Assignment 1: Matlab Basics

Optimal Control For Robotics

Assigned: Jan. 18 — Due: Jan. 25

Problem 1: fun with random sequences (25 pts)

In this assignment you will write a computer program that generates a sequence of points in a plane, which will converge to a well-knot fractal. There are two parts to this problem, each with slightly different rules for generating the sequence of points, resulting in two different fractals.

In addition to submitting plots for each part (on a single figure), you will also be asked to submit your code and detail how long the assignment took you to complete. Make sure that your code is well documented (use comments!) and clearly written so that someone else can make sense of it.

Part One: a special triangle

Start by selecting three control points (A, B, C) that are uniformly spaced around the edge of a circle. Next, generate a sequence of points $P_0 ... P_N$ using a random number generator and the control points. Use $\alpha = \frac{1}{2}$ and the choice of A, B, or C should be randomly selected with equal probability.

$$P_0 = A \tag{1}$$

$$P_{k+1} = \alpha \cdot P_k + (1 - \alpha) \cdot \texttt{RandomChoice}(A, B, C) \tag{2}$$

Create a plot to visualize the sequence, using the following guidelines:

- plot the circle as a thin curved black line
- plot each control point (A, B, C) as a small red circle
- plot the initial point P_0 as a small green "X"
- plot $P_1 ... P_N$ as tiny blue dots, where N is 5000
- make sure that the axes are scaled so that the circle looks like a circle

Part Two: a special rectangle

Repeat the entire procedure from Part One, but make two changes:

- Select four uniformly spaced control points around the edge of the circle, instead of three.
- Set $\alpha = \frac{1}{3}$

Deliverables:

1. Submit your matlab code as a single file: `prob_01_studentName.m`
2. Submit a single figure (with two sub-plots) as a pdf: `prob_01_studentName.pdf`
3. Create a short write-up for the problem: `prob_01_studentName.txt`
    - Header: full name, studentName, date, problem name and number
    - List any other students that you worked with.
    - How long did this problem take you?
    - Briefly describe or outline your code (roughly 50 – 100 words).

Comments:

- In part one you generated the *Sierpinski Triangle* fractal. This algorithm is just one of many ways to create this fractal.
- This problem was inspired by the numberphile video "Chaos Game": `https://www.youtube.com/watch?v=kbKtFN71Lfs`
- See the Hints section at the end of the assignment for how to save a Matlab figure into a pdf and how to include multiple matlab functions in a single file.

Problem 2: simple plots and derivatives (25 pts)

In this problem you will generate a single figure with a set of six-subplots. The figure is described below, along with the functions that you will be plotting. As with the previous problem, please keep track of the time you spend on this problem and make your code well-documented and organized. You may use any method that you like to compute the derivatives of $x(t)$ and $y(t)$.

$$x(t) = \left(1 + (t-2)^2\right) \cdot \sin(3t) \tag{3}$$

$$y(t) = t^3 - 6t^2 + 2t + 5 \tag{4}$$

$$\dot{z}(t) \equiv \frac{d}{dt} z(t) \qquad \ddot{z}(t) \equiv \frac{d^2}{dt^2} z(t) \tag{5}$$

What to plot:

- plot $x(t)$, $\dot{x}(t)$, and $\ddot{x}(t)$ on the domain $t \in [0, 5]$
- plot $y(t)$, $\dot{y}(t)$, and $\ddot{y}(t)$ on the domain $t \in [0, 5]$
- the plots for $x(t)$ and its derivatives should be in the left column
- the plots for $y(t)$ and its derivatives should be in the right column
- $x(t)$ and $y(t)$ should be in top two plots
- $\dot{x}(t)$ and $\dot{y}(t)$ should be in middle two plots
- $\ddot{x}(t)$ and $\ddot{y}(t)$ should be in bottom two plots
- Plot a blue circle at the point where $y(t)$ reaches its maximum value on the domain $t \in [0, 5]$
- Plot a red "X" at the point where $y(t)$ reaches its minimum value on the domain $t \in [0, 5]$
- All plots should have both axes labeled and a title.

Deliverables:

1. Submit your matlab code as a single file: `prob_02_studentName.m`
2. Create a single figure, with six sub-plots: `prob_02_studentName.pdf`
3. Create a short write-up for the problem: `prob_02_studentName.txt`
   - Header: full name, studentName, date, problem name and number
   - List any other students that you worked with.
   - How long did this problem take you?
   - Briefly describe your code (roughly $50 - 100$ words).
   - Show work for derivative calculations (use pseudo-code, *e.g.* $\dot{x} \rightarrow$ `dx`).
   - How did you compute the minimum and maximum values?

Matlab Hints:

You can export a matlab figure to a .pdf file using the following function:

```
function saveFigureToPdf(fileName, hFig)
% saveFigureToPdf(fileName, hFig)
%
% This function saves a figure to a pdf file
%
% INPUTS:
%   fileName = string = save the figure under this file name
%   hFig = figure handle optional  (default:  gcf)
%

if nargin < 2
    hFig = gcf;
end

hFig.PaperPositionMode = 'auto';
pos = hFig.PaperPosition;
hFig.PaperSize = [pos(3), pos(4)];
print(hFig, fileName,'-dpdf')

end
```

You can use multiple functions in a single file:

```
function prob_01_studentName()

disp('Hello World!  Let''s add two numbers together:');

a = 4;
b = 7;
c = addTwoNumbers(a, b);
fprintf('%d + %d = %d\n', a, b, c);

end

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~%

function c = addTwoNumbers(a, b)
%
% This function adds the two inputs together:  c = a + b
%

c = a + b;

end
```