Showcase of Adaptive Monte Carlo Particle Filter for Robot Locatization

Mo Messidi

Abstract—This paper showcases a successfully implementation of a robot localization exercise using the Adaptive Monte Carlo Localization (AMCL) algorithm. Two robots, a benchmark robot and a custom robot, were made to self-localize and navigate a maze in a simulation environment. The robots successfully navigated the maze and reached their desired final location by solely utilizing their onboard sensors data and the AMCL algorithm.

Index Terms—Localization, Kalman Filter, Monte Carlo Filter, ROS, Robotic Software Engineer Nanodegree Program, Udacity.

1 Introduction

In the framework of robotics, localization denotes a robot's ability to establish its own position and orientation within a specific frame of reference given uncertain senor data.

This paper showcases a successfully implementation of a robot localization exercise using the Adaptive Monte Carlo Localization (AMCL) algorithm. Two robots, a benchmark robot and a custom robot, were made to self-localize and navigate a maze in a simulation environment. The robots successfully navigated the maze. The benchmark definition of one of the robots was given as part of the project, while the second was created independently. The benchmark robot was developed with guidance from Udacity while the custom robot was coded independently. The maze environment used in this project is a map created by jackal race that was developed by Clearpath Robotics. The source software for this project can be found here: https://github.com/mo-messidi/RoboND-Localization-Project.

2 Background

Localization is a fundamental issue in robotics. The most common solution approaches to robot localization problems are Kalman Filters and Monte Carlo Simulation.

2.1 Kalman Filters

The Kalman Filter is an estimate algorithm that can be used to estimate the value of a noisy variable very quickly as multiple time variant data points are collected. It can be an tool in estimating actual values from noisy input date in certain situations; However, It assumes linear models with unimodal Gaussian noise distributions making its real world applications severely limited in its original designed state.

The Extended Kalman filter (EKF) is a modified type of Kalman Filter algorithm that addresses the limitations of traditional kalman filters by linearizing nonlinear inputs using multidimensional Taylor series. The increased use cases come at the cost of increased computational resource requirements.

2.2 Particle Filters

Particle Filters are another localization solution that works by uniformly distributing virtual particles in a map and then gradualy removing the particles that are least likelihood of representing the current robot location.

The Monte Carlo Filter is a type of particle filter that uses Monte Carlo simulation for its location likelihood calculation.

The Adaptive Monte Carlo Localization method used in this project is a modified type of Monte Carlo filter where the number of virtual particles in the map is dynamically adjusted over time as the robot navigates a map.

2.3 Comparison / Contrast

Particle Filters tend to be much more efficient than Kalman Filters and ,in the case of Monte Carlo particle filters, they are not subject to the limiting assumptions of Kalman Filters as outlined above.

3 Models

The localization exercise was performed using two different robot models. The same localization method and parameters were used for both robots. Both robots consisted of a base chassis and two wheels. They both had two sensors, a camera and a LIDAR, which were used for localization. The benchmark robot had a rectangular chassis while the custom robot had a cylindrical chassis and a camera sensor that is mounted higher relative to ground level.Figure 1 shows the benchmark bot model in simulation and figure 2 shows the custom bot model in simulation.

3.1 Parameters

After the robots and the environment were created, ROS's AMCL and Base-mover packages was used for movement and localization with parameters tuned to best fit the task of getting the robots to autonomously navigate the maze and reach the goal position.

The following list describes the most critical parameters that were tuned and their individual effect on the robots' performance:

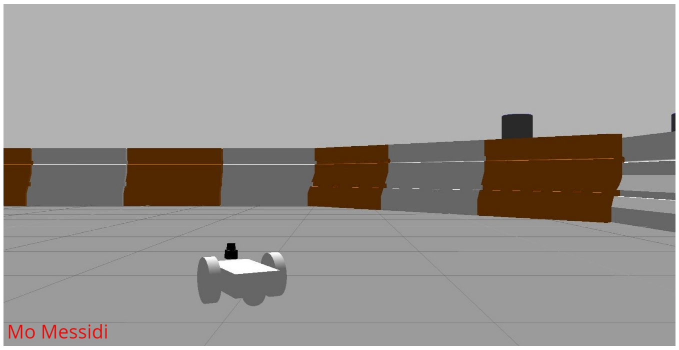

Fig. 1. Benchmark Bot Model

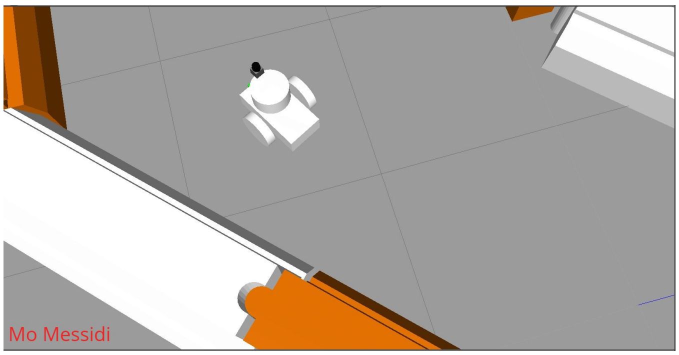

Fig. 2. Custom Bot Model

- robot radius This parameter describes the radius of the robot in meters. It was set to 2.3.
- inflation radius - This parameter describes the inflation radius in meters from the robot center. It is used to avoid collisions. It was set to 2.3.
- obstacle range This parameter sets the maximum distance ,relative to the robot, an object/obstacle can be added to the map. It was set to 5.0 meters.
- raytrace range This parameter sets the maximum range in meters at which to raytrace objects/obstacles from the map.It was set to 8.0.
- max particles This parameter, along with min particles, specifies the particle count used by the AMCL algorithm. A higher particle count leads to higher accuracy but adds more processing power requirements. The (min, max) number of particles was set to (20,500).
- transform tolerance: This parameter specifies the delay in transform data that is still acceptable. It accounts for the expected sensor measurement errors. It was tuned to 0.3.
- odom alpha 1 to 5 These parameters specify the expected noise of the odometry measurements. They account for the expect senor measurement errors. They were all set to 0.05.

4 Results

Both robots managed to successfully accomplish their objective and reach the desired target. The benchmark robot managed to complete the course in slightly over 15 minutes. while the custom robot managed to complete the same course in slightly over 14 minutes. figures 3 and 4 show each respective robot at its final desired location on the map.

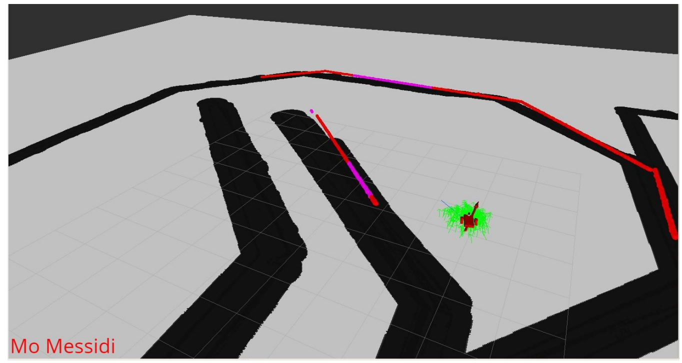

Fig. 3. Benchmark Bot Model

5 Discussion and Conclusion

This paper showcased an Adaptive Monte Carlo localization implementation on two robot in simulation. Both robots were able to localize autonomously in the given maze map and successfully reach the set target position while simultaneously avoiding all obstacles by relying solely on their noisy camera and LIDAR sensor information. It was noted that ,for both robots, the path taken to the target destination, while very similar, was not the most optimal path possible. The path similarity leads to the conclusion that robot size, weight and sensor placement configuration have little effect on the performance of the localization algorithm. A possible factor that could be investigated in future work is the effect of the quantity and/or quality of input data from various onboard sensors on the performance of the robots. It should also be noted that the localization algorithms mentioned in this paper can only be utilized if the overall map of the environment is know i.e. if the robots used in this project were "kidnapped" and teleported to an unmapped envirment that is unknow to them then localization using Kalman or particle filters would not be possible. One possible application for the localization methods highlighted in this paper could be an indoor floor sweeping robot where the indoor map of the house is know. Both AMC or EKF are possible localization methods in that application.

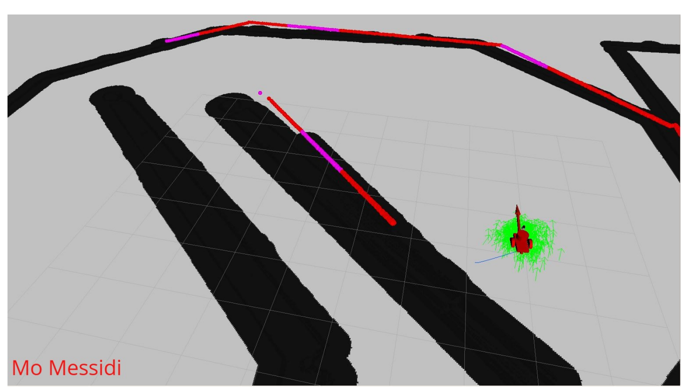

Fig. 4. Custom Bot Model