

## ECE-320 Lab 2

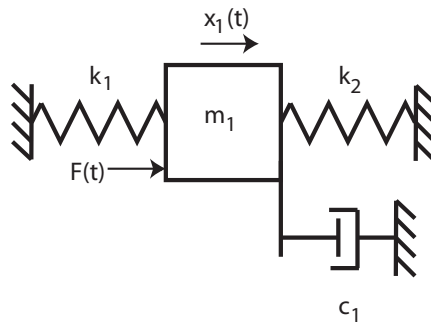
### Time Domain Modeling of One Degree of Freedom Systems

#### Overview

In this lab you will be modeling two one degree of freedom systems using time-domain analysis. The goal is to review the log-decrement method and step response analysis for modeling 2nd order systems. If your station has both a rectilinear (model 210) and torsional (model 205) you should make one model of each type of system. Otherwise make two different systems (and corresponding models) using either the model 210 or model 205.

#### Background

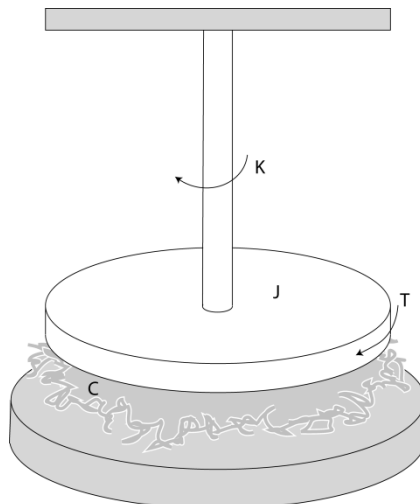
A one degree of freedom rectilinear mass-spring-damper system can be modeled as



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 transfer functions for both of the one degree of freedom systems can be put into the standard form

$$G(s) = \frac{K}{\frac{1}{\omega_n^2} s^2 + \frac{2\zeta}{\omega_n} s + 1}$$

Here  $K$  is the static gain,  $\omega_n$  is the natural frequency, and  $\zeta$  is the damping ratio. These are the parameters we need to determine for these models.

You will need to set up a folder for Lab 2 and copy all of the files from the **basic\_files** folder into this new folder.

### **Part A: One degree of Freedom Rectilinear Systems**

You will need to go through the following steps for *two different* configurations (different masses and/or springs) unless you are going to use one torsional and one rectilinear system.

Be sure to load the correct controller personality file for the ECP system (and reset the controller)!!!

Step 1: Set Up the 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. Do not use the damper. **Be sure you write down all of the information you need to duplicate this configuration.**

Step 2: Log Decrement Estimate of  $\zeta$  and  $\omega_n$

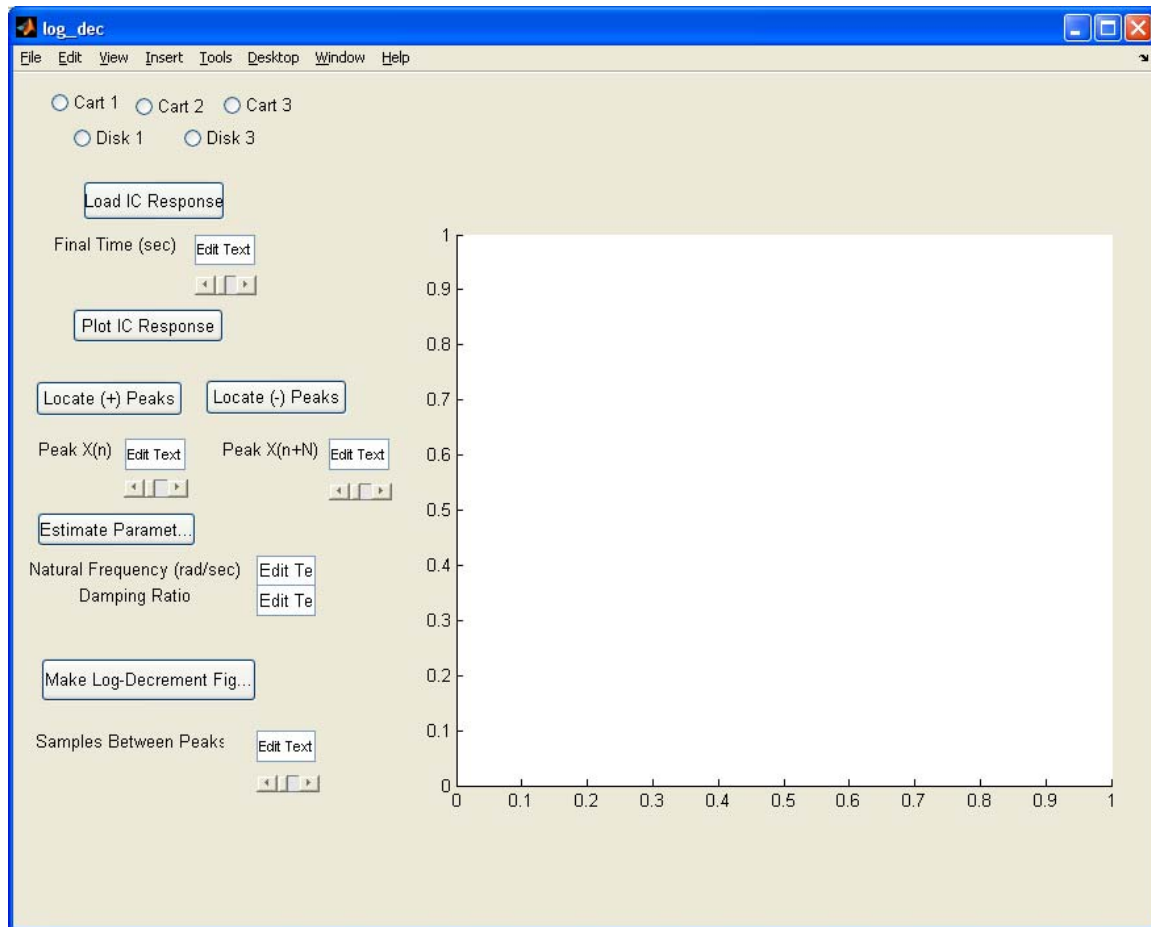
As you recall, the log decrement method is a way of estimating the natural frequency  $\omega_n$  and damping ratio  $\zeta$  of a second order system. However, this method does not determine the static gain.

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 *y* 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  $\zeta$  and  $\omega_n$
- Select **Make Log-Decrement Figure** to get a plot and summary of the results. You need to include this figure in your memo.

### Step 3: Estimating the Static Gain $K$

You will go through the following steps:

- Reset the system using **ECPDSPresetmdl.mdl**.
- Modify **Model210\_Openloop.mdl** so the input is a step. You may have to set the mode to **Normal**.
- Set the amplitude to something small, like 0.01 or 0.02 cm.
- Compile **Model210\_Openloop.mdl**, if necessary.
- Connect **Model210\_Openloop.mdl** to the ECP system. (The mode should be **External**.)
- Run **Model210\_Openloop.mdl**. If the cart does not seem to move much, increase the amplitude of the step. If the cart moves too much, decrease the amplitude of the step. You may have to recompile after the change.
- You only need to run the system until it comes to steady state, then stop it.

Estimate the static gain as

$$K = \frac{y_{ss}}{A}$$

where  $y_{ss}$  is the steady state value of the cart position, and  $A$  is the input amplitude. You should determine the value of  $y_{ss}$  in Matlab, **don't use the X-Y Graph**. The variables  $y$  and  $time$  should be in your workspace. You should use **three different input amplitudes** and produce three different estimates for the static gain. Try and make your systems move so the steady state values are between 0.5 and 1.0 cm. Average your three estimates to produce a final estimate of the static gain. Your static gain should generally be between 10 and 30. If yours is not, it is likely you did not use the same units for  $A$  and  $y_{ss}$ .

### Step 4: Second Order Step Response

We will now try to identify the required system parameters by determining the step response of the real (ECP) system, simulating an ideal second order system, and comparing the two responses. If you cannot make the ideal system match the real system, make the ideal system match the real system as well as possible *at the beginning of the response*.

This part of the lab should be done *independently* of **Parts 2** and **3**. That is, ***do not*** just use the values for the natural frequency  $\omega_n$ , damping ratio  $\zeta$ , and static gain  $K$  you obtained previously. Part of the reason for doing this is so you can develop a better feel for how changing these parameters will change the step response. You might want to have one partner do **Parts 2** and **3** and the other do this part for the same system, and then switch when you do the other system.

You will go through the following steps:

- Reset the system using **ECPDSPresetmdl.mdl**.
- Modify **Model210\_Openloop.mdl** so the input is a step. You may have to set the mode to **Normal**.
- Set the amplitude to something small, like 0.01 or 0.02 cm. Be sure this amplitude is large enough that the steady state value is between 0.5 and 1.0 cm. You may have to try a few values.
- Compile **Model210\_Openloop.mdl**, if necessary.
- Connect **Model210\_Openloop.mdl** to the ECP system. (The mode should be **External**.)
- Run **Model210\_Openloop.mdl**. If the cart does not seem to move much, increase the amplitude of the step. If the cart moves too much, decrease the amplitude of the step. You may have to recompile after the change.
- You only need to run the system until it comes to steady state, then stop it.

Once the system has run, edit **open\_loop\_driver.m** from Lab 1 and comment out (put % at the beginning of lines that don't start with %) the lines

```
load state_model_1dof
C = [1 0];
[num_Gp,den_Gp] = ss2tf(A,B,C,D);
%
%
% clean up the transfer function by setting small values to 0
%
tol = 1e-7; % anything smaller than this is set to zero
%
% you are not responsible for the following voo doo
%
num_Gp = (abs(num_Gp) > tol*ones(1,length(num_Gp))).*num_Gp;
den_Gp = (abs(den_Gp) > tol*ones(1,length(den_Gp))).*den_Gp;
```

Next you need to input an estimated transfer function for the second order system, so add something like the following to **open\_loop\_driver.m** ...

```
omega_n = 20;
zeta = 0.1;
K = 2;
```

```
num_Gp = K;  
den_Gp = [1/omega_n^2 2*zeta/omega_n 1];
```

Next simulate your open loop system. Once it is done, run the program **compare1.m** (available on the class website). This program compares the response of the real system with the response of the model system. The estimated (model) parameters all begin with an **m** (**m\_time**, **m\_y**) while the values for the real system do not (**time**, **x1**). We will use **compare1.m** a lot this term.

Finally iterate (guess and check) on the values of  $K$ ,  $\omega_n$  and  $\zeta$  to get the best fit between the model and the real system. Remember that fitting during the early part of the step is most important. Record the values of static gain  $K$ , natural frequency  $\omega_n$ , and damping ratio  $\zeta$  that results in the best fit. Include the plot of the fit of your best model to the real system in your memo.

### **Part B: One degree of Freedom Rotational Systems**

*Be sure to load the correct controller personality file for the ECP system (and reset the controller)!!!*

You will need to go through the following steps for *two different* configurations (different masses and/or springs) unless you are going to use one torsional and one rectilinear system.

Note that for the rotational systems, the mass positions should be place symmetrically about the rotational axis. There are lines on the disks to help you accomplish this.

*Step 1: Set Up the System.* The middle disk should be securely fixed in place. **Be sure you write down all of the information you need to duplicate this configuration.**

*Step 2: Log Decrement Estimate of  $\zeta$  and  $\omega_n$*

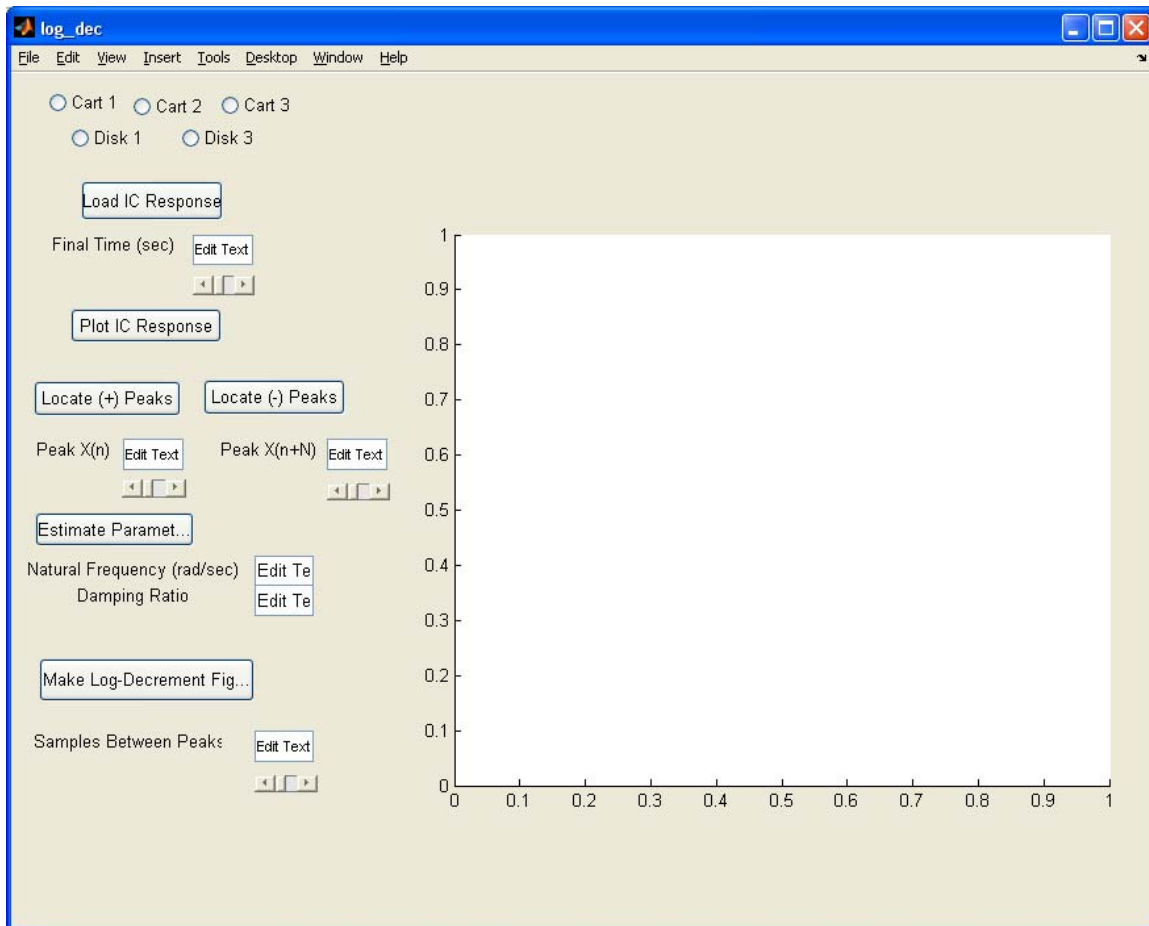
As you recall, the log decrement method is a way of estimating the natural frequency  $\omega_n$  and damping ratio  $\zeta$  of a second order system. However, this method does not determine the static gain.

You will go through the following steps:

- Reset the system using **ECPDSPresetmdl.mdl**.
- Modify **Model205\_Openloop.mdl** so the input has **zero** amplitude.
- Compile **Model205\_Openloop.mdl** if necessary.
- Connect **Model205\_Openloop.mdl** to the ECP system. (The mode should be **External**.)
- Displace the first disk, and hold it. Don't displace it more than about 15 degrees.

- Start (**play**) **Model205\_Openloop.mdl** and let the mass go.
- Run the m-file **log\_dec.m**. This should be in the same directory as **Model205\_Openloop.mdl** and **log\_dec.fig**. This routine assumes the position of the first disk is labeled *theta1* and the time is labeled *time*. (These are the defaults in **Model205\_Openloop.mdl**.)

The program **log\_dec** comes up with the following GUI:



You need to

- Select **Disk 1**
- Select **Load IC (initial condition) Response** (the variables *time* and *y* 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  $\zeta$  and  $\omega_n$
- Select **Make Log-Decrement Figure** to get a plot and summary of the results. You need to include this figure in your memo.

### Step 3: Estimating the Static Gain $K$

You will go through the following steps:

- Reset the system using **ECPDSPresetmdl.mdl**.
- Modify **Model205\_Openloop.mdl** so the input is a step. You may have to set the mode to **Normal**.
- Set the amplitude to something small, like 2 or 5 degrees. *Be sure to convert to radians!*
- Compile **Model205\_Openloop.mdl**, if necessary.
- Connect **Model205\_Openloop.mdl** to the ECP system. (The mode should be **External**.)
- Run **Model205\_Openloop.mdl**. If the disk does not seem to move much, increase the amplitude of the step. If the disk moves too much, decrease the amplitude of the step. You may have to recompile after the change.
- You only need to run the system until it comes to steady state, then stop it.

Estimate the static gain as

$$K = \frac{y_{ss}}{A}$$

where  $y_{ss}$  is the steady state value of the disk position, and  $A$  is the input amplitude. You should determine the value of  $y_{ss}$  in Matlab, **don't use the X-Y Graph**. The variables  $y$  and  $time$  should be in your workspace. You should use **three different input amplitudes** and produce three different estimates for the static gain. Try and make your systems move so the steady state values are between 5 and 10 degrees. Average your three estimates to produce a final estimate of the static gain. Your static gain should generally be between 2 and 10. If yours is not, it is likely you did not use the same units for  $A$  and  $y_{ss}$ .

### Step 4: Second Order Step Response

We will now try to identify the required system parameters by determining the step response of the real (ECP) system, simulating an ideal second order system, and comparing the two responses. If you cannot make the ideal system match the real system, make the ideal system match the real system as well as possible *at the beginning of the response*.



This part of the lab should be done *independently* of **Parts 2** and **3**. That is, ***do not*** just use the values for the natural frequency  $\omega_n$ , damping ratio  $\zeta$ , and static gain  $K$  you obtained previously. Part of the reason for doing this is so you can develop a better feel for how changing these parameters will change the step response. You might want to have one partner do **Parts 2** and **3** and the other do this part for the same system, and then switch when you do the other system.

You will go through the following steps:

- Reset the system using **ECPDSPresetmdl.mdl**.
- Modify **Model205\_Openloop.mdl** so the input is a step. You may have to set the mode to **Normal**.
- Set the amplitude to something small, like 1 or 5 degrees (*be sure to convert to radians!*). Be sure this amplitude is large enough that the steady state value is between 5 and 10 degrees. You may have to try a few values.
- Compile **Model205\_Openloop.mdl**, if necessary.
- Connect **Model205\_Openloop.mdl** to the ECP system. (The mode should be **External**.)
- Run **Model205\_Openloop.mdl**. If the disk does not seem to move much, increase the amplitude of the step. If the disk moves too much, decrease the amplitude of the step. You may have to recompile after the change.
- You only need to run the system until it comes to steady state, then stop it.

Once the system has run, edit **open\_loop\_driver.m** from Lab 1 and comment out (put % at the beginning of lines that don't start with %) the lines

```
load state_model_1dof
C = [1 0];
[num_Gp,den_Gp] = ss2tf(A,B,C,D);
%
%
% clean up the transfer function by setting small values to 0
%
tol = 1e-7; % anything smaller than this is set to zero
%
% you are not responsible for the following voo doo
%
num_Gp = (abs(num_Gp) > tol*ones(1,length(num_Gp))).*num_Gp;
den_Gp = (abs(den_Gp) > tol*ones(1,length(den_Gp))).*den_Gp;
```

Next you need to input an estimated transfer function for the second order system, so add something like the following to **open\_loop\_driver.m** ...

```
omega_n = 20;
zeta = 0.1;
```

```
K = 2;  
num_Gp = K;  
den_Gp = [1/omega_n^2 2*zeta/omega_n 1];
```

Next simulate your open loop system. Once it is done, run the program **compare1.m** (available on the class website). This program compares the response of the real system with the response of the model system. The estimated (model) parameters all begin with an **m** (**m\_time**, **m\_y**) while the values for the real system do not (**time**, **theta1**). We will use **compare1.m** a lot this term. *You should plot all of your results in degrees, so you may have to change the compare1.m program to convert from radians to degrees. You will have to modify the program to use theta1 instead of x1.*

Finally iterate (guess and check) on the values of  $K$ ,  $\omega_n$  and  $\zeta$  to get the best fit between the model and the real system. Remember that fitting during the early part of the step is most important. Record the values of static gain  $K$ , natural frequency  $\omega_n$ , and damping ratio  $\zeta$  that results in the best fit. Include the plot of the fit of your best model to the real system in your memo.

*Your memo should include descriptions of each system (so you can set them up again), a **table** comparing the estimated values of  $K$ ,  $\omega_n$  and  $\zeta$  (two different methods for each parameter). You should include as attachments 4 graphs (log-decrement and step response graphs for two systems), each with a Figure number and caption. The graphs should be included in your Word document, do not print out graphs and attach them. Be sure the step responses for the torsional systems are plotted in degrees!*