LABORATORY IV

Particle Filter Localization and Mapping for Lidars

ROB521 Introduction to Robotics
Spring 2023

1 Introduction

This is the fourth laboratory exercise of ROB521-Autonomous Mobile Robotics. The course will encompass a total of four labs, all of which are to be completed in the scheduled practicum periods. Each of the four labs will grow in complexity and demonstrate important robotic concepts from the lectures. Our robot of choice: *Turtlebot 3 Waffle Pi* running the operating system *ROS*.

2 Objective

The objective of this lab is to explore the properties of particle filters in the context of localization and mapping. The two ros packages that you will be using today are Adaptable Monte-Carlo Localization (AMCL) and Gmapping. Both algorithms use a particle filter to estimate the state of the robot in a occupancy map. AMCL estimates the pose of the robot by fusing lidar scan match estimates with the robot inputs.[1] Gmapping uses a Rao-Blackwellized Particle Filter (RBPF) to model the probability of the map and robot trajectory given the lidar measurements and robot inputs.[2] Papers describing the RBPF used in gmapping can be found in Section 6. In this lab you will:

- *Use particle filtering for localization*
- *Investigate potential issues with particle filtering*
- *Remap the environment from Lab 3 and discuss the differences between your new map and the previous*
- *Map a portion of the environment from Lab 2 and discuss the challenges that you encountered while mapping.*
- *Map the real-life environment in Myhal and discuss the differences between simulation and real-life deployment.*

To understand the effects of the various parameters that you will be tuning for AMCL and Gmapping, you should read the original AMCL and Gmapping papers. These papers are Monte Carlo Localization: Efficient Position Estimation for Mobile Robots (`https://www.ri.cmu.edu/pub_files/pub1/fox_dieter_1999_1/fox_dieter_1999_1.pdf`) and Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling (`http://www2.informatik.uni-freiburg.de/~stachnis/pdf/grisetti05icra.pdf`), respectively. Additionally, these papers will help you understand how AMCL and Gmapping operate, which will assist in answering some of the deliverables.

[1] The code for AMCL is available at `http://wiki.ros.org/amcl` and `https://github.com/ros-planning/navigation`.

[2] The gmapping code is available at `https://github.com/ros-perception/slam_gmapping` and `https://github.com/OpenSLAM-org/openslam_gmapping`.

2.1 Lab Deliverables

You will be required to demonstrate functionality during in-person lab sessions and submitting a report.

In this (and future) labs, look for deliverables that will be in **bolded red text** throughout the manual, and follow the instructions for each. Additionally, a summary of the deliverables and their mark distribution is at the end of the document. **In this lab, you will submit a pdf lab report and a video demonstrating the deliverables**.

In your lab report, for each deliverable, include a short description of what your code is doing, and what the robot does as a result of running the code. Provide context that will allow the TA's to understand what your robot is doing, in order for them to properly gauge your success. Importantly, state whether you were able to complete the deliverable. If not, explain why you weren't able to complete the task. Do not write more than a paragraph for each deliverable.

Finally, include a copy of your code for the TA's to look over. The code itself will not be marked, but it must be presented to demonstrate that each group has independently written their solution. If the code does not look complete, TA's will have access to all of the computers, and will be able to run the code themselves to confirm that the deliverables have been completed.

3 Assignment

Task 1: Lidar Localization in turtlebot3_world

In *Lecture 19 - Filter-based Localization - Particle Filters*, the use of particle filters for localization was discussed. For this task, you will explore using the ros AMCL package.

First, use `roslaunch rob521_lab3 turtlebot3_world.launch` to start the Gazebo simulation for this task. Next, use `roslaunch turtlebot3_navigation turtlebot3_navigation.launch map_file:=$HOME/catkin_ws/src/turtlebot3/turtlebot3_navigation/maps/map.yaml` to launch AMCL. In the launched Rviz window you will need to provide a rough initial pose estimate for AMCL. This can be achieved by clicking the 2D pose estimate button in the rviz menu - highlighted by a red box in Figure 1. Click on the map where the robot is located and, while holding the mouse button down, drag the cursor the direction that the robot is facing. The robot should be roughly aligned with the map now. You should drive the robot around the map using `rosrun turtlebot3_teleop turtlebot3_teleop_key.launch`. If you have followed these steps correctly, then the robot should converge to the correct position on the map.

Now that you have AMCL running and converged, you should try changing the particle filter parameters to see when the filter diverges from reality. Start the parameter reconfiguration node using `rosrun rqt_reconfigure rqt_reconfigure`. A window will open that allows you to modify the particle parameters without restarting the previous ros nodes! Depending on the parameter changed, you may need to provide another rough initial pose estimate.

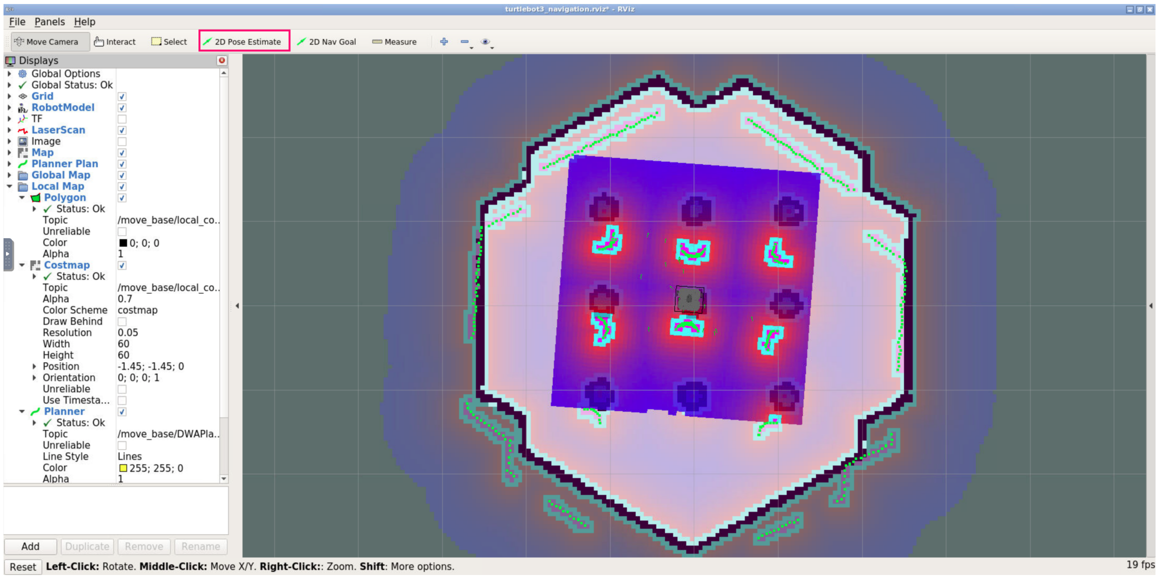

Figure 1: The rviz window that appears after launching AMCL. A red box surrounds the 2D Pose Estimate Button.

The first set of parameters that you should change are max_particles and min_particles - make sure to record the default parameter values. You should decrease the max_particles and min_particles parameters and drive the robot around the environment. What is the minimum number of particles required for successful localization? Return the max_particle and min_particles to their original settings. The second set of parameters that you should change are odom_alpha parameters. These parameters scale the uncertainty of the robot motion model. You should increase the odom_alpha parameters and drive around in the environment. What effect did that have on the localization solution? What is the maximum odom_alpha values that you can achieve and still have the particle filter converge? When you are finished with this task, stop all of the nodes except the Gazebo simulator and turtlebot teleoperation node.

In your report, you must:

- present a picture of the AMCL rviz window with a converged filter.
- what was the minimum number of particles for successful localization? Are your results repeatable (i.e if you give the robot a new initial pose estimate does the particle filter converge)?
- what were the maximum odom_alpha parameters before localization failed?

Task 2: Mapping turtlebot3_world

For Lab 3, you performed mapping using purely wheel odometry measurements. In that lab, you found that wheel odometry measurement noise caused the map to blur. In this task, you will map the `turtlebot3_world` from Lab 3 using Gmapping. Gmapping performs a lidar scan matching step to correct the robot motion errors. You can launch the gmapping node using `roslaunch turtlebot3_slam turtlebot3_slam.launch`. After launching the gmapping node, use the teleoperation node to map the entire environment. How does this new map compare to the one you made in Lab 3?

Similar to AMCL, Gmapping has a variety of parameters that control the particle filter. The parameters file can be found at `/catkin_ws/src/turtlebot3/turtlebot3_slam/config/gmapping_params.yaml`.[3] If you wish to change these parameters, then you will need to relaunch `turtlebot_slam.launch` for each time you change the parameters. Once you have completed this task, return the parameters back to their original settings and shut down all ROS windows.

In your report, you must:

- present a picture of your Lab 3 map and your new map. Figure 2 is a solution map for Lab 3. You may use this map if your group has not completed Lab 3.
- discuss how the localization component of gmapping improved your new map compared to the map from Lab 3.

Task 3: Mapping willowgarge_world

In Lab 2, the map of `willowgarge_world` was provided for path planning. Now, you will map a portion of `willowgarge_world` yourself. You will likely need to change the Gmapping parameters for this task.

(HINT: Try increasing the number of particles and maxUrange.)

(HINT: Try decreasing srr, srt, str, and stt.)

(HINT: Explore some of the side rooms along the path to increase the number of loop closure opportunities.)

(HINT: Don't drive too fast.)

First, you will launch the simulator using `roslaunch rob521_lab2 willowgarge_world.launch`. Next, you will use the commands from Task 2 to start Gmapping and teleoperation nodes.

Given the size of the map from Lab 2, we **do not** expect you to map the entire environment. However, we would like you to map a two loops. The first path to map is the blue path shown in Figure 3. This path will demonstrate the benefits and challenges of performing loop closure in SLAM. After completing the blue path, you should follow the purple path. Gmapping will likely fail while you drive this path. Why do you think that Gmapping fails?

For this section, the deliverables are to:

- present your map and the map provided for Lab 2.

[3]Descriptions of the parameters can be found at: `http://wiki.ros.org/gmapping` and `https://emanual.robotis.com/docs/en/platform/turtlebot3/slam/`.

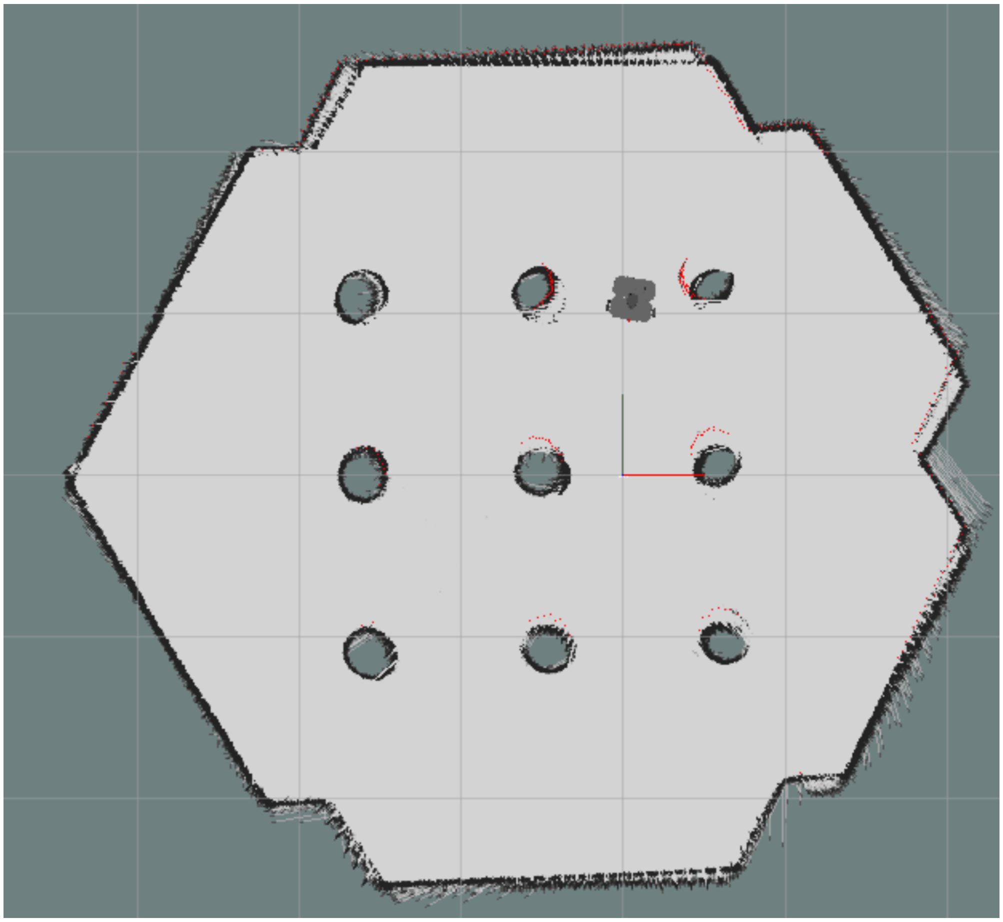

Figure 2: Lab 3 Lidar map solution.

- compare your map to the corresponding part of the map from Lab 2.
- describe any mapping issues you encountered and discuss their cause.
- describe any issues that you encountered when you attempted to close a loop. If the loop closed perfectly, describe issues that you expect you could have encountered.

Task 4: Mapping Myhal 570 Testing Environment

Lastly, you can test out the algorithms during the in-person lab session to map the set-up environment in the Myhal 570 room using the same commands as Task 2.

- present your map of the Myhal testing environment.
- describe any mapping issues you encountered and discuss their cause. Discuss any differences between the real-life deployment and simulation.

4 Concluding Remarks

This lab exposes you to the challenges of mapping environments using a single sensor and noisy wheel odometry. It hopefully solidified the benefits of performing SLAM and the limitations of lidar-based localization. Congratulations on finishing the final lab of ROB521!

5 Deliverable Summary

The deliverables are listed below for a total of 10 points.

- (/4) Report Part 1: turtlebot3_world localization
  - present a picture of the AMCL rviz window with a converged filter.
  - what was the minimum number of particles for successful localization? Are your results repeatable (i.e if you give the robot a new initial pose estimate does the particle filter converge)?
  - what were the maximum odom_alpha parameters before localization failed?
- (/2) Report Part 2: turtlebot3_world mapping
  - present a picture of your Lab 3 map and your new map. Figure 2 is a solution map for Lab 3. You may use this map if your group has not completed Lab 3.
  - discuss how the localization component of gmapping improved your new map compared to the map from Lab 3.
- (/4) Report Part 3: willowgarge_world mapping
  - present your map and the map provided for Lab 2.
  - compare your map to the corresponding part of the map from Lab 2.
  - describe any mapping issues you encountered and discuss their cause.
  - describe any issues that you encountered when you attempted to close a loop. If the loop closed perfectly, describe issues that you expect you could have encountered
- (/2) Report Part 4: Myhal environment mapping
  - present your map of Myhal.
  - describe any mapping issues you encountered and discuss their cause. Discuss any differences between the real-life deployment and simulation.

6 Additional Resources

1. Introduction to ROS, ROB521 Handout, 2019.
2. Robotis e-Manual, http://emanual.robotis.com/docs/en/platform/turtlebot3/overview/.
3. “SSH: Remote control your Raspberry Pi,” The MagPi Magazine, https://www.raspberrypi.org/magpi/ssh-remote-control-raspberry-pi/.
4. Official ROS Website, https://www.ros.org/.
5. ROS Wiki, http://wiki.ros.org/ROS/Introduction.
6. Useful tutorials to run through from ROS Wiki, http://wiki.ros.org/ROS/Tutorials.
7. ROS Robot Programming Textbook, by the TurtleBot3 developers, http://www.pishrobot.com/wp-content/uploads/2018/02/ROS-robot-programming-book-by-turtlebo3-developers-EN.pdf.
8. Monte Carlo Localization: Efficient Position Estimation for Mobile Robots, https://www.ri.cmu.edu/pub_files/pub1/fox_dieter_1999_1/fox_dieter_1999_1.pdf
9. Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters, http://www2.informatik.uni-freiburg.de/~stachnis/pdf/grisetti07tro.pdf
10. Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling, http://www2.informatik.uni-freiburg.de/~stachnis/pdf/grisetti05icra.pdf

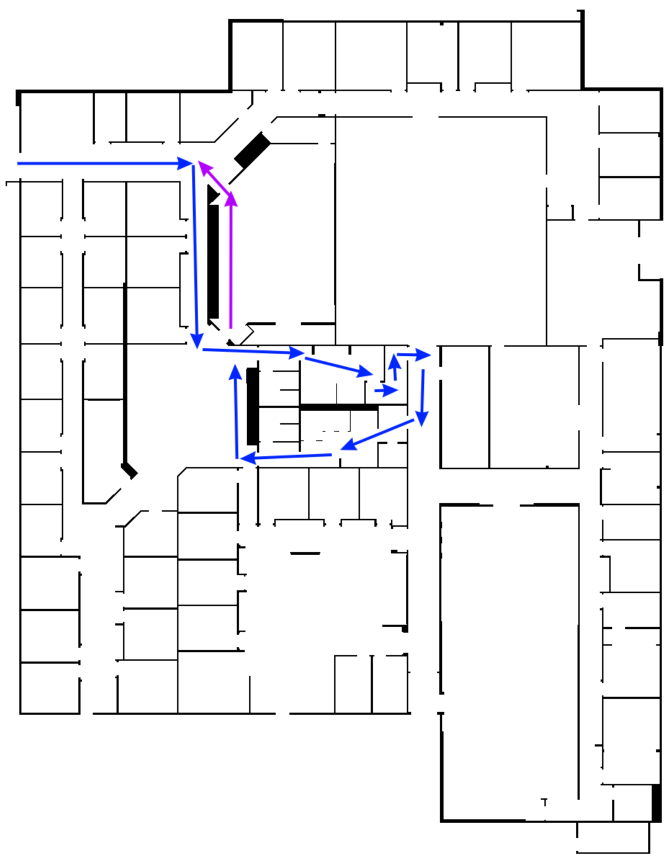

Figure 3: Willow garage mapping paths. The first path to map is the blue path. The second path to map is the purple path.