ASSIGNMENT 2*

MTRN2500 Computing for Mechatronic Engineers - T3 2020

Due Date: 5:00pm, Friday 27th November 2020

Weighting of final course mark: 20%

1 Learning Outcomes

This assignment specifically targets the following learning outcome:

- To develop a fundamental knowledge of the Matlab framework and develop a skill to choose and use Matlab tools to solve problems in Mechatronic engineering.

2 Background

In this assignment, you will simulate the motion of a pendulum which has a time-varying friction. You will use several MATLAB tools in this assignment including plots and App Designer.

Consider the pendulum shown in Figure 1, where ℓ denotes the length of the rod and m denotes the mass of the bob. We assume the rod is rigid and has zero mass. Let θ denote the angle subtended by the rod and the vertical axis through the pivot point as shown in Figure 1. The pendulum is free to swing in the vertical plane and the bob of the pendulum moves in a circle of radius ℓ.

Using Newton's second law of motion, we can write the equation of motion as

$$m\ell\ddot{\theta}(t) = -mg\sin(\theta(t)) - \ell h(t)\dot{\theta}(t), \tag{1}$$

where g is the gravity constant and $h(t)$ is the time-varying friction defined as

$$h(t) = 2e^{-0.3t} + k, \tag{2}$$

with k being the steady state friction in the system.

At $t = 0$, the bob has an initial angular position and angular velocity denoted by $\theta(0)$ and $\dot{\theta}(0)$. Due to these initial conditions and the gravitational force, the pendulum starts to swing. We are interested in understanding how the pendulum swings for $t > 0$. In other words, we want to find $\theta(t)$ and $\dot{\theta}(t)$ for $t > 0$. So $\theta(t)$ and $\dot{\theta}(t)$ are the outputs of the system.

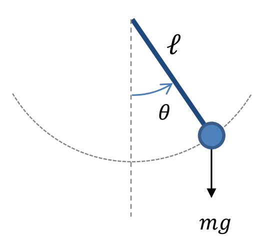

Figure 1: Pendulum system

*Prepared by Mohammad Deghat, November 2020.

Define the state variables $v_1(t) = \theta(t)$ and $v_2(t) = \dot{\theta}(t)$. Then (1) and (2) can be written as

$$\begin{aligned} \dot{v}_1(t) &= v_2(t) \\ \dot{v}_2(t) &= -\frac{g}{\ell}\sin(v_1(t)) - \frac{2e^{-0.3t}+k}{m} v_2(t). \end{aligned} \tag{3}$$

Note that (3) is in fact the same as (1) and (2), except that we wrote the second-order differential equation in (1) as two first-order differential equations in (3). Equation (3) is called the state-space representation of (1).

3 Assignment Specification

This assignment is split into four components: a style component and three other tasks. This document will explain what you are required to do and the contributions each part will make to your overall assignment mark. The total marks available for this assignment are 20. You will complete this assignment individually.

The marks allocated to each part of the assignment are as follows:

- Part 1 (4 marks): Plotting $\theta(t)$ for a range of initial conditions.
- Part 2 (7 marks): Dynamic figure
- Part 3 (6 marks): App Designer
- Style Component (3 marks): Things we will be looking for include
  1. (1 mark) Consistent and neat structure
  2. (1 mark) Choice of names for variables
  3. (1 mark) Commenting and readability of the code

3.1 Part 1 (4 marks)

Your mission in this part is to plot the solutions of the differential equation (3) for different initial conditions. We aim to plot the solution $v_1(t) = \theta(t)$ of (3) for 100 initial values:

(a) Define variables in MATLAB that contain the following parameters of the system: (1 mark)

- Length of the rod : 1 (m)
- Mass of the bob: 1 (kg)
- The gravitational acceleration : 9.81 (m/s^2)
- The constant k in (2): 4 (N)
- A 2-by-1 time-span vector containing the initial time which is zero and the end time of 5 (sec). This vector will be used in the `ode45` function.
- A vector of initial conditions $v_1(0) = \theta(0)$ with 100 evenly spaced values between -2π and 2π.

(b) Create a function named "System" that represents (3). Use the space allocated in the template MATLAB file to define this function (do not create a separate program file; all your MATLAB code for Part 1 and 2 should be inside a single script program). (1 mark)

(c) Call the function "System" in `ode45` to plot the solutions $v_1(t) = \theta(t)$ for all 100 initial conditions defined above in Part 1(a). Assume the initial value $v_2(0) = \dot{\theta}(0)$ is zero. All 100 plots should be plotted in the same figure (figure(1)). Use appropriate labels for x and y axes and a title for the figure. (2 marks)

3.2 Part 2 (7 marks)

In this part, you create a figure that includes 3 subplots: a dynamic figure, the trajectory of $\theta(t)$ and the trajectory of $\dot{\theta}(t)$. The structure of the subplots is shown in Figure 2.

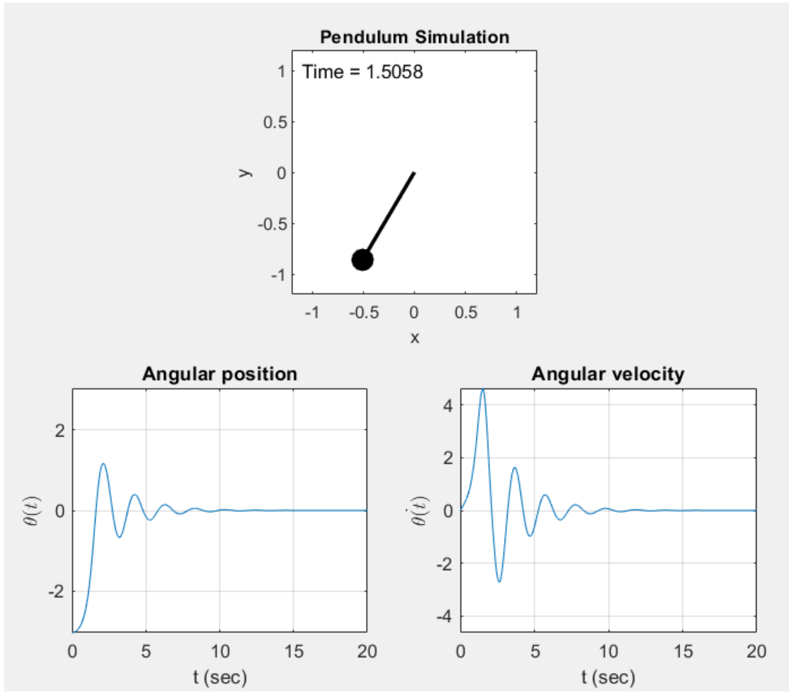

Figure 2: Simulation layout

(a) Let the initial conditions in this part be $v_1(0) = \theta(0) = 2\pi - 0.1$ and $v_2(0) = \dot{\theta}(0) = 0$. Also change the friction constant value, k, to 0.5 and the end time in the time span vector to 20 (sec).

(b) Open `figure(2)` and plot the trajectory of $v_1(t) = \theta(t)$ in a subplot located in the second row and the first column of the figure. Use an appropriate title, x-label and y-label to describe the subplot. The x-axis should be from 0 to 20 (sec) and the y-axis should be such that it is tight and symmetric. In other words, the y-axis limit should be such that the absolute value of the maximum and minimum y-axis limits are equal. Furthermore, the y-axis limit should be equal to the maximum of $|\theta(t)|$ for $t \in [0, 20]$. See, for example, the y-limits of the two sublpots in the second row of Figure 2. (2 marks)

(c) Plot the trajectory of $v_2(t) = \dot{\theta}(t)$ in a subplot located in the second row and the second column of the figure. Use an appropriate title, x-label and y-label and change the x-axis and y-axis limits in a similar way that you did in Part 2(b). (1 mark)

(d) Plot a dynamic figure that spans both columns in the first row. The dynamic figure should show the motion of the bob for all times from 0 to 20 (sec). (4 marks)
In particular, the dynamic plot should include:

- A text box at the top left corner of the subplot that shows the time (for example, shows "Time = 2.1021" when $t = 2.1021$)
- The location of the bob. The bob should be shown as a black full circle.
- The position of the rod (shown as a black line).
- Appropriate x-label, y-label and title.

Note that you are only allowed to use the `plot` function to plot the rod. Other plotting functions is MATLAB such as `line` cannot be used.

3.3 Part 3 (6 marks)

In this part you will use App Designer to create a simulation app.

(a) Drag the components shown in Figure 3 from the Component Library onto the canvas.

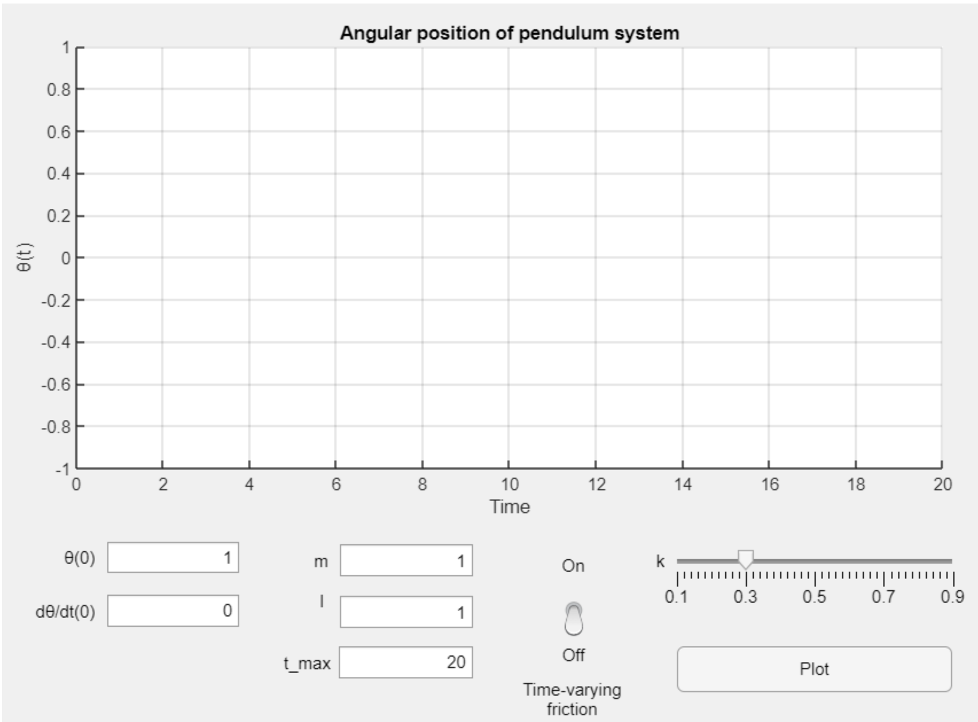

Figure 3: Simulation layout

- $\theta(0)$ and $\dot{\theta}(0)$, denoted by $d\theta/dt(0)$ in Figure 3, can take any value from -inf to inf.
- m and l can take values from 0.1 to inf.
- t_max can take values from 0 to inf.
- The switch changes the friction. If it is 'On' then the friction is time-varying and is equal to (2). Otherwise, the friction is constant and is equal to k. In this case, the second equation of (3) changes to:

$$\dot{v}_2(t) = -\frac{g}{\ell}\sin(v_1(t)) - \frac{k}{m}v_2(t). \tag{4}$$

- The slider should take k values from 0.1 to 0.9. The tick values should be 0.1, 0.3, 0.5, 0.7 and 0.9.
- The default values for $\theta(0)$, $d\theta/dt(0)$, m, l, t_max, the time-varying friction switch, and k should be the same as the ones shown in Figure 3. The default axes limit for the figure should be the same as the values in Figure 3: x_min=0, x_max=20, y_min=-1, y_max=1.

(b) A plot should be updates when the plot button is clicked. If there is no plot currently, one should be created. Similarly to Part 2(b), the y-axis should be symmetric and tight (see the explanation in Part 2(b)).

3.4 Submission of your results

You are required to submit two files: (a) A single .m file that includes the results for Part 1 and 2, and (b) a .mlapp file that contains your App Designer file for Part 3. Each file must be named with your full zID (i.e. z1234567.m and z1234567.mlapp) and submitted to Moodle by 5:00pm on Friday 27th November 2020. There will be two submission boxes in Moodle where you can submit each of your MATLAB files.

Feedback on this assignment will be provided within 2 weeks of the submission deadline.

4 Progress Check

You will have your progress checked with your demonstrator in a 5-minute meeting on the afternoon of Friday 13 November. You should book a time by writing your information in the Excel file in the MATLAB Assignment channel on Teams.

To pass the progress check, you must complete Part 1 and demonstrate your result to the tutor.

5 Additional Information

- A plagiarism check will be performed on all assignments and any instances of plagiarism will be dealt with under the UNSW plagiarism policy (linked from the course outline).
- Your MATLAB files should compile without error. Make sure your files work fine before you submit them.
- Please refer to the course outline for late submission policy.
- Please post questions about this assignment on MS Teams.
- Finally, enjoy learning how to work with Matlab.