

Robotics Project : Where am I?

Shotaro Oyama

Abstract—The robot localization in simulation using ROS, Gazebo, and RViz. The robot finds and reaches the desired goal pose and orientation in the given situation. The Adaptive Monte Carlo Localization, AMCL in short, a Particle filter-based algorithm is used for the robot localization. This project also configures the specification of Robot by URDF file format and tweak the parameters in order for the robot to work well. The robot is implemented camera and hokuyo sensors node and the data from them is used for AMCL node and move_base node which is used for navigation of the robot.

Index Terms—Where am I project, Monte Carlo Localization, Particle Filter, Robotics Nanodegree, Udacity

1 INTRODUCTION

Localization is the challenge of determining the pose of robot in a mapped environment. There are three challenges, local localization, global localization, and the kidnapped robot problems. In local localization, or position tracking, robot knows its initial pose and the problem is to keep track of the robot's pose as it moves. In global localization, the robot's initial pose is unknown, and the robot tries to determine its pose relative to the ground truth map. In the kidnapped robot problem, robot's initial pose is unknown, and to be kidnapped at any time and moved to another location of the map. These challenges assume the situations in which robot gets involved.

In this project, robot tries to solve global localization problem. The two robots traverse a known mapped environment then navigate to the desired goal, though robot does not know its initial pose. Using the sensor data, and mitigating the noise by the Adaptive Monte Carlo Localization algorithm, robot tries to solve the problem.

2 BACKGROUND / FORMULATION

Localization problems are mainly caused by noisy sensors and actuators. Instead of totally relying on the data from them, there are several approaches. In this project, Kalman Filter and Monte Carlo Localization (Particle filter) are focused.

2.1 Kalman Filter

Kalman Filter is an estimation filter technique used to estimate a value of a variable in real time as the data is being collected such as a robot's pose. This filter is very prominent in control systems due to its accuracy and computational efficiency. However, Kalman Filter requires the assumptions that (1) Motion and measurement models are linear, State space can be represented by a unimodal Gaussian distribution. In the nonlinear problem, Extended Kalman Filter, which linearize a nonlinear motion or measurement function by Taylor Expansion.

Copied from the udacity Course

	MCL	EKF
Measurements	Raw Measurements	Landmarks
Measurement Noise	Any	Gaussian
Posterior	Particles	Gaussian
Efficiency(memory)	✓	✓✓
Efficiency(time)	✓	✓✓
Ease of Implementation	✓✓	✓
Resolution	✓	✓✓
Robustness	✓✓	x
Memory & Resolution Control	Yes	No
Global Localization	Yes	No
State Space	Multimodel Discrete	Unimodal Continuous

Fig. 1. Kalman Filter and Particle Filter Comparison

2.2 Monte Carlo Filter(Particle Filter)

Particle Filter is the algorithm that randomly and uniformly spreads particle within the entire state space. Each particle represents a guess of the robot's pose and orientation, and also contains a weight which is the difference between the actual and estimated. The value of the weight for each particle corresponds to the accuracy of each guess. These particles are re-sampled every time the robot moves by sensing the environment through range-finder sensors. After iterations, these re-sampled particles eventually converge with the robot's pose, allowing the robot to know its location and orientation. Particle Filter is not limited to linear models.

2.3 Comparison

Extended Kalman Filter has advantage for time, memory efficiency and accuracy. However, it requires Linear Gaussian state space assumption. Monte Carlo Localization algorithm is easier to program and setup than Extended Kalman Filter, and it can be used to represent any model. This advantage makes it since the world cannot always be modeled by Gaussian distributions. (See Fig. 1).

3 RESULTS

When My_robot is made initially, it had only 2 wheels, and it was unstable. Thus 2 casters were added to keep it

