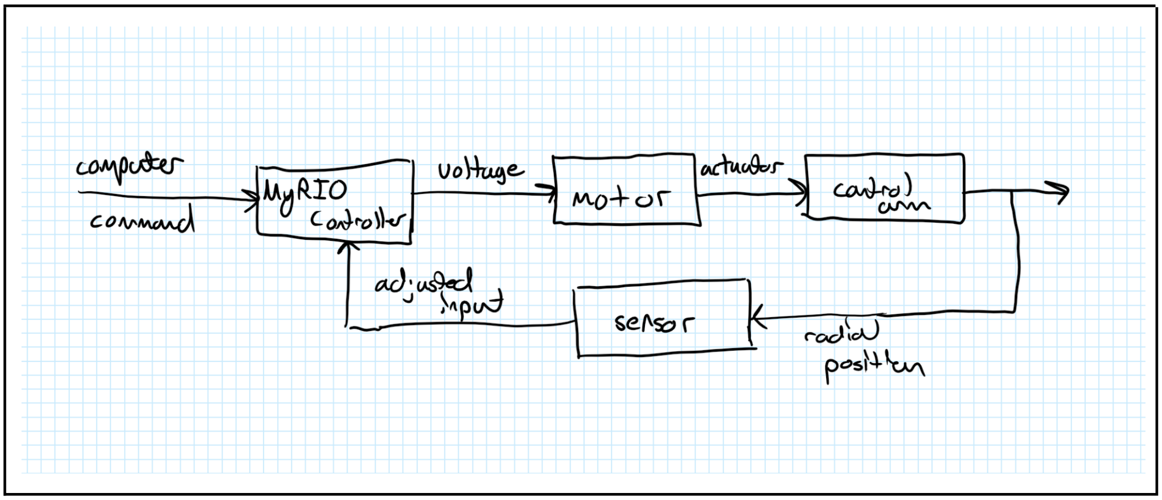
Overshoot can be increased by changing the derivative gain. A negative derivative gain will create more overshoot and a positive derivative gain will decrease overshoot. Our MATLAB simulation actually shows this well with our graphs.

4.1.d How can the settling time be increased/decreased? What are the similarities and differences between the hardware, your MATLAB simulation and the Simulink simulation?

Settling time can be increased by decreasing gain, and decreased by both increasing gain and minimizing overshoot. Our MATLAB agrees with what we noticed in the experiment.

Question 4.2 - Overall System And Software

4.2.a Sketch a functional block diagram of the rigid arm system below:



4.2.b What is the input and output of each block? What is the function of each block?

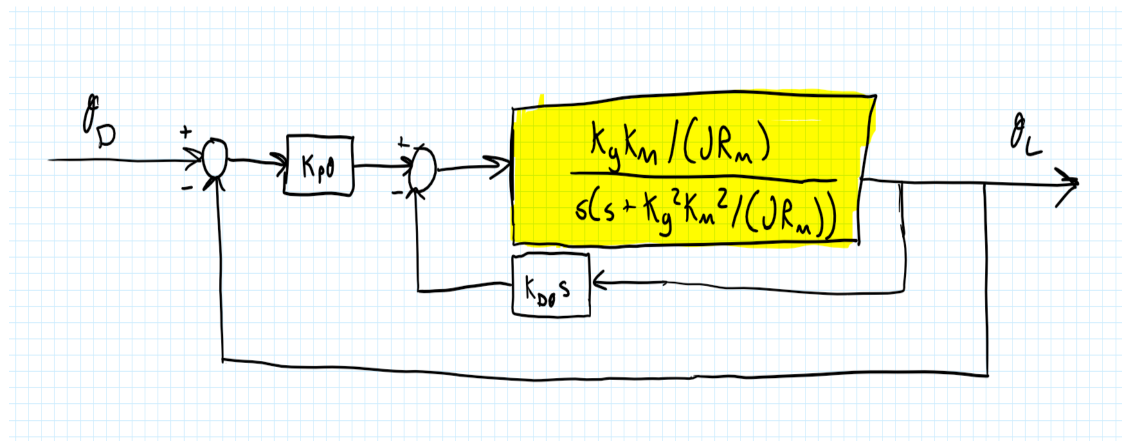
MyRio Controller
Input: Computer command on where to go
Output: Voltage

Motor
Input: Voltage
Output: Energy to Actuator

Control Arm
Input: Energy from the actuator
Output: Moving radial positions

Sensor Input: Radial position Output: Adjustments for input

4.2.c Sketch the control block diagram, and highlight the transfer function(s):



4.2.d What does each part of the system do (the arm, sensors, myRIO, actuator) and how do they interact with each other?

The MyRIO controller: Tells the motor to turn the arm to a certain point **The motor:** Powers actuator **Actuator:** Turns the arm to the desired location **Sensor:** Communicates Radial position to myRio controller **The arm:** Responds to the force applied by the actuator

Question 4.3 - Design Controls and Behavior Response

Now you are going to design your own control gains such that your system meets a set of requirements. Your controlled rigid arm should:

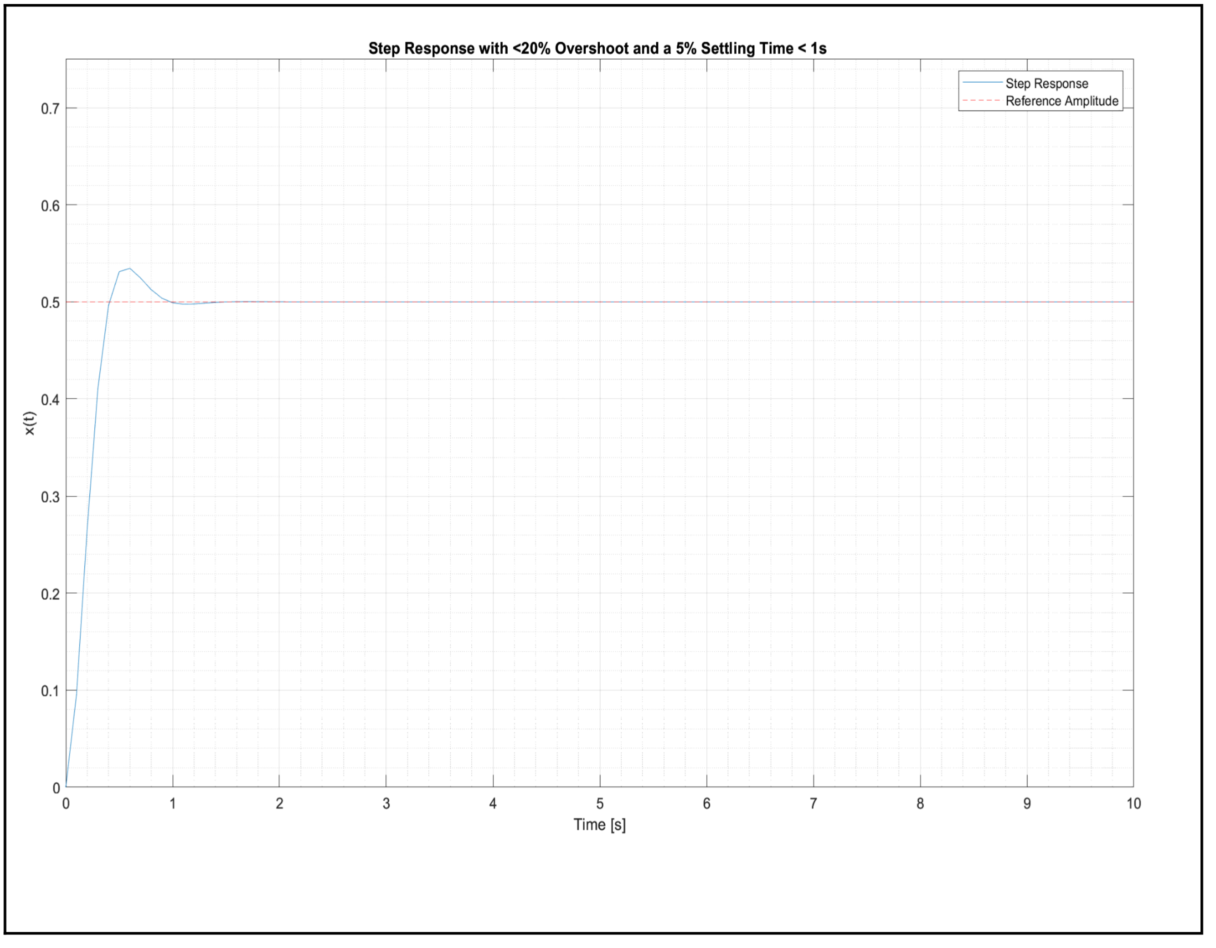
- **Have less than 20% overshoot**
- **Achieve a 5% settling time in less than 1 second**

Your reference step signal should have an **amplitude of 0.5 rad** and a **period of 10 s**. Make sure to **show at least 1 cycle** in your plots. Make sure to add the reference step input in all your plots.

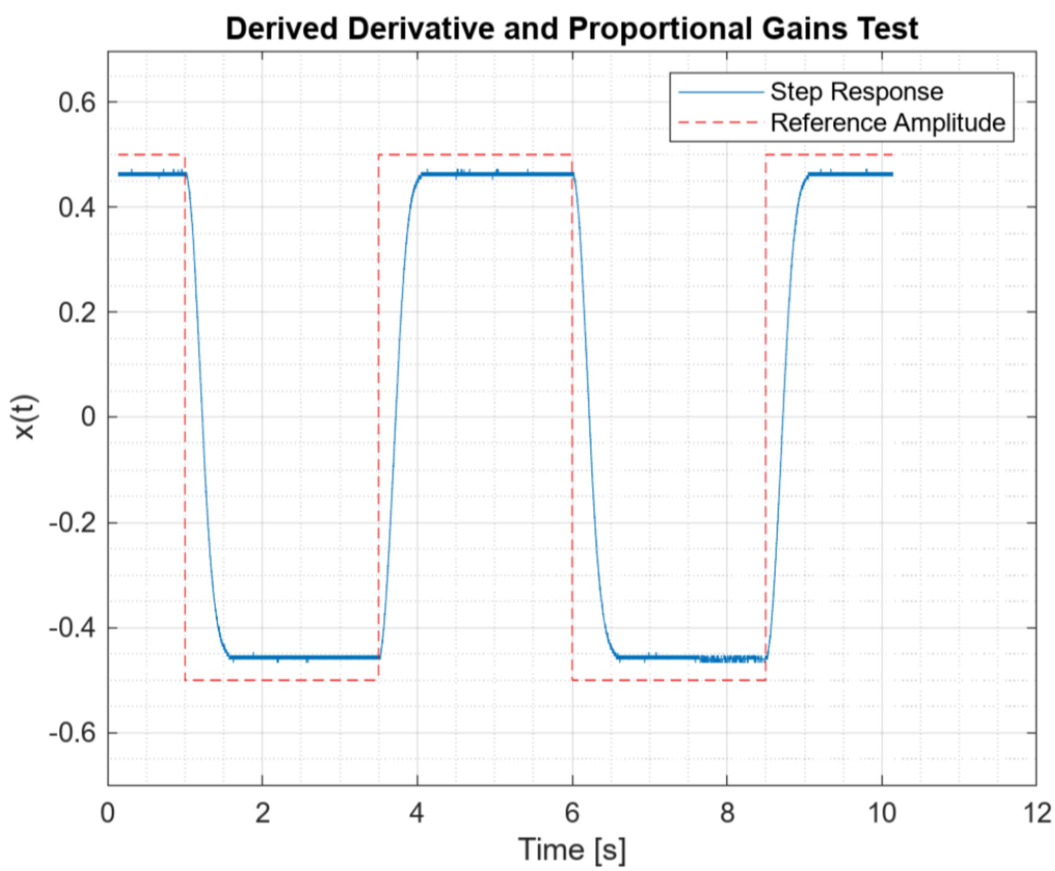
4.3.a What are the proportional and derivative gains that you arrived at?

K1 - Proportional Gain	13.45
K2 - Derivative Gain	1.04

4.3.b Input these gains into **your MATLAB simulation**. Do you meet the requirements? Check by inspection your time response plot and include it below. If you don't meet the requirements, explain why.

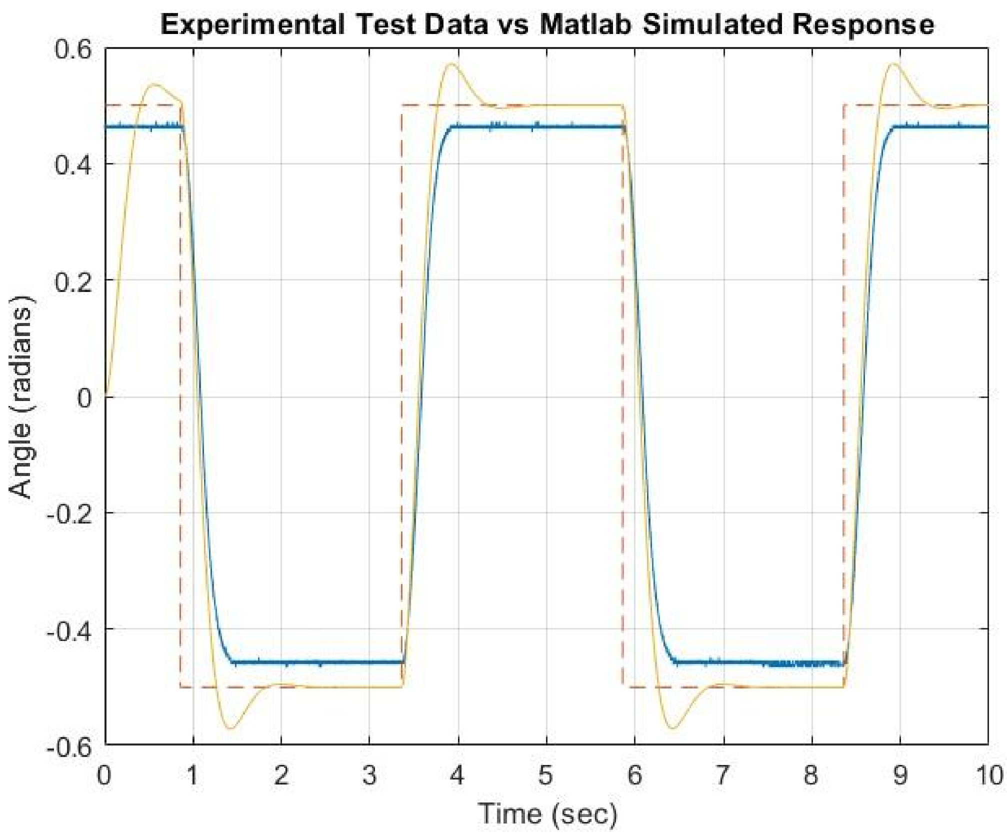


4.3.c Now input these gains into the rigid arm **hardware** VI. Record data and then plot that data in MATLAB (include the figure of your time response below). Do you meet the requirements in the hardware? If you don't, explain why.



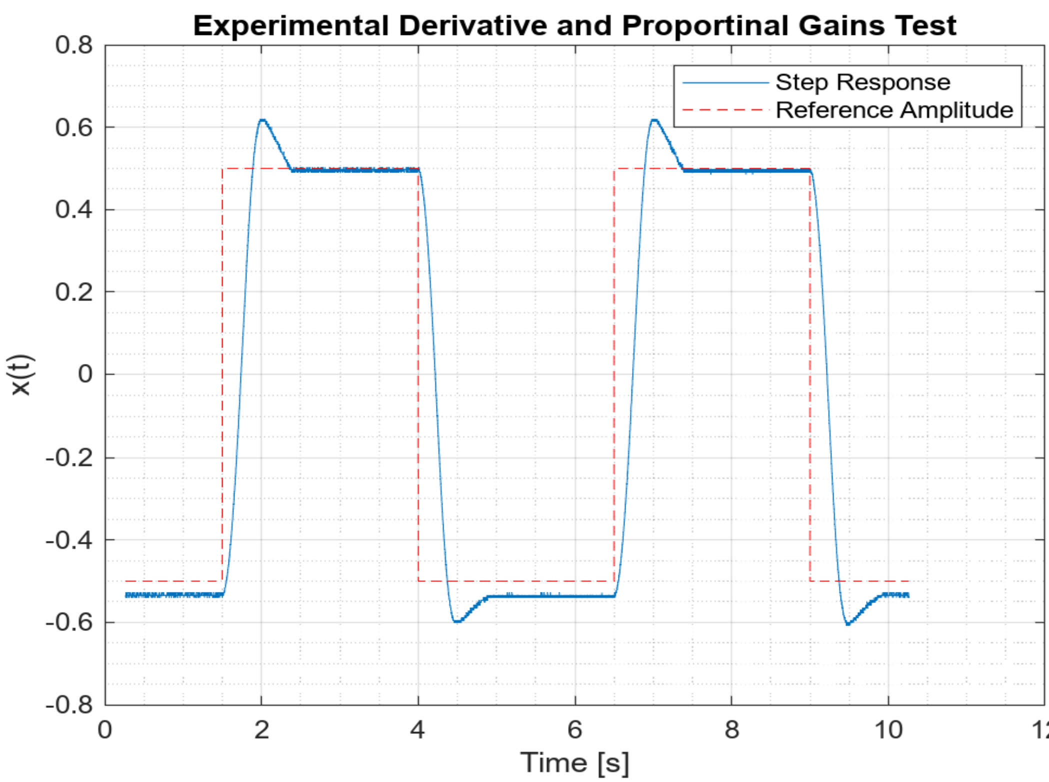
We do not meet the requirements because the system is still not underdamped, so there is no overshoot or settling time. This is due to unaccounted for damping in the actual system vs our MATLAB model.

4.3.d Now compare the response you get from your MATLAB model with the response you get from the hardware. Plot both time responses together and explain the differences between the two. What do you think is causing that difference?



We believe the main factor of the MATLAB model not matching the hardware is due to friction and heat losses. With a damping coefficient of ~0.65, we see that the system is actually still not underdamped. This suggests there is another source of damping not present in the MATLAB model (i.e. friction and heat).

4.3.e How can you modify your gains to meet the requirements in the hardware? Try these new gains in the hardware and show if they meet the requirements by plotting the output of the VI in MATLAB.



In order to meet the requirements for the hardware, the derivative gains were modified from 1.04 to 0.1 and the proportional gains were kept at 13.45. We did this to lower the damping coefficient, which will eventually cause overshoot once the system is underdamped.

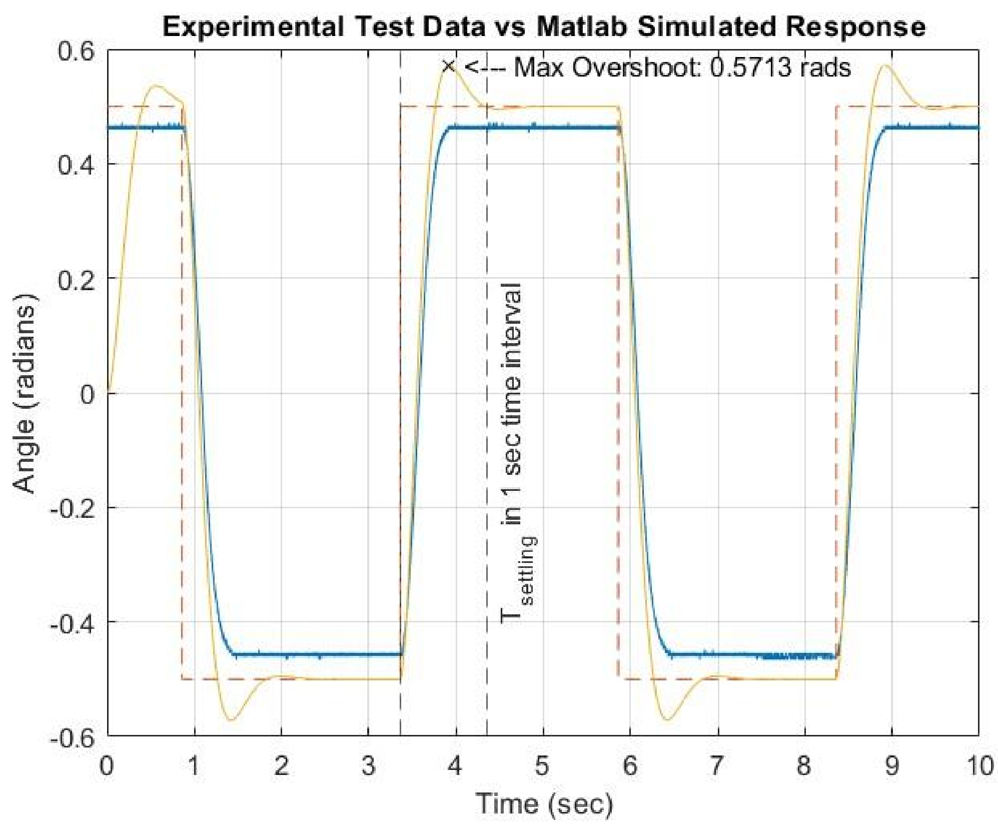
4.3.f Beware of the 10V limit on V_{IN}. How does your model and experimental data differ if the control output reaches this value? Look at the Control Voltage plot in the hardware VI.

In the model, if it hits the 10V limit it has reached max speed but still retains accuracy. With the experimental data, once it reaches 10V it has also reached max speed but some accuracy is lost.

Section 5 - Results and Analysis

NOTE: It has been observed that the physical units may exhibit different behavior when moving clockwise vs. counterclockwise. You only need to analyze a single transition. This means you are free to pick the direction that reasonably matches your model and ignore the opposing transition.

5.1.a For the rigid arm, plot and compare the experimental results with the model results from the gains you selected to meet the performance objectives. Label the overshoot and 5% settling time on the plot. Upload the result and explanation below:



5.1.b Plot the response of the experimentally modified gains if the gains were modified to meet the requirements. Label the overshoot and 5% settling time on the plot. What gain(s) did you modify and why? Does this make sense given the unmodeled dynamics of the hardware?

New gain: K_DΘ = 0.1. We chose this value because we needed a lower damping coefficient. With our initial selected values, the system was still not underdamped, implying that there is some source of damping not present in the MATLAB simulation. In order to lower the damping coefficient, the derivative gains could be lowered or the proportional gains could be raised. However, since the proportional gains have their square root taken, raising the proportional gains has less of an effect than lowering the derivative gains. This is the main reason why we did not change the proportional gains.

5.2.a How and why do the experimental results differ from the theory?

When comparing the experimental results and theory, it can be observed that the general shape of the graphs are the same, although there is clearly some degree of error between the two where MATLAB and the hardware do not match up perfectly. The real-life system from which experimental data was gathered is affected by things that the theoretical model did not take into account, such as friction. Additionally, the real system has a level of natural damping that was assumed to be negligible in the model. If this natural damping is not negligible, then, of course, the model will not be completely accurate compared to the actual experimental results.

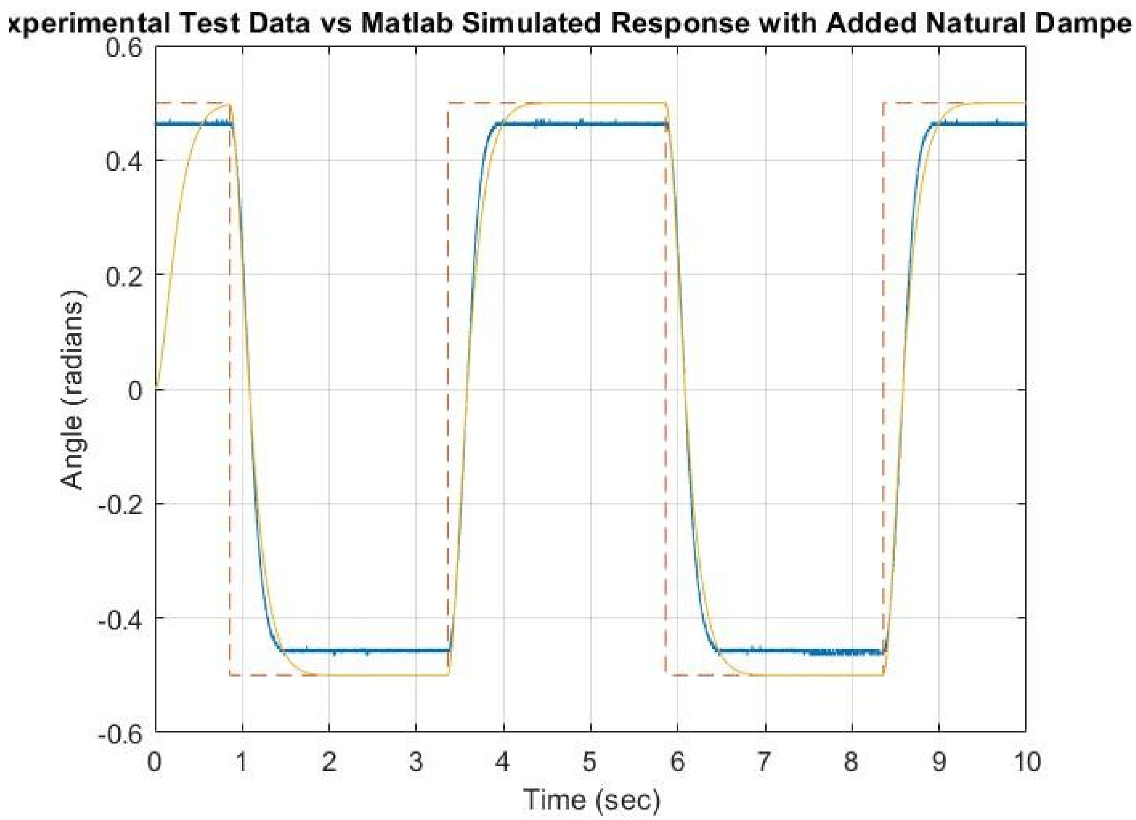
5.2.b We assumed negligible natural damping (B) to be zero. Is this a good assumption for this experiment? Why or why not?

Negligible natural damping in this scenario is not necessarily a good assumption. There is a certain level of damping due to friction, and this appears in the data as plateaus just below the target.

5.2.c How can you determine an approximation for the natural damping in the real system?

We could solve for the difference between our predicted model which assumes zero damping, and the real data which, of course, includes natural damping.

5.2.d Compare your improved model (the one that includes the effects of natural damping in the real system) to the model without friction and to the real system.



In order to compare our improved model to the real system, we found the difference in damping coefficient between the old model and the real system and added it to the model.