

ECE-521 Control Systems II

Laboratory 5

The plot thickens, now a 2 DOF system!

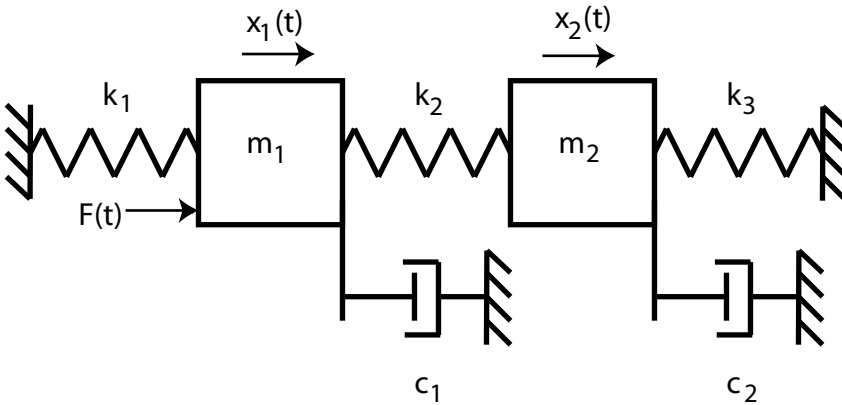
Preview In this Lab you will first obtain a fourth order model of a two degree of freedom spring/mass/damper system (two carts can move), design a state variable feedback controller for it, implement the controller on the real system, and then compare the predicted response with the actual response. You will probably make at least two observations during this lab:

- models are not perfect, they are just guides
- real motors have real limits, which can make designing more difficult

Pre-Lab

1) Print out this lab and **read** it.

Consider the following model of the two DOF system we will be using.



2) Draw freebody diagrams of the forces acting on the two masses.

3) The equations of motion for the two masses can be written

$$m_1 \ddot{x}_1 + c_1 \dot{x}_1 + (k_1 + k_2)x_1 = F + k_2 x_2 \quad (1)$$

$$m_2 \ddot{x}_2 + c_2 \dot{x}_2 + (k_2 + k_3)x_2 = k_2 x_1 \quad (2)$$

If we define $q_1 = x_1$, $q_2 = \dot{x}_1$, $q_3 = x_2$, and $q_4 = \dot{x}_2$, show that we get the following state equations

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\left(\frac{k_1+k_2}{m_1}\right) & -\left(\frac{c_1}{m_1}\right) & \left(\frac{k_2}{m_1}\right) & 0 \\ 0 & 0 & 0 & 1 \\ \left(\frac{k_2}{m_2}\right) & 0 & -\left(\frac{k_2+k_3}{m_2}\right) & -\left(\frac{c_2}{m_2}\right) \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m_1} \\ 0 \\ 0 \end{bmatrix} F$$

In order to get the A and B matrices, we need to determine all of the quantities in the above matrices.

4) Now we want to rewrite equations 1 and 2 in terms of ζ_1 , ω_1 , ζ_2 , and ω_2 as

$$\ddot{x}_1 + 2\zeta_1\omega_1\dot{x}_1 + \omega_1^2x_1 = \frac{k_2}{m_1}x_2 + \frac{1}{m_1}F \quad (3)$$

$$\ddot{x}_2 + 2\zeta_2\omega_2\dot{x}_2 + \omega_2^2x_2 = \frac{k_2}{m_2}x_1 \quad (4)$$

We will get our initial estimates of the parameters ζ_1 , ω_1 , ζ_2 and ω_2 using the log-decrement method. Assuming we measure these parameters, show how $A_{2,1}$, $A_{2,2}$, $A_{4,3}$ and $A_{4,4}$ can be determined.

5) By taking the Laplace transforms of equations 3 and 4, show that we get the following transfer function

$$\frac{X_2(s)}{F(s)} = \frac{\left(\frac{k_2}{m_1m_2}\right)}{(s^2 + 2\zeta_1\omega_1s + \omega_1^2)(s^2 + 2\zeta_2\omega_2s + \omega_2^2) - \frac{k_2^2}{m_1m_2}} \quad (5)$$

6) It is more convenient to write this transfer function as

$$\frac{X_2(s)}{F(s)} = \frac{\left(\frac{k_2}{m_1m_2}\right)}{(s^2 + 2\zeta_a\omega_a s + \omega_a^2)(s^2 + 2\zeta_b\omega_b s + \omega_b^2)} \quad (6)$$

By equating coefficients of powers of s in the denominators in these two transfer functions (equations 5 and 6), you should be able to write down four equations. The equations corresponding to the coefficients of s^3 , s^2 , and s do not seem to give us any new information, but they will be used later to get consistent estimates of ζ_1 and ω_1 . The equation for the coefficients of s^0 will give us a new relationship for $\frac{k_2^2}{m_1m_2}$ in terms of parameters we will be measuring.

7) We will actually be fitting the frequency response data to the following form of the transfer function:

$$\frac{X_2(s)}{F(s)} = \frac{K_2}{\left(\frac{1}{\omega_a^2}s^2 + \frac{2\zeta_a}{\omega_a}s + 1\right)\left(\frac{1}{\omega_b^2}s^2 + \frac{2\zeta_b}{\omega_b}s + 1\right)} \quad (7)$$

What is K_2 in terms of the parameters given in equation 6?

8) Using equation 6 and the Laplace transform of equation 4, show that we can write

$$\frac{X_1(s)}{F(s)} = \frac{\frac{1}{m_1}(s^2 + 2\zeta_2\omega_2s + \omega_2^2)}{(s^2 + 2\zeta_a\omega_a s + \omega_a^2)(s^2 + 2\zeta_b\omega_b s + \omega_b^2)} \quad (8)$$

9) This equation is more convenient to write in the form

$$\frac{X_1(s)}{F(s)} = \frac{K_1\left(\frac{1}{\omega_2^2}s^2 + \frac{2\zeta_2}{\omega_2}s + 1\right)}{\left(\frac{1}{\omega_a^2}s^2 + \frac{2\zeta_a}{\omega_a}s + 1\right)\left(\frac{1}{\omega_b^2}s^2 + \frac{2\zeta_b}{\omega_b}s + 1\right)} \quad (9)$$

What is K_1 in terms of the quantities given in equation 8?

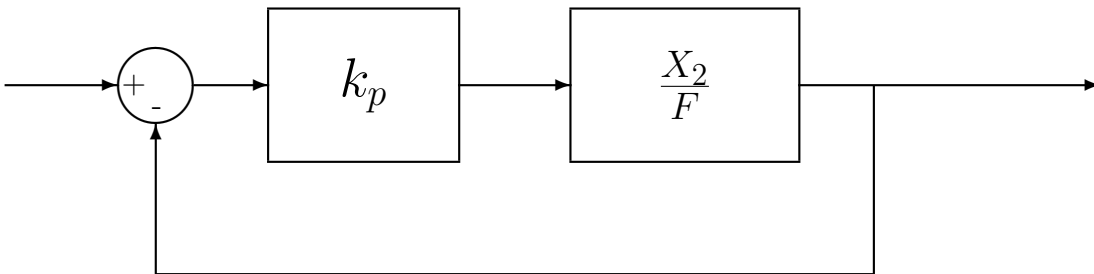
10) Show that

$$A_{4,1} = \frac{k_2}{m_2} = \frac{K_2}{K_1} \omega_2^2 \quad (10)$$

11) Show that

$$A_{2,3} = \frac{k_2}{m_1} = \frac{\omega_1^2 \omega_2^2 - \omega_a^2 \omega_b^2}{A_{4,1}} \quad (11)$$

12) All that's left is to find $\frac{1}{m_1}$, which is b_2 . It's important to understand that this parameter also includes "scaling" on $F(s)$. Now assume we look at the closed loop response to a simple proportional type controller. Hence we have the system shown below:



For a step response of amplitude Amp , show that the steady state value of $x_2(t)$, $x_{2,ss}$ is

$$x_{2,ss} = \frac{K_2 k_p Amp}{1 + K_2 k_p} \quad (12)$$

(Hint: It's easiest to use equation 7 for X_2/F)

and that by rearranging this equation we get

$$b_2 = \frac{1}{m_1} = \frac{x_{2,ss}}{k_p (Amp - x_{2,ss})} \frac{\omega_a^2 \omega_b^2}{A_{4,1}} \quad (13)$$

Summary

1) Fit frequency response data to

$$\frac{X_2(s)}{F(s)} = \frac{K_2}{\left(\frac{1}{\omega_a^2}s^2 + \frac{2\zeta_a}{\omega_a}s + 1\right)\left(\frac{1}{\omega_b^2}s^2 + \frac{2\zeta_b}{\omega_b}s + 1\right)} \quad (14)$$

this will give us initial estimates for K_2 , ζ_a , ω_a , ζ_b , and ω_b .

2) Using the above parameters as initial estimates, fit frequency response data to

$$\frac{X_1(s)}{F(s)} = \frac{K_1\left(\frac{1}{\omega_2^2}s^2 + \frac{2\zeta_2}{\omega_2}s + 1\right)}{\left(\frac{1}{\omega_a^2}s^2 + \frac{2\zeta_a}{\omega_a}s + 1\right)\left(\frac{1}{\omega_b^2}s^2 + \frac{2\zeta_b}{\omega_b}s + 1\right)} \quad (15)$$

this will give us estimates for K_1 , ζ_2 , ω_2 , and new estimates (they will not be very different from the old estimates) for ζ_a , ω_a , ζ_b , and ω_b .

3) Using the relationships derived in part 8, find the values of ζ_1 and ω_1 consistent with the rest of the parameters.

4) Estimate all of the parameters in A .

5) Look at the step response to estimate b_2 .

We need to first **identify** the system:

1. For this lab, be sure
 - The spring between the first cart and the motor is a stiff spring
 - The mass on the first cart is greater than or equal to the mass on the second cart.
2. Estimate an initial second order system model using time domain analysis for each mass (cart). Only the cart you are analyzing should be free to move (lock the other carts in place). Use the **log_dec** program to get initial estimates of ζ_1 , ω_1 , ζ_2 and ω_2 .
3. Be sure both of the first two carts are free to move. Measure the frequency response (make one measurement at 1Hz, 2Hz, ..., 7 Hz, and at least 4 points near the two resonant peaks). You need to measure the amplitudes of both the first and the second carts. Be very careful here, if you plot them both on the same graph there are two different scales (one on the left axis, on on the right).
4. Enter the frequency data into the program **process_data_2carts**, and then run the program with the output going to variable *data*, i.e., type

```
> data = process_data_2carts;
```

5. Now use the program **model_2carts** The arguments to this file procedure generally initial guesses of the final parameters. The arguments (and reasonable guesses) are:
 - *data*, data from the **process_data_2carts** routine
 - *K2*, the estimated gain, try 1000 for an initial guess
 - ω_a , use the estimated first resonant frequency in rad/sec
 - ζ_a , use 0.1
 - ω_b , use the estimated second resonant frequency in rad/sec
 - ζ_b , use 0.1
 - ω_1 , use your results from **log_dec**
 - ζ_1 , use your results from **log_dec**
 - ω_2 , use your results from **log_dec**
 - ζ_2 , use your results from **log_dec**

The output of the routine is the *A* matrix, so you should invoke the routine with

```
> A = model_2carts(...
```

6. You should get two plots from this routine, the first is the estimated transfer function for X_2 and the second is the estimated transfer function for X_1 . *You need to include these two initial plots in your memo, so save them now!* At this point you will probably need to add more data points. In particular, you need to be sure there are at least 4 data points near each peak and 4 data points near each null. Most likely you will need to add data points for the null of X_1 . Edit **process_data_2carts** and rerun it, then rerun **model_2carts**.

7. Before you go on, make sure I see the fit of your data to the measured frequency response.
You need to include these two final plots in your memo, so save them now!
8. Finally, we need to estimate b_2 , to be described below.

Estimating b_2

0. Set the units

Click **Setup** → **User Units** and set the units to **cm**.

1. Setting up the controller

Click **Setup** → **Control Algorithm**. Be sure the system is set for *Continuous Time*. Select **PID** under **Control Algorithm**. Click on **Setup Algorithm**. Be sure **Feedback** is from **Encoder 2**. Set k_p to a small number (less than or equal to 0.05) and be sure $k_d = 0$ and $k_i = 0$. Then click **OK**. Next Click **Implement Algorithm**. The click **OK**.

2. Setting up the closed loop trajectory

Click **Command** → **Trajectory**. Select **Step** and click on **Setup**. Select **Closed Loop Step** and set **Step Size** to 0.5 to 1.5 cm. Be sure to record this step size (we'll refer to the amplitude as *Amp* below). Set the **Dwell Time** to something like 2000 ms, this is the time the system will be recording data. Finally click **OK**, then **OK** and you should be back to the main menu.

3. Executing the closed loop step

Click **Command Execute**. A menu box will come up with a number of options, and a big green **Run** button. Click on the **Run** button. When the system has finished collecting data, a box will appear indicating the how many sample points of data have been collected. (If you have hit a stop, the system stops recording data. This usually means you're input amplitude was too large or k_p was too large.) Click on **OK** to get back to the main menu.

4. Determining the steady state value

Click **Plotting** → **Setup Plot**, or just **Plotting Data** → **Plot Data**. Look at the steady state value ($x_{2,ss}$). You may need to change the dwell time if your system has not reached steady state.

5. Estimating b_2

Estimate b_2 using the formula derived previously:

$$b_2 = \frac{1}{m_1} = \frac{x_{2,ss}}{k_p(Amp - x_{2,ss})} \frac{\omega_a^2 \omega_b^2}{A_{4,1}} \quad (16)$$

Note that the value $\frac{\omega_a^2 \omega_b^2}{A_{4,1}}$ is printed out by **model_2carts** and is referred to as the variable *scale*.

You need to go through this procedure at least three times for each configuration. You must use at least two different values of k_p and two different values of input amplitude Amp . If none of the steady state values is larger than 0.4 cm, increase either k_p or Amp . Average the three results to get your b_2 (they should be similar). For the trials I've run, I've got b_2 between 6000 and 9000. Yours may be outside this range though.

Designing and Implementing the State variable Controllers

Our general goals for the state variable controllers are as follows:

- produce a position error of less than 0.15
- reach steady state within 1 seconds (the faster the better)
- have as little overshoot as you can manage

Here are some general ideas:

- You need to try and use positive values for k_1, k_2, k_3 and k_4 ($\underline{k} = [k_1 \ k_2 \ k_3 \ k_4]$). The system does not respond very well to negative values, though negative values with small magnitudes are usually alright. In particular, the steady state values may be off.
- k_1 should be larger than k_2 , and k_3 should be larger than k_4 . k_2 and k_4 are multiplying the derivatives, and estimates of derivatives tends to be noisy.
- Try to keep k_2 and k_4 less than 0.05, and k_1 and k_3 less than 1.0.
- Be sure that the gains for the third carriage are set to zero
- You may need to reset the controller often, such as every time you want to implement a new controller. Click **Utility** → **Reset Controller**. Only do this **before** you have implemented a controller.
- You may need to rephase the motor. Click **Utility** → **Rephase Motor**
- Be sure to **Implement** the controller you have designed.
- Try and track a step with an input amplitude of 0.5 to 1 cm.

Direct (Trial and Error) Method

0. Determine Which Cart Location is the Output

You will need to determine which cart location is the output of the system. This will allow you to construct the appropriate C matrix. Most likely you'll want the position of cart 2 to be the output, but you should also try controlling the position of cart 1 also.

1. Estimate the Gains

Use the program **state_variables_2carts** to guess values for $k_1 \dots k_4$. The program will print out the corresponding locations of the closed loop poles and the correct gain k_{pf} , as well as produce a plot of the system with state variable feedback.

The arguments to this program are:

- the amplitude of the input signal (in cm)
- the A matrix
- the B vector
- the C vector
- the feedback gain matrix $\underline{k} = [k_1 \ k_2 \ k_3 \ k_4]$
- the length of time to run the simulation for
- the file name with containing the response of the real system in single quotes. At this point, the filename is just "

2. Implement the Gains on the ECP System

Once your simulated system has a reasonable response, and probably more importantly, reasonable gains, try running the ECP system with these gains. If the gains are not too large and the system works, save the results to a file. If the system buzzes and doesn't work, go back to step 1 and try again.

3. Comparing the Simulation and the ECP system

Edit the file you saved in part 2 so Matlab can read it. Run the program **state_variable_2carts** again, with the same gains as you used on the system. This time the last argument to the program is the name of the file you saved the response of the system into. You should get a plot containing both the real system and the simulated system. You may want to reduce the final time of the plot so there is not alot of time at steady state showing.

4. Practice Makes Perfect

Try at least three different combinations of gains before you move on to the next method. Be sure to produce a plot for each system, and record the gains and closed loop poles for each system.

Linear Quadratic Regulator Method

1. Estimating the Feedback Gains

Use the Matlab routine **lqr** to estimate the feedback gains k_1, k_2, k_3 and k_4 . The arguments to this routine are

- the A matrix of the system
- the B matrix of the system
- a penalty matrix Q
- a penalty matrix R

(Note there is one more possible argument, but we won't use it. Type help lqr for more information).

The Linear Quadratic Regulator finds the gain \underline{k} to minimize

$$J = \int_0^{\infty} [\underline{x}^T(t)Q\underline{x}(t) + u(t)Ru(t)] dt$$

where

$$\begin{aligned}\dot{\underline{x}}(t) &= A\underline{x}(t) + Bu(t) \\ u(t) &= -\underline{k}\underline{x}(t)\end{aligned}$$

In our case Q is a four by four positive definite matrix, and R is a scalar. Since Q is most likely a diagonal matrix, it's easiest to iterate using the following command in Matlab

```
> K = lqr(A,B,diag([q1 q2 q3 q4]),R)
```

where $q1, q2, q3$ and $q4$ are the desired diagonal elements of Q and R is a scalar. In general, as R gets larger (it may have to get very large), the size of the gains goes down. Also, if a gain is negative, try setting the weight on that gain very large. Note that if all of the gains are large there is no real effect! Iterate on values of Q and R until you think you have something that works.

2. Determining k_{pf}

Again used the program **state_variables_2carts**. The values for \underline{k} have been determined by the lqr routine above. The program will print out the corresponding locations of the closed loop poles and the correct gain k_{pf} , as well as produce a plot of the system with state variable feedback.

The arguments to this program are:

- the amplitude of the input signal (in cm)
- the A matrix
- the B matrix
- the C matrix
- the feedback gain matrix $\underline{k} = [k_1 \ k_2 \ k_3 \ k_4]$
- the length of time to run the simulation for
- the file name with containing the response of the real system in single quotes. At this point, the filename is just ”

3. Implement the Gains on the ECP System

Once your simulated system has a reasonable response, and probably more importantly, reasonable gains, try running the ECP system with these gains. If the gains are not too large and the system works, save the results to a file. If the system buzzes and doesn't work, go back to step 1 and try again.

4. Comparing the Simulation and the ECP system

Edit the file you saved in part 3 so Matlab can read it. Run the program **state_variables_2carts** again, with the same gains as you used on the system. This time the last argument to the program is the name of the file you saved the response of the system into. You should get a plot containing both the real system and the simulated system. You may want to reduce the final time of the plot so there is not alot of time at steady state showing.

4. Practice Makes Perfect

Try at least three different combinations of gains (corresponding to three different values of Q and R). Be sure to produce a plot for each system, and record the gains and closed loop poles for each system.

Memo

Your memo should compare (briefly) the response of the model and the response of the real system for the different gains you tried. You should have some description of the configuration of the system you were trying to control.

You should include the following items as attachments. Most of these are figures which should have reasonable captions.

- The initial fit of the transfer functions to the measured frequency response of the system.
- The final fit of the transfer functions to the measured frequency response of the system.
- The data used to determine b_2 .
- The final model of the system.
- The predicted and actual response of the system to each of the different controllers where you guessed the values of \underline{k} , and the corresponding closed loop pole locations.
- The predicted and actual response of the system to each of the different controllers where you used the lqr algorithm to determine the values of \underline{k} . Also record values of Q and R used and the corresponding closed loop pole locations.