# Lab 2: 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. You will be assigned to model either rectilinear or torsional systems.

Background

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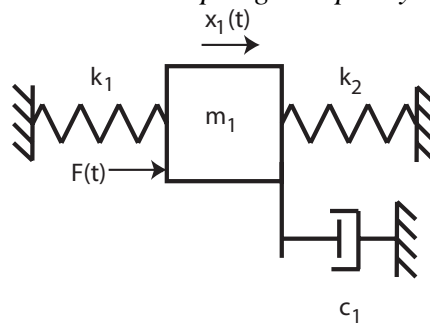
or a rotational mass-spring-damper system
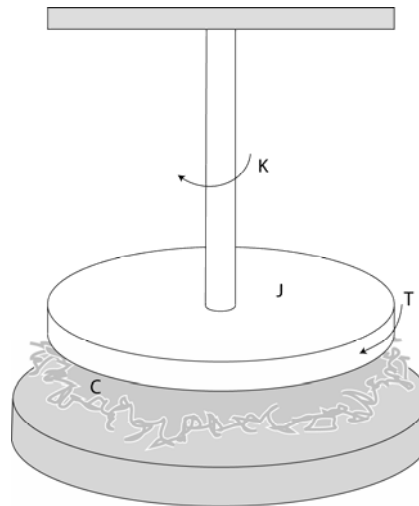
can be modeled as

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

Here K is the static gain, 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 for Lab 2 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 connecting the first cart to the second cart (you may also have an additional spring between the motor and the first cart) and at least two masses on the cart. Do not use the damper. *If you use two springs, the stiffer of the two springs should be between the first and second cart!* **Be sure you write down all of the information you need to duplicate this configuration.**

Step 2: Log Decrement (Time Domain) Estimate of ζ and ω_n

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.

You will go through the following steps:

- Reset the system using **ECPDSPresetmdl.mdl.**
- Modify **Model210_Openloop.mdl** so the input has *<u>zero</u>* 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:
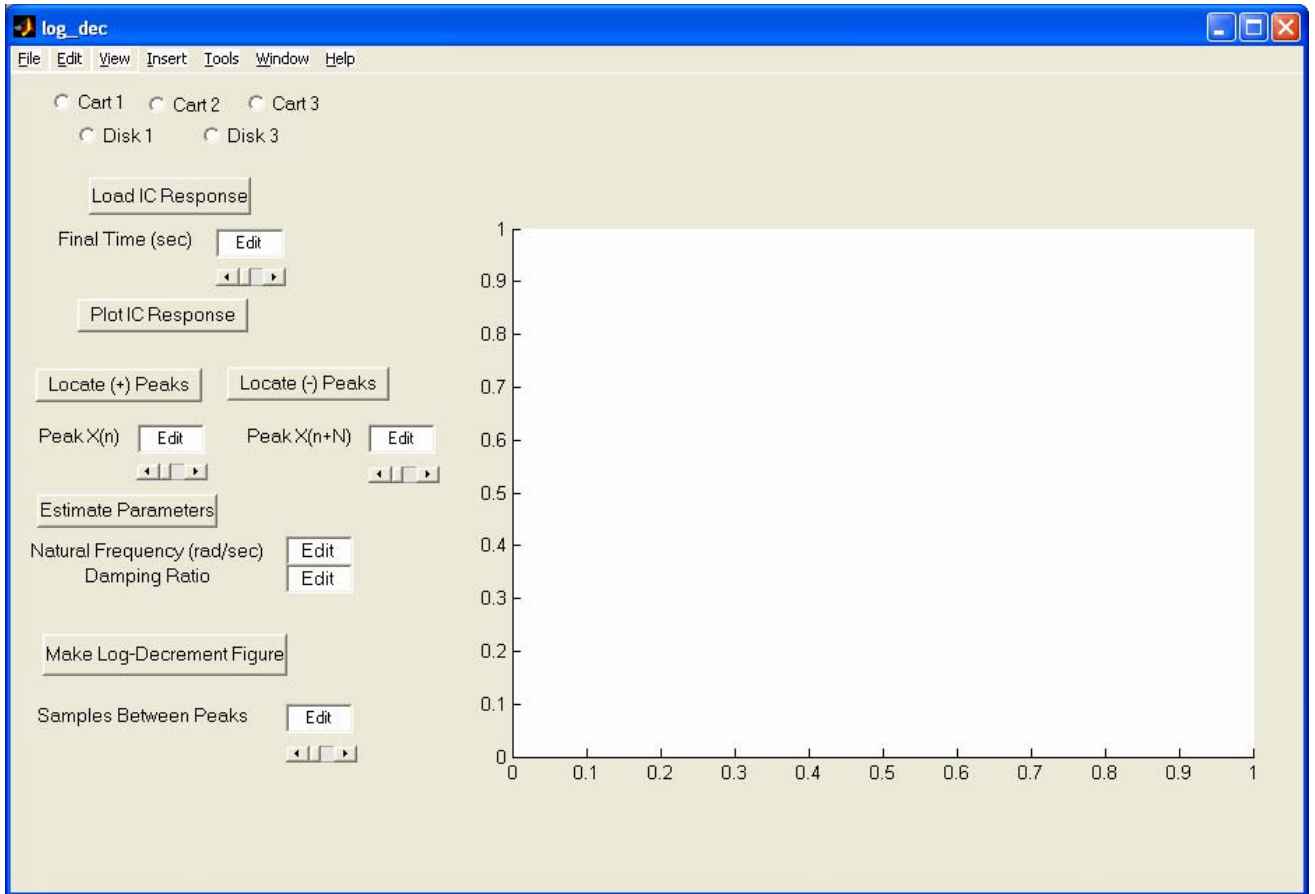
You need to

- Select the **cart** to be analyzed (cart one in this case)
- Select **Load IC (initial condition) Response** (the variables *time* and *x1* 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 put this figure in your memo.

You will go through the following steps:

- Reset the system using **ECPDSPresetmdl.mdl.**
- Modify **Model210_Openloop.mdl** so the input is a step. To make any changes to **Model210_Openloop**.mdl, the mode must be **Normal**.
- Set the amplitude initially to something small, like 0.01 or 0.02 cm.
- Compile **Model210_Openloop.mdl**
- 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, the system will tell you if you have to.
- You only need to run the system until it comes to steady state, then stop it.
- Estimate the static gain as

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

where x_{ss} is the steady state value of the cart position, and A is the input amplitude. You should do this in Matlab, don't use the X-Y Graph. The variables *x1* and *time* should be in your workspace.

You need to increase the value of the input amplitude until the cart is moving a maximum of about 2 cm or so. Use the static gain values associated with this input amplitude as the static gain.

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, ζ, and ω_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 θ.

You will go through the following steps

For frequencies $f = 0.5, 1, 1.5...7.5$ 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.
- 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. 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_A.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_A) 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
- ω_n the estimated natural frequency (from the log decrement analysis)
- ζ 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 , ζ , and ω_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.*

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. Only the first disk should move, all other disks should be fixed. ***Be sure you write down all of the information you need to duplicate this configuration.***

Step 2: Log Decrement Estimate of ζ and ω_n

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.

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:
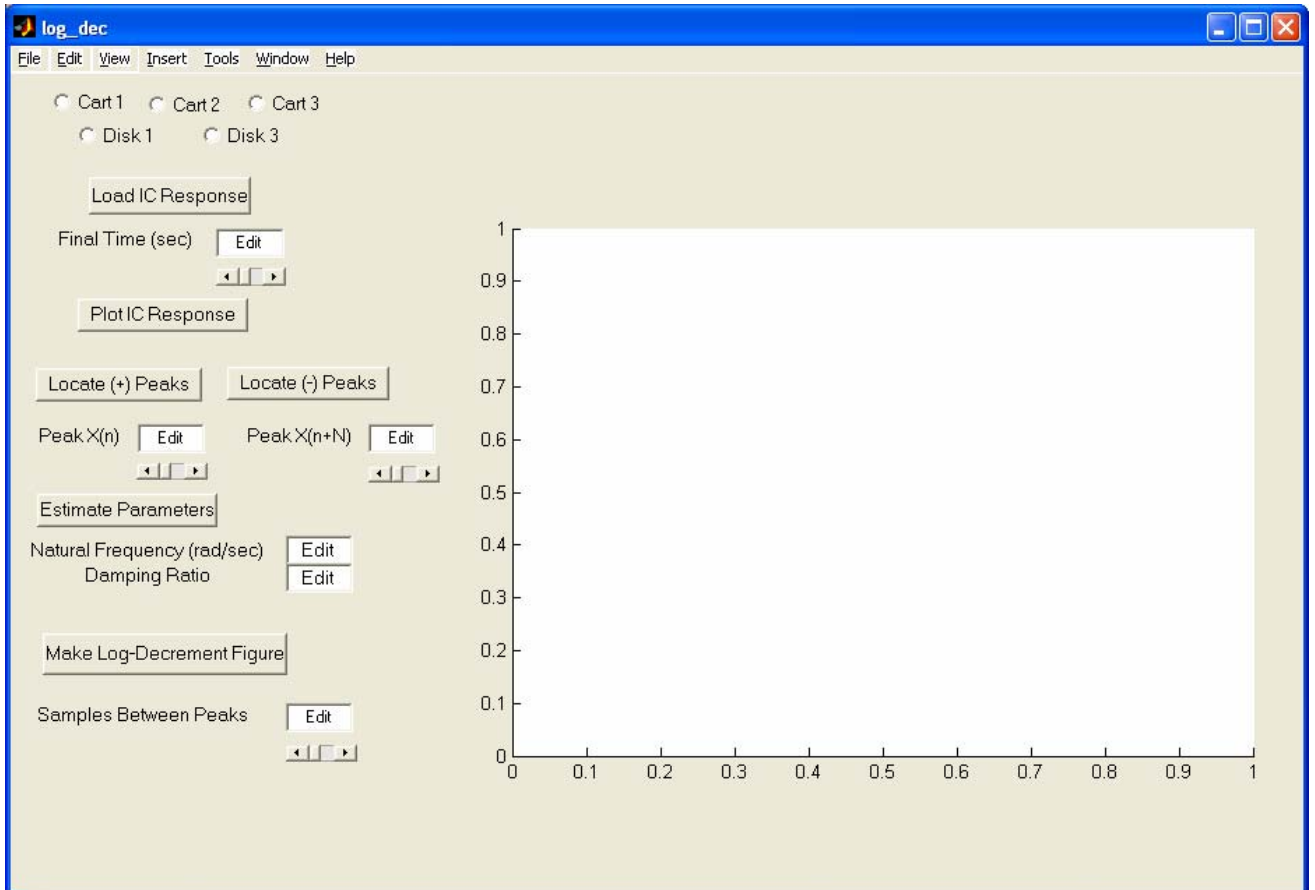
You need to
- Select the **disk** to be analyzed (disk one in this case)
- Select **Load IC (initial condition) Response** (the variables *time* and *theta1* will be loaded from the workspace). At this point some initial estimates will be made.
- Set/modify the **Final Time**
- Select **Plot IC Response** 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 put this figure in your memo.

You will go through the following steps:

- Reset the system using **ECPDSPresetmdl.mdl.**
- Modify **Model205_Openloop.mdl** so the input is a step. To make any changes to **Model205_Openloop**.mdl, the mode must be **Normal**.
- Set the amplitude initially to something small, like 1 or 5 degrees (note that the ECP system expects the input to be in *radians*!).
- Compile **Model205_Openloop.mdl**
- 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, the system will tell you if you have to.
- You only need to run the system until it comes to steady state, then stop it.
- Estimate the static gain as

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

where θ_{ss} is the steady state value of the disk position, and A is the input amplitude. Be sure you are using the *same units* for both θ_{ss} and A! The variables *theta1* and *time* should be in your workspace.

You need to increase the value of the input amplitude until the disk is moving about 20 degrees or so. Use the static gain values associated with this input amplitude as the static gain.

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, ζ, and ω_n. For each frequency $\omega = 2\pi f$ we have as input $u(t) = A\cos(\omega t)$ where, for out 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 also measured in cm. Since we will be looking only at the magnitude portion of the Bode plot, we will ignore the phase angle ψ.

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 $f = 0.5, 1, 1.5...7.5$ 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.
- 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_A.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_A) 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
- ω_n the estimated natural frequency (from the log decrement analysis)
- ζ 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 , ζ , and ω_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.*

Your memo should include descriptions of each system (so you can set them up again), a table comparing the estimated values of K, ω_n and ζ 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 rotational systems), each with a Figure number and caption. You should also include the data used for estimating the static gain.

Names _____

<u>*1 dof rectilinear systems (model 210)*</u>

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

state_model_1dof.mat *is now named :*

Instructor Verification and Time _____

<u>*1 dof rectilinear systems (model 210)*</u>

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

state_model_1dof.mat *is now named :*

Instructor Verification and Time _____

Names _____

<u>*1 dof torsional systems (model 205)*</u>

Number of Masses: 0   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 _____

<u>*1 dof torsional systems (model 205)*</u>

Number of Masses: 0   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 _____