CSCE 274 Project 4 – Team 05

Chris McKinney – Nicholas Wrobel – Josh Benton

This project required us to program our robot such that it would find and dock on the charging station on its own. To do this, we have two stages. The first stage simply is a repeat of project 3, where we use our PID controller to follow a wall on the right hand side and drive forward. The second stage begins when we first begin to detect the force field.

To detect the force field, we maintain 3 running averages. The dock has 3 zones: a red buoy, a green buoy, and a force field. The robot's sensors can give us a binary value for each zone that can tell us whether we are in that zone or not. However, because the robot's sensors are noisy, we average the values for each of the zones. This gives us a smoothing effect and lets us know a probability that we are in that zone. If the probability that we are in a particular zone is greater than some particular threshold, then we can determine with some confidence that we are within that region. This threshold was chosen to favor false positives (thinking we are in a region) than false negatives (thinking we aren't when we actually are). The averages are calculated in the updateIR() method. If any of these averages are above the threshold, we are now in the 'docking' state, which our algorithm has a flag for.

Once in the docking state, we first need to determine if we are coming into the dock directly head on, or from the side, so that we can perform necessary adjustments to get the robot to dock correctly. If the red buoy is triggered, then we know that we are coming at the dock from the front. We then turn 90 degrees left and drive forward. When we enter the green buoy, we drive forward an extra robot radius, and turn 90 degrees back towards the dock. We then we know that our robot is lined up head on with the docking station.

We then are in the final docking stage, and we drive forward, making course corrections as we go (when we exit one of the buoy zones). When the bump sensor is activated, we know we are on the dock. Once we are stationed inside of the dock, although the robot is there, it may not be aligned correctly to receive the charge. We pull the charge sensor value. If the charge is being received, then we are docked and done.

If the charge is not yet received, to line the robot up properly, we simply rotate the robot in place while on the docking station, constantly checking the 'charging status' from the packet until it is true. This rotation is done in the jimmy() method, which rotates the robot back and forth, increasing the angle every time the method is called. This continues in a loop until our functions that read the charging state return true. Once we know we are charging, we can stop.

To help us develop and debug, we also use the LEDs to detect which dock zone the robot is in.

Evaluation

Our robot is able to find the dock with a fairly high degree of accuracy. However, even with our jimmy() function at the end, sometimes our robot fails to receive the charging signal when it is on the dock. For some robots, however, this works fine.

Allocation of Effort

Chris wrote the code with the assistance of Josh. Nick wrote the paper.