Lab 5: SFWR ENG 3DX4

Steady-State Error and PID Control of Electromechanical Systems

First week of lab: March 25, 2024
Prelab Due start of Lab Period, week of: April 1, 2024
Demo Due End of Lab Period, week of: April 1, 2024

Announcements

In this lab, you will be evaluated on prelab questions and demonstration of working laboratory exercises. Evaluation of lab exercises will be performed on a checkpoint system, with checkpoints indicated in your laboratory instructions. In order to receive full marks, you must show a TA your work/output at each checkpoint, and your work must be correct and complete. In addition, lab work must be completed in the allotted laboratory time. Prelab questions are to be completed *individually*. This lab consists of **3 checkpoints**.

Goals

- Introduction to computer-based control of a voltage-controlled electromechanical system - a DC servo motor with ball and beam apparatus.
- Learn how to design an ideal PID compensator to satisfy specific requirements using root locus techniques.
- Learn the difference between PD compensator behaviour and PID compensator behaviour under load conditions.
- Take empirical measurements of a physical system.

Background Reading

Before you go for your lab session, please read the following documents at your convenience, in addition to the class notes:

- Ball and Beam User Manual
  (download the `BallBeamUserManual.pdf` from the course webpage if the link does not work)
- Ball and Beam Student Workbook
  (download the `BallBeamStudentWorkbook.pdf` from the course webpage if the link does not work)

Equipment Arrangement

The equipment setup for Lab 5 is nearly the same as for Lab 4. The equipment consists of the Quanser SRV02 rotary servo plant, with attached Quanser BB01 Ball and Beam plant. The Quanser SRV02 rotary servo plant consists of a DC motor, a gear box, a potentiometer, and a quadrature encoder. Some models also have a tachometer. The Quanser BB01 Ball and Beam plant consists of a steel ball, two rails along which the ball freely rolls and a sensor apparatus (collectively the *rail assembly*), one stationary post providing one rotational degree of freedom to the rail assembly, and a link connecting the rail assembly to the centre gear of the SRV02 rotary servo plant. When the angular position of the SRV02 plant changes, it causes a corresponding change in the angle of inclination of the rail. The Analog Output signal provided to the SRV02 is to adjust its gear's angular position so as to adjust the angle of the rail and the position of the ball. The sensor apparatus included in the rail assembly consists of a potentiometer, where the steel ball acts as the wiper. In particular, one of the rails may be thought of as a resistor with a resistance proportional to its length. The other rail is the analog input for the controller. The steel ball, being conductive, provides the voltage at its position on the resistor rail to the other rail, which is sampled as input.

In this lab we will study how a PID controller will decrease or eliminate the steady-state error of a system. In our lab set up, we put some weight on one end of the beam to introduce a disturbance. This disturbance will create a steady-state error if the system controller is not designed appropriately.

Mathematical Background

The ball and beam setup is composed of two distinct systems, the BB01 ball and rail plant, and the SRV02 servo motor plant. This setup may be modelled mathematically using transfer functions. Since these two systems are in series (or in cascade form), combining their transfer functions ($P_{bb}(s)$ and $P_s(s)$ respectively) is a simple multiplication:

$$P(s) = P_{bb}(s)P_s(s) \tag{1}$$

The transfer function for the SRV02 rotary servo plant is given as follows:

$$P_s(s) = \frac{K}{s(\tau s + 1)} \tag{2}$$

where

$$K = 1.53\, rad/(V \cdot s) \tag{3}$$

and

$$\tau = 0.0248\ s \tag{4}$$

The transfer function for the Ball and Beam assembly without the SRV02 plant is derived from first principles in the Ball and Beam Student Workbook, and the result of that derivation is as follows:

$$P_{bb}(s) = \frac{K_{bb}}{s^2} \tag{5}$$

where

$$K_{bb} = 0.419\frac{m}{s^2 rad} \tag{6}$$

The control system for the Ball and Beam and the SRV02 plant consists of two parts: an inner loop to control the SRV02's gear angle, and an outer loop to control the ball's position. As in the last lab, we will assume that the servo controller in the inner loop is perfect, even though it isn't (it is simply a proportional controller).

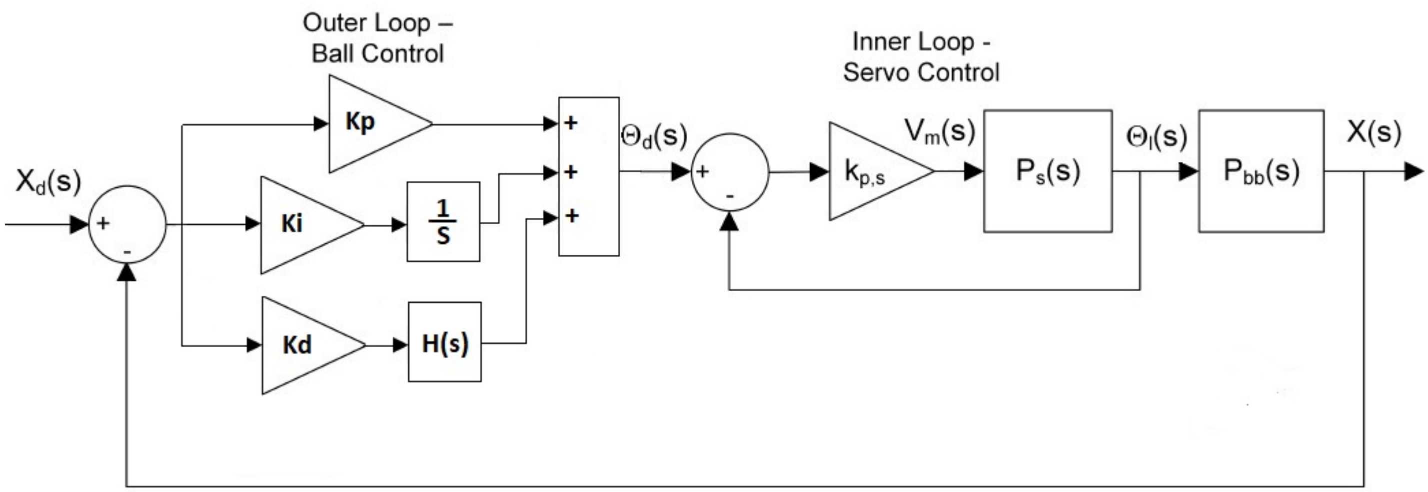

Figure 1: Cascade control system with a PID controller for the outer loop for ball's position

For the outer loop to control the ball's position, we will use a PID controller in this lab. Just like in Lab 4, we will use a high-pass filter to replace the derivative term in the PID controller. The control system is then as shown in Figure 1, where

$$H(s) = \frac{\omega_f s}{s + \omega_f} \tag{7}$$

and ω_f is the cutoff frequency. For our lab setup, $\omega_f > 5Hz$ will be adequate for filtering of the noise.

In Figure 1, the PID compensator is given by:

$$C(s) = K_p + K_i \frac{1}{s} + K_d \frac{\omega_f s}{s + \omega_f} = \frac{(K_p + \omega_f K_d)s^2 + (\omega_f K_p + K_i)s + \omega_f K_i}{s(s + \omega_f)} \tag{8}$$

Because we assume the servo controller in the inner loop is perfect, the plant's transfer function in our control system can be simplified as :

$$P_{bb}(s) = \frac{0.419}{s^2} \tag{9}$$

In this lab, we will use MATLAB's Control System Designer to design our PID controller. Our control requirements are:

$$T_s = 3.0s \tag{10}$$

$$\%OS \leq 20.0\% \tag{11}$$

To design a PID controller for this second order system, we will need to know the damping ratio ζ and the natural frequency ω_n, which can be calculated from the settling time T_s and the percent overshoot $\%OS$ by using the following formula, which should be familiar by this point:

$$\zeta = \frac{-ln(\frac{\%OS}{100})}{\sqrt{\pi^2 + ln^2(\frac{\%OS}{100})}} \tag{12}$$

$$T_s = \frac{4}{\zeta\omega_n} \tag{13}$$

MATLAB's Control System Designer can assist us to design a controller that meets our control requirements. By appropriately specifying poles and zeros for the compensator, MATLAB's Control System Designer can give us a compensator in the following format:

$$C = \frac{K(s+z_1)(s+z_2)}{s(s+p)} \tag{14}$$

That is:

$$C = \frac{Ks^2 + K(z_1+z_2)s + Kz_1z_2}{s(s+p)} \tag{15}$$

Comparing Equation (15) with Equation (8), the following equations can be obtained:

$$K = K_p + \omega_f K_d \tag{16}$$

$$K(z_1+z_2) = \omega_f K_p + K_i \tag{17}$$

$$Kz_1z_2 = \omega_f K_i \tag{18}$$

$$p = \omega_f \tag{19}$$

The parameters ω_f, K_p, K_i, and K_d can then be derived from the above equations and can be used to implement the controller to control the ball's position.

In-Lab Activities

Design a PID controller using the Control System Designer

1. In order to use the control system designer, we must first have a model for the system we intend to control. Create a transfer function object in MATLAB using Equation (9) and the `tf()` command.

2. Load your transfer function into the control system designer using `rltool()`. You should see something similar to Figure 2.

3. It will be useful to tile the root locus and step response graphs, so that we can monitor both at the same time. This can be accomplished by clicking and dragging the graphs.

4. We can quickly add poles and zeros to our root locus using the Root Locus Editor ribbon. This will appear at the top if the root locus graph is selected (see Figure 2).

5. Let's also make our step response more readable.

   - **Characteristics → Peak Response**
   - **Characteristics → Settling Time**

6. For this system, we are aiming for a settling time of 3s and an overshoot of no more than 20%.

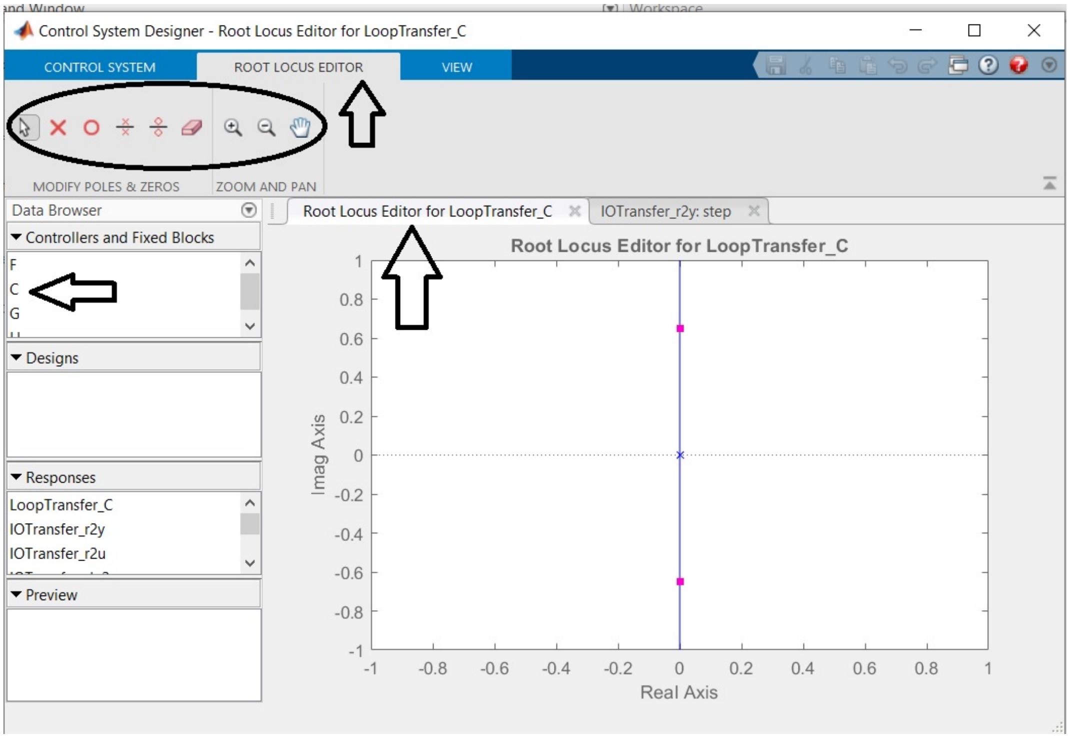

Figure 2: Control System Designer and Root Locus Editor plot

7. Calculate the corresponding ζ and ω_n values, and plot them on the root locus graph using the right-click menu.

8. From Equation (8), we know that our compensator will have two poles and two zeros.

   - Our poles will be at 0 and $-\omega_f$. Just like in the previous lab, ω_f will be $31.42\frac{rad}{s}$.
   - We may infer from Equation (16) to (19) that our zeros will be real numbers with no complex component.

9. Add two real poles as specified above. If you would prefer positioning these poles precisely to dragging them around the graph, you can modify their positions by double clicking on "C" under "Controllers and Fixed Blocks" to bring up the "Compensator Editor".

10. Now add the two zeros. These should be placed fairly close to the origin (between -1 and 0). Remember, you can place poles and zeros more precisely if you zoom in on the root locus graph first! Slide these two zeros around until the root locus response passes through the intersection of ζ and ω_n. Hitting the intersection exactly is impossible, so don't worry about zooming in to get it exact. Approximately hitting the intersection is good enough for our purposes.

11. Modify your system gain until your step response conforms to the design parameters. Feel free to further modify your zero locations if you can't get the response you need. Feel free to use the zoom tools for more precise positioning.

12. Find and write down the equation for "C" in the "Preview" box at the left side of the "Control System Designer", or at the top of the "Compensator Editor" window, If the equation for "C" is not available, double click "C" in the "Controllers and Fixed Blocks" box.

13. MATLAB gives us a controller in the "Preview" box of the form:

$$\frac{K(s+z_1)(s+z_2)}{s(s+p)}$$

or a controller at the top of the "Compensator Editor" window in the form:

$$\frac{A(1+\alpha s)(1+\beta s)}{s(1+\gamma s)}$$

14. Given the values for z_1, z_2, p and K derived from your model, as well as Equations (16) through (19) in the mathematical background section of this document, derive values for ω_f, K_p, K_i, and K_d.

15. Show these values to your TA to complete CHECKPOINT 1.

Setting up the Simulink Model

1. In order to save some time, we will start off with a Simulink model with a PD controller, which is similar to the model in Lab 4. Download the Simulink model ***lab5starter.slx*** from Avenue.

2. Open the model, replace the setpoint input parts (a square wave generator and two other blocks), with a constant block reading zero. The zero position of the ball is in the centre of the ball-beam apparatus. **Save a copy of this model**, as we will use it in the last section of this lab to compare the performance of a PD controller with a PID controller.

3. Continue to take the following steps to transform this PD compensator into a PID controller:

   - Create environment variables for K_p, K_i and K_d.
   - Set the gain on the K_C gain block used in the previous lab to 1 (or simply remove this gain).
   - The branch of the PD controller that only has a gain block and no transfer function will be our proportional control. Assign the new K_p variable to this gain block.
   - Our previous derivative branch lacks a gain block. Add a gain block to this branch before the transfer function, and assign the environment variable K_d to it.
   - Create a new branch with a gain block (with K_i assigned) and an integrator block. Connect the output of this integrator block to the summation block used by the proportional and the differential branches.

4. You have finished your model with a PID controller. **Save this model with a new name**. Show your model to your TA to complete CHECKPOINT 2.

Test the Controller on the Ball and Beam Plant

In this part of the lab, you will examine the effect of PD and PID controllers on steady-state error. To introduce disturbance to make steady-state error more observable, and to compare the effect of the PD and PID controllers on the steady-state error, we will attach a weight at one end of the beam set.

1. Conduct a series of trials using your models, according to the following chart. For your PID values, use the model you constructed in this lab. For your PD values, use the model you derived in Lab 4. Between each trial, stop the model and let the beam rest at its natural initial position.

2. Because the purpose of this lab is to examine the effect of the PD and PID controllers on the system's steady-state errors, when starting your trials, start with any combination of the parameters for your PD controller or any combination of the parameters for your PID controller, then tune your parameters until you believe you get the best steady-state error. Record your trial results.

3. The weight can be adjusted by removing or adding some pieces of metal plates.

Controller Type	Weight Applied	% Overshoot	Settling Time	Steady State Error
PID	No weight			
PID	No weight			
PID	No weight			
PD	No weight			
PD	No weight			
PD	No weight			
PID	half weight			
PID	half weight			
PID	half weight			
PD	half weight			
PD	half weight			
PD	half weight			
PID	full weight			
PID	full weight			
PID	full weight			
PD	full weight			
PD	full weight			
PD	full weight			

4. Show your results to your TA to complete CHECKPOINT 3. Be prepared to answer the following questions:

   - Which controller had better performance in each of the three categories?
   - What was the effect of increasing the mass of the counterweight on both controllers?