Homework 2: Locomotion

Due: Beginning of class, Monday 30-Jan-2017

This assignment is based on in-class discussion and Section 2 notes.

1. What are the advantages and disadvantages of legged robots over wheeled robots? What are some of the applications of each type of robot?
2. Chapter 2, Problem 1 (Page 25) from *Seigwart et al.*
3. Describe how an airfoil generates lift.

MATLAB Programming Problem

Consider a planar hexapod robot with a center of gravity (CG) located at:

```
% position of the center of gravity
cgX = 2.5;
cgY = 3;
```

and with all six legs planted on the ground at each corner of the robot's body:

```
% Case 1: Robot has legs planted at each corner
legsX = [2 0 1 5 6 4]';
legsY = [0 3 8 8 3 0]';
```

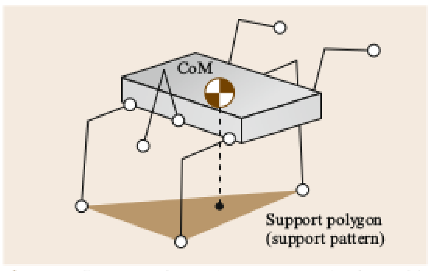
The support polygon is the polygon formed by connecting the lines that join successive legs that are planted on the ground. Thus, for m legs planted on the ground there are m edges in the support polygon. The static margin S_n is defined as the minimum of the CG to the edge of the support polygon.

In the example shown in Fig. 1 there are three legs planted on the ground and thus there are three perpendicular distances to each edge d_1, d_2, d_3. The static margin is the minimum of these three:

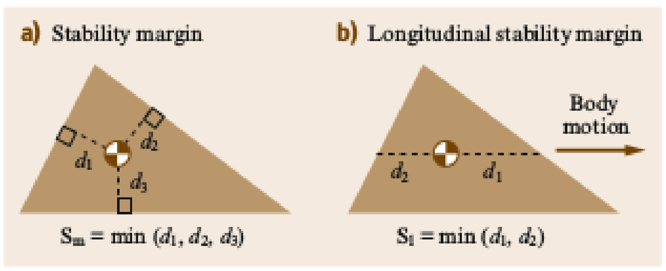
$$S_n = \min(d_1, d_2, d_3)$$

For a robot with m legs you will, in general, need to compare m signed distances [i.e., $S_n = \min(d_1, \cdots, d_m)$].

The distances should be defined as positive when the CG is on the interior of the polygon and negative if it is outside. Thus the bigger the value $S_n > 0$ the more "stable" the robot is. If $S_n = 0$ then the CG is on the boundary of the support polygon. If $S_n < 0$ the CG is outside the support polygon and the robot will topple over.



(a) Support polygon



(b) Static margin

Figure 1: Legged robot stability concepts

A numerical approach to computing S_n is to approximate each edge into a large number n of points. For example, if an edge is joining leg `i` to leg `j` then the edge can be approximated with `numLegPts` points as:

```
    legsXpts = linspace(legsX(i),legsX(j),numLegPts);
    legsYpts = linspace(legsY(i),legsY(j),numLegPts);
```

Then by creating a vector consisting of all of the points on the edge of the support polygon, and calculating the planar distance from each point in this set to the CG, the smallest value will give S_n. This approach does not actually compute the d value for each edge, but instead compares all of the points, across all edges, to obtain S_n directly.

But there is more than one way to solve this problem, so computing the d values for each edge, and comparing those, is also valid.

In either case, you iterate for each edge using a `for` loop. Pay careful attention to how the edges are defined, especially for the "last" leg connecting back to the "first". Once all of d values have been computed, S_n can be found from the formula given above. Note: the stability margin must be *signed* – so to check if S_n should be negative (when the CG is outside the support polygon) you may use

```
CGinside = inpolygon (cgX, cgY, legsX, legsY);
```

and modify S_n accordingly.

Part A. Write a MATLAB function of the form

```
[minStabMargin] = stabMargin( legsX, legsY, cgX, cgY )
```

that returns the static stability margin using the numerical approach described above. Verify the MATLAB output indicates a static margin of 2.08 for the nominal case with all legs planted on the ground. Grading will be based in part on clarity of code and quality of comments.

Part B. Use your script to compute the static margin in the following cases:

- Case i: The robot lifts its sixth leg.
- Case ii: The robot lifts its fifth and sixth leg.
- Case iii: The robot lifts its fourth, fifth and sixth leg.

In which cases would the robot topple over? Note: Simulate legs being lifted by simply removing them from the vector `legsX` and `legsY`.

Summary of What You Should Submit: After completing Part A and Part B you shoud submit the following:

- 1 hardcopy of the `stabMargin.m` script file
- 1 hardcopy of the `main.m` script (i.e., the script you used to call the `stabMargin.m` function file)
- 4 plots of the robot's stance (1 for each case, generated from the `drawLeggedRobot.p` script)
- 4 outputs of the S_n values (1 for each case, generated from successive calls to `stabMargin.m`). The output should be produced using an `fprintf` statements that is human-readable.

  For example, if the S_n value you calculated for Case 1 is a variable called `SnCase1`, then include the following as an output:

  ```
  fprintf('Static Margin for Case 1 (all feet planted) is: %4.4f \n',SnCase1)
  ```

  which prints to the Command Window:

  ```
  Static Margin for Case 1 (all feet planted) is: 2.0801
  ```

Hints:

1. Use the `drawLeggedRobot.p` script provided to verify your S_n values, as shown in Fig. 2. The file `drawLeggedRobot.p` is an encrypted m-file (you cannot see its contents). It will not "open" in Blackboard. You must download it, place it in your working directiory, and call it according to the syntax from within MATLAB as you would any other MATLAB file.

   The `drawLeggedRobot.p` script is used as follows:

```
% Usage:  drawLeggedRobot(legsX, legsY, cgX, cgY)
%
% Purpose: produces a simple drawing of a hexapod robot with a
% given center of gravity (CG) and leg positions
%
% Inputs:
% legsX : an (n x 1) vector of robot leg x-positions
% legsY : an (n x 1) vector of robot leg y-positions
% cgX : a scalar indicating robot cg x-position
% cgY : a scalar indicating robot cg y-position
% * Note : the leg positions must be defined consecutively along the
%   perimeter of the robot forming a convex polygon
%
% Outputs:
% a plot of the robot's body, support polygon, CG, and static margin
```

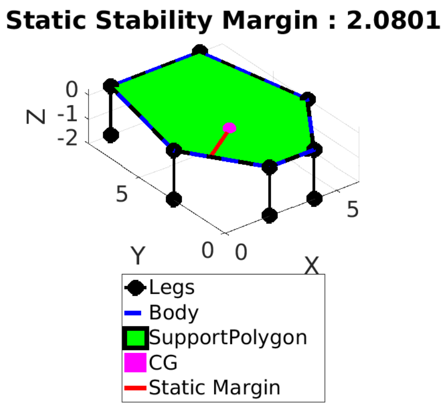


Figure 2: Robot with all legs planted on the ground