**Path Planning**

The robot navigates the course using the A\* path planning algorithm. The course is sectioned into 2’ x 2’ grid spaces which are prepopulated with the course walls. The A\* algorithm plans the optimal path from the current position of the robot to the target waypoint, and the robot follows this path through the course. The implemented A\* algorithm uses Manhattan distance to calculate the heuristic and does not allow diagonal movement to prevent the robot from running into obstacles. Furthermore, an extra movement cost for turns is implemented to minimize the number of turns in the planned path. As the robot moves through the course it uses LADAR measurements to detect new obstacles. If a new obstacle is detected, then the robot calculates a new path using the updated obstacle map. A new path is also calculated if the robot diverges from the current path and collects a golf ball. Finally, the robot calculates a new path to the next target waypoint once the current target waypoint is reached. This process repeats until the robot has reached all five waypoints in order to complete the course.

**Obstacle Detection**

The robot detects obstacles using LADAR measurements. As the robot navigates the course, LADAR measurements are recorded, and the obstacle map is updated. Extra distance is added to every LADAR measurement so that every data point is pushed into a definite grid space away from the robot. Every LADAR sweep, the detection algorithm sums up the number of LADAR data points for each grid space and assigns a hit to a grid space if the sum is greater than a threshold. If the number of hits for a grid space becomes greater than a threshold overtime, then the obstacle is detected and the obstacle map is updated. This two-layered system adds robustness to the obstacle detection and minimizes false obstacle identifications.

**LabVIEW GUI**

A LabVIEW Virtual Instrument was developed to provide visual feedback of the robot’s navigation through the course. The VI displays the course map with walls and grid spaces, the robot’s current position with a trail of past positions, detected obstacles, and collected balls with their respective position and color. This functionality is extremely valuable for debugging, especially for obstacle detection, because it provides visual data on what the robot was seeing. The VI communicates with the robot via a wireless TCP/IP connection, so real-time data is able to be viewed while the robot navigates the course.



