QEA Module 3: The Gauntlet and Triple Threat

Braden Oh, Kyle Emmi

1 Strategy

Upon approaching the Gauntlet challenge we first sought to identify the sequence of steps necessary to succeed. These steps included; identifying objects in the LIDAR environment, creating a pseudo-electrostatic field of point charges to attract and repel the robot, and implementing a control system that would move the robot accordingly. Beyond this, in order to complete the "Triple Threat" Challenge (autonomously completing the Bridge of Death, Mount Doom, and Gauntlet in sequence) we would need to implement additional systems to stop gradient ascent at the top of Mount Doom and to let the robot navigate off Mount Doom into the Gauntlet.

2 Process

Given our desire to complete Triple Threat, we decided that time was going to be a serious constraint and so decided to skip Gauntlet Challenges 1-4 and immediately attempt Challenge 5 (as that was going to be required to complete our ultimate goal).

2.1 Pre-Processing and "Object" Categorizing

We began by removing zero-radius values from the raw (polar) LIDAR data and then transformed each polar coordinate into a Cartesian coordinate. We then took advantage of the fact that the Cartesian points are inherently ordered by nature of how they were scanned, meaning that each point in the data array was always adjacent to the two points immediately to either side of it in space. Using this fact, we were able to identify individual "objects" in space by measuring the Euclidean distance between each point and the points adjacent to it and clustering the raw data points into "objects" by setting a threshold Euclidian distance between adjacent points (i.e. if two points are close together, cluster them into the same object; if two points are far apart, cluster them into different objects). The result of this process was a set of n objects with few to zero outliers.

2.2 Line/Circle Identification

Given these point-cluster "objects" we next needed to identify the bucket. For each object we picked 3 random points within the cluster and generated a circle between them. If the circle was significantly larger than the (known) bucket radius, or if a large percentage of points were not within a certain distance threshold from the circle, we discarded it. This process was repeated two more times for each object, resulting in each object being tested 3 separate times to determine whether or not it was a circle, and moreover, whether or not it could be the bucket.

For each object deemed not a circle, we performed linear regression and stored the starting and ending points of this best fit line, as well as its slopes and y-intercepts. With both circle and line identification algorithms implemented, we had created a full map of lines and circles.

2.3 Potential Field Generation

Rather than creating a single mathematical function to describe the potential field space, we decided to implement a pseudo-electrostatic repulsion field, creating a field of ”point charges” that would attract or repel the robot according to each point’s charge and distance from the NEATO. We began by generating a matrix of 250 by 250 zeros and assigned coordinates to each of those points using the maximum and minimum x and y values scanned. For each line feature we calculated a series of 100 discrete points in Cartesian space and mapped each of the 100 points to its closest corresponding cell in the field matrix, setting its ”charge” value to +1. For each circle, we calculated the coordinates of its center and found the closest cell in the field matrix to that center, setting the cell’s ”charge” value to -500. Finally, we set the charge of the robot at the origin to +1 so that it would be repelled by the walls and attracted to the bucket by the inverse-square equation

$$F_{mag} = \frac{-Q_{robot} * Q_{point}}{d^2}$$

where Q_{robot} is the charge of the robot (+1), Q_{point} is the charge of a given point in the field matrix, and d is the Euclidian distance between the robot and the point-charge.

To generate the net potential gradient vector, we summed the x and y components of every charge vector in the field space; we first calculated the magnitude of the force, F, generated on the robot by every point in the field space and then determined the angle θ between that point and the robot. The x and y components were given by

$$F_x = F\cos\theta$$
$$F_y = F\sin\theta$$

The net force vector was given by a simple summation of the vector components for each force vector F_i:

$$\Sigma F_x = \Sigma F_i \cos\theta$$
$$\Sigma F_y = \Sigma F_i \sin\theta$$

The turning angle α for the robot to follow the gradient is given by

$$\alpha = \arctan\frac{\Sigma F_y}{\Sigma F_x}$$

After the turn, the robot moved forward a set distance and repeated the process until it sensed an object with its bump sensor.

2.4 Triple Threat

The two problems remaining were those of the Triple Threat. To stop our summit of Mount Doom, we simply set the robot to stop its gradient ascent program as soon as it deemed itself within a certain threshold of being flat. Getting off of Mount Doom and into the Gauntlet, we took advantage of the fact that the direction of the Gauntlet was parallel to the wall on the NEATO’s right side; we wrote a simple three-state bang-bang line following algorithm (within a certain threshold from the wall/too far from wall/too close to wall) performing a geometric calculation with two LIDAR points to determine the absolute horizontal distance from the wall. Once these problems were solved, we successfully completed the Triple Threat.

3 Experimental Results

Our robot, Wumbo Joe, generally completed Challenge 5 (wherein the bucket and obstacle coordinates are unknown) in 1-2 minutes. The shortest time it found the bucket was 30 seconds, but Wumbo Joe would generally reach the bucket at around 1 min 30 seconds. We defined success as the act of touching the bucket, so the valuable metric to us was the success rate, rather than the speed. Once our system was tuned, Wumbo Joe touched the bucket every time we tested it in the Gauntlet and struck obstacles less than 20% of the time.

4 The Proof