Robotic Localization and Dynamic Navigation using Adaptive Monte Carlo Localization (AMCL) with ROS

Tyler Wolfe

Abstract—Amcl (Adaptive Monte Carlo Localization) is a Robot Operating System (ROS) navigation package which utilizes particle filters to track the pose of a moving robot with a known 2D map. To test amcl's localization algorithm capabilities multiple robots were simulated using Gazebo and RViz where their movements were tracked and their ability to move toward a goal position was evaluated.

Index Terms—Robotics, Localization, AMCL, ROS.

1 Introduction

ROBOTIC localization is the process of evaluating where a robot is physically located with its current environment. Only by first having an accurate understanding of its current location can a robotic system be expected to perform movement autonomously. As humans it is easy to take for granted that each time someone gets up from the couch and makes their way to the refrigerator to fetch a drink they are localizing their current position, planning the best possible route to their destination, and continuously scanning for obstacles along the way; which are largely the same behaviours expected of a mobile robot moving within an environment. ROS (Robot Operating System), Gazebo, and RViz working together provide a framework of open source algorithms, packages, simulation, and mapping tools to test the capabilities of various mobile robot configurations to localize within a mapped environment and based on sensory feedback take movement actions toward goal positions.

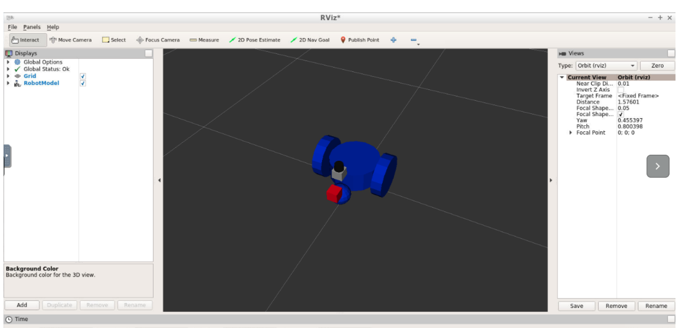

Fig. 1. Custom Robot Model in RViz.

2 Background / Formulation

In order to test amcl's probabilistic localization system several elements must be built in order to have a robot and navigable environment. First, a mapped world is necessary for robots to operate within inside the simulation environment. For this project a pre-made world was utilized provided by Clearpath Robotics known as Jackal. Second, a robot model must exist to provide a simulated robot in order to perform the localization, mapping, and movement tasks within the simulated environment. For this project a cylindrical robot base was selected with two casters, two wheels, a laser scanner, and camera. Lastly, amcl must be implemented in order to provide the robot model with the ability to use the map and laser scans in order to determine estimated poses which guide the robots movement toward a desired position within the map.

Key parameters guiding how the robot will localize within the map are contained within the costmap common params.yaml file and several are listed below:

- map type: costmap
- obstacle range: 3.1
- inflation radius: 0.5
- observation sources: laser scan sensor

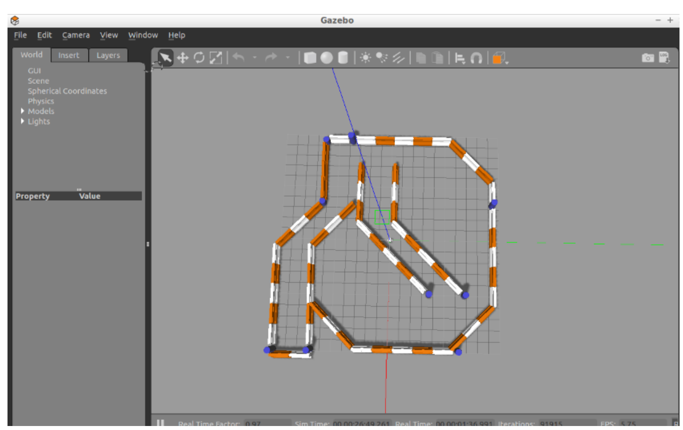

Fig. 2. Clearpath Jackal Mapped Environment for Robot Navigation.

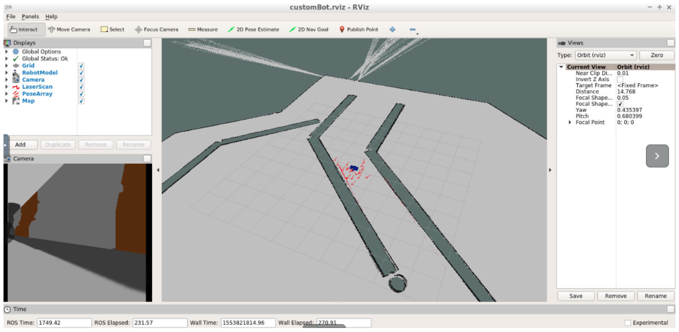

Fig. 3. Mapped Robot using Laser Scans in Particle Cloud.

3 Results

During initial tests RViz's built-in feature to assign 2D Nav Goals within the map was utilized to test the robot model's ability to successfully localize and move from a starting position to the goal position. The robot model was able to reach the goal position if nearby, however it repeatedly failed if it required escaping a hallway or turn a corner as the robot would become stuck near obstacles. However, after optimizing costmap parameters the robot is able to avoid getting stuck on barriers and successfully navigates to goal positions within the Jackal mapped environment.

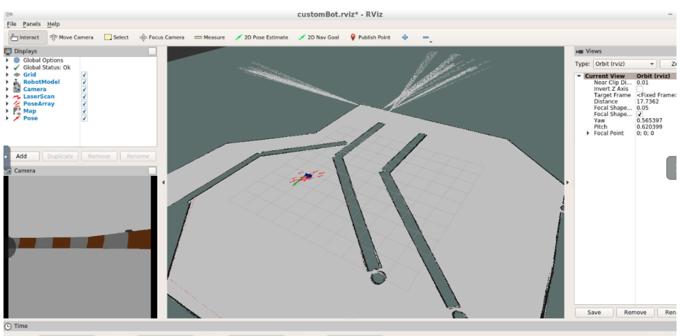

Fig. 4. Robot Reaches Goal Position in RViz.

The goal position is indicated by a green arrow using the Pose indicator in RViz.

A video was recorded of the another box-based robot model reaching the goal position with parameters optimized and can be seen here on YouTube.

4 Discussion

During the testing the amcl package efficiently support the robot quickly localizing within the map and with the costmap parameters optimized the robot is enabled to predictably reach the goal position in the majority of test events.

5 Conclusion / Future work

While the amcl-powered localization is efficient, there is opportunity to improve the robot's navigation and movement toward the goal position as demonstrated by frequent observations during test runs of inefficient path planning, frequent delays, and wide circling maneuvers when correcting for movement errors during the approach. With

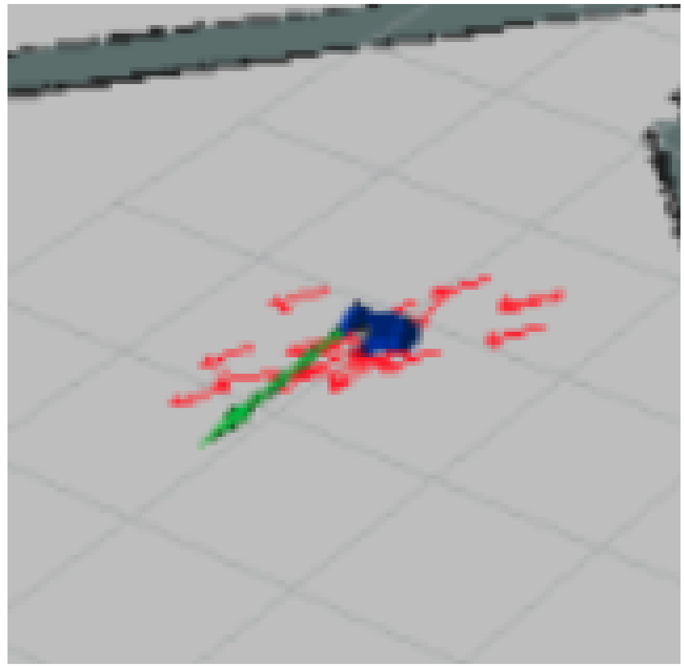

Fig. 5. Goal Position in focus with Green Arrow.

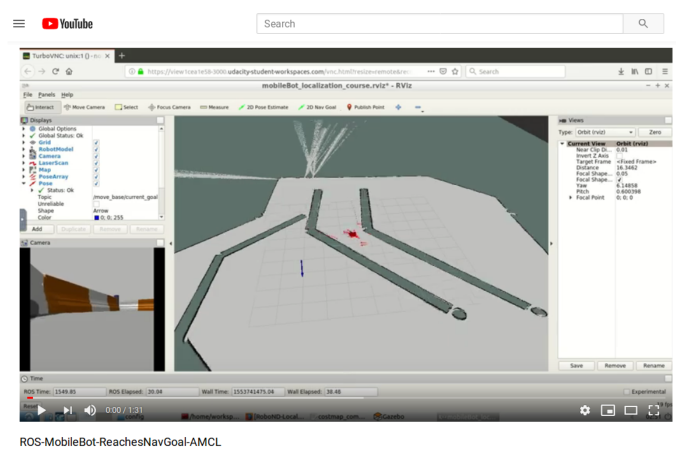

Fig. 6. Video of Robot Navigation via YouTube.

that said, the current setup of multiple robot models with their supporting costmap parameters are able to predictably reach the goal position as supplied in the navigation goal executable.