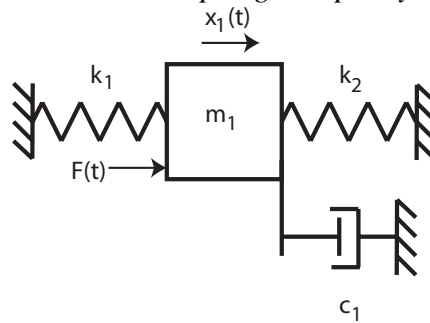


# Lab 3: Time and Frequency Domain Modeling of One Degree of Freedom Systems

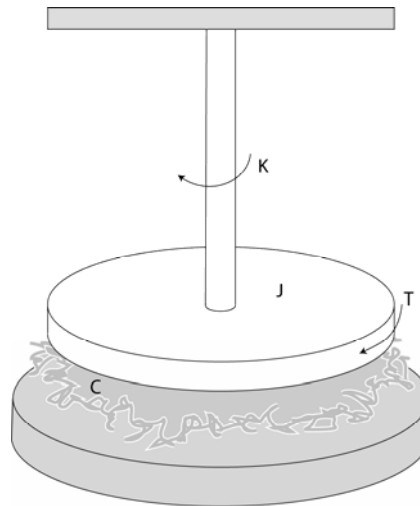
## Overview

In this lab you will be modeling two one degree of freedom systems using time-domain analysis and frequency domain analysis. The steps we will go through in this lab are very commonly used in system identification (determining the transfer function) of unknown systems. We will utilize these models in later labs so do a good job in this lab, your results in later labs will be affected by how well you perform in this lab. If your lab station has only a rectilinear (model 210) or only a torsional (model 205) system you will make two different models using those systems. If your lab station has both a torsional and a rectilinear system then you need to make one model of each type of system

A one degree of freedom rectilinear mass-spring-damper system



or a rotational mass-spring-damper system



can be modeled as

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

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 Lab3 and copy all files from the folder **basic\_files** into this folder.

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

You will need to go through the following steps for *each* configuration (different masses and/or springs). **You need to fill out the data sheet indicating each configuration on the last page of the lab and turn it in! It needs to be signed before you start the second 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.**

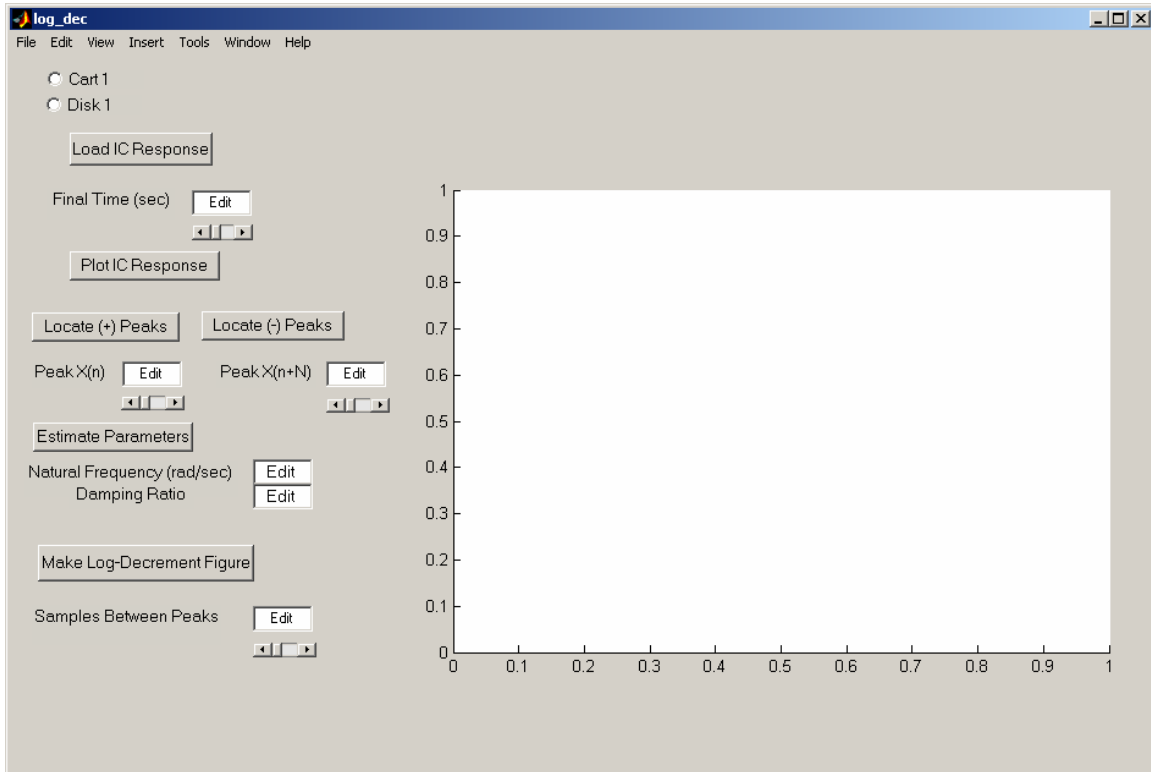
Step 2: Log Decrement Estimate of  $\zeta$  and  $\omega_n$

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  $y$  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: Fitting the Estimated Frequency Response to the Measured Frequency Response

We will be constructing the magnitude portion of the Bode plot and fitting this measured frequency response to the frequency response of the expected transfer function to determine  $K$ ,  $\zeta$ , and  $\omega_n$ . For each frequency  $\omega = 2\pi f$  we have as input  $u(t) = A \cos(\omega t)$  where, for our systems,  $A$  is measured in centimeters. After a transition period, the steady state output will be  $x_1(t) = B \cos(\omega t + \theta)$ , where  $B$  is also measured in cm. Since we will be looking only at the magnitude portion of the Bode plot, we will ignore the phase angle  $\theta$ .

You will go through the following steps

For frequencies  $f = 0.5, 1, 1.5 \dots 7.5$  Hz

- Modify **Model210\_Openloop.mdl** so the input is a sinusoid. You may have to set the mode to **Normal**.

- Set the frequency and amplitude of the sinusoid. Try a small amplitude to start, like 0.01 cm. Generally this amplitude should be as large as you can make it without the system hitting a limit. This amplitude will probably vary with each frequency. *The frequency needs to be entered in radians/sec!*
- Compile **Model210\_Openloop.mdl**, if necessary. (Assume it is not necessary. The system will let you know if it is 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 input sinusoid.. If the cart moves too much, decrease the amplitude of the input sinusoid.

Record the input frequency ( $f$ ), the amplitude of the input ( $A$ ), and the amplitude of the output ( $B$ ) when the system is in steady state. Note that the output may not be a sine wave symmetric about zero. Hence you need the average of the positive and negative values. The Matlab file **get\_B.m** will help with this. *You need to include a description of how this function operates in your memo. Be sure the system is in sinusoidal steady state (look at the plot made by **get\_B.m**) before you collect the data!*

Enter the values of  $f$ ,  $A$ , and  $B$  into the program **process\_data\_1dof.m** (you need to edit the file)

At the Matlab prompt, type **data = process\_data\_1dof;**

Run the program **model\_1dof.m**. There are four input arguments to this program:

- **data**, the measured data as determined by **process\_data\_1dof.m**
- $K$  the estimated static gain
- $\omega_n$  the estimated natural frequency (from the log decrement analysis)
- $\zeta$  the estimated damping ratio (from the log decrement analysis)

The program **model\_1dof.m** will produce the following:

- A graph indicating the fit of the identified transfer function to the measured data. (You need to include this graph in your memo.)
- The optimal estimates of  $K$ ,  $\zeta$ , and  $\omega_n$  (written at the top of the graph)
- A file **state\_model\_1dof.mat** in your directory. This file contains the A, B, C, and D matrices for the state variable model of the system. If you subsequently type **load state\_model\_1dof** you will load these matrices into your workspace.

**Add additional data points so you have 4 points close to the resonant peak of the transfer function.**

*Before you model the second system, you should **save** and **rename** the files **process\_data\_1dof.m**, and **state\_model\_1dof.mat** in a way that you will be able to*

identify them later. Write these on the data sheet at the end of this lab. These are the files that contain a state variable model of your system and will be used in nearly all future labs!

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

You will need to go through the following steps for *each* configuration (different mass positions).

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

***You need to fill out the data sheet indicating each configuration on the last page of the lab and turn it in! It needs to be signed before you start the second system!***

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

Step 1: Set Up the System. Be sure the middle disk is securely fixed in place. You need to have at least two masses on the first disk. ***Be sure you write down all of the information you need to duplicate this configuration.***

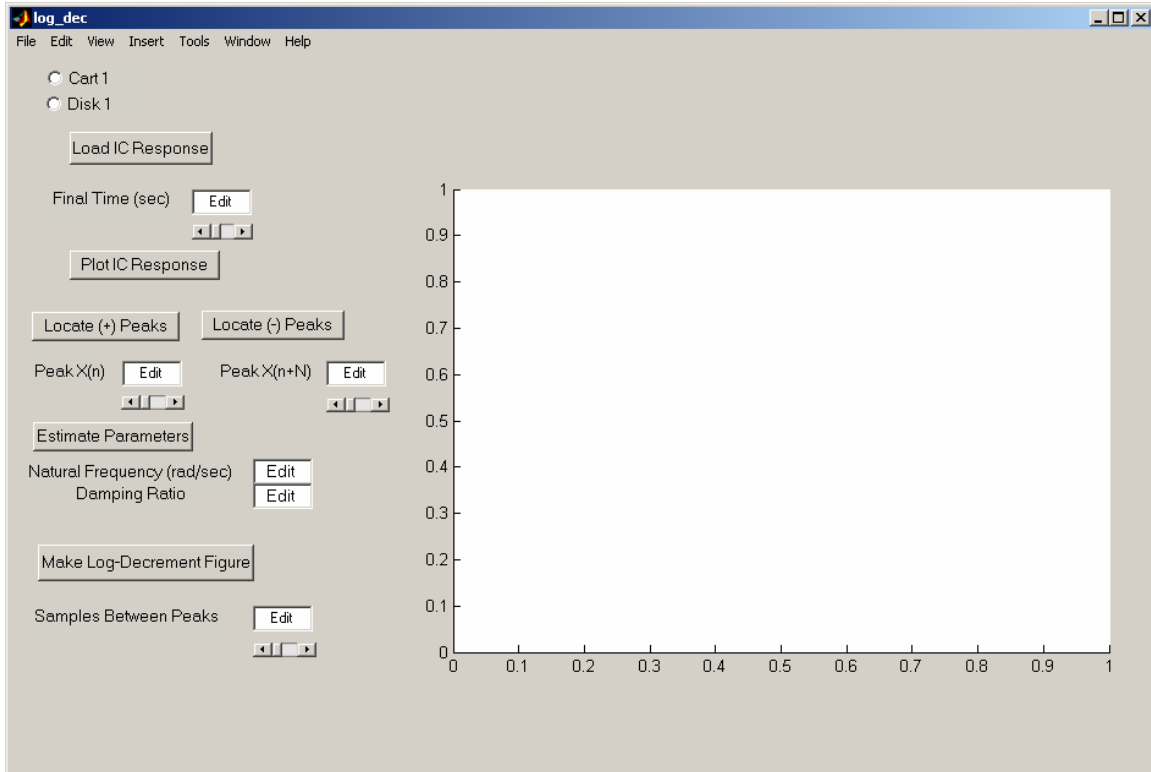
Step 2: Log Decrement Estimate of  $\zeta$  and  $\omega_n$

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 *y* 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 (more than 25 or 30 degrees) 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 15 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: Fitting the Estimated Frequency Response to the Measured Frequency Response

We will be constructing the magnitude portion of the Bode plot and fitting this measured frequency response to the frequency response of the expected transfer function to determine  $K$ ,  $\zeta$ , and  $\omega_n$ . For each frequency  $\omega = 2\pi f$  we have as input  $u(t) = A \cos(\omega t)$  where, for our systems,  $A$  is measured in radians or degrees. After a transition period, the steady state output will be  $\theta_1(t) = B \cos(\omega t + \psi)$ , where  $B$  is measured in radians (or degrees). Since we will be looking only at the magnitude portion of the Bode plot, we will ignore the phase angle  $\psi$ .

**In this part, limit the motion of the torsional system to +/- 30 degrees. If it appears to be turning more than this stop the system and put in a smaller amplitude!**



You will go through the following steps

For frequencies  $f = 0.5, 1, 1.5 \dots 7.5$  Hz

- Modify **Model205\_Openloop.mdl** so the input is a sinusoid. You may have to set the mode to **Normal**.
- Set the frequency and amplitude of the sinusoid. Try a small amplitude to start, like 1 degree (the input to the ECP needs to be in *radians*). Generally this amplitude should be as large as you can make it without the system hitting a limit. *The frequency needs to be entered in radians/sec!*
- Compile **Model205\_Openloop.mdl**, if necessary. (Assume it is not necessary. The system will let you know if it is 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 input sinusoid.. If the disk moves too much, decrease the amplitude of the input sinusoid.

Record the input frequency ( $f$ ), the amplitude of the input ( $A$ ), and the amplitude of the output ( $B$ ) when the system is in steady state. Not that the output may not be a sine wave symmetric about zero. Hence you need the average of the positive and negative values. The Matlab file **get\_B.m** will help with this, though you may need to modify the file. *You need to include a description of how this function operates in your memo. Be sure the system is in sinusoidal steady state (look at the plot made by **get\_B.m**) before you collect the data! Be sure to use the same units for the input amplitude and the output amplitude. **get\_B.m** will return radians unless you change it!*

Enter the values of  $f$ ,  $A$ , and  $B$  into the program **process\_data\_1dof.m** (you need to edit the file)

At the Matlab prompt, type **data = process\_data\_1dof;**

Run the program **model\_1dof.m**. There are four input arguments to this program:

- **data**, the measured data as determined by **process\_data\_1dof.m**
- $K$  the estimated static gain
- $\omega_n$  the estimated natural frequency (from the log decrement analysis)
- $\zeta$  the estimated damping ratio (from the log decrement analysis)

The program **model\_1dof.m** will produce the following:

- A graph indicating the fit of the identified transfer function to the measured data. (You need to include this graph in your memo.)

- The optimal estimates of  $K$ ,  $\zeta$ , and  $\omega_n$  (written at the top of the graph)
- A file **state\_model\_1dof.mat** in your directory. This file contains the A, B, C, and D matrices for the state variable model of the system. If you subsequently type **load state\_model\_1dof** you will load these matrices into your workspace.

**Add additional data points so you have 4 points close to the resonant peak of the transfer function.**

*Before you model the second system, you should **save** and **rename** the files **process\_data\_1dof.m**, and **state\_model\_1dof.mat** in a way that you will be able to identify them later. Write these on the data sheet at the end of this lab. These are the files that contain a state variable model of your system and will be used in nearly all future labs!*

*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  $\zeta$  using the two different methods (time domain and frequency domain), and a brief comparison of the values. The damping ratios are often quite different, so that's OK. The other values should be close. You should include as attachments 4 graphs ( log-decrement and frequency response graphs for the rectilinear and/or rotational systems), each with a Figure number and caption. You should also include the data used for estimating the static gain.*

Names \_\_\_\_\_

1 dof rectilinear systems (model 210)

Left Spring:                   stiff/light/none   Spring Number =  
Number of Large Masses: 1 2 3 4  
Number of Small Masses: 1 2 3 4  
Right Spring:                   stiff/light/none   Spring Number =

**state\_model\_1dof.mat** is now named :

Instructor Verification and Time \_\_\_\_\_

1 dof rectilinear systems (model 210)

Left Spring:                   stiff/light/none   Spring Number =  
Number of Large Masses: 1 2 3 4  
Number of Small Masses: 1 2 3 4  
Right Spring:                   stiff/light /none   Spring Number =

**state\_model\_1dof.mat** is now named :

Instructor Verification and Time \_\_\_\_\_

Names \_\_\_\_\_

1 dof torsional systems (model 205)

Number of Masses: 2 4

Each mass

a) is as far in (towards the center) as it can go

b) is as far out (away from the center) as it can go

c) is aligned so the outer edge is on the \_\_\_\_\_ ring from the outside

**state\_model\_1dof.mat** is now named :

Instructor Verification and Time \_\_\_\_\_

1 dof torsional systems (model 205)

Number of Masses: 2 4

Each mass

a) is as far in (towards the center) as it can go

b) is as far out (away from the center) as it can go

c) is aligned so the outer edge is on the \_\_\_\_\_ ring from the outside

**state\_model\_1dof.mat** is now named :

Instructor Verification and Time \_\_\_\_\_