Behaviors

Teleop

Established a node that reads in key presses from the machine running the node. On key press, the movement mapped to from the key will occur until a different input is given. ‘w’ moves the robot forward, ‘s’ moves it backwards, ‘a’ causes it to rotate ccw, ‘d’ causes it to rotate cw, and any other input key will cause it to stop all movement.

Drive Square

We developed a node that will begin by turning 90° and then go forward 1 meter. It repeats this behavior three more times, until it has completed driving in a square. Implementing this relatively simple behavior allowed us to familiarize ourselves with publishing information to the neato.

Person Follower

Established a node that subscribes to the LaserScan data published to /scan, attempting to locate a person using the distance of the person and the angle range that the person is in, relative to the robot. The robot begins by assuming the closest object identified as the person to follow, then continues to identify that specific object as the person by using a set of distance and angle tolerances to ignore other potential objects. To find a person, all of the points from LaserScan are iterated over and the distances are collected into a queue and a running average which are both operated on to determine whether the distances qualify for the identification of a single object. If it is tracking a person that it has already identified, only a certain range of angles are operated on for object identification. Simultaneously, the robot is constantly attempting to move towards whatever values have been stored as the object by turning until the object is within a certain range of the robot, and once within that range, moving forward so long as the object remains in the angle range.

Object Avoidance

The object avoidance node examines the 180° in front of the robot (i.e. from 270°-360° and 0°-90°). Using information obtained from LaserScan, the robot will turn away from the object it detected that it is closest to. It will then go forward in that direction. It continues to follow this pattern of scanning for objects, turning away from the closest object, and driving in that direction.

Notably, this behavior would not handle being entirely surrounded in the front by objects. For example, if the neato were contained in a circular barrier with one exit behind it, it would not find the gap behind it. It would simply find that it was completely surrounded in front, turn, and then

drive into the wall. We did not have time to improve on this behavior much, as it would have required a complete overhaul of the algorithm we developed.

Wall Follower

The wall follower uses the Hough Transform functionality of OpenCV to detect a straight line within the data obtained from LaserScan. It marks the longest of any such identified line as a wall.

State Controller

The state controller combines the functionality of teleop and person following, switching between the two on a time interval basis. Every 10 seconds after the last pass through the run() function of the state controller node, the robot will switch to the other function.

Code Structure

Our code generally followed an object-oriented style in which every behavior was defined in its own class, which in turn instantiated a node to execute the behavior. Each class had a defined run function, and a listener callback function if the relevant behavior required data that was being published to a topic.

Challenges and Takeaways

The process of visualization was a challenge throughout this project. Particularly in the wall follower portion, we implemented behavior and wall detection before actually visualizing the wall we had detected. This prevented us from learning about issues within our code earlier and made visualization into a more complex problem because we had not structured our results to work with it. With future robotic programming projects, we should be more proactive with debugging tools such as visualization. They are not only easier to integrate early on, they allow us to better recognize potential problems as they arise.