

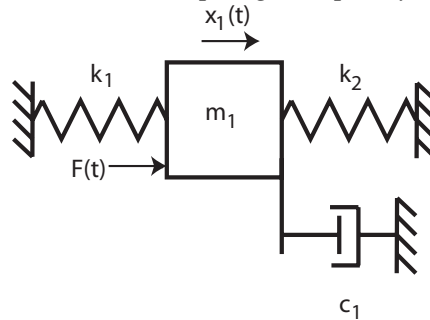
## Lab 3: Time and Frequency Domain Modeling of One Degree of Freedom Systems

### Overview

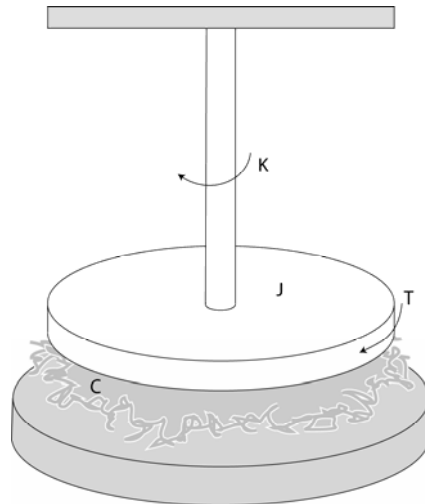
In this lab you will be modeling three one degree of freedom systems using time-domain analysis and frequency domain analysis. The steps we will go through in this lab are very commonly used in system identification (determining the transfer function) of unknown systems. We will utilize these models 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.

### 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,  $\omega_n$  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 3 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).

*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 (Time Domain) 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**.)

### Step 3: Time Domain Estimate of the Static Gain

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.
- 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 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  $K$ ,  $\zeta$ , 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.
- 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 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 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.

**You need to be sure you have 4 points close to the resonant peak of the transfer function. At this point you probably should go back and add a few points near the resonant peak.**

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.

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

You and your partner should change functions/positions now, i.e. whoever was running Simulink and the ECP system for the first part should not do this for the second part.

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.

*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  $\zeta$  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**.)

### Step 3: Time Domain Estimate of the Static Gain

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.
- 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  $\theta_{ss}$  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  $\theta_{ss}$  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.

### 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 also measured in cm. Since we will be looking only at the magnitude portion of the Bode plot, we will ignore the phase angle  $\psi$ .

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.
- 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 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 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.
- 

**You need to be sure you have 4 points close to the resonant peak of the transfer function. At this point you probably should go back and add a few points near the resonant peak.**

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

*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 6 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.*