ENPM690-HW3
Hitesh Kyatham

Autonomous Behavior

The autonomous behavior I implemented is **obstacle avoidance**, as I had already worked extensively on it during my planning course. I reused most of my previous code but opted to use **LiDAR** as the primary sensor for obstacle detection and avoidance. Additionally, my implementation includes **path planning using A***, which can be triggered using the `Astar_maze_navigation.py` launch file. This path planning approach utilizes odometry sensor callbacks to guide the robot from one waypoint to the next. I chose LiDAR because it provides a straightforward way to measure obstacle distances, allowing the robot to effectively avoid collisions.

Tunable Parameter: Velocity

I introduced **velocity** as the tunable parameter by creating a dedicated ROS2 node. This node functions similarly to **teleoperation**, where pressing the **'w'** key increases speed and the **'s'** key decreases it. This allows real-time manual adjustment of the robot's velocity during operation.

Observations on Tunable Parameter

Through testing, I observed that:

- The robot **stops completely at a velocity of 0.0** (expected behavior).
- At **0.8 m/s, the robot frequently collides with obstacles**.
- The robot functions **safely within the speed range of 0.0 to 0.7 m/s** without losing its obstacle avoidance capability.

I hypothesize that at higher speeds (≥ 0.8 m/s), the robot **does not have enough time to process LiDAR data and react** before a collision occurs. To mitigate this issue, I could **increase the LiDAR detection range**, allowing the robot to anticipate obstacles earlier and make timely avoidance maneuvers.