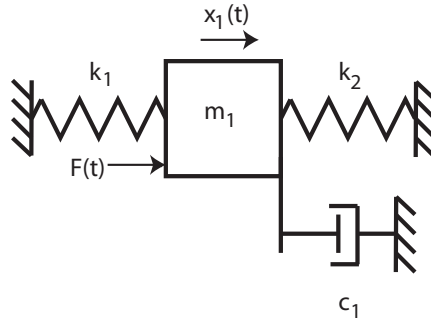


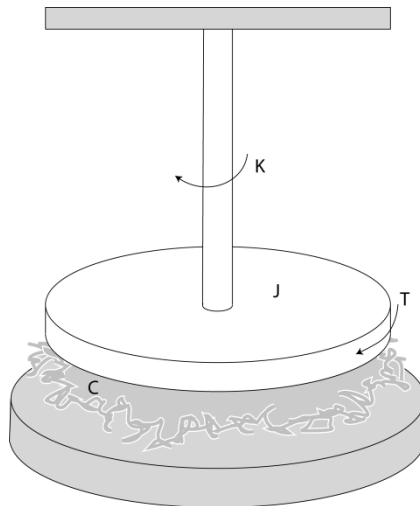
ECE-520 Lab 1: Modeling of One Degree of Freedom Systems

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.

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 Lab1 and copy and unzip all of the files from **Lab1 files.rar** from the class website.

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/or rotational systems), each with a Figure number and caption. You should also include the data used for estimating the static gain. However, do not include the data for constructing the Bode plot.

You may need to review the file *shortguide.doc*. You may also need to set the **base address** for your system.

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 cart and the motor!* **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 **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**. (be sure the file **log_dec.fig** is in the same folder). *Include the final log-dec 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 $K = \frac{x_{1,ss}}{A}$ where $x_{1,ss}$ is the steady state value of the cart position, and A is the input amplitude. You should determine the value $x_{1,ss}$ in Matlab, **don't use the X-Y Graph**. The variables $x1$ 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 $x_{1,ss}$.

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 , 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 $f = 0.5, 1, 1.5 \dots 7.5$ 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.

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 names 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 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**. (be sure the file **log_dec.fig** is in the same folder). *Include the final log-dec 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

$$K = \frac{\theta_{1,ss}}{A}$$

where $\theta_{1,ss}$ is the steady state value of the disk position, and A is the input amplitude. You should determine the value of θ_1 in Matlab, **don't use the X-Y Graph**. The variables *theta1* 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 $\theta_{1,ss}$.

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 , 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 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 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!

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. 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, *though you may need to modify the file.*

Enter the values of f , A , and B into the program **process_data_1dof.m** (you need to edit the file, be sure the units of A and B are the same! I usually do this all in Hz)

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.

*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 names 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!*

Name(s) _____

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 _____