Into the Gauntlet:
Venturing Through the Darkness of Docker

Grant Miner and Max Stopyra

May 6, 2021

1 Introduction

We were faced with the challenge of navigating our robot through a series of obstacles to a specific object. Of the methods we explored, we chose to start simple, using the provided positions of the obstacles to generate a gradient field.

After many hours of attempting to represent the square obstacles as actual squares, and the "Barrel of Benevolence," the target object, as a circle, it became clear that generating these fields was creating infinities and other hard-to-handle values. The vast range of the data, which originally went from values on the order of 10^{-15} to values on the order of 10^{15}.

To combat this issue, the obstacles and Barrel were all represented instead as point sources and sinks. That decision greatly reduced the range of our data, and made the gradient and potential fields much easier to work with. However, the walls were not so easily solvable. They cannot be represented as point sources and sinks because they are so long. Instead, they must be expressed as a summation of several gradient and potential fields calculated at various points along them. These design choices yielded a field that did not have saddle points or other critical points along the path between the robot's starting position and its beloved Barrel of Benevolence.

There was still a decent amount of scaling work to be done. By delicately sweeping through various numbers of points to represent the walls and the scale factors for the walls and the Barrel, a potential and gradient field were produced that solved the issues with the first iteration's driving algorithm. The graphs below demonstrate the potential and gradient field of the first iteration to more accurately illustrate the Gauntlet (the obstacle course the robot navigated).

$$\text{Object Field} = -\ln\sqrt{(x+0.25)^2+(y+1)^2} - \ln\sqrt{(x-1)^2+(y+0.7)^2} - \ln\sqrt{(x-1.41)^2+(y+2)^2} \quad (1)$$

The next equation factors in the barrel, and scales it appropriately to remove the saddle point.

$$\text{Object Field} = \text{Potential Field} + 10\ln\sqrt{(x-0.75)^2+(y+2.5)^2} \quad (2)$$

The last step of developing the gradient is to add in the walls. By stepping along them using a step size of 0.1 meters, and by scaling the magnitude of their potential fields by 1/4.

$$\text{First Wall Set} = -0.25 * \sum_{a=-1.5}^{2.5}\left(\ln\sqrt{(x-a)^2+(y-1)^2} + \ln\sqrt{(x-a)^2+(y+3.37)^2}\right) \quad (3)$$

$$\text{Second Wall Set} = -0.25 * \sum_{a=-3.37}^{1}\left(\ln\sqrt{(x-2.5)^2+(y-a)^2} + \ln\sqrt{(x+1.5)^2+(y-a)^2}\right) \quad (4)$$

$$\text{Total Potential Field} = \text{Object Field} + \text{First Wall Set} + \text{Second Wall Set} \quad (5)$$

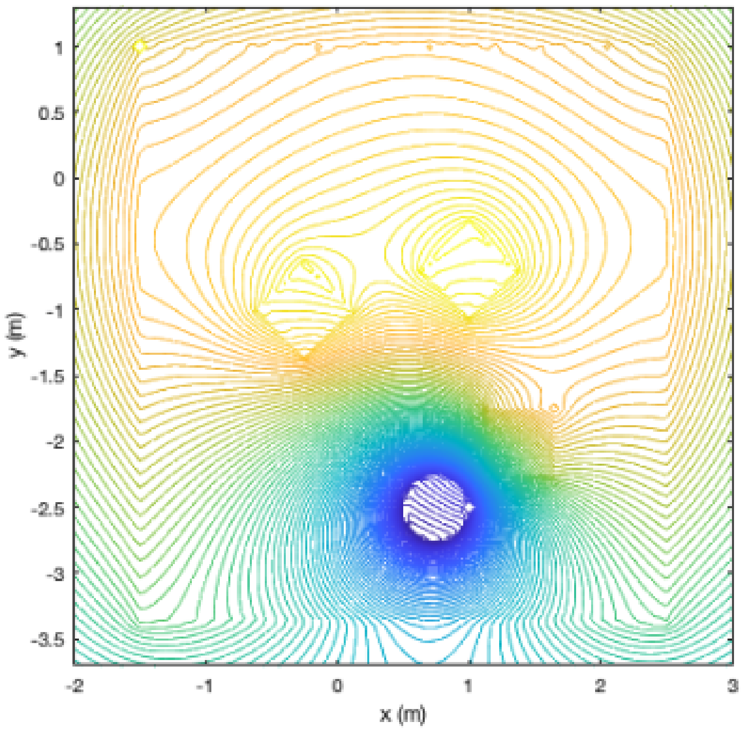

Figure 1: Complete Contour of Gradient Using Box Sources

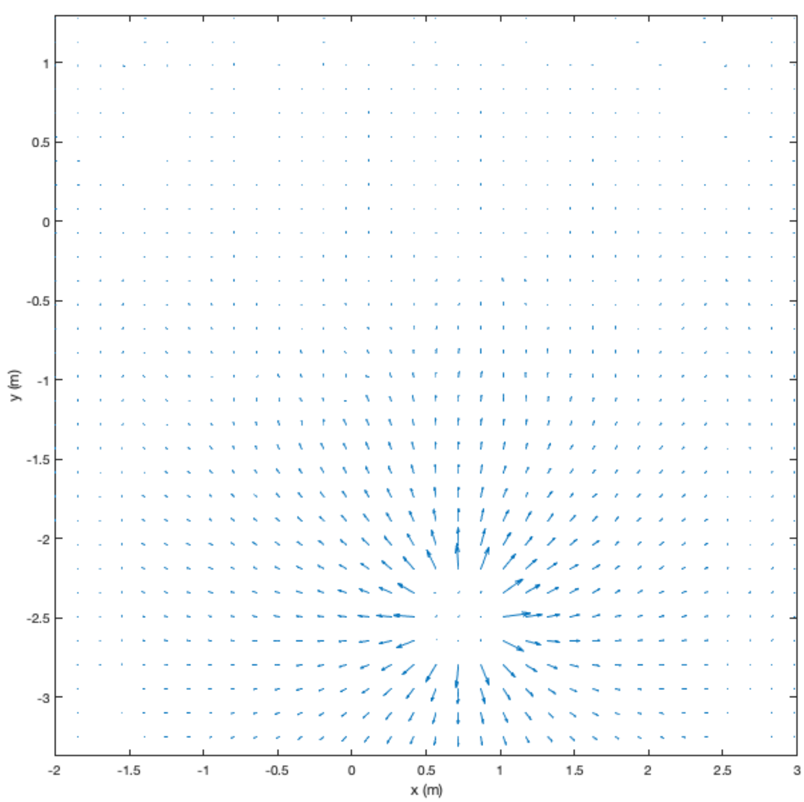

Figure 2: Complete Gradient Map of Gauntlet with Box Sources

2 Methods

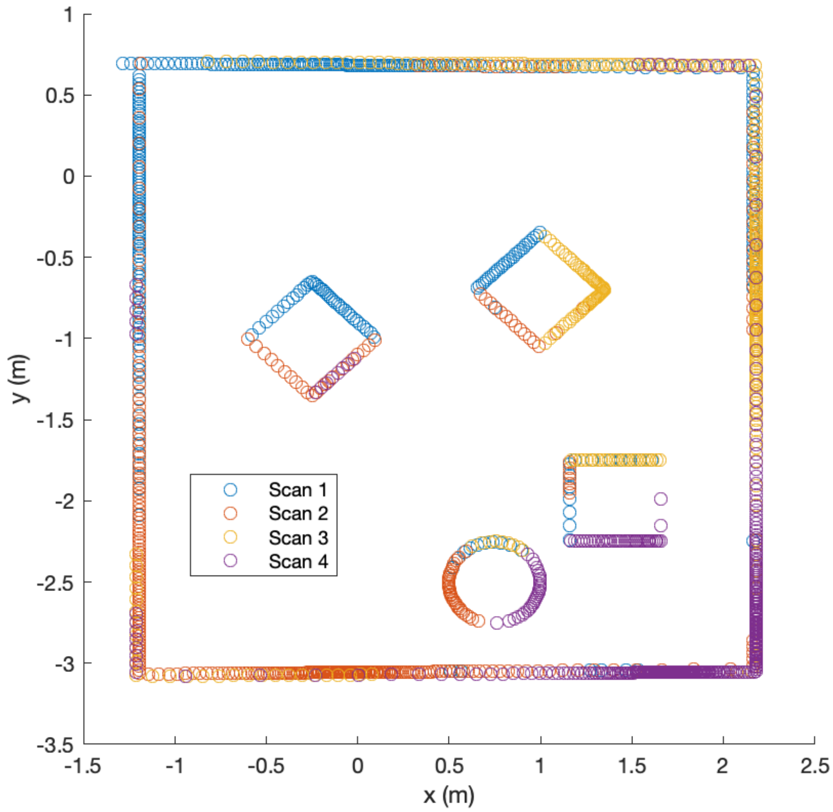

Figure 3: Superimposed LIDAR Scans from Various Points Around the Gauntlet

Figure 3 illustrates data collected by the Robot's LIDAR sensor at various places around the Gauntlet. The positions used were (0, 0), (0, -2.7), (1.7, -0.7), and (1.8, -2.8). Although our algorithm did not require these specific positions to be known, having a scaffolding of what the Gauntlet would look like helped us scale the potential field to aid the Neato on its journey.

Calculating its path was an entirely different challenge. Using code from a previous project that drove the robot up a gradient, we simply added a single negative sign and tweaked a few easy values. Instead of recording the path up the gradient, we multiplied the gradient at the robot's location by -1, to make it travel the opposite direction, or to make it descend the gradient. Along with some small experimentation with multipliers for the gradient, that one change was sufficient to guide the robot to the Barrel.

The code relies on having a basis for the gradient field, which was calculated using the equations above. Using simple vector addition, and cross products with a dash of trigonometry, we were able to have the Neato "step" forward (or, in this case, backwards) along a gradient vector, recalculate the gradient, and then step along that vector, until it reached its destination.

"You can take a horse to water, but you can't make him drink." This proverb is especially relevant here, where we point the Neato at the barrel, but cannot simply pick it up out of the simulation or cut its power when it gets to the Barrel. Instead, the Neato must know for itself when to stop. We accomplished this by adding a simple function that checked for the moment when any of the Neato's bump sensors was triggered.

Figure 4 shows the theoretical trajectory of the Neato from its starting point at (0, 0) to the side of the Barrel of Benevolence

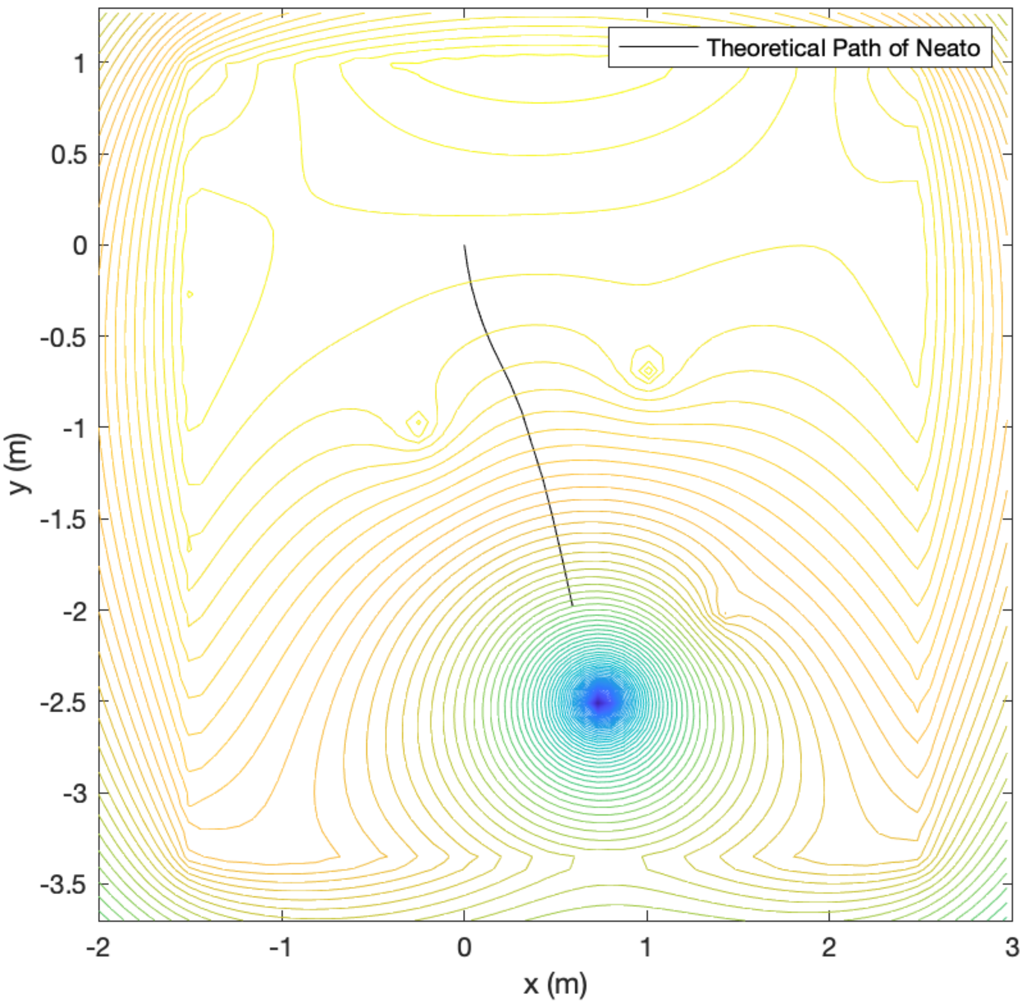

Figure 4: Theoretical Movement of Neato Through Gauntlet

3 Validation

In order to validate our data, we compared our experimental data for the path of the Neato, to the theoretical data we generated before navigating the Neato through the Gauntlet. When put on the same axes in Figure 5, we found that our experimental trip was extremely close to that of the theoretical trip.

We obtained the experimental data using a program that saved the distance traveled by each wheel of the Neato, along with the times at which those data were recorded. Using these datasets, we were able to create a vector that represents the approximate velocity at each time step, and also determine which direction, if any, the Neato was turning at a given time. By adding to the current position a vector that represented the displacement to the next time step, we put together a vector that included the position data of the center of the Neato. We also determined that the Neato took roughly 45.38 seconds to travel the total 2.258 meters on its winding path down the slope of the gradient to the object of its and our affection, the Barrel.

On the graph, we note what may seem to be a major issue in scaling with the potential map, or some sort of calculation error due to the gap between the scanned walls and the equations of the walls. This is thanks to the shape of the walls, which look like the cement barriers along roads in construction, which have much wider bases than tops. The scan data shows the locations of the walls at the Neato's LIDAR sensor's vertical coordinate, when the equations of the walls, and thus the edges in the potential field, represent the middle line of the walls from above.

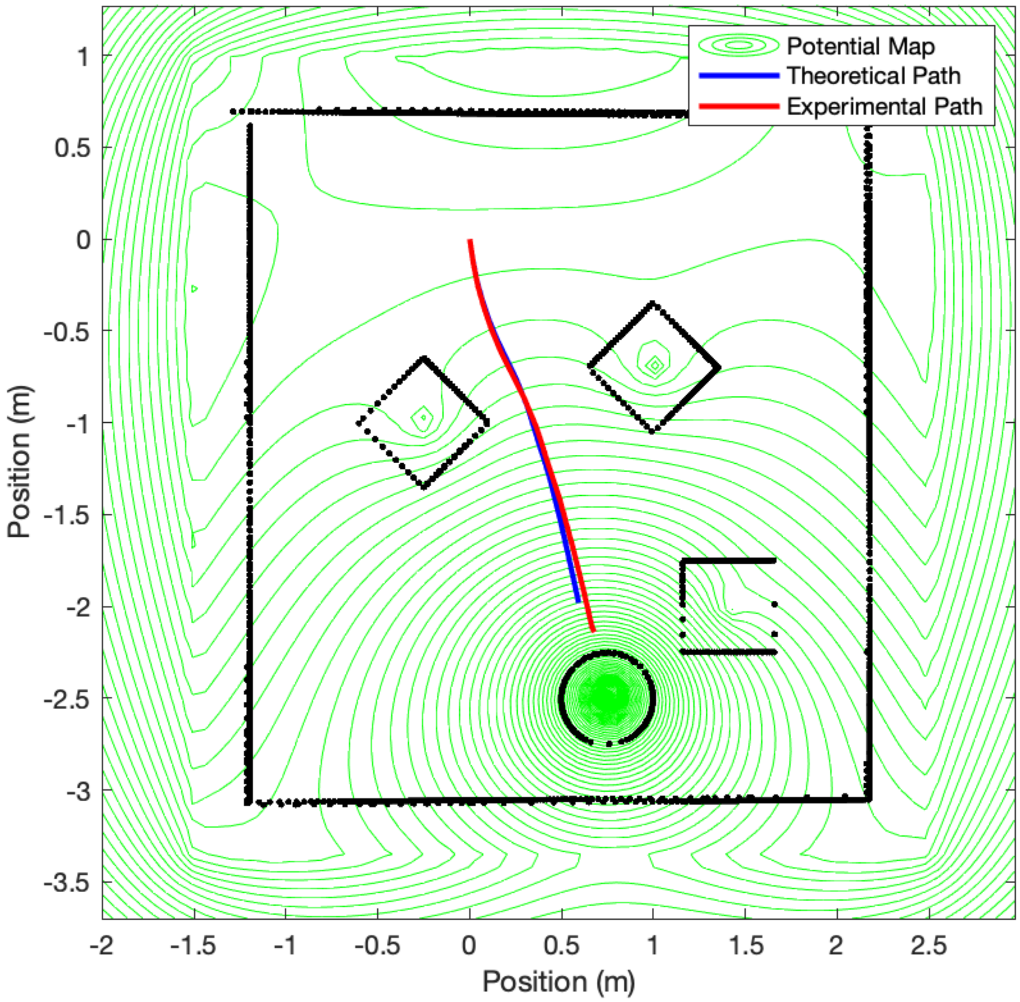

Figure 5: Experimental and Theoretical Paths Compared Over Gauntlet Map

4 Resources

Our code can be found here.
The video of our Neato's journey can be found here.
Special thanks to Paul Ruvolo for writing the software that runs the simulator we used for this exploration.