Lab 4 Report - Particle Filter Localization and Mapping for Lidars

Team: Aditya Jain, Geet Patel, Sugumar Prabhakaran

1 Introduction

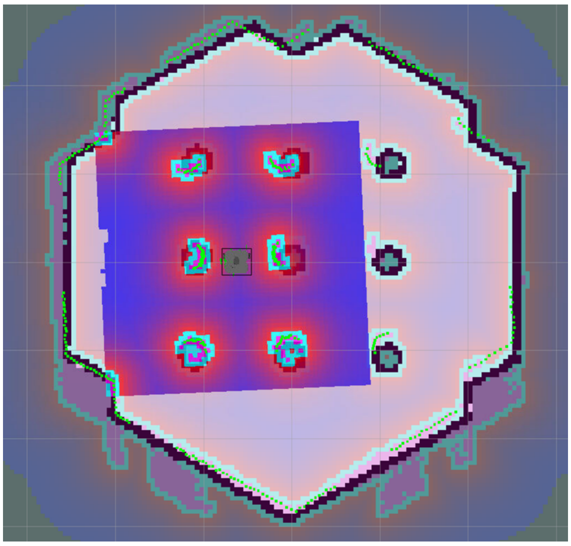
The purpose of this final lab is to explore the the use of different particle filters to support localization and mapping of the Turtlebot3 Waffle Pi. In particular, the built-in ROS packages Adaptable Monte-Carlo Localization (AMCL) and gmapping were used. In task 1, we used AMCL to localize the robot in the turtlebot3_world environment using the lidar data. For task 2 and 3, we used the gmapping package to create maps of the turtlebot3and willow garage_world environments.

2 Task 1: Lidar Localization in turtlebot3_world

In task 1, we used the AMCL package to localize the turtlebot3 in the turtlebot3_world environment and experimented with different parameter values to understand the limits of the AMCL approach.

- Figure 1 shows a picture of the AMCL rviz window with a converged filter, obtained using the default parameters.

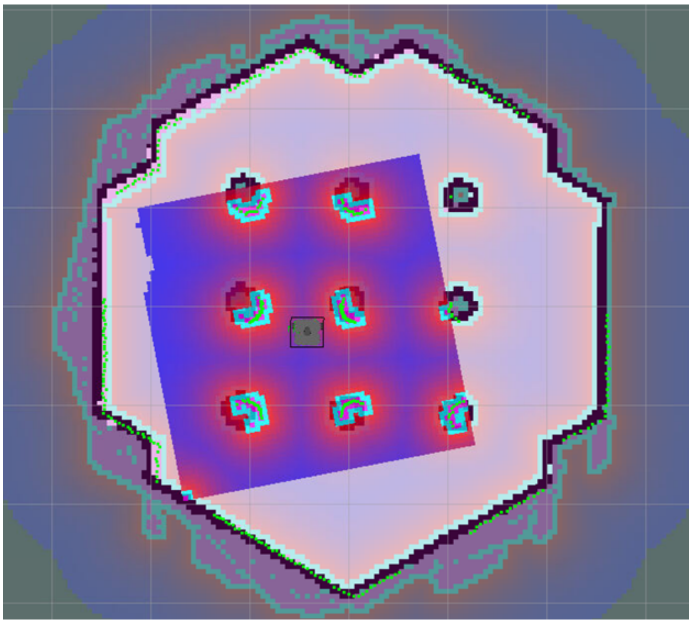
Figure 1: AMCL localization in turtlebot3_world



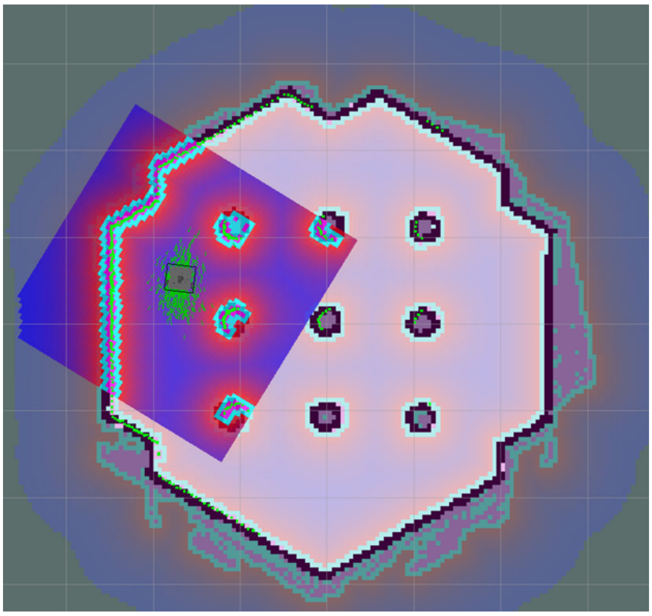
- **Minimum number of particles for successful localization:** Reducing the number of particles reduced the accuracy of the localization results with respect to the actual robot's pose. From experimentation, we found that AMCL needs a minimum of 100 particles for successful localization. Figure 2a shows the result obtained with min_particles set to 100 and max_particles to 120. Moreover, these results were proven repeatable as this figure was obtained after re-initializing the pose estimate of the robot.

- **Maximum odom_alpha parameter before localization failed:** Increasing odom_alpha parameters, increased the uncertainty in our localization result. Our experimentation found that odom_alpha value above 1.8 led to unsuccessful localization, where the particles will not converge to a single cluster of pose. The result for all five odom_alpha parameters being set to 1.8 are shown in figure 2b.

Figure 2: AMCL task results



(a) Converged filter for min_particles = 100 and max_particles = 120



(b) Fairly converged filter for odom_alpha parameters = 1.8

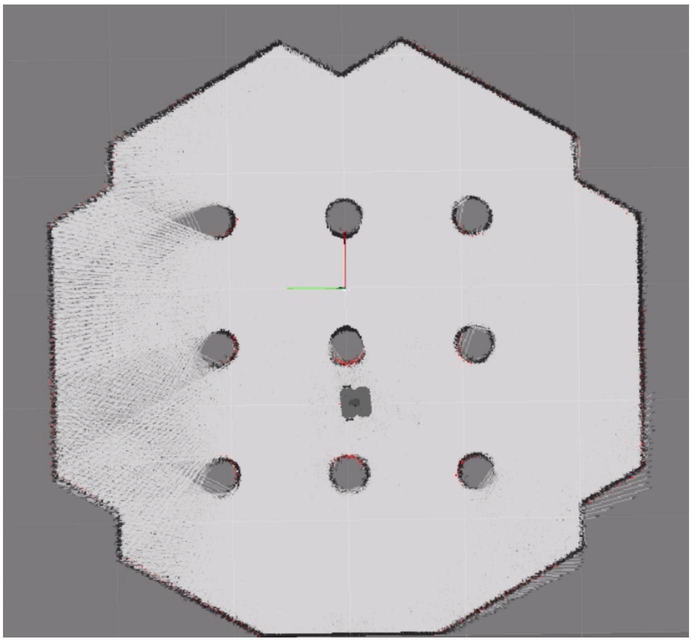
3 Task 2: Mapping turtlebot3_world

In task 2, we used the built-in ROS gmapping SLAM package to generate a map of the turtlebot3_world environment and then compared that with the results from the lidar and odometry based map generated in lab 3:

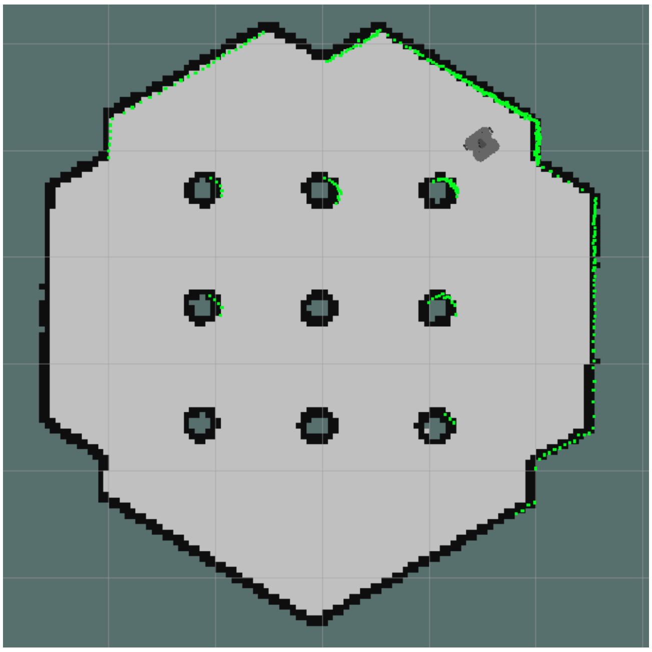
- Figure 3a shows the map built in lab 3 using only wheel-odometry and lidar data, while figure 3b is the map built in lab 4 using the ROS gmapping laser-based SLAM package.

- **Comparison:** In lab 3, the approach used was to localize was entirely based on odometry, which accumulates error as the robot continues to move further away from it's initial start point. From experimentation, we found that a constant linear and angular velocity needed to be maintained to reduce error and generate a good map. A sudden change in velocity led to significant distortion of the map, likely as a result of the imperfect motion model. Furthermore, the optimal map was produced using the shortest possible path in order to reduce the accumulation of odometry error. In lab 4, the ROS gmapping package uses the Rao-Blackwellized particle filter to localize and so the resulting map is more certain. With the superior localization, the mapping process was invariant to changes in velocity. Sudden changes in the linear velocity can distort the map momentarily, but are then corrected using lidar-scan matching.

Figure 3: Occupancy grid map built for turtlebot3_world in lab 3 and lab 4 respectively



(a) Lab 3: turtlebot3_world using Lidar localization



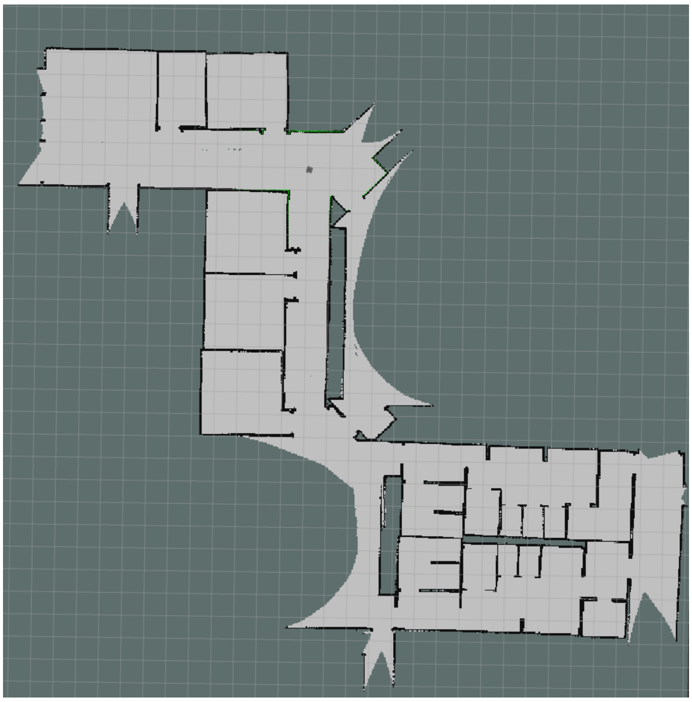
(b) Lab 4: turtlebot3_world using Gmapping

4 Task 3: Mapping willow garage_world

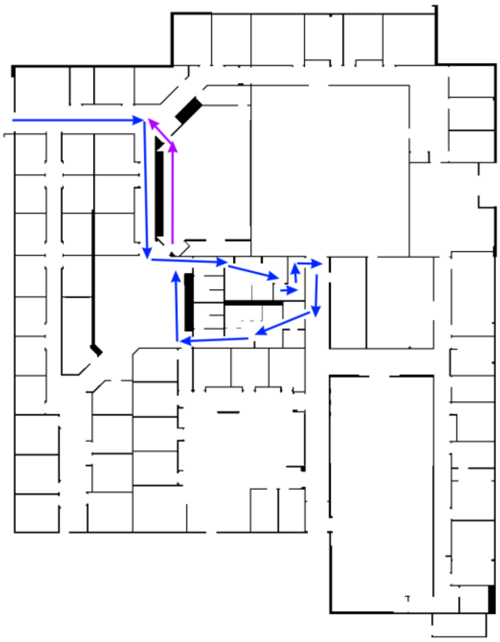
In Task 3, we used the same built-in ROS gmapping SLAM package to generate a map of a section of the willow garage_world map from lab 2:

- Figure 4a is the final map generated using the gmapping package for the section of the willow garage map outlined in blue and purple in figure 4b.
- As we can see from the two maps in figure 4, the ROS gmapping package was able to fairly accurately map the section of the willow garage_world environment. There is a slight skew at the bottom right section of the map, which is reasonable since this is the furthest area from the start point and is likely the result of accumlated odometry errors.
- **Issues:** We initially attempted to generate the map using the default gmapping parameters, which were set at particles = 100, max_laser_range = 3.0, and the four odometry error terms: srr, str = 0.1, srt, stt = 0.2. This resulted in a less accurate map that was unable to successfully close the loop indicated by the purple path in figure 4b. This most likely occurred because this particular path (purple) consists of a straight wall without major distinguishing features, making the localization more reliant on the odometry. Since odometry errors accumulate, the loop closure resulted in the robot thinking it was further "south" of where it actually was and proceeded to move through walls. This was rectified by improved exploration of the region where the purple arrow meets the blue arrow before proceeding down the blue corridor. Further, the following parameters were adjusted to improve the localization: particles = 400, max_laser_range = 6.0, srr, str, srt, stt = 0.1.

Figure 4: Occupancy grid for willow garage_world using ROS gmapping package



(a) Lab 4: willow garage_world using gmapping



(b) Lab 2: willow garage_world map