09-723 Proximal probe techniques. Name_____

Homework #2 .

Due by 6:30 PM, Thursday, September 16

Download the HarmonicExplorer.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 then 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

HarmonicExplorer (Some Assembly Required)

In its basic state, HarmonicExplorer 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)* based on the following equation:

$$M_{tip}\ddot{z} + B_{damp}\dot{z} + K_{spring}z = 0$$

solving for $\ddot{z}$

$$\ddot{z} = -\frac{K_{spring}}{M_{tip}}z - \frac{B_{damp}}{M_{tip}}\dot{z}$$

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 a 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 (Fixe-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

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. Create a new blank model.

> **This model needs to be saved as HarmOscModel.mdl for the GUI to work later. Make sure the model is saved in the path of Matlab.** 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.

II. Build the model.

III. Set up model parameters

> Make sure to change in the Simulation Configuration Parameters menu the Stop time to SimTime, the solver options type to Fixed-step, the Fixed step size to SimStep, and the solver to ode4(Runga-Kutta). Set the initial conditions of the integrators to Zprime and Zstart respectively. Set the properties of the gains appropriately. Use an addition box to add these two gains back into the first integrator. Send Ztraj to the workspace. By using a clock, also send the time to the workspace.

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 `SimPrep` from command line.

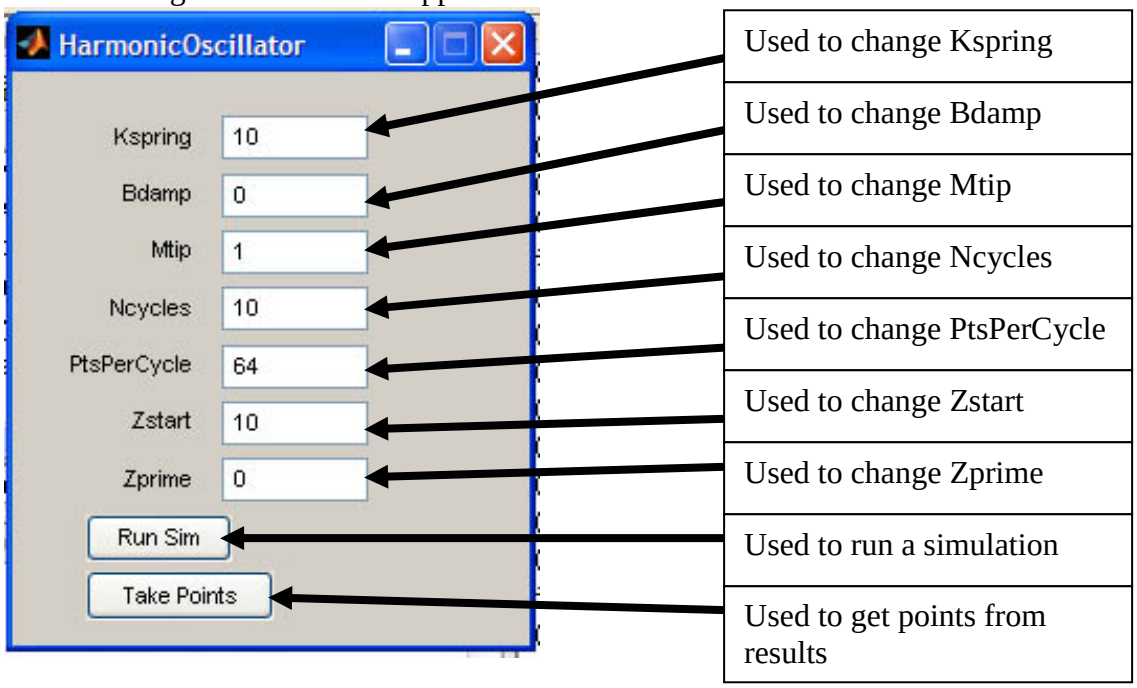
>> `SimPrep`

(If you look inside this m-file, you will see a list of variable being declared as globals.)

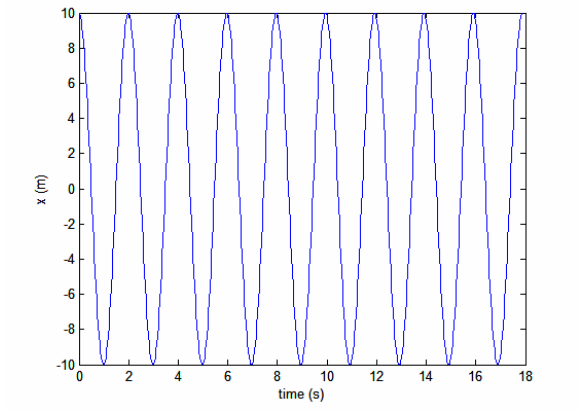
To open the GUI, from command line type `HarmOscGUI`

>> `HarmOscGUI`

The following interface should appear.



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:



Now, you can freely change parameter 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.

Later, you will be asked to load special GUIs as homework problems. Make sure that you close any open GUIs before opening a new one, and make sure that you run `SimPrep` before opening any GUI. (hint: it may also be useful to clear the workspace before running `SimPrep`). Feel free to play around with parameters just to see what happens.

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*.

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

Part I

Explorations with user-defined parameters.

1.1 Set **Kspring** to 10, **Bdamp** to 0, **Mtip** to 1, **Ncycles** to 100, **PtsPerCycle** to 64, **Zstart** to 10 and **Zprime** to 0. In the Simulation Configuration Parameters menu of your model, change the Solver to ode1(Euler). Run the simulation. Knowing that the trajectory should be harmonic, comment on the solution achieved using the Euler method. How could this result be improved using the Euler method? Now explore the other options for the Solver (i.e. ode2(Heun), ode3(Bogacki-Shanpine), and ode4(Runge-Kutta)). Comment on the relative ability of all of these methods to produce the expected harmonic result.

1.2 Set **Bdamp** to 0, **Ncycles** to 10, **PtsPerCycle** to 64, **Zstart** to 10 and **Zprime** to 0. Generate a few solutions with various values of **Kspring** and **Mtip** to demonstrate how the natural frequency of the oscillator depends on these parameters. In each case determine the natural frequency from the plot by measuring spacing between the maxima. Derive a simple relationship between **Kspring**, **Mtip**, and the natural frequency.

1.3 Generate solutions demonstrating the role of initial conditions, both for **Zstart** and **Zprime**.

1.4 Set **Kspring** = 4, **Mtip** = 1, **Ncycles** = 20. We will now explore the role of damping by varying **Bdamp**. Run simulations with **Bdamp** set to 0.01, 0.1, 1, and 2. Comment on the observed effect. This regime is called the underdamped harmonic oscillator. If a damping coefficient is defined as $\gamma = \frac{Bdamp}{2Mtip}$ and the fundamental angular frequency(ω) is related to **Kspring** and **Mtip** by: $\omega_0^2 = \frac{Kspring}{Mtip}$, this is where $\gamma^2 - \omega_0^2 < 0$. Notice that the decay goes like $e^{-\gamma t}$ where *t* is time. Now, set **Bdamp** = 4. Comment on the trajectory. This is the critically damped harmonic oscillator ($\gamma^2 - \omega_0^2 = 0$). In successive simulations, set **Bdamp** to 5, 10, 15, and 25. Comment on the observed effect. This is the overdamped harmonic oscillator ($\gamma^2 - \omega_0^2 > 0$).

Using parameters of your choice (not the ones from above), generate trajectory plots for the:

a) Underdamped

b) critically damped

and

c) overdamped

oscillator.

Part II

These are the problems with preset values of parameters, some of which are going to be unknown to you. Your task is to determine these parameters by analyzing the resulting trajectory plot. Each problem is brought up by typing `Problem_#` from the command line, where # = 1 – 5 is the problem number. Before executing each of these problems you will need to "clean the slate" by clearing the Matlab environment, as shown below. Subsequently, run `SimPrep` from command line, execute the `Problem_#` file from command line, and press the **Run Sim** button to produce the trajectory. You can now take measurements to find answers to questions posed in the problem.

Example:

- *Close any open HarmonicOscillator GUI windows*

- *"Clean the slate" by closing all open figures and clearing the variables in the workspace*

```
>>Clear all;
>>Close all
```

- *Prepare the workspace parameters*

```
>>SimPrep;
```

- *Start the problem*

```
>>Problem_1
```

-*Run the simulation*

Press **Run Sim** button

!!! **Note**: Unlike in the `HarmOscGUI`, due to the removal of parameters from the workspace, the simulation will not work if you hit the **Run Sim** button a second time. To rerun the simulation, you will need to run `SimPrep` first.

2.1. `Problem_1`

From the given parameters and trajectory, determine *Bdamp* and *Kspring*.

2.2. `Problem_2`

This is *almost* the same as the previous one. What is the difference?

2.3. `Problem_3`

This one differs from the previous one by just one parameter. Determine its value.

2.4. `Problem_4`

Again, only one parameter has been changed compared with the previous one. Find out its new value.

2.5. `Problem_5`

Just one parameter has been changed compared to the previous one. What is its new value?