CS1112 Fall 2014 Project 3 Part B **due Friday 10/3 at 11pm**

You must work either on your own or with one partner. If you work with a partner you must first register as a group in CMS and then submit your work as a group. *Adhere to the Code of Academic Integrity.* For a group, "you" below refers to "your group." You may discuss background issues and general strategies with others, but the work that you submit must be your own. In particular, you may discuss general ideas with others but you may not work out the detailed solutions with others. It is not OK for you to see or hear another student's code and it is certainly not OK to copy code from another person or from published/Internet sources. If you feel that you cannot complete the assignment on you own, seek help from the course staff.

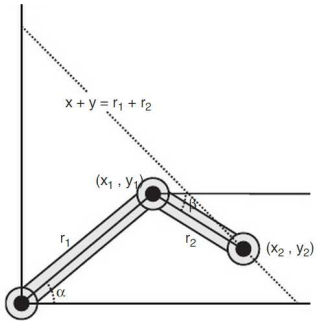
Objectives

Completing this project will solidify your understanding of user-defined functions and vectors. You will also do more graphics.

Part A, specifying problem 1, is in a separate document.

2 Sensitivity Analysis

A *sensitivity analysis* is used to explore how the result of a model changes when the inputs to the model are systematically varied, i.e., we ask the question "how *sensitive* is the result to variations in the model parameters?"



In Project 2, we considered the two-hinged robotic arm problem where the distance $D(\beta)$, the difference between the end-effector and the target line $x+y = r_1 + r_2$ is given by

$$D(\beta) = r_1 \cos\alpha + r_2 \cos\beta + r_1 \sin\alpha + r_2 \sin\beta - (r_1 + r_2).$$

Given the values of r_1, r_2, and α, one can plot the distance D as a function of β. An example of such a curve was shown in Project 2. Now we will consider how the curve would change when r_2, the length of the second segment of the arm, is varied. (For example, the end-effector may include an extender sleeve). Specifically, you will plot the difference between the end-effector and the target line versus β for these values of r_2: $1, 2, \ldots, 6$. Let $r_1 = 4$ and $\alpha = \pi/6$ (as we did in Project 2).

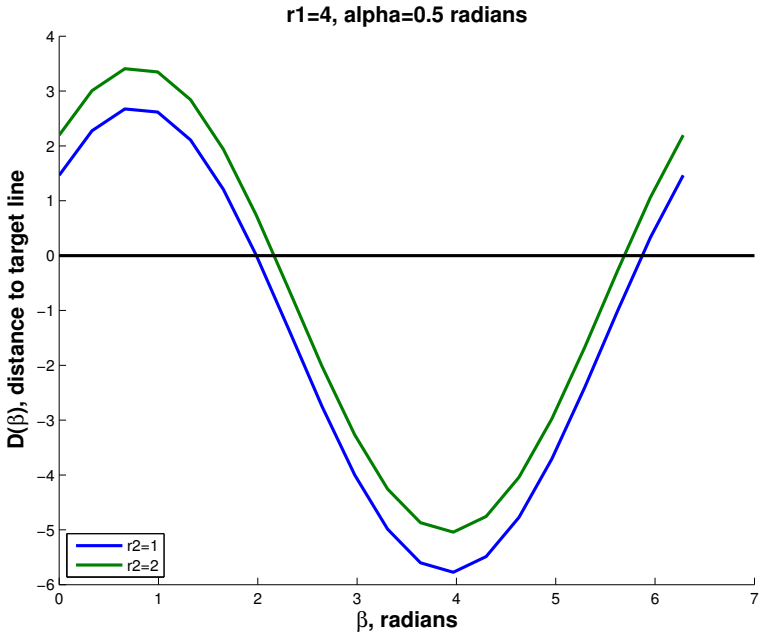
Write a script `varySegment2` to perform the analysis. *For full credit, make effective use of nested loops in your solution.* I.e., use a loop to compute the distances for $\beta = 0, \ldots, 2\pi$. The six curves must be plotted on the same set of axes inside one figure window. Show a legend in addition to a title and axis labels. The following skeleton shows the relevant graphics commands for plotting the six curves one at a time:

```
close all
figure
hold all  % Hold formats as well as plots on current axes

for ____  % loop over the different values of r2

    % Compute the distance for beta in the range of 0 to 2pi and
    % store the values in vectors beta and dis (beta and distance).

    % Plot the curve
    plot(beta, dis)
end
legend('r2=1','r2=2','r2=3','r2=4','r2=5','r2=6', 'Location','Southwest')
xlabel(____)
ylabel(____)
title(____)
hold off
```

Note the use of the command `hold` *all* instead of `hold on`. `hold all` keeps the line format cycling instead of using the default blue line for every call to the `plot` function. In the `legend` command, you can use other directions, e.g., `'Southeast'`, to specify the placement of the legend in the figure.

The figure looks clearer with a horizontal line at distance=0. So *after* using the `legend` command, use another `plot` statement to add the horizontal line.

Below is an example graph that demonstrates the *format* only—your figure should have six curves and they may not look like these.



3 The coin game "Gap N"

Complete Exercise **P6.1.18** from *Insight*. First read the problem description on page 142 of *Insight* and additionally follow the specifications below.

The problem statement asks (a) what is the expected value of the score given N and (b) what is the probability that the game is over on or before the $4N^{\text{th}}$ flip of the coin. You will answer these questions by producing two figures: (a) expected value of the score vs. N and (b) probability that the game ends by the $4N^{\text{th}}$ flip of the coin vs. N. Each figure should display two curves, one for the case of a fair coin being used in the game and another for the case of a coin that has a probability of heads of 0.75 (i.e., heads is three times as likely as tails to show up).

3.1 Simulate one game

Implement the following function:

```
function s = gameGapN(N, ph)
% Simulate one game of "Gap N", which involves flipping one coin until the
% difference between the number of heads and the number of tails obtained
% is N.
% N:  the desired difference between #heads and #tails obtained.
% ph: the probability of getting heads on a toss of the coin.
% s:  the score at the end of the game, which is the number of tosses
%     required to achieve the difference N.
```

3.2 Statistical properties of the game

Write a script `statsGapN` to produce the two figures that will answer the questions asked above. You will use *simulation* to solve this problem; don't try to work out the analytical solution `:)`. When the number of trials is large, the *expected value* is just the average value and the *probability* of an event is just the number of times the event occurs divided by the total number of trials.

Graphics notes: In this problem, you may find it easier to use a single call to the `plot` function to draw two curves in one figure (see notes from Lecture 10) than to use the draw-one-curve-at-a-time approach used in the previous problem. To use a single call to `plot` to make two curves on one set of axes simply requires that the vectors of data for both curves are available when you call `plot`, unlike the previous problem when you had the data vector for only one curve at a time. To start a new figure, use the command `figure`.

Program development note: Decompose the problem! Don't try to deal with both questions (a) and (b) and both kinds of coins all at the same time! For example, pick either question (a) or (b)—pick the one that seems simpler to you—and deal with the fair coin case first. After implementing and testing just that part, then add/modify the code to deal with the unfair coin case. Finally, work on the remaining question.

Submit your files `varySegment2.m`, `gameGapN.m`, and `statsGapN.m` in CMS.