Jackie Scanlon
Algorithmic Robotics - PA3
Due: 3/28/19

Part 1: Pose-to-Pose Motion Control

1. *Provide a screenshot of your robot executing the paths it found for the demonstration defined just above.*

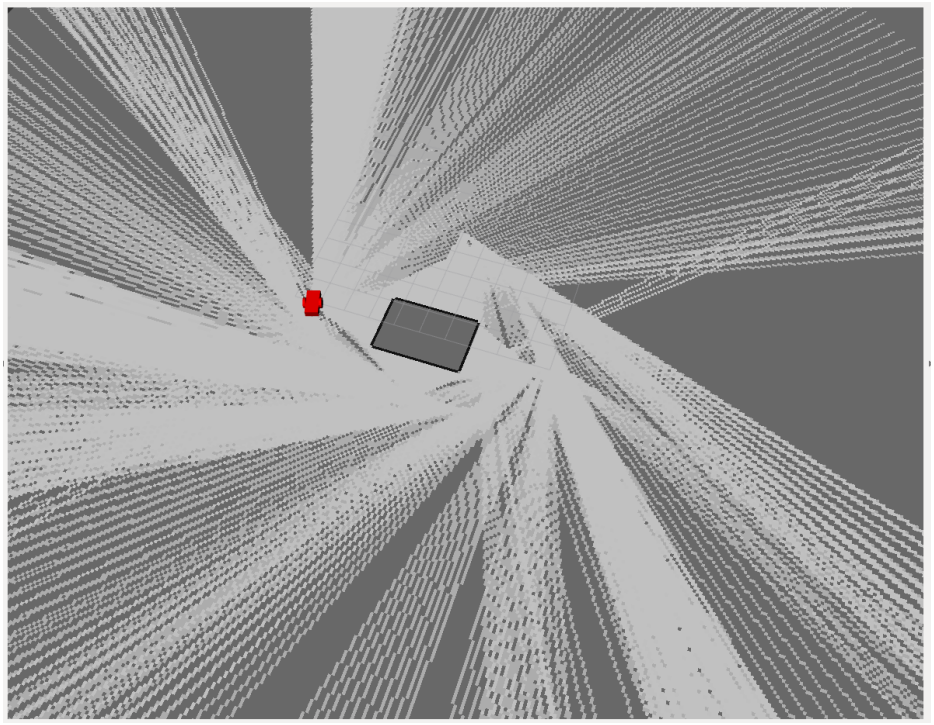
2. *If you implemented Corke's move-to-pose controller, what values of c, kα, kρ, and kβ < 0 did you choose? Explain why you chose them.*
   I chose kρ = 1, kα = 7, kβ = -4 and c = .25. The first 3 were to match Corke's so I could compare. C of .25 seemed to work well for getting my turtle to the right position.
3. *If you implemented Corke's move-to-pose controller, describe the robot's behavior when the controller had "bad" gains.*
   One set of bad gains I tried was c = 5, kd = 7, ka = 10, kb = -1. With this set of gains, the turtle would get to the correct position very quickly but then would need to spend a lot of time turning in circles to get to the right theta.

Questions 4-6: didn't do because I used Corke's controller.

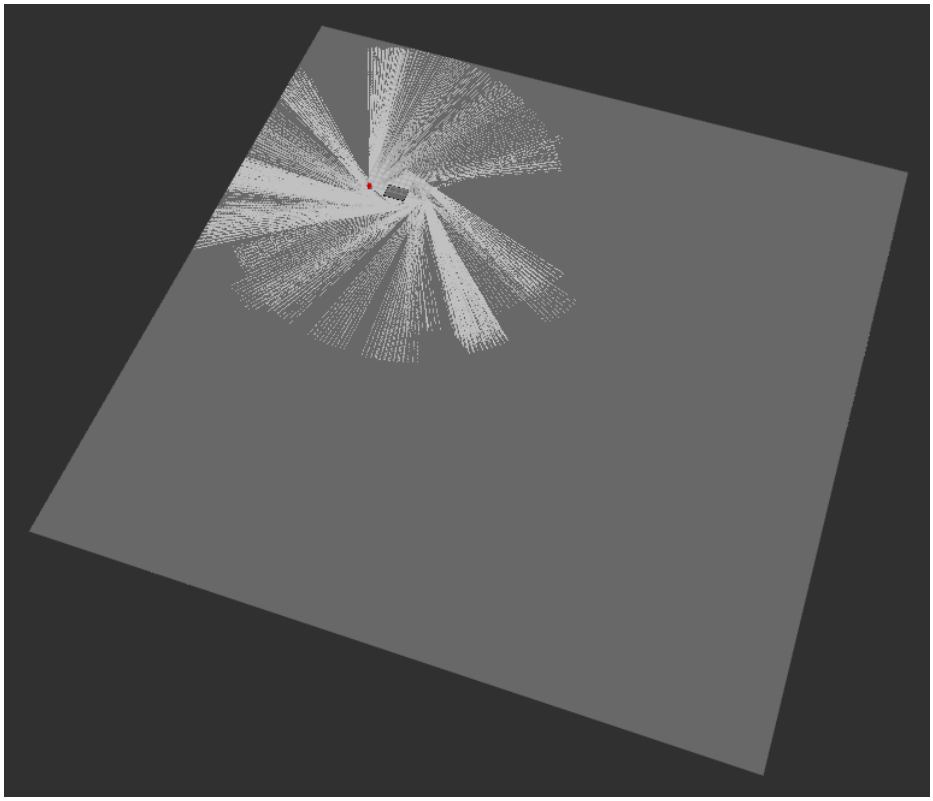
Part 2: Occupancy Grid Construction

Demo

Zoomed in:



Zoomed out:



Questions:
1. In a few short sentences, describe the intuition behind Bresenham's line algorithm.
Essentially, the algorithm increments x starting from the start point. It keeps track of an error variable that increases by the slope each time. If error is ever increased by more than half a step, y is also incremented. Thus it builds a list of x and y values that are along the line. The main point is that the "error" accumulated indicates how far along the line we are.

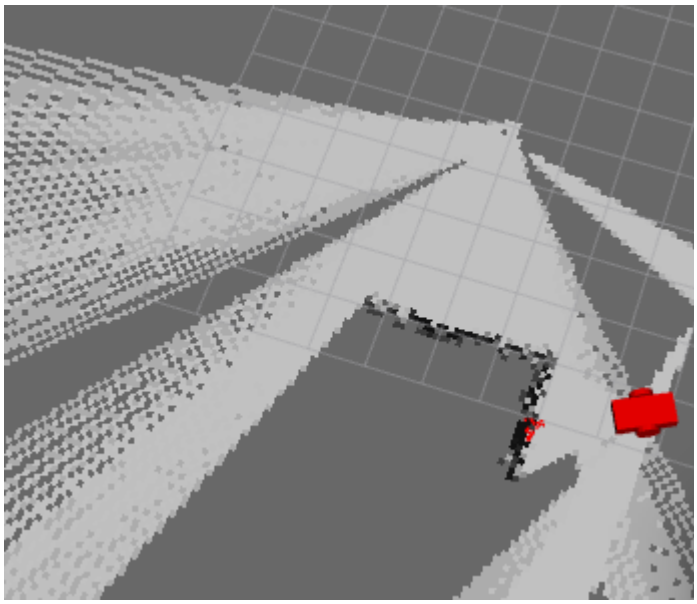
2. Describe one reasonable approach for handling scans that partially go beyond the boundaries of the map.
I have it set up such that scans can be taken anywhere. However, right before the grid value is updated for a specific cell, I check that the cell exists within the bounds of the grid. If it is not, I discard that cell (not the whole line).

3. How does the occupancy grid algorithm perform when $\beta = 1 - \alpha$?
Pretty terribly: the probabilities update so slowly that they don't really appear on the map ever. The only values that do show up are the spots where the robot was, because we know those spots have to be empty.

Part 3: Sensor and Controller Noise

Questions:
1. How does the twist noise impact the performance of your pose-to-pose motion controller?
The turtle is pretty jittery, especially when standing still. It seems to especially affect the theta value (less so x and y).

2. How does the sensor noise impact the construction of your occupancy grid?



The edges of the wall get pretty fuzzy. But, still somewhat useful.

Acknowledgment statement:
I used code from the Wikipedia page for Bresenham's Line Algorithm:
https://en.wikipedia.org/wiki/Bresenham's_line_algorithm

Other useful things I discovered:

- Quaternian/Roll-pitch-yaw conversion
- How ROS Occupancy grids work
- Tons of numpy stuff, like matrix multiplication, differences between matrix and vector in numpy, etc.
- A lot about lists and lists of lists or lists of matrices