Lab Exercise #4

INSTRUCTIONS

Complete the exercises below and upload them to Canvas as a single MATLAB script file using the naming convention "ENGR131_21F_Lab##_abc123.m", replacing abc123 with your Case ID, and ## with the two-digit lab number.

For example, if Dr. Williams were submitting Lab 4 it would be ENGR131_21F_Lab04_mrw8.m

For your script, please perform the following:

1. Separate each question into separate, runnable sections using the "%%" comment notation.
2. You may use the code and notes from class, the textbook, MATLAB's documentation, and anything you find using Google to solve these problems.
3. Use comments as appropriate to indicate your thoughts and how your code works (or is supposed to work). This is 5 points (10%) of your grade.

QUESTIONS

There are 3 questions for this lab.

1. MODULAR PROGRAMMING (20 PTS)

For this Lab, you will be building on the work of Lab 3. In this lab, you will use nested for loops to plot the behavior of the system shown in Labs 2 & 3 over several damping coefficients and oscillation frequencies. You will also be revising your Lab 3 code to make it more modular. Modifications to your Lab 3 code will consist of the following:

- Removing the user input commands and instead loading the system parameters from a file
    - This file, once loaded, will two variables, DC, and w in your workspace. These are the damping coefficients and oscillation frequencies you will be evaluating.
- Revising the CalcPosition function to use w as an input argument
- Moving the plotting commands to a separate function that uses the loop variables as well as the time, computed height, and w as input arguments.

The overall output of this question will produce a plot similar to that shown in Fig. 1.

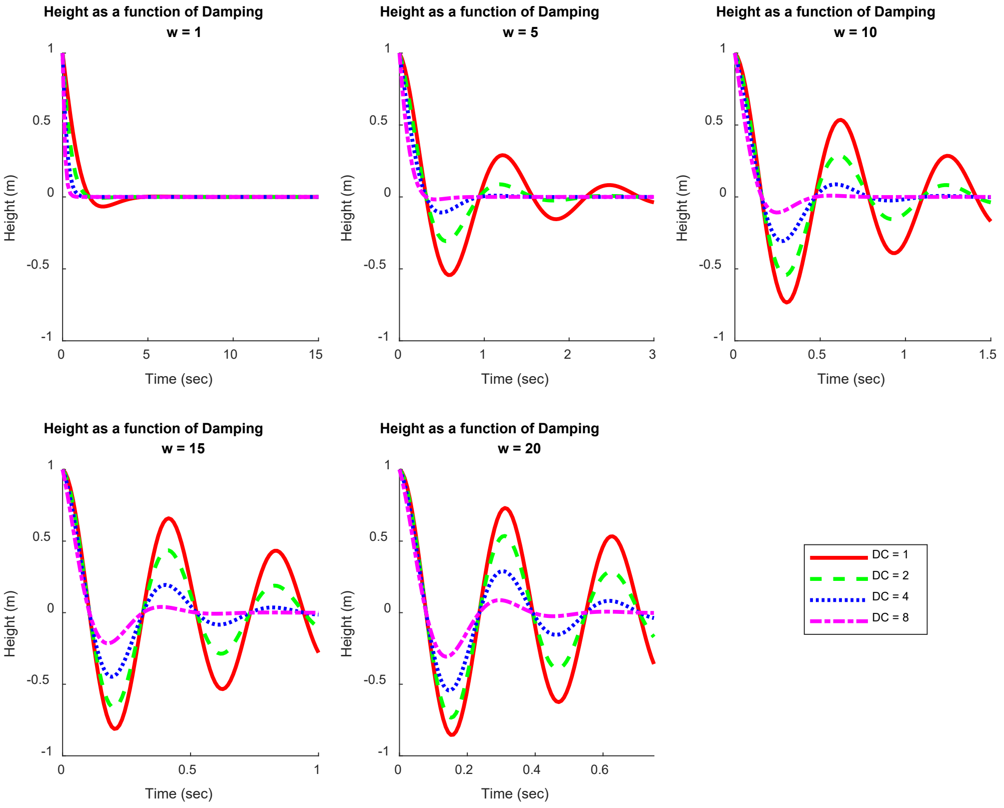

Fig. 1. Example of output for question 1 showing five plots, each with a different frequency of oscilation (w) across four different Damping Coefficients. Both the title and the legend entries use the str2num function to include numeric data from the variables used for the plots.

Recreate the plot in Fig. 1 by following the steps below:

A. Create a new script and copy your CalcPosition function from Lab 1 over to this new script (remember, functions stored inside the script file are not available to other scripts). Include a third input argument, w (1 pts)

B. Load the data file ENGR131_Lab4_Data.m into the workspace (3 pts).

C. Since we'll be looping over two variables, we'll need two loops with one inside the other. For the first one (outer loop), create a loop that will range from the first to the last value of the oscillation frequency (w) (4 pts).

D. Create a Time vector that ranges from 0 to 15/w for each value of w and contains 100 evenly spaced values.

E. For each value of w, calculate and plot the height of the system over time for each of the damping coefficients that you calculated

   1. Use a for-loop that runs from 1 to the length of the damping coefficient vector (this is your inner loop) (3 pts)
   2. Compute the height for each damping coefficient by calling the CalcPosition and passing in your Time vector, the damping coefficient for that iteration, and the w for that iteration.
   3. Ex: Time, DampingCoeff(1,j), w(1,i)
   4. Call your plotting function (Covered below in part f) and pass both loops variables, the time and height vectors, and the current value of w.

F. Create function to handle plotting your data. It should have your two loop variables, the time and computed height vectors, and the current value of w for that outer loop iteration as input arguments

   1. "scoop up" the values of the damping coefficients by using the evalin command (this is super lazy programming but a good learning example) (1 pts).
   2. Similar to Lab 3, use a switch statement based on the inner-loop iterator variable to assign a character array to the color and linetype code you wish for that iteration. (1 pts)

      ```
      Ex: switch i
                case 1
                      LineColor = 'b-';
                ….
            end
      ```

G. Set figure 1 as the active figure (the first time you use this, it will create the figure, the subsequent times, it will "grab" the figure and make it active for the following plot commands.

H. Use the subplot command to make a 2 x 3 set of subplots and select the current one using the outer-loop iterator variable you passed in (3 pts).

I. Set the hold on toggle (so you don't overwrite your plots with each iteration)

J. Plot the system behavior using the plot command using the character array you set in the switch statement to specify the color and line type codes.

K. Add appropriate axis labels (1 pts)

L. Create a two-line title with the first line stating what is being shown and the second indicated the value of the oscillation frequency (*hint: ['w = ',num2str(w)]*) (1 pts)

M. Check to see if the last plot has been plotted. If so, place a legend on the plot that lists value of each damping coefficient. (hint: this too will involve the num2str command). Include this property statement when you call the legend command: 'Position', [0.75 0.25 0.1 0.1] (2 pts)

N. Set the axis limits for x to be 0 to the last value in the time vector and y to -1 to 1

O. Turn off the hold toggle

2. Formatting Plots (20 pts)

Speaking of springs, Fall is in the air, Spring is far away, and soon caterpillars will be sleeping across the landscape. We can use Matlab to make our own that might look like those in Fig. 2. Note that this particular type of caterpillar has solid red triangle on its end segments and yellow x's on its middle segments.

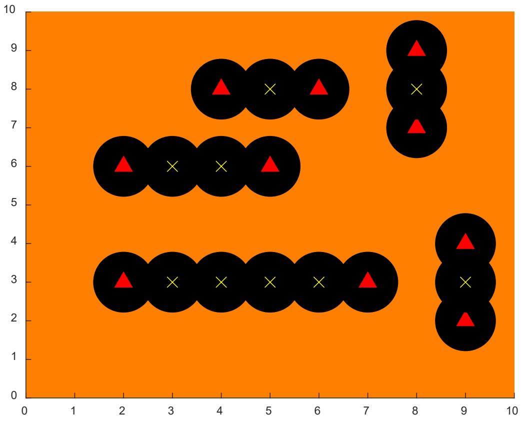

Fig. 2. Matlab plotted caterpillars on a 10 x 10 grid. The diamonds indicate the head. Filled in diamonds are hungry caterpillars.

To recreate something similar to Fig. 2, perform the following:

A. Load the file ENGR131_Lab4_CatMap.mat into the workspace. This will produce a variable, CatMap, which is a 10 x 10 map of where the caterpillars are.

B. Create a function that determines where the ends and middle segments are. Pass in CatMap as an input argument and output a modified map, CatMarks (10 pts)

C. To start, assign CatMap to a variable called CatMarks (ie. they're the same array)

1. Use nested for-loops with ranges equal to the number of rows and columns of the CatMap, ignoring the edges and scan the array element by element (ie. Do not scan through the first and last rows or columns)
2. In the case that an element in the map is a body segment (value of 1), add up the number of segments touching it.
3. Determine if this is an end segment based on the number of segments touching it. If it is, assign the value 2 to CatMarks in that location to indicate that it is an end.

D. Create a new figure

E. Change the background of the figure to orange. Use the hold on command after this (5 pts).

F. Call the function you created in part b.

G. Because plots increase "upwards" where the rows of an array increase "downwards" flip CatMarks vertically and assign to a new variable for plotting.

H. Scan the new plotting map using two nested loops – one for the rows and one for the columns of the plotting map

1. If the current element is above 0, mark it with a big, black circle using the various plotting properties to set the size, shape, and color (2.5 pts). *Hint: you will be plotting a single point, not a vector as we have been doing.*
2. Decide what marking to add to the segment (2.5 pts).
   a. A filled triangle if it is an end segment (a value of 2)
   b. An x if it is a middle segment.

I. Set your axis limits to 10 x 10

J. Turn off the hold

3. Computing Volumes (5 Pts)

The profile of a coffee cup is shown in Figure 3. The profile (radius) follows the equation r = log(h) + 0.5 where h is the height in cm. Compute the volume of the mug using both the trapezoidal and summation integration methods for determining the volume of a revolved solid using a step size of 1 mm and report these values to the command window. Plot the profile as shown.

- *Hint #1: Do not let the figure fool you.*
- *Hint #2: To plot the figure, swap the order of the independent and dependent axes.*

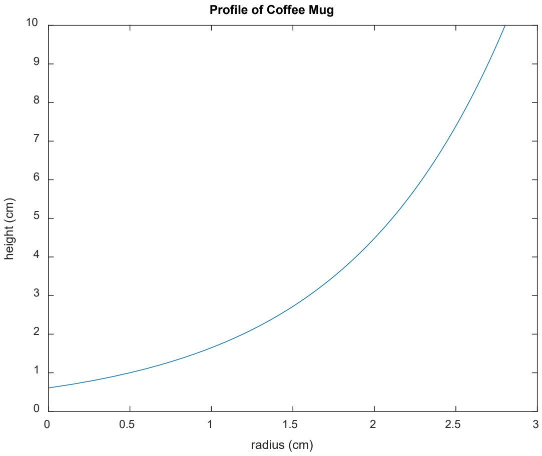

Fig. 3. Profile of coffee mug

Revision	Description	Date
A	Original Document	9/16/2021