

ECE-205 Lab 3

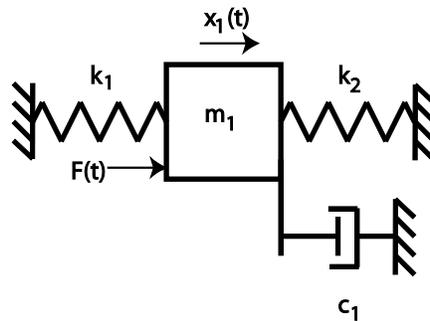
Time Domain Modeling of One Degree of Freedom Systems

Overview

In this lab you will be modeling two one degree of freedom (second order) systems using time-domain analysis. The goal is develop some intuition into how the parameters in a second order system affect the output by varying parameters in a model to match the step response of two second order systems. You will also use the log-decrement method to estimate these parameters. Your station has either a rectilinear (model 210) or torsional (model 205) system connected. You should make two different systems (and corresponding models) using either the model 210 or model 205.

Background

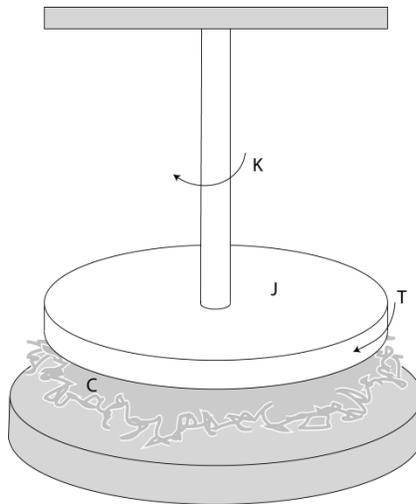
A one degree of freedom rectilinear mass-spring-damper system can be modeled as



By drawing a free body diagram and balancing forces, we get the equation of motion:

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

A one degree of freedom rotational mass-spring-damper system can be modeled as



By drawing a free body diagram and balancing torques, we get the equation of motion

$$J \ddot{\theta}(t) + c \dot{\theta}(t) + k \theta(t) = T(t)$$

Despite the fact that the systems appear quite different, the governing differential equation for both of the one degree of freedom systems can be put into the standard form we have been using to model our circuits:

$$\ddot{y}(t) + 2\zeta\omega_n\dot{y}(t) + \omega_n^2y(t) = K\omega_n^2x(t)$$

Here K is the static gain, ω_n is the natural frequency, and ζ is the damping ratio. These are the parameters we need to determine for these models.

You will need to set up a folder on the desktop for ECE-205, and then a folder for Lab 3. Then copy **Lab3 files.rar** (from the class website) into this folder and install the files. Start Matlab and change the default folder to the folder where these files are located.

You will need to go through the following steps for *two different* configurations (different masses and/or springs or locations of masses for the torsional systems).

PART I : Setting up communications

In this part we get the systems ready to run and start (hopefully) communications between Simulink, the miniPMAC card, and the ECP system. Sometimes this can take a few tries, so be patient and ask for help.

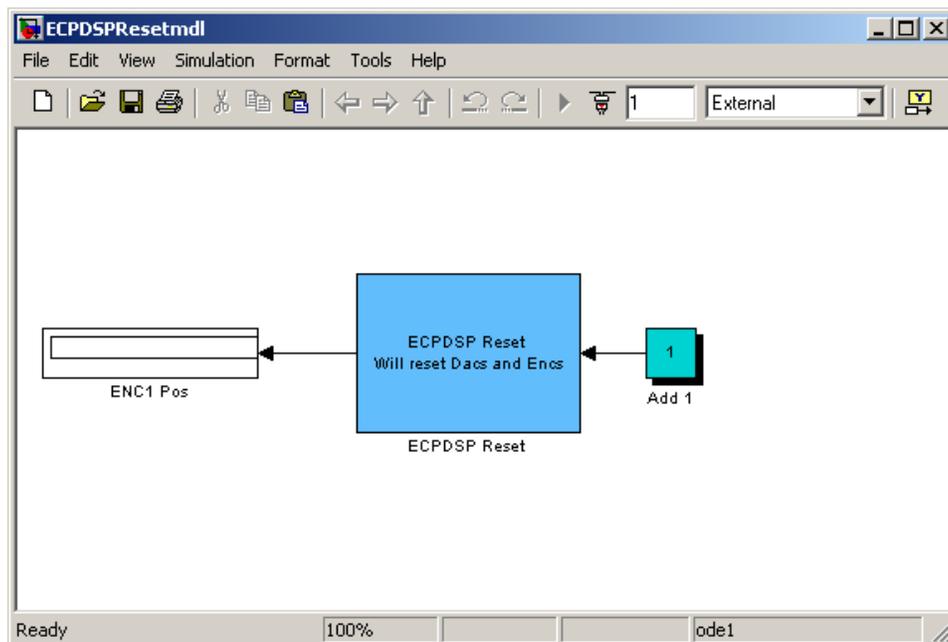
1) We need to inform the ECP system that we will be using Simulink and the real-time windows target. To do this, click **Start -> Programs -> ECP**

First select on **Utility-> Download Controller Personality File.**

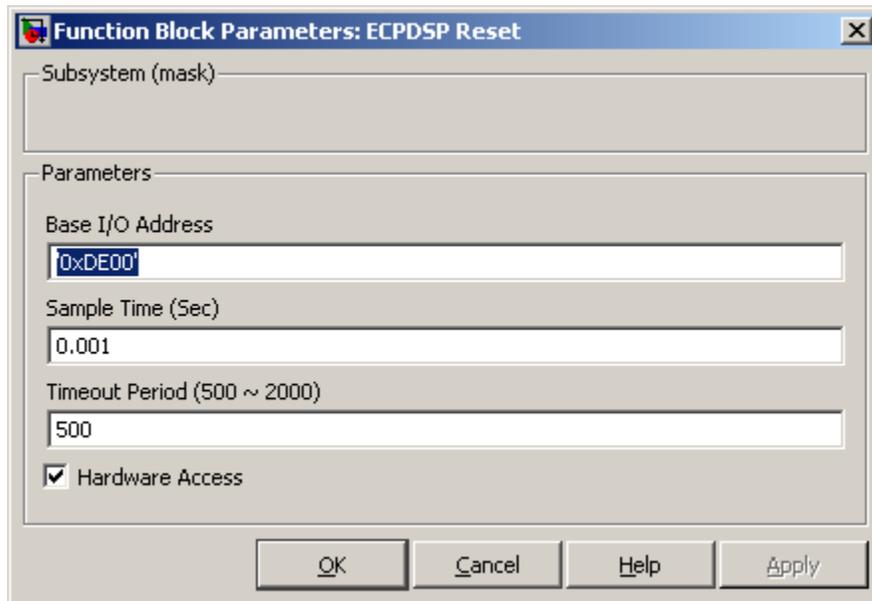
Then select **C: -> Program Files -> ECP Systems -> cn** (it may default to this)

Finally select **m210_rtwt_3.pmc** for the Model 210 (or **m205_rtwt_3.pmc** for the Model 205) and click on **open**. Wait for the ECP system to load the personality file, then close the window (do not just minimize it).

2) Now we need to reset the system. You need to do this each time before you run the system. From Matlab, open the Simulink file **ECPDSPReset.mdl**. It should look like the following:

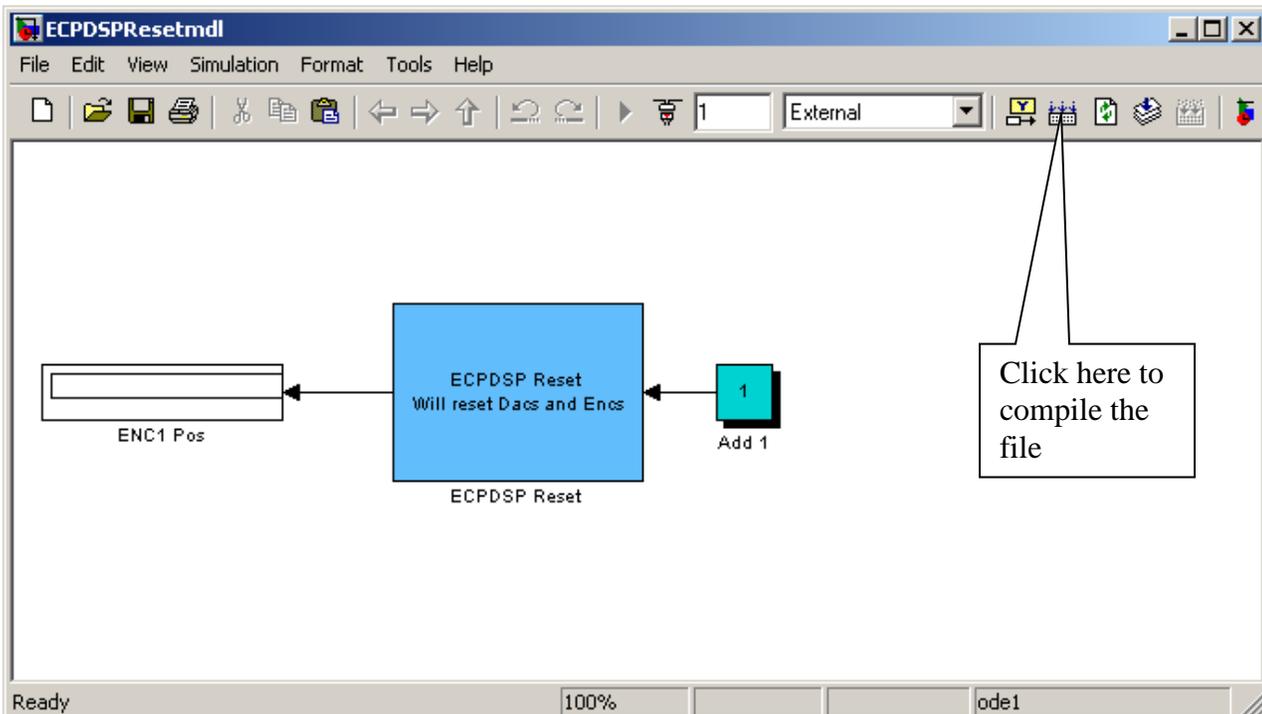


We first need to be sure the **Base I/O Address** is correct for your work station. Double click on the blue ECPDSP Reset box, and you will get something like the following (your base address is probably 0xD800):

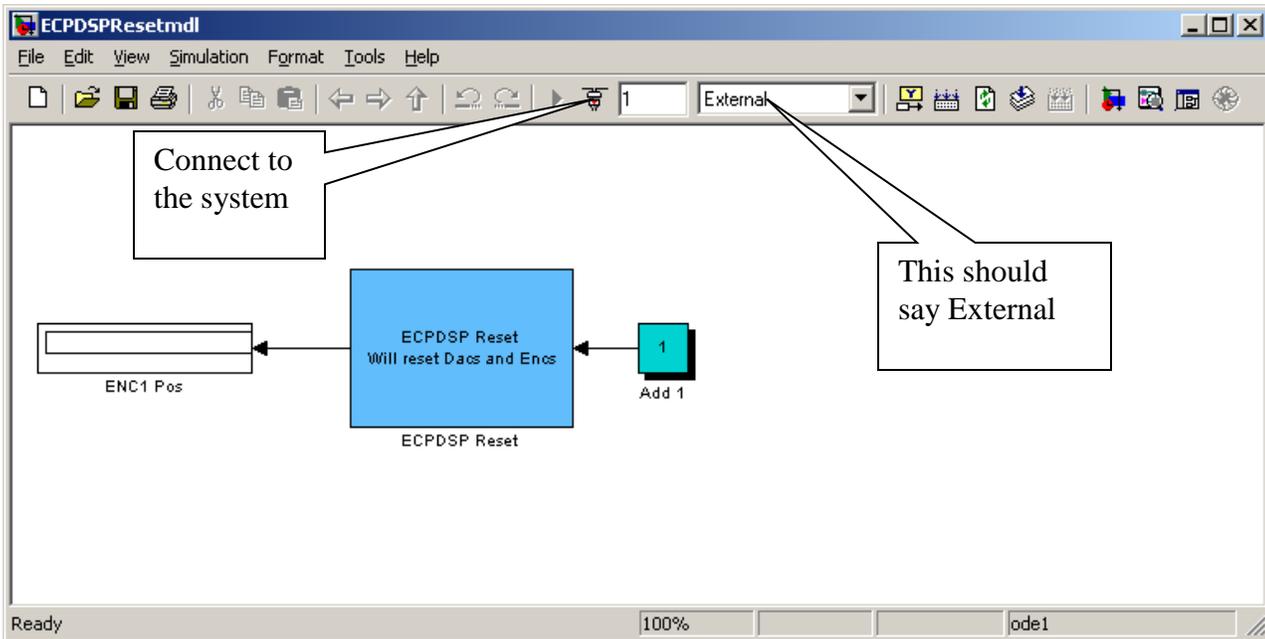


Be sure to save your file if you change the Base I/O Address.

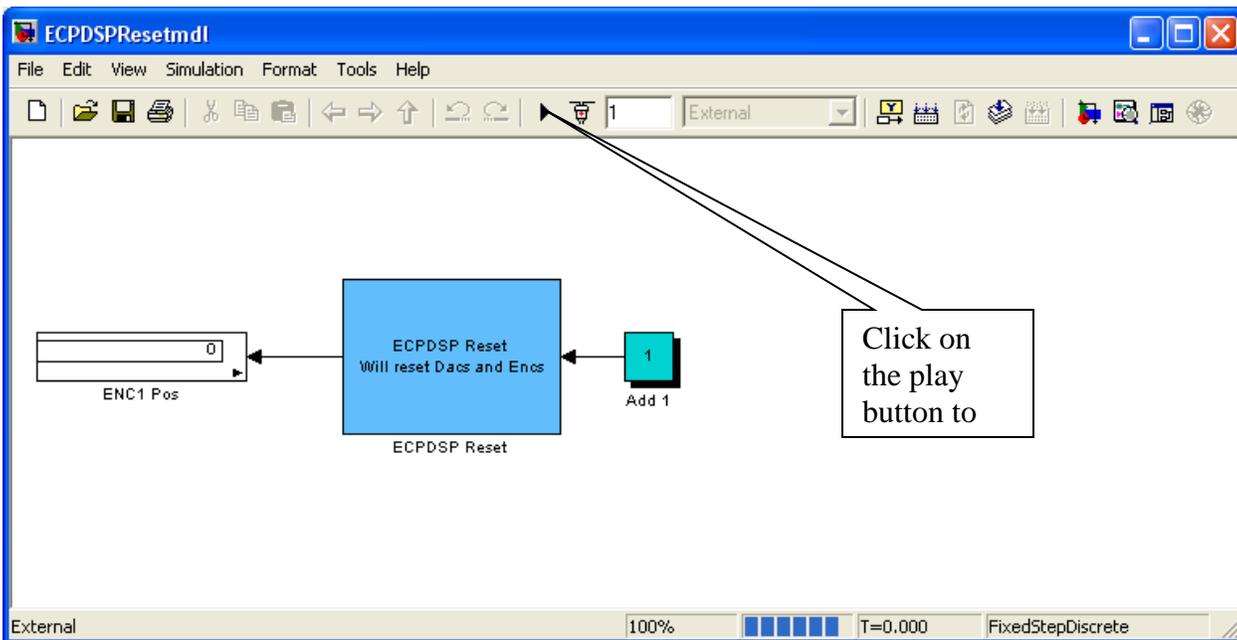
Now we need to **compile** this file. Once you have clicked on the icon (shown in the figure below), wait until Matlab indicates the file was successfully generated (it will be at the end of a lot of messages in the command window).



Finally, we need to run this file. First **connect** to the system,



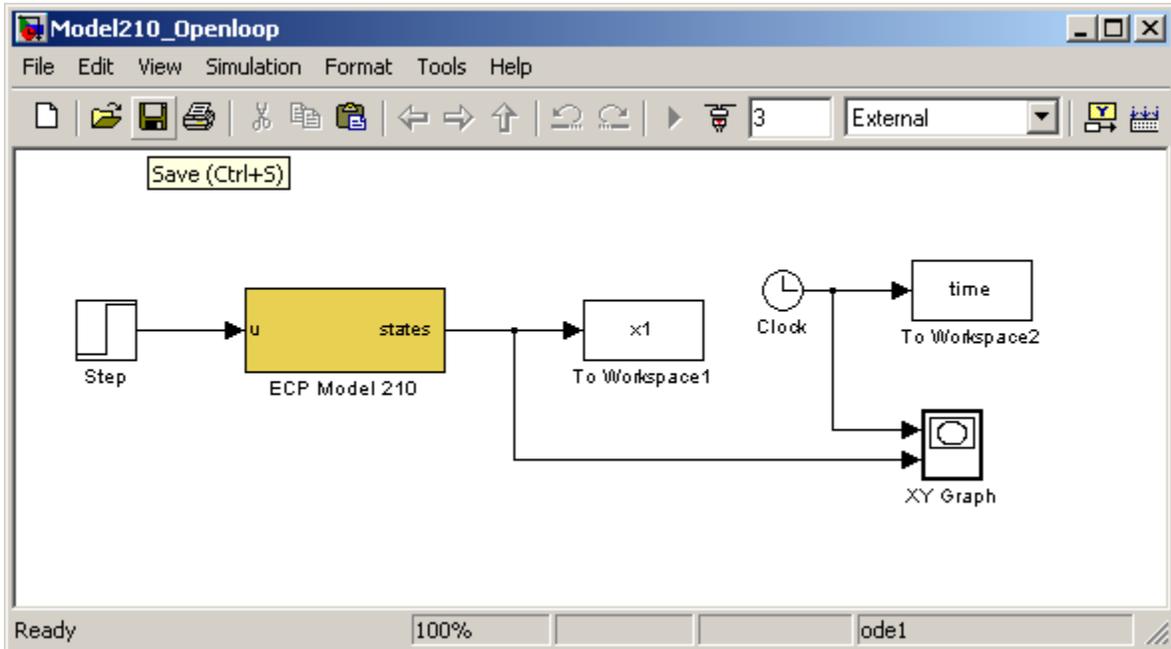
Then click on the **play** button



If you will do all of this correctly, nothing (obvious) will happen. In the Matlab command window it will say you have connected and disconnected to the system. However, you have reset the counters and zeroed the system.

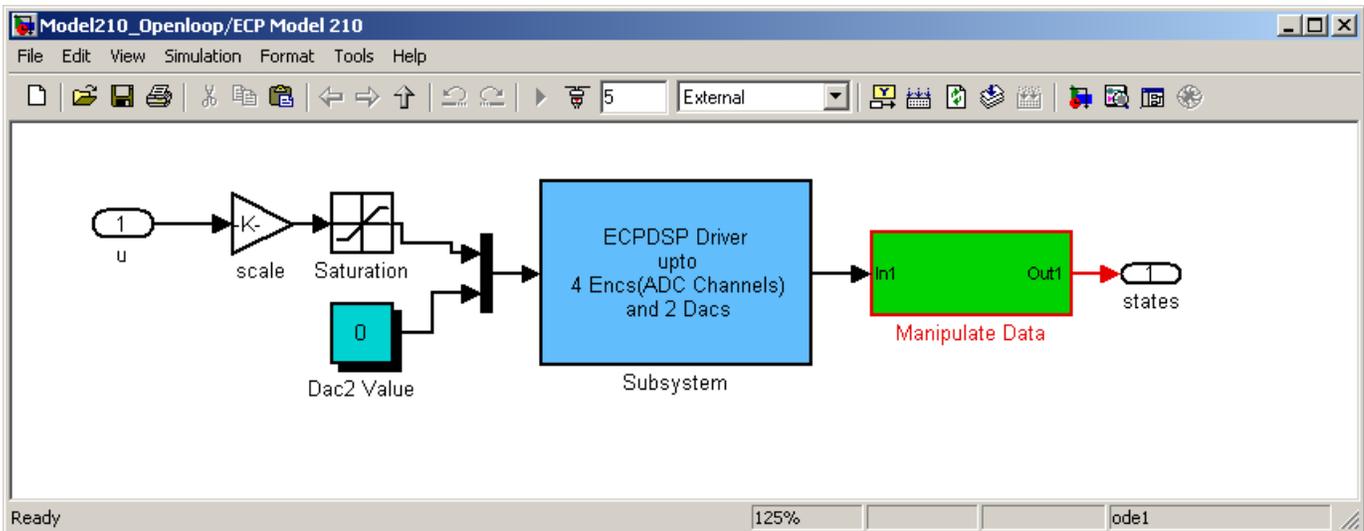
PART II: Second order rectilinear system (ECP Model 210)

- 1) *Set Up the Mechanical System.* Only the first cart should move, all other carts should be fixed. You need to have at least one spring connected to the cart and at least one mass on the cart.
- 2) Open up the Simulink file **Model210_Openloop.mdl**, it should look like the following:



The yellow block in the middle is what actually connects to the ECP system. Note that the output of this system is labeled x1. **Do not change this!**

- 3) Double click on the yellow block, and you will get the following:



Double click on the blue ECPDSP Driver block to be sure your Base I/O Address is correct. Then save the files.

4) Now we want to change the input to the system. Double click on the **Step** block, and enter an input value. Note that some of the systems have a static gain of more than 40, and they don't like moving more than about 2 cm, so start with something small, like a final value of 0.02. Click **OK** and then save the file.

5) **Compile** the file, just as you did in PART 1, then connect to the system, and then run it. The cart *should* move. There are three (usual) outcomes here:

a) If the cart does not move, go back though PART I. You don't need to change the Base I/O Address or recompile, but you do need to reload the controller personality file and reset the system.

b) If the cart moved, but the XY Graph does not look smooth, the communications are still not correctly set up (this is not your fault). Go back though PART I. You don't need to change the Base I/O Address or recompile, but you do need to reload the controller personality file and reset the system.

c) If the cart moved and the XY Graph looks reasonable, be sure the system recorded data for three seconds. If it did not record for three seconds, first look at the output level. If the maximum value of the output is between 1 and 2 cm (or larger), try resetting (ECPDSPReset) and rerunning the system a few times. If the maximum value of the output is less than about 1.5 cm, then increase the value of the amplitude in the **Step**, reset the system (ECPDSPReset) and rerun the system.

Before you go on, be sure your system has run for 2-3 seconds and recorded enough good data.

6) Modify the file **second_order_driver.m** to plot the data from the ECP system and the results from the simulation on the same graph. Be sure to use different line types and a legend. There are two different time arrays (one from the simulation and one from the ECP system), and two different outputs. You may need to use **axis** command to limit the time axis if your ECP system did not run long enough. Also, be sure the y-axis indicates the output is displacement (in cm).

7) Modify the values of the static gain, then the damping ratio, and then the natural frequency to try and get the model to match the system as closely as possible *during the first part of the step* (it is likely to get worse as time progresses.) You will have to iterate a bit on this. Once you have a reasonably good fit, copy the Matlab figure into a word file (use **Edit**, then **Copy Figure**).

8) Before you go on, we need to rename our variables from the ECP system so we can use them again. In the Matlab **command window**, type

```
time_save = time;  
x1_save = x1;
```

PART III: The log decrement method

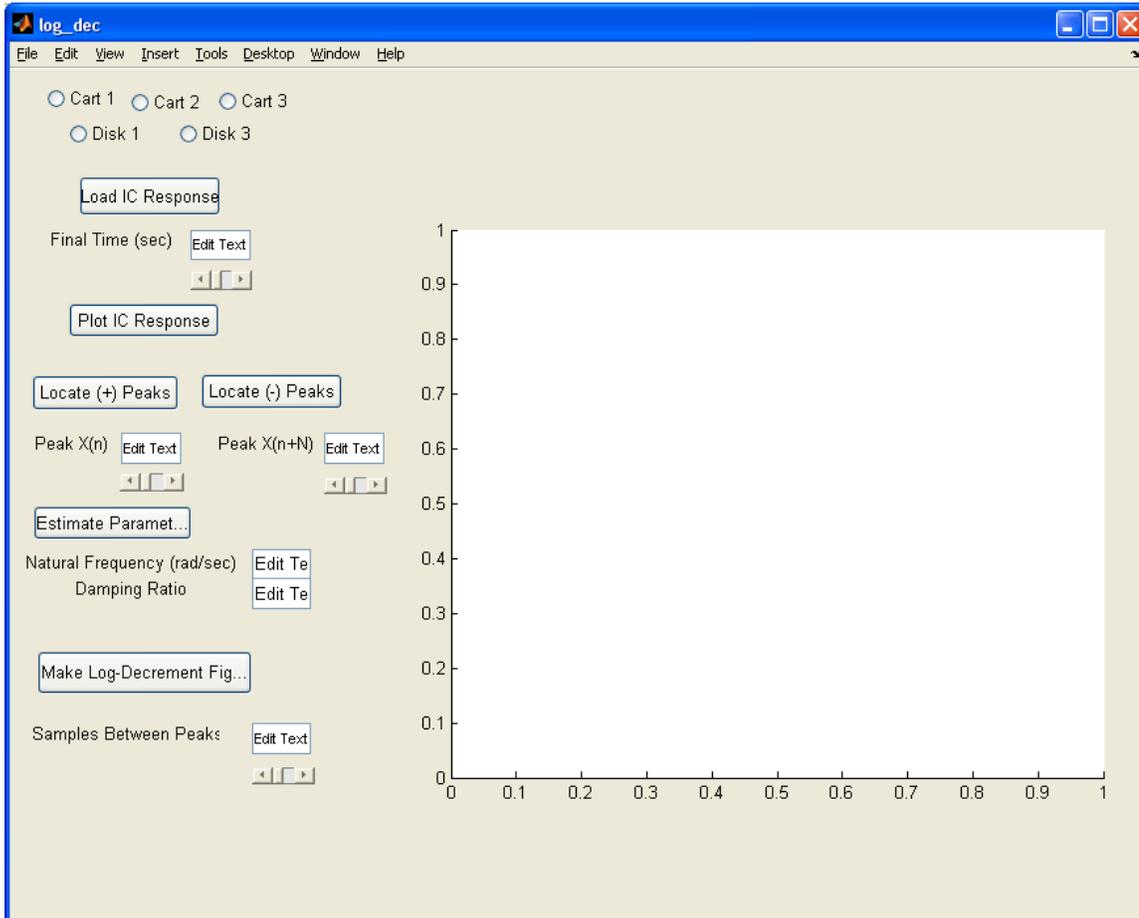
As you recall, the log decrement method is a way of estimating the natural frequency ω_n and damping ratio ζ of a second order system. However, this method does not determine the static gain.

1) We have automated the log-decrement algorithm since it is so useful for these systems. You will go through the following steps:

- Reset the system using **ECPDSPresetmdl.mdl**.
- Modify **Model210_Openloop.mdl** so the input has zero amplitude.
- Compile **Model210_Openloop.mdl** if necessary.

- Connect **Model210_Openloop.mdl** to the ECP system. (The mode should be **External**.)
- Displace the first mass, and hold it.
- Start (**play**) **Model210_Openloop.mdl** and let the mass go.
- Run the m-file **log_dec.m**. This should be in the same directory as **Model210_Openloop.mdl** and **log_dec.fig**. This routine assumes the position of the first cart is labeled $x1$ and the time is labeled $time$. (These are the defaults in **Model210_Openloop.mdl**.)

The program **log_dec** comes up with the following GUI:



You need to

- Select **Cart 1**
- Select **Load IC (initial condition) Response** (the variables $time$ and $x1$ or $theta1$ will be loaded from the workspace). At this point some initial estimates will be made.
- Set/modify the **Final Time**
- Select **Plot IC Response** to plot the initial condition response
- Choose to identify the positive peaks (**Locate + Peaks**) or negative peaks (**Locate - Peaks**). If the peaks are not numbered consecutively, you need to decrease the **Samples Between Peaks** and try again until all peaks have been identified.
- Choose the initial peak (**Peak $x(n)$**) and final peak (**Peak $x(n+N)$**) to use in the log-decrement analysis. These should be fairly close to the beginning of the initial condition response. Don't try and use more than a few peaks.
- Select **Estimate Parameters** to get the initial estimates of ζ and ω_n
- Select **Make Log-Decrement Figure** to get a plot and summary of the results. You need to include this figure in your memo.

2) Use the values of the natural frequency and damping ratio determined using the log decrement method in your simulation. Before you run your simulation you will need to recover the saved variables by typing in the Matlab command window:

```
time = time_save;  
x1 = x1_save;
```

Include your new graph in your memo. *This graph should include the model results with your parameters determined from the log decrement method, and the step response of the real system.* Be sure to include a figure number and a figure caption.

PART IV: Second order torsional system (ECP Model 205)

Redo all of the above procedures with the following exceptions (read them before going on!):

0) Be sure to set the ECP controller personality file to **m205_rtw_3.pmc**

1) The input to the torsional system is in radians, so you need to multiply the desired degree angle by $\frac{\pi}{180}$

2) Start with small angles (like 1 degree) until the maximum output is around 15 to 20 degrees.

3) Be sure all of your computations are done in either degrees or radians. *The variable written to Matlab from the ECP system will be in degrees.*

4) Be sure to save your variables (*theta1* and *time*) before you do the log decrement!!

5) Be sure your output is labeled correctly (in radians or degrees)

Your memo should contain 6 graphs, each of which is neatly labeled with a Figure number and a caption. The body of your short memo should include a table comparing values used trying to match the step response and using the log decrement method and what you may think causes any differences.