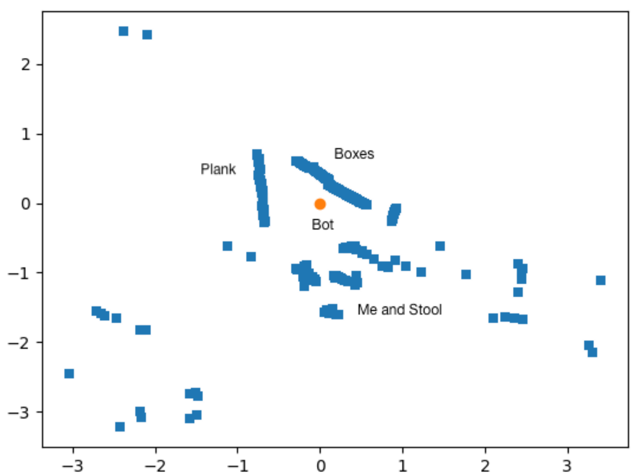
CS81 Homework 4

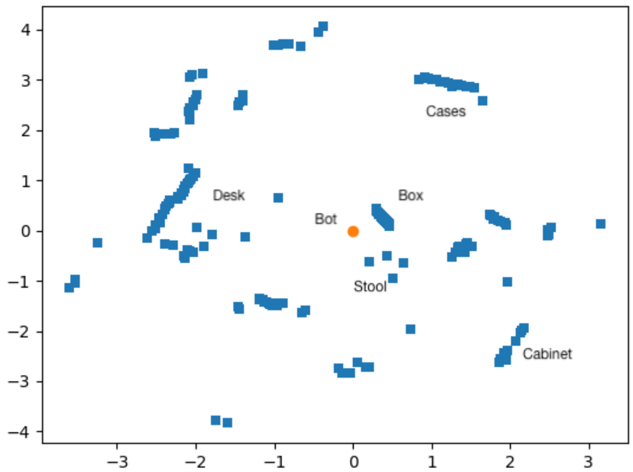
Dev Jhaveri, Anne Sherrill, Lisa Oh

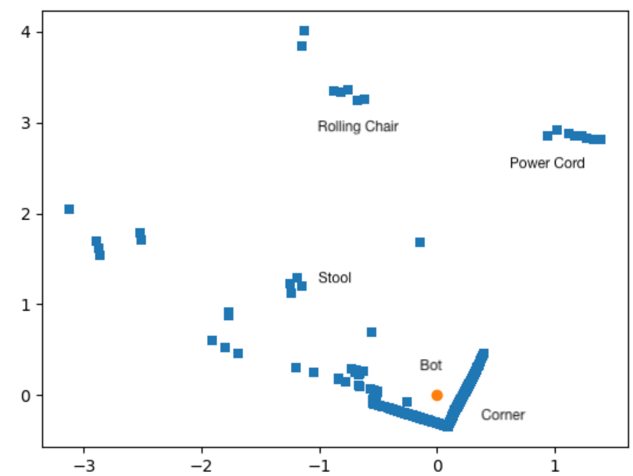
October 31, 2018

1 Problem 1 - Visualization

The below three graphs show the robot's surroundings in the global x-y frame. The orange dot is the robot.
These graphs were created by calculating x and y values of each item found by the rangefinder with the formulas:
$x = r\cos(\theta)$
$y = r\sin(\theta)$
and then plotting the points with a matplotlib scatter.

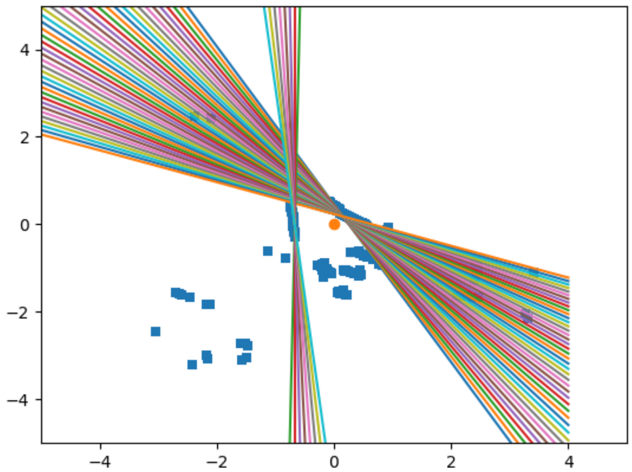


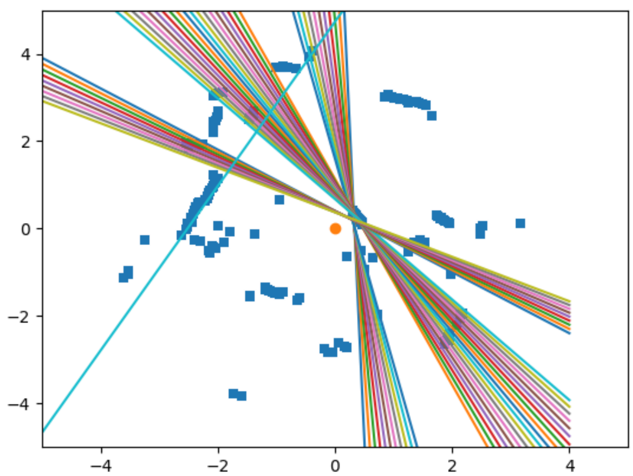


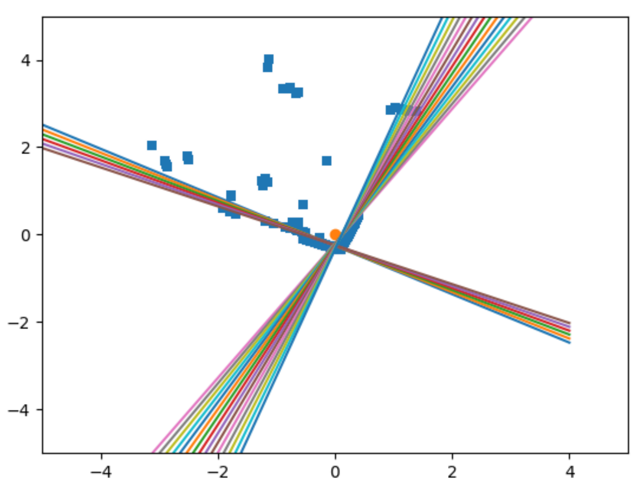


2 Problem 2 - Hough Transform

The below three graphs show the same visualization graphs from part 1, with overlaid lines representing walls identified by our program. The Hough Transform was implemented by creating a 2d array to represent an $rX\theta$ space and then, for each point and θ, voting into the correspondent (r, θ) array point as per the formula:
$r = x\cos(\theta) + y\sin(\theta)$.
For all the (r, θ) points that reached a certain threshold (threshold varying per situation), we plotted the line using the point-slope formula:
$y - y_0 = \frac{-x_0}{y_0}(x - x_0)$
Where x_0 and y_0 were calculated using the same x and y formulas as above.


4
2
0
−2
−4
−4
−2
0
2
4


4
2
0
−2
−4
−4
−2
0
2
4


4
2
0
−2
−4
−4
−2
0
2
4

3 Robot Alignment

See submitted video.
The robot alignment algorithm takes in a Hough array as computed above. It loops through each r value, which corresponds to the distances from the robot. For each of these distances, it finds the angle corresponding to the place at that distance that received the highest score, which is what the robot interprets to be most likely to be a wall at that distance. If the score is higher than some threshold, the robot takes that angle and distance to be the closest wall, and turns to it.

4 Angular Improvement