Robot Localization and Navigation

Jiangdong Chen

Abstract—Two robot models using differential drive system as actuator are designed and the robot navigation task is successfully implemented in simulation environment. Localization of the robots is performed using the ROS AMCL package with input from a simulated laser range finder and odometry data. Navigation is implemented using ROS move_base package, which uses Dijkstra algorithm as global path planner and dynamic window algorithm as local path planner. By adjusting the key parameters, both robots can reach the goal pose with high accuracy.

Index Terms—Robot, IEEEtran, Udacity, LaTeX, Localization, Navigation.

1 INTRODUCTION

MOBILE robot have vaster application prospect because of its flexibility, small size and low cost. Environment perception, behavior decision-making and action are the three important parts of robotics architecture and are also critical for mobile robots. A truly autonomous mobile robot must has a good localization performance in order to navigate safely within its environment. This is a challenging task, because the environment is complex and the measurement from sensors is noisy. Furthermore, the robot must do path planning in realtime to efficiently avoid obstacles and arrive the goal. Solutions based on inertial measurement units or global positioning system (GPS) can provide position approximations and their corresponding uncertainties. However, this solution is impractical in indoor applications where GPS signals are not reliable [1]. Therefore, research of mobile robots localization algorithm and related sensors play an important role in the field of robotics

Probabilistic robotics is the focus of robot localization research work. The key idea of probabilistic robotics is to express uncertainty with probability theory. In many real world applications, probabilistic algorithms outperform other techniques [2]. These probability algorithm is mainly based on Bayesian filtering, include Kalman filter and Particle filter (also known as Monte Carlo localization). Both two algorithms are widely used in real robot application. General schematic for mobile robot localization is shown as Fig. 1 [3].

With the development of computer software, simulation has become an indispensable research means in scientific research. Similarly, we can use ROS in the Gazebo environment to test the reliability of localization algorithm.

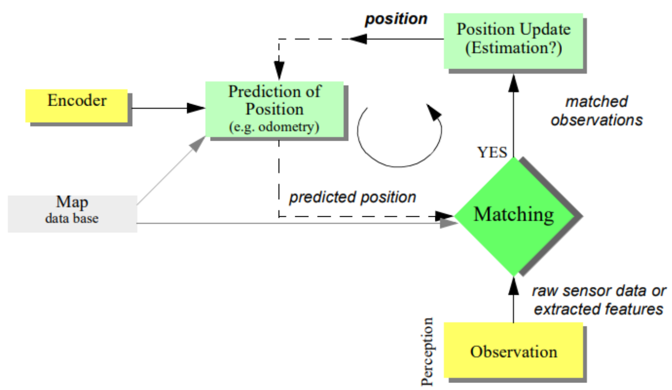


Fig. 1: General schematic for mobile robot localization.

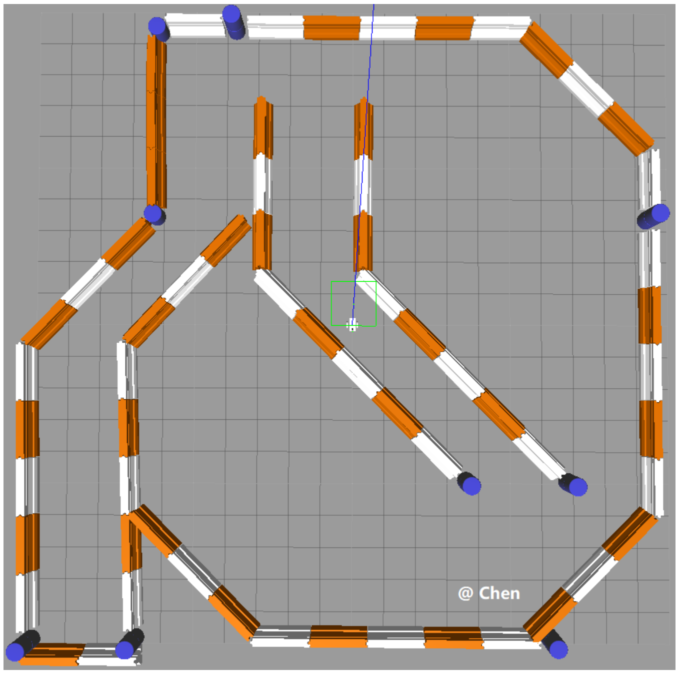


Fig. 2: Map used in gazebo simulation environment.

2 BACKGROUND

The aim of this work is to utilize ROS packages to accurately localize a mobile robot inside a provided map in the Gazebo and RViz simulation environments, as Fig.2 shown. The essential task can be broken into 4 sections, include robot model design (focus on sensors and actuators), global localization algorithm, local planner configuration and parameters Tuning.

2.1 Kalman Filters

The Kalman filter, also known as linear quadratic estimation(LQE), is an efficient recursive filter that estimates the internal state of a liner dynamic system from a series of noisy measurements. The algorithm is implemented in two steps, which are prediction and update. In the first section, it produces estimates of current state variables, along with their uncertainties. In the second section, these estimates are updated using a weighted average once the outcome of the next measurement is observed. However, the Kalman filter can only handle Gaussian distribution, and is limited to a linear assumption. Since non-linear systems are common for robots, Extended Kalman filter (EKF) is introduced. It works by linearizing the non-linear function through Tyler Series method.

2.2 Particle Filters

Particle filter is an alternative nonparametric implementation of the Bayes filter. When used in Localization problem, particle filter is also called as Monte Carlo Localization (MCL). In particle filters, the samples of a posterior distribution are called particles, which carry the information of robot like position and attitude, along with probability. The goal of MCL is to produce robot's pose represented by the belief. In first section, motion and sensor update, then in second section, the resampling process is implemented, and the particles with high probability survive and are re-drawn in the next iteration.

2.3 Comparison / Contrast

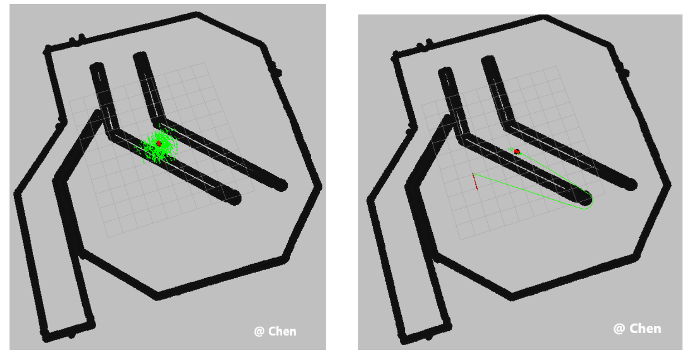
Particle filter has many advantages over Kalman filter. It can approximate almost any state space distribution, while Kalman filter only works with a Gaussian distribution. Furthermore, particle filter is easier to implement, more robust, and can be used in Global Localization. EKF also has some good characteristic such as high resolution and efficiency. In this project, we aim to solve a global localization problem, and require a robust algorithm, thus the Kalman filter is not suitable and the particle filter is used.

3 SIMULATIONS

A benchmark model and a personal model are built for simulated tasks both in UDRF format. A ROS package that launches a mobile robot model in a Gazebo world and utilizes packages like AMCL and the Navigation Stack is created. The goal is to explore, add, and tune specific parameter corresponding to each package to achieve the best possible localization results. All the simulation is conducted on a VM with Ubuntu 16.04 LTS system and ROS-kinetic installed.

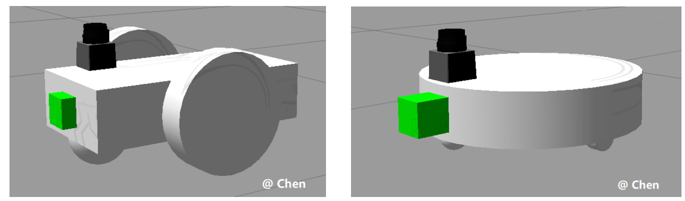
3.1 Achievements

2D global localization and navigation are achieved on both the benchmark model and the personal model, with different parameters configured. Fig.3a shows the initial robot pose (both benchmark model and personal model) while Fig.3b shows the goal robot pose. The graphs are viewed in Rviz. The green arrows represent particles created by the amcl package, and the red arrow specifies the pre-set goal robot pose.



(a) Start Pose (b) Goal Pose

Fig. 3: Start and Goal Poses



(a) Benchmark Model (b) Personal Model

Fig. 4: Robot Models

3.2 Benchmark Model

3.2.1 Model design

The benchmark robot has a box shape chassis with dimensions 0.4x0.2x0.1. At the bottom of the robot, there are two casters. There are two wheels connected to the sides of chassis. The laser sensor located at the middle of the robot while the camera at the front. An image of the benchmark model is shown in Fig.4a.

3.2.2 Packages Used

The packages used are as follows:

- ros-kinetic-map-server.
- ros-kinetic-amcl.
- ros-kinetic-costmap-2d
- ros-kinetic-global-planner
- ros-kinetic-base-local-planner
- ros-kinetic-move-base.
- ros-kinetic-navigation.

Fig.5 shows the ROS topics for the benchmark model.

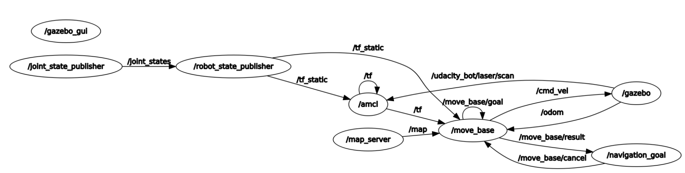


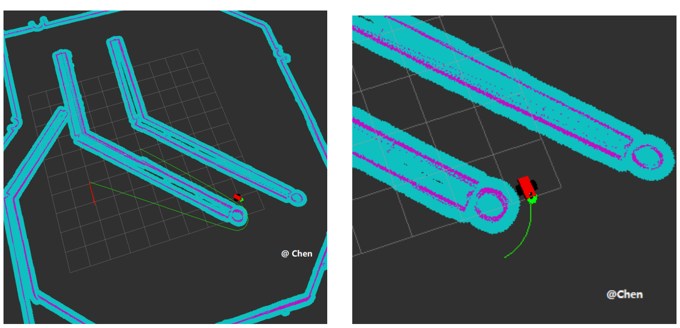
Fig. 5: Rosgrpah.

3.2.3 Parameters

Parameters in AMCL and move_base configuration file are shown in Table 1.

TABLE 1: AMCL and MOVE_BASE Parameters

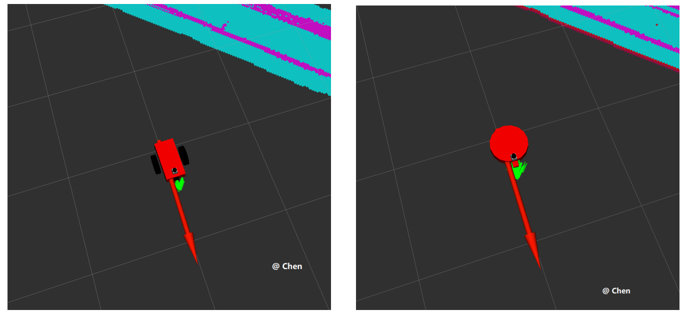
Parameter	Value
AMCL Parameters	
odom_alpha1	0.01
odom_alpha1	0.01
odom_alpha1	0.001
odom_alpha1	0.001
update_min_d	0.05
update_min_a	0.05
transform_tolerance	0.5
min_particles	10
max_particles	300
laser_min_range	0.1
laser_max_range	30
base_local_planner	
holonomic_robot	false
yaw_goal_tolerance	0.1
xy_goal_tolerance	0.15
sim_time	3
vx_samples	16
sim_granularity	0.02
pdist_scale	0.8
gdist_scale	0.45
occdist_scale	0.01
meter_scoring	true
costmap_common_paras	
obstacle_range	8.0
raytrace_range	10.0
transform_tolerance	0.5
inflation_radius	0.2



(a) Global Planner (b) Local Planner

Fig. 6: Path Planner

The above parameters has relatively great influence on robot performance compared to others. The amcl parameters determine localization performance, while other two types of parameters are essential for navigation performance. Among these parameters, some parameters are sensitive to computer performance, such as sim_time and transform_tolerance, which may due to different simulation results in different computers. Global planner uses Dijkstra's method and its parameters are not turned and remain the default values. Green line in Fig.6a is the global path created by global planner. Local planner provides implementations of the Dynamic Window Approach(DWA) and Trajectory Rollout approaches to local control. The short green line in Fig.6b is the local path created by local planner. Costmap parameters obstacle_range and raytrace_range are increased to improve the efficiency of creating the local and global costmaps and can affect the behavior of global and local planner.



(a) Benchmark Model (b) Personal Model

Fig. 7: Simulation Result

3.3 Localization Results

3.4 Personal Model

3.4.1 Model design

The personal robot is a sweeping robot whose structural parameters refer to a sweeping robot that exists on the market. The actuator, laser range finder, and camera are remained the same with the benchmark model, but the installation placement are different. An image of the personal model is shown in Fig.4b.

3.4.2 Packages Used

Packages used for the personal models are the same as those used for the benchmark model.

3.4.3 Parameters

Most parameters for the personal model are the same as those for the benchmark model. The modified parameters are shown in the TABLE 2.

TABLE 2: Modified AMCL and MOVE_BASE Parameters

Parameter	Value
base_local_planner	
gdist_scale	0.3
costmap_common_paras	
robot_radius	0.175

4 Results

Both the benchmark model and the personal model reaches the goal efficiently with sufficient localization accuracy. When simulation starts, the robot turn in place to sense its surrounding and later the particle filter converges. Although the robot drifts away from the global path when it navigates to the goal pose, the robot move on the correct path in most time.

As shown in Fig.7, both the benchmark model and the personal model are localized accurately at the goal position.

4.1 Technical Comparison

The benchmark robot has a rectangle chasis systems while the personal robot a radius chasis systems, but the overall size does not differ a lot. Both two models use differential-driven wheels. The noise parameters of the sensor model and the motion model remains the same. This allows for similar localization effect with fewer changes in parameters although the sensor layout is changed. Hence, the localization accuracy stays almost the same.

5 DISCUSSION

The adaptive particle filter provided by AMCL package and navigation stack provided by move base package have excellent performance. With the same particle filter parameters and minimal changes to the navigation parameters the personal robot was able to navigate successfully to the goal pose. However, in the beginning, it's not easy to quickly choose and tuning the appropriate parameters to make the robot perform well. Parameters of base local planner have a great influence on the performance of robot navigation and must be tuning patiently because these parameters are related to the bias of local and global planners. Move to AMCL parameters, the particle filter's behavior is relatively easier to control.

From this work it was demonstrated that the adaptive particle filter approach could be competent for global localization task, and has good versatility. The laser range finder sensor could supersede the perception section for indoor environment, Overcoming the shortcomings of GPS and WIFI sensors. Differential wheel system is a reliable configuration to obtain odometer data.

The Monte Carlo Localization method can be use in many area, such as home service robot, AGV robot used for piling can, rescue robot.

6 CONCLUSION / FUTURE WORK

The benchmark model and personal model, performing 2D global localization and navigation are built and simulated in gazebo environment. A ROS package is created and the amcl node and move base node are configured appropriately. The parameters are tuning, and both robot models successfully reach the goal with high localization accuracy, although not fully following the global path.

For future work, the created package is deployed on an actual device to perform home sweeping task. Some parameters should be configured patiently to meet the system requirement and deal with the complexity of the real world. Also, attention should also be paid to the structure design and parts installation of the real robot.

REFERENCES

[1] S. M. Malagon-Soldara, M. Toledano-Ayala, G. Soto-Zarazua, R. V. Carrillo-Serrano, and E. A. Rivas-Araiza, "Mobile robot localization: A review of probabilistic map-based techniques," *IAES International Journal of Robotics and Automation*, vol. 4, no. 1, p. 73, 2015.
[2] T. Sebastian, B. Wolfram, and L. Dieter, *Probabilistic Robotics*. MIT Press, 2006.
[3] R. Siegwart, *Introduction to Autonomous Mobile Robots*. MIT Press, 2004.