CS 331 Lab/Homework #3
Control Systems & Signal Processing

Spring 2003

Week #1: 3/12/2003 – 3/14/2003
Report due by 5:00pm: 3/21/2003

1 Administrative

This assignment will count as both Lab 3 and Homework 3. You must do the work individually and complete your own lab report.

Since there is only one week of lab time for this assignment, we suggest that you concentrate on finishing the simulation diagrams while the TA is available to answer questions. Once you have all the simulations working, go back and play with the parameters to answer the questions.

The following TA office hours will be held in the Engineering Hall EWS Linux lab. If additional office hours are scheduled, they will be announced on the course newsgroup (`uiuc.class.cs331`).

Monday, March 17	2:00 – 4:00pm	Xiaolei
Tuesday, March 18	2:30 – 3:30pm	Tim
Thursday, March 20	2:30 – 3:30pm	Tim

Reports may be submitted to a TA or under the 1102 DCL door at any time up until 5:00pm on Friday, March 21, 2003. To receive credit for late submissions, you *must* email your report as plain text, PDF, or PS attachments to both eriksson@uiuc.edu and xli10@uiuc.edu.

2 Overview

In this lab you will use MATLAB to experiment with some control systems and signal processing topics. The first two exercises use the Simulink tool to simulate the marble problem discussed in lecture and the lab's water seesaw. The final exercise uses a digital low-pass Butterworth filter to reduce the noise in a signal.

This assignment is designed to be followed linearly. You should be able to answer each of the 21 *italicized* "*Q*" questions at the point where it appears in the exercises. Your answers to these questions constitute a significant portion of the lab report (see Section 5 for details).

3 Getting started with MATLAB & Simulink

Log into an Engineering Hall EWS Linux machine[1] and run "`matlab &`" to start MATLAB with the GUI desktop. Alternatively, you can run "`matlab -nodesktop`" to use the older text-based interface. Once at the >> prompt, type "`simulink`" to open the Simulink library.

From the Simulink window, navigate the menus *File* → *New* → *Model*. Then double-click on some groups to see what blocks are available. (You will need to use blocks from the following five groups to complete this assignment: Sources, Sinks, Continuous, Math Operations, and Discontinuities.) To add a block to your model, simply drag-and-drop it from the library. Lines can then be drawn between the blocks to connect them.

[1] `ehlnx`*[1-32]*`.ews.uiuc.edu` if you are working remotely.

4 Exercises

4.1 Marble Simulation

To start, you will simulate the marble control problem presented in lecture. Create a new Simulink model and complete the diagram seen in Figure 1. The triangle blocks are gain multipliers (Math Operations group) and the $\frac{1}{s}$ blocks are integrators (Continuous group). Blocks may be flipped and rotated by right-clicking on them.

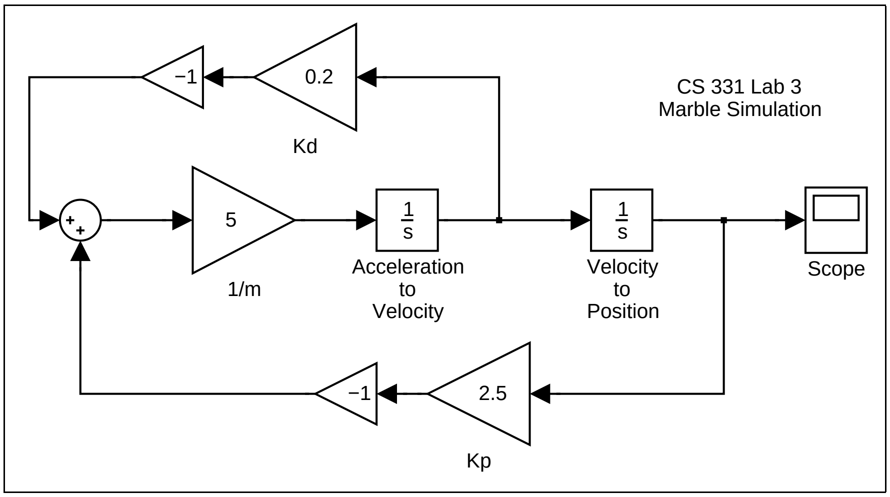

Figure 1. Basic marble PD control simulation diagram. The two inputs to the summation block are the proportional and derivative feedback forces. The net force is divided by the mass of the marble to get the acceleration ($F = ma$). The integral of the acceleration is velocity and the integral of velocity is position. Notice that the proportional gain (K_p) should be positive so that the proportional feedback force pushes the marble toward the set point. Likewise, notice that the derivative gain (K_d) should be positive so that the derivative feedback force pushes to slow the marble down.

Once you have connected all of the blocks as shown, double-click on the $\frac{1}{m}$ multiplier and set its gain to 5 ($m = 0.2$ kg). In the same manner, set $K_p = 2.5$, $K_d = 0.2$, and the two -1 multipliers. Then double-click on the Velocity to Position integrator and set the initial value to 4. This starts the marble off at a position $x_0 = +4$ meters from the set point. You can also set the initial velocity to a non-zero value at the Acceleration to Velocity integrator, but leave $v_0 = 0$ for now. When finished, save your model as "`marble.mdl`".

To run the simulation, double-click on the Scope and then navigate the menus *Simulation* → *Start*. The Scope will show the position error over time. If the system is stable, the error will asymptotically approach zero. Play with the model for a few minutes trying out different values for the five parameters: m, K_p, K_d, x_0, and v_0.

If the axes in the Scope plot are not optimal, right-click within the plot and choose *Autoscale*. To change the duration of time that the simulation runs for, navigate the menus *Simulation* → *Simulation Parameters*, and enter a new stop time.

⇒*Q1.* *Please read Section 5 regarding the lab report prior to writing final answers to these questions.*

Starting with the values given above ($m = 0.2$, $K_p = 2.5$, $K_d = 0.2$, $x_0 = 4$, $v_0 = 0$) each time, describe how the position error graph seen on the Scope responds when you ... (a) Increase the marble's mass (decrease $\frac{1}{m}$). (b) Increase K_p. (c) Increase K_d by a small amount ($K_d \leq 1.0$). (d) Set $K_p = 0$. (e) Set $K_d = 0$. (f) Make K_p negative. (g) K_d negative.

With $K_p = 2.5$ and $m = 0.2$, slowly increase K_d beyond 1.0. Eventually the oscillations will disappear from the position error graph seen on the Scope.

⇒*Q2.* *With $K_p = 2.5$ and $m = 0.2$, beyond what value of K_d do the oscillations disappear from the position error graph? Showing your work, compute the theoretical value of K_d at which the oscillations disappear. How do does the experimental value compare to the theoretical one?*

Gains of $K_p = 2.5$ and $K_d = 0.2$ cause the $m = 0.2$ kg marble system to be stable for all initial conditions (x_0, v_0). We know this by observing the eigenvalues of the system.

To find the eigenvalues of a Simulink system, make sure that your model is saved as "`marble.mdl`" and then return to the MATLAB window with the `>>` prompt. Type "`[A,B,C,D] = linmod('marble')`" to have MATLAB compute the state-space linear model of the *saved* `marble.mdl` diagram. Since the diagram contains PD control feedback, matrix `A` is the closed loop system matrix. "`eig(A)`" will therefore return the closed loop eigenvalues of the system.

⇒*Q3.* *With $m = 0.2$, $K_p = 2.5$, and $K_d = 0.2$, what are the eigenvalues of the marble system? How do we know that the system is stable? How do the eigenvalues and stability change as the initial conditions x_0 and v_0 are varied?*

⇒*Q4.* *Starting with ($m = 0.2$, $K_p = 2.5$, $K_d = 0.2$, $x_0 = 4$, $v_0 = -7$), describe how the position error graph seen on the Scope responds when you set $K_p = 0$. Where did the marble settle and why?*

If s_1 and s_2 are distinct eigenvalues, then the position error is $x(t) = c_1 e^{s_1 t} + c_2 e^{s_2 t}$ where c_1 and c_2 are constants that can be determined from the initial conditions.

⇒*Q5.* *With ($m = 0.2$, $K_p = 0$, $K_d = 0.2$, $x_0 = 4$, $v_0 = -7$), what are the eigenvalues of the system? Showing your work, derive the equation for $x(t)$, this system's position error over time, and solve for c_1 and c_2. What is the numerical limit of $x(t)$ as $t \to \infty$? Does this agree with your answer to Q4?*

Reset $K_p = 2.5$ and $v_0 = 0$. Insert a saturation block[2] from the Discontinuities group into your diagram right after the summation block. Set the maximal force to be ± 5 Newtons. This means that your actuator is limited in the force it can exert.

⇒*Q6.* *Identify an initial condition (x_0, v_0) at which this system behaves significantly different from the case where there is no limitation on the force. Print the two position error graphs (with and without the saturation block) from the Scope and include them in your report. Be sure to label and reference the graphs.*

⇒*Q7.* *Based on your observations, why is it generally not a good idea to push an actuator to the point of saturation in the real world?*

4.2 Water Seesaw Simulation

Now you will model the water seesaw from the lab. Create a new Simulink model and build the partially completed diagram seen in Figure 2. The mass of the marbles is $m_{marbles} = 0.5$ kg and the length of the seesaw bar is $L = 0.8$ meters. The moment of inertia is $I = (m_{marbles} + m_{water}) \times \left(\frac{L}{2}\right)^2$ (the mass of the bar is relatively light, so we ignore it). The seesaw bar hits the table at -15^o and $+15^o$ ($-\frac{\pi}{12}$ and $\frac{\pi}{12}$ radians)[3]. Use the saturation block to take this into account. Set the initial angle of the bar to $+\frac{\pi}{12}$ radians and the initial water mass equal to the mass of the marbles. The coefficient of static friction at the pivot point is $F = 5$.

Add proportional, derivative, and integral control feedback to the diagram such that values for K_p, K_d, and K_i will all be non-negative. When finished, save your model as "`seesaw.mdl`". Now play with the PID control gains. Your goal is to get the seesaw to balance nicely.

Once you have the system balancing nicely, make your diagram presentable. (Label every block appropriately; make sure the gain multipliers are large enough for their value to be visible; etc.)

[2]Note that the `linmod` command ignores all saturation blocks in your model.

[3]Use "`pi`" in MATLAB for the value π.

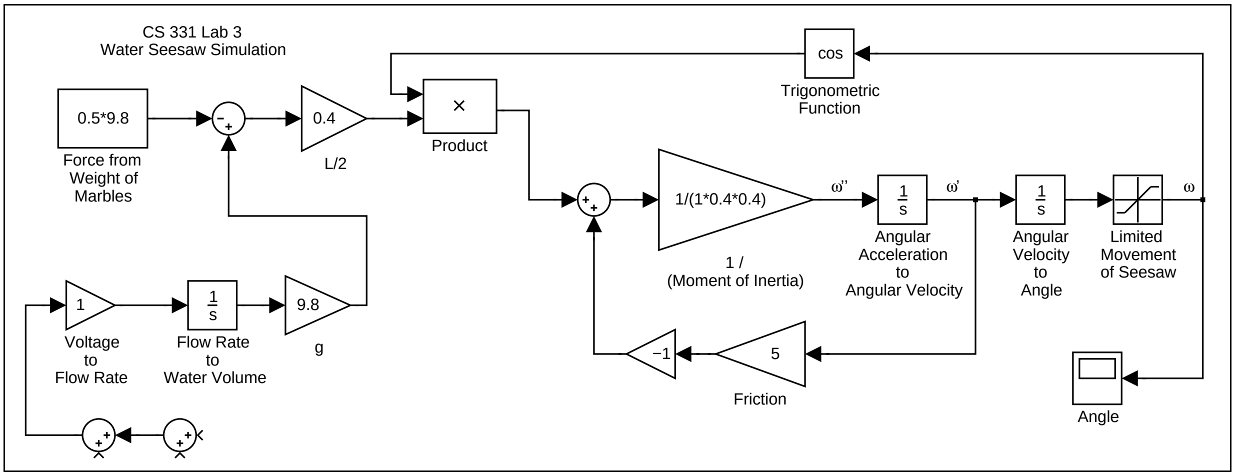

Figure 2. Basic water seesaw simulation diagram with no feedback control. ω is the angle of the seesaw arm (in radians), ω' is the angular velocity, and ω'' is the angular acceleration.

⇒*Q8.* *Print out a copy of your model diagram and include it in your report.*

⇒*Q9.* *List your values for K_p, K_d, and K_i that cause the seesaw to balance nicely. Print the graph of the angle error from the Scope and include it in your report. Be sure to label and reference the graph.*

⇒*Q10.* *What are the eigenvalues of the system from Q9? Based on the eigenvalues, is the system stable?*

⇒*Q11.* *Starting with your system from Q9 each time, describe how the angle error graph seen on the Scope responds when you… (a) Increase K_p. (b) Increase K_d. (c) Increase K_i a little. (d) Increase K_i a lot.*

With big values for K_p and K_d it is possible to balance the seesaw model very quickly (under 2 seconds). Experiment until you find such values.

⇒*Q12.* *What K_p and K_d did you find to balance the seesaw very quickly? Could this be realistically achieved using the physical seesaws in the lab? Explain.*

In the real world, a common use of integral control is to overcome the force of *static* friction.

⇒*Q13.* *In the system that you have modeled with Simulink, is it possible for the seesaw to get stuck near but not at the set point? Is integral control required to ensure that the modeled seesew can balance perfectly? Explain.*

Return to your PID values from *Q9*. Now change the coefficient of friction to $F = 2$.

⇒*Q14.* *How does the angle error graph seen on the Scope respond when you decrease the friction? Why does this happen?*

4.3 Low Pass Filtering

At this point, if you are not familiar with MATLAB, please look over the MATLAB tutorial available at "`http://spicerack.sr.unh.edu/~mathadm/tutorial/software/matlab/`". The sections Vectors, Vector Operations, Plots, and Executable Files will be of interest. Also note that you can type "`help` *`command`*" into MATLAB to get usage information about any command.

For this experiment we will work with three signals, x, y, and z, composed of sine waves at the given frequencies. x has frequency components at 2 Hz, 3 Hz, and 7 Hz. y has frequency components at 18 Hz, 22 Hz, and 24 Hz. z is a weighted sum of x and y: $z = x + \frac{5}{7}y$.

You want to sample z and then use a low-pass Butterworth filter to digitally remove y, the "noise".

⇒*Q15.* *What is the minimum practical sampling rate to capture z without aliasing?*

To sample z, start by constructing a vector t that contains the time points at which samples will be measured, starting at 0 seconds and going to 2 seconds. (The time between sample points should be the period corresponding to *Q15*'s rate.) Then sample x and y by adding together sine waves at the specified frequencies. ($\sin(\Omega \times 2\pi \times t)$ would sample an Ω Hz sine wave at the times specified in the vector t.) Finally, construct z as the weighted sum of x and y.

Plot x, y, and z using the `plot` command. Make sure the horizontal axis is properly scaled with the time in seconds.

⇒*Q16.* *Include the MATLAB code used to generate x, y, z, and the plot of z in your report. Be sure to label and reference the graph.*

Use the `butter` command to design a filter to remove the noise (y) from z. Note that the second argument to `butter` is the desired cutoff frequency divided by half of the sampling rate. Then plot the magnitude of the filter's frequency response with the following commands:

```
[H,F] = freqz(B,A,512,Σ);
plot(F,abs(H)); grid on
```

where `[B,A]` are the filter coefficients returned by `butter` and Σ is your sampling rate. Apply the filter to z with the `filter` command to get z_f. Plot z_f and x on the same graph.

Repeat the above procedure for different values of the filter order N and cutoff frequency ω_c. Take note which combinations do the best job of matching z_f to x.

⇒*Q17.* *At a fixed order, what effect does increasing the cutoff frequency of the filter have on the magnitude of the filter's frequency response?*

⇒*Q18.* *At a fixed cutoff frequency, what effect does increasing the order of the filter have... (a) On the magnitude of the filter's frequency response? (b) On z_f with respect to z?*

⇒*Q19.* *What are some advantages and disadvantages of having a filter with a large order?*

⇒*Q20.* *In your opinion, what is the best order and cutoff frequency for removing the noise from z in this particular case? How did you arrive at these values?*

⇒*Q21.* *Include the MATLAB code used to generate your filter from Q20 and apply it to z. Also include a plot of the magnitude of the filter's frequency response, and a plot of z_f and x on the same graph. Be sure to label and reference the graphs.*

5 Lab Report

The Lab 3 report must be completed individually. Reports may be submitted to a TA or under the 1102 DCL door at any time up until 5:00pm on Friday, March 21, 2003. To receive credit for late submissions, you *must* email your report as plain text, PDF, or PS attachments to both eriksson@uiuc.edu and xli10@uiuc.edu. Keep in mind that this report will count as both a lab and homework assignment.

The report should be typed and include answers to the questions *Q1* through *Q21*. Each response should be made of complete sentences in the form of a coherent paragraph. No credit will be given for imprecise or ambiguous answers. Please partition your report into three sections, one for each of the exercise, and clearly label where you are answering each question with its Q number.

Some of the questions ask you to print out graphs or diagrams. Please make sure these are titled, have their axes labeled, and are referenced appropriately in your response to the particular question.