09-723 Proximal probe techniques. Name___________________________

Homework #3.

Due by 6:30 PM, Tuesday, September 21, 2004

Download the DrivenExplorer.zip file which contains all necessary Matlab files by clicking on it and choosing to save it on your computer (e.g. on your desktop). Open the archive on your computer and extract it to some folder. Make sure that a path is set in Matlab to this folder. To set a path, choose the set path option in the file menu of Matlab. Click the Add with subfolders button and locate the folder to be added to the path in the pop-up window and click OK. Then click Save and Close.

As usual, please contact me if you have any problems with the installation or any other problems with using the program.
justinl@andrew.cmu.edu
phone: 8-9175

DrivenHarmonicExplorer (Some Assembly Required)

In its basic state, DrivenHarmonicExplorer allows you to generate solutions of the equations of motion of mass *(Mtip)* attached to a spring with a spring constant *(Kspring)*, subjected to damping *(Bdamp)*, driven at a specific frequency with specified drive amplitude based on the following equation:

$$M_{tip}\ddot{z} + B_{damp}\dot{z} + K_{spring}z = F_o \cos(2\pi F_{oper}t + \varphi)$$

All necessary parameters are entered through the provided Graphical User Interface (GUI), which also controls the execution of the Simulink model, which you will need to construct in Simulink, following the directions provided in class. Make sure that in your model you call the variables as follows:

Mass: `Mtip` (This is to make it easier to build this as the tip mass in later models)
Spring constant: `Kspring`
Damping Coefficient: `Bdamp`
The simulation step (Fixed-step size): `SimStep`
The total simulation time (i.e. Stop Time)= `SimTime`
The initial velocity = `Zprime` (initial condition for the first integrator)
The initial position = `Zstart` (initial condition for the second integrator)
The position trajectory = `Ztraj`
The time = `time`
The driving force amplitude = `Fo`
The drive frequency (operating frequency in Hz) = `Foper`
The phase φ = `Phi`

If these names are not used, the GUI will not work.
All other variables will be defined through the GUI.
Steps in construction of your model:

I. Add a drive to your harmonic oscillator model from homework #2.

This model needs to be saved as DriveHarmOscModel.mdl for the GUI to work later. Make sure the model is saved in the path of Matlab. (DO NOT SAVE OVER YOUR OLD MODEL.) The easiest way to do this is to save the model in the folder that you extracted HarmonicExplorer into. You should have already set a path to this folder.

Before running the model you need to provide the values of other parameters using the provided GUI. To use the GUI provided, the workspace needs to be prepared to receive the parameters generated by the GUI. This is accomplished by running `SimPrep2` from command line.

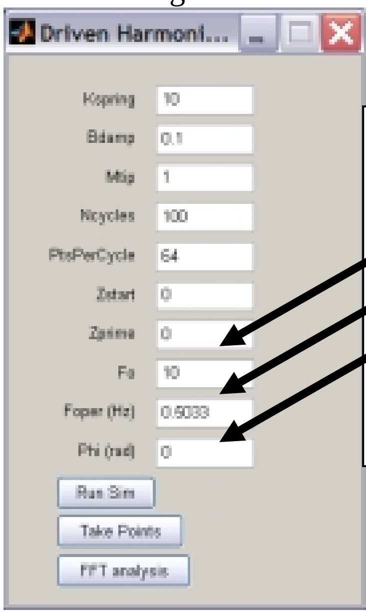
```
>> SimPrep2
```

(If you look inside this m-file, you will see a list of variable being declared as globals. New variables were needed to add the FFT function to the GUI, which required a new m-file to set the workspace)

To open the GUI, from command line type `DHO_GUI`

```
>> DHO_GUI
```

The following interface should appear.

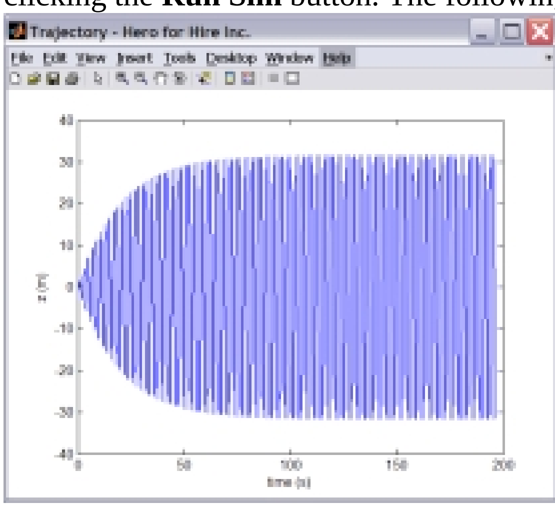


This GUI is similar to the `HarmOscGUI` from Homework #2, but it has textboxes to adjust the three new parameters:

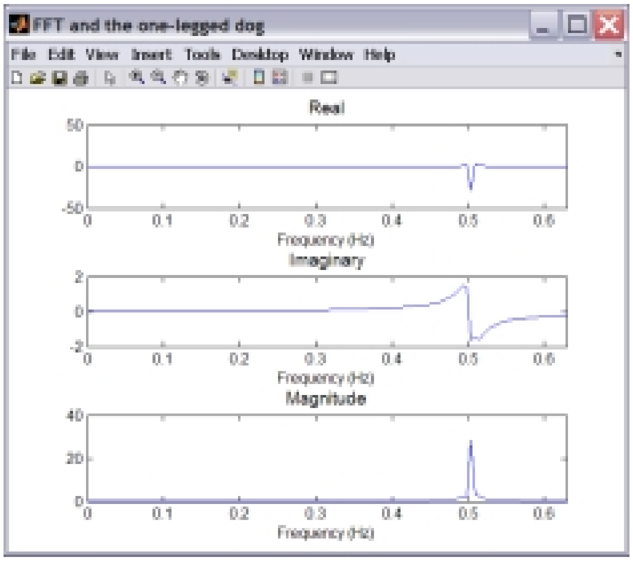
`Fo`
`Foper`
`Phi`

Also, it has an **FFT analysis** button for Fourier transform calculations.

If everything is set up, you should be able to run a simulation using the default setting by clicking the **Run Sim** button. The following figure should appear:



If you see this plot, you should be able to run a FFT analysis using the **FFT analysis** button. The following figure should appear:



Now, you can freely change parameters to see what happens. **Ncycles** determines the number of cycles you will simulate. **PtsPerCycle** will determine the resolution at which each cycle will be simulated. All other scaling is taken care of in the GUI.

Specific values of points can be recovered from any trajectory by using the **Take Points** button. You will be prompted for the number of points you wish to take. Enter the number of points you would like to take and press OK. For example, if you want 5 points, enter 5 and press OK. Next, crosshairs will appear on the trajectory figure. Line the cross hairs up on the point you want and left click the mouse. You will need to do this for each point you want. For example, if you chose to collect 5 points, you will need to click 5 points on the figure before they will be retrieved. These points are placed in the workspace in a matrix called `PointMat`. The first column corresponds to time, the second to z. You may also use the **Data Cursor** icon on the figure to determine points on any plot (this is actually more accurate).

The following exercises will have you use simulations to explore the relationships between different parameters and the trajectory of a driven harmonic oscillator. Eventually, these relationships will be rigorously derived in class.

Part I

Explorations with user-defined parameters.

1.1 Set **Kspring** to 1, **Bdamp** to 0.3, **Mtip** to 1, **Ncycles** to 50, **PtsPerCycle** to 64, **Zstart** to 0, **Zprime** to 0, **Fo** to 1, **Foper** to $\frac{1}{2\pi} \approx 0.1592$, and **Phi** to 0.

a) The resonance frequency f_r at which the system will be on resonance is $f_r = \frac{\omega_r}{2\pi} = \frac{1}{2\pi}$ Hz. Map the resonance peak by running the program at this frequency and frequencies $f_r \pm \Delta f$ below and above f_r (five above and five below, use the value of $\Delta f = 0.003$ Hz). For each case measure the oscillation amplitude *A* from the plot. Make the plot of $A(f)$.

b) Make analogous calculations for **Bdamp**=0.05. This time use the smaller step $\Delta f = 0.0005$ Hz .

c) Compare both resonance peaks and try to reason why their heights and widths are so different.

d) Compare side by side the trajectory plots of *z(t)* on resonance for **Bdamp** = 0.3 and 0.05. In which case the system reaches the steady state faster?

1.2 Take any lightly damped oscillator and determine what is the driving force amplitude *Fo* to achieve the same target oscillation amplitude at different frequencies f ranging from far below to far above resonance. Make a plot of $Fo(f)$. Compare its shape with the shape of the curve obtained in 1.1 (a or b) and comment on your observations.

Part 2

Exploring FFT

In this series of problems you will be getting acquainted with Fourier transform $\Im(\omega)$, which decomposes the function $f(t)$ defined in time domain into its harmonic components in the frequency domain:

$$\Im(\omega) = \int_{-\infty}^{+\infty} f(t)\exp(-i\omega t)\,dt$$

This capability of Fourier transform makes it extremely useful in analyzing periodic time series (or periodic structures in space). We will be using it for both of these purposes.

For discrete signals $f_k \equiv f(t_k)$, defined at equally-spaced times $t_k \equiv k\Delta$, with k = 0, ..., N-1 where Δ is the time step we deal with the discrete Fourier transform (DFT). The frequency range of DFT is defined as

$$\nu_n = \frac{n}{N\Delta} \text{ with } n = -\frac{N}{2},\ldots 0,\ldots \frac{N}{2}.$$

and the values of Fourier transform at ν_n are given by

$$F_n = \sum_{k=0}^{N-1} f_k e^{-2\pi i n k/N}.$$

Inverse transform is defined as:

$$f_k = \frac{1}{N}\sum_{n=0}^{N-1} F_n e^{2\pi i n k/N}.$$

Solve the following problems using the supplied m-file `ExploreFFT.m` that was shown in class. Illustrate your answers with graphical output. **Note:** There is no GUI for this m-file. You are responsible for modifying it appropriately (most of the time, this will involve manipulating the form of your waveform, which is given in line `10).` In some instances you will also have to change the values of some parameters, such as `a`, calculation step `TimeStep`, etc. To do so, find the respective lines in the m-file and set the values by changing them appropriately. In order to modify and execute the m-file, make sure that it is in the path and open it in Matlab. After modifying the file you need to save it before executing it (saving/executing can be done in one step by using `F5` key, which is a shortcut to `Save and Run`).

2.1. Calculate the FFT of `sin(a*t)` for different values of `a` (and in each case note the position of peak maximum. Inspect the magnitude plot. At what frequencies does it show the maxima? By recording the position of a maximum occurring at positive values of frequency (the other one lies symmetrically) as a function of `a,` determine the relationship between the maximum position and `a`. What is the conversion factor? What does its value indicate?

2.2. Calculate the FFT of the sum of two or more sine or cosine waves with different frequencies. Where are the maxima now?

2.3. Calculate the FFT of a few weighted sums of sine and cosine waves (e.g., $2\sin(5t)+3\cos(2t)$). In addition to positions of the maxima, now note also their heights. What is the relationship between the height of maxima and weighting factors?

2.4. Now add some constant factor to your signal (e.g. $\sin(t)+10$). What happened? Justify why the new feature of the spectrum appeared at the frequency where it did. Explore how the height of this feature depends on the value of the constant factor.

2.5. Explore what is the impact of multiplying the sine wave by the decaying exponent (e.g., $\sin(t)\exp(-t/2)$). Try to determine what is the impact of the exponent factor on the shape of the FFT peak. Do you see any analogies with resonance peaks of damped harmonic oscillators? Can you justify it?

2.6. So far we have been focusing on the magnitude of FFT. Now we will explore the role of phase of the signal on the real and imaginary part of FFT. Calculate FFTs of signals shifted by various amounts (most conveniently by some integer products of $\pi/2$). Summarize your observations and describe how FFT can be used to determine the phase of the signal.

2.7. Now let us examine the impact of the length of time series (or more accurately the role of the number of cycles it contains) on the quality of the FFT of the signal. For a chosen signal vary the time scale, to include a range of cycles (from a few to a few hundred). Start from some small number of cycles (e.g. 10), calculate the FFT and manually set the frequency range to center one of the peaks and have its full width at the bottom pretty much fill the axis range. Now carry out your calculations for longer and longer time series. Summarize your observations.

2.8. In this last problem we will try to determine what is the role of time step with respect to the length of the cycle. Calculate the FFT of some chosen signal under conditions when you have at least 100 points per cycle. As before, center the peak and observe what happens when you decrease the number of points per cycle. What has the larger impact on the quality of the FFT peak: the sampling density (number of points per cycle) or the total number of cycles in the series? Can you justify your answer using some common sense?

Part 3

FFT of Simulated Trajectories

A series of problems exploring how Fourier analysis can be used to study simulated trajectories.

3.1 Set **Bdamp** to 0.0, **Ncycles** to 50, **PtsPerCycle** to 64, **Zstart** to 1, **Zprime** to 0, **Drive** to 0, **Foper** to 0, and **Phi** to 0.

(a) For Mtip=1, see how the natural frequency shifts when Kspring is 0.1, 1, and 10 using fourier analysis. Comment.
(b) For Kspring =1, see how the natural frequency shifts when Mtip is 0.1, 1, and 10 using fourier analysis. Comment.

3.2 Set **Kspring** to 1, **Mtip** to 1, **Ncycles** to 1000, **PtsPerCycle** to 256, **Zstart** to 1, **Zprime** to 0, **Drive** to 0, **Foper** to 0, and **Phi** to 0.

Using Fourier analysis, see the effect of **Bdamp** (set it to 0.1, 0.05, and 0.01) on the width of the harmonic peak in the FFT. Comment.