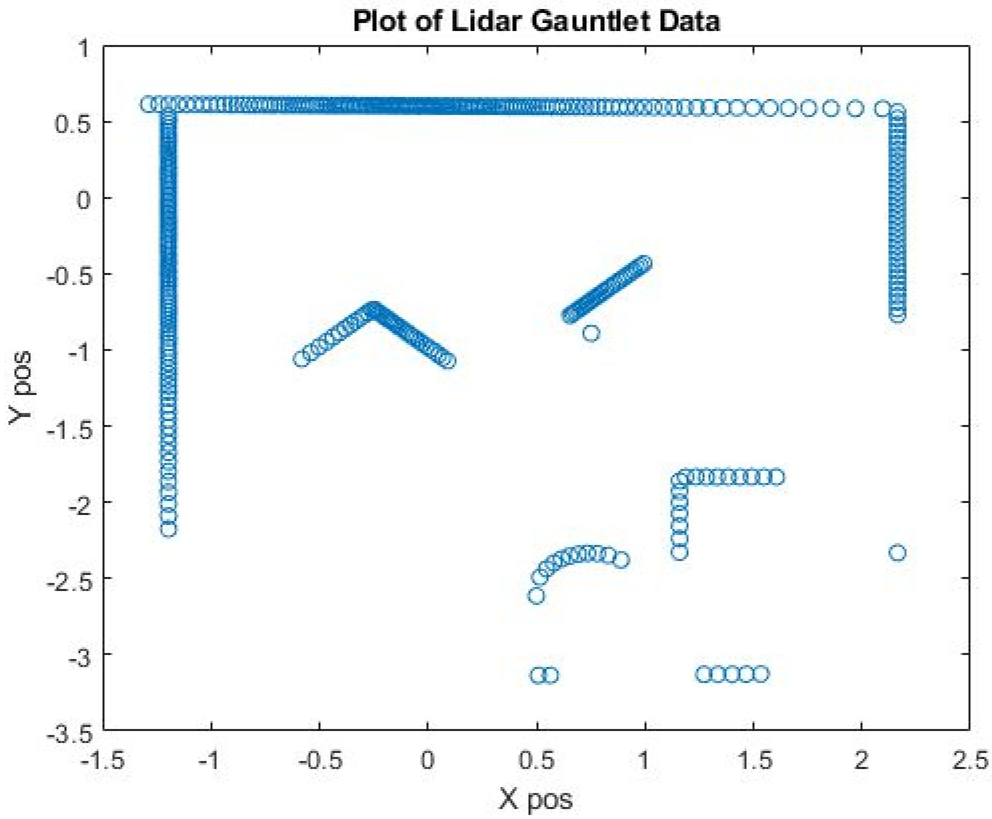
Adi R. & Patrick O.
QEA ROBO | 5.6.20

ROBO QEA Gauntlet Challenge Writeup Doc

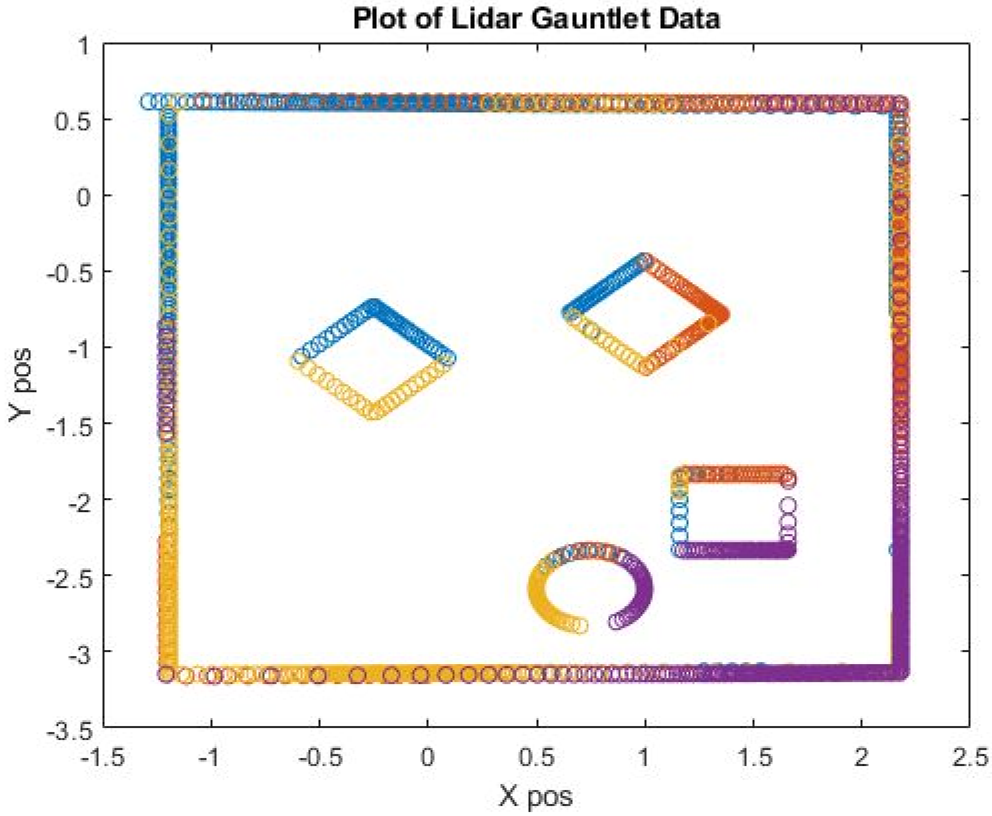
Mapping & Path Planning

1. Lidar plot of pen

Below, you can see the Lidar scan taken from the initial Neato Position of the entire Gauntlet Pen. This is the scan before any line fitting algorithms have been applied and just features dots plotted.

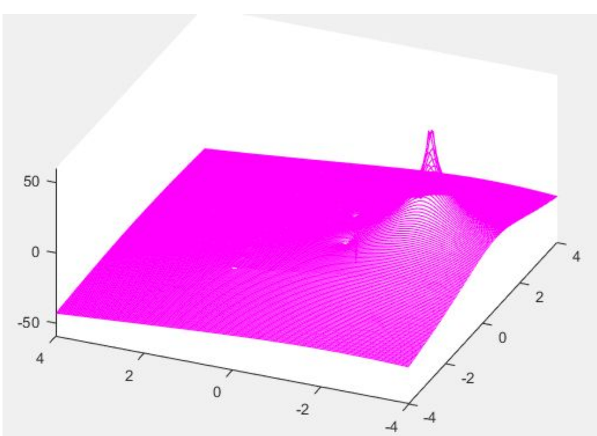


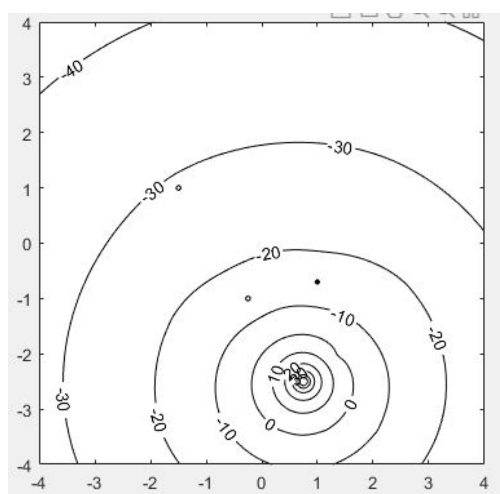
And below we've included a Complete composite Lidar Scan, taken from multiple different locations in the pen. The colors indicate scan information from the different scans.



2. Equation and contour of pot field

Below, you can see the 3D equation used to simulate the Level 1 Field. The equation can be broken up into a section for the Barrel, the Obstacles, and the borders. The borders use a number of points along the same line to denote the borders of the pen, and drop off, as we use gradient ascent to pilot our Neato in our code. We also use a high coefficient for the BoB source and low coefficients for the obstacle sinks to ensure that the Neato doesn't get confused or distracted when following its gradient ascent path.

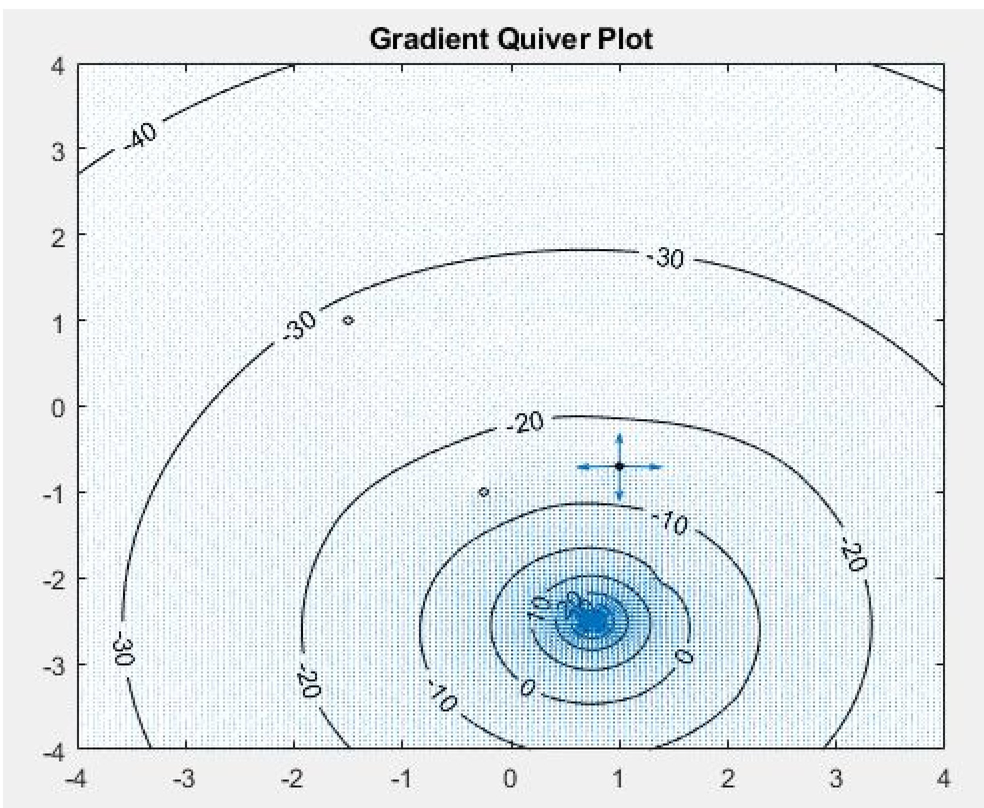




```
[x,y]=meshgrid(-4:0.05:4,-4:0.05:4);
Func = 0;

 % Adding in barrel
Func =  -20*log(sqrt((x-.75).^2 + (y+2.5).^2));

% Adding in obstacles
Func = Func + 1*log(sqrt((x-1).^2 + (y+.7).^2));
Func = Func + 1*log(sqrt((x-1.41).^2 + (y+2).^2));
Func = Func + 1*log(sqrt((x+.25).^2 + (y+1).^2)) ;

% Adding in borders
b = -1.5;
for a = -3.37:0.5:1
    Func = Func - .1*(log(sqrt((x-2.5).^2 + (y-a).^2))) - .1*(log(sqrt((x+1.5).^2 + (y-a).^2))) ...
        - .1*(log(sqrt((x-b).^2 + (y+3.37).^2))) - .1*(log(sqrt((x-b).^2 + (y-1).^2)));
    b = b+ 4/9;
end
```

3. Quiver pot of pot field

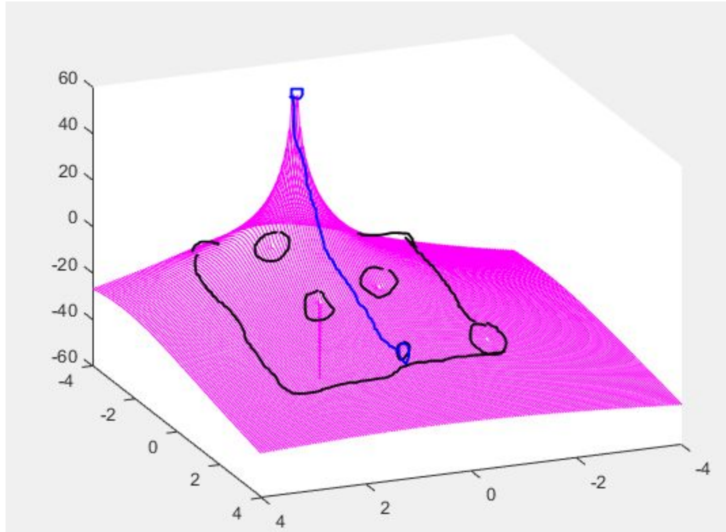
Below you can see a quiver plot of the vector field of the 3D function (function is shown above). The BoB is clearly represented as the large spike in the contour map. The obstacles, because of their reduced coefficients and intensity when they were defined in the function, are much harder to see in the contour plot. They are visible as the small black squares / dots in the



contour plot.

4. Grad Descent (Ascent) path

For reference, the function is plotted in magenta. The Sinks are circled in black, located at the positions of the obstacles in the 2D plane, and the single source is clearly visible, located at the BoB position. The theoretical path of gradient ascent, as should be followed by the Neato, is outlined in blue, and starts at the starting point of (0,0).



Navigating the Gauntlet

Introduction

Using the skill set in basic multivariable calculus, path planning, gradient descent, etc. that we've developed over the past few night assignments and projects working during the ROBO section of QEA, we'll be working to operate a virtual Neato through an obstacle course to reach a final endpoint.

During this project, we are to perform a number of basic tasks that add up and enable us to create a Neato that follows a path successfully. We must be able to read and work with Lidar scan data obtained from the Neato, transform the data between different coordinate systems, operate the Neato through a 3D field and have it find its way using a custom gradient descent movement algorithm.

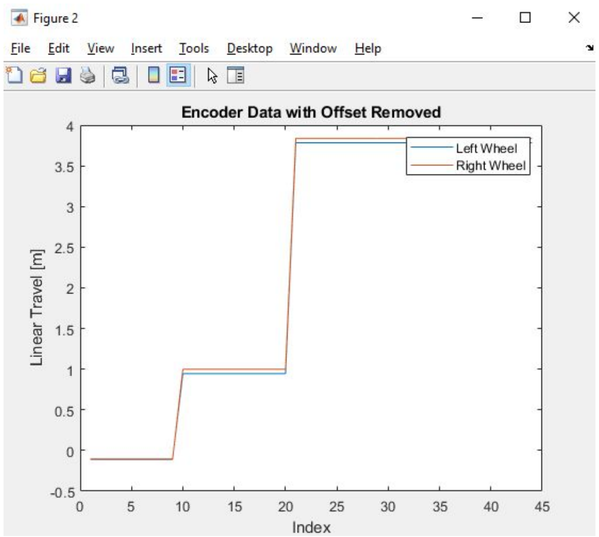
Experimental Data, Explanation

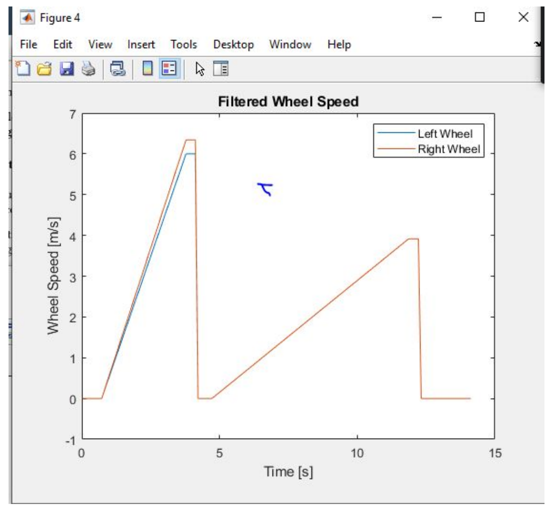
Time & distance Neato traveled for:
To complete the Gauntlet challenge, the Neato traveled continuously for around 12 seconds before making contact with the BoB.

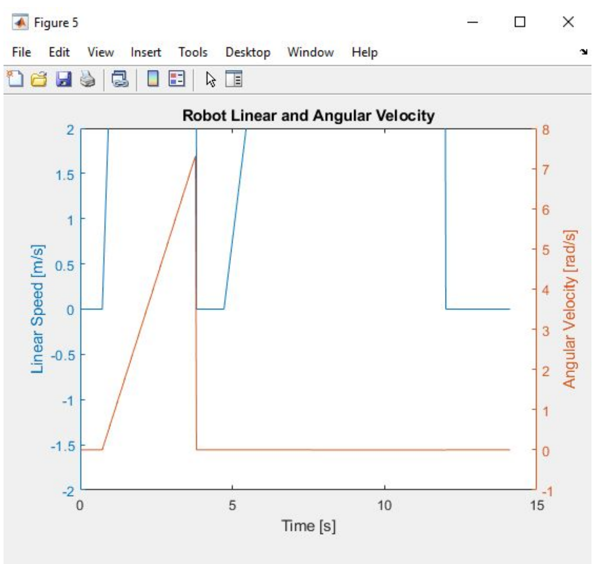
Below, you see our plot of the Neato's path, obtained with Matlab's collect_dataset_sim.m function. In order to obtain our Neato's path of travel plotted out, we obtained a dataset from collect_dataset_sim.m and used this data pipeline.
Wheel linear travel → Wheel velocities → linear speed, angular velocities → Robot Position & Tangent Vectors

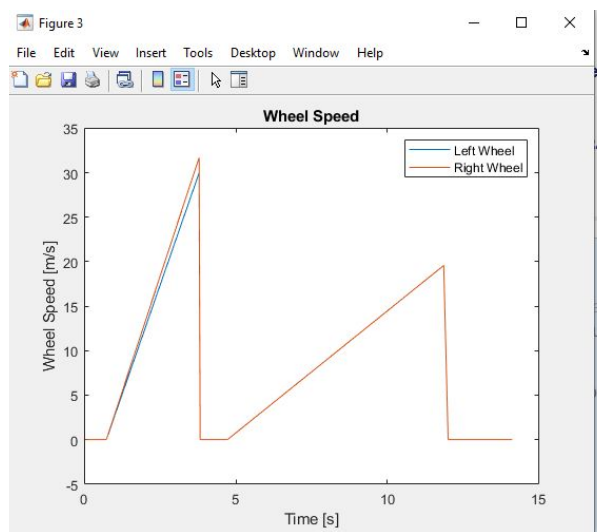
Adi R. & Patrick O.
QEA ROBO | 5.6.20

We were unable to collect data via the collect_dataset_sim.m function at a high enough frequency so the quality of the plots and graphed data is poor at best.
Below we've included some basic plots of the wheel speeds and velocities that we obtained from the encoder data.









Video

[Video linked here.]

Code attached in Canvas Assignment.