Where-Am-I

Hytham Tag

Abstract—This project Solves one of the major problems in autonomous robots which is it's location.

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

1 Introduction

THE localization problem is one of the major problems that faces alot of peoble as the robot must know it's location because some tasks may be very hard if the robot didn't learn how to tell it's location, Here i am solving the localization problem of a know environment using advanced monte carlo localization.

2 Background

As from the above i am solving the localization problem in a know environment but the sensors contain some noise so i am using some techniques to solve this issue.

2.1 Kalman Filters

Kalman Filters is a filter used to overcome noisy measurements and to estimate the location of the robot. Kalmen filter is done in 2 steps. First predict and then update. The predict step calculate the position after movement is done and the update step uses the measurements to update the location. Kalman filters is applied only using Gaussian distributions so only linear operation are allowed. And the robot model is not linear, so we need to convert from nonlinear equations to linear equations. Taylor Series is the method for linear approximation. By applying linearity, the Kalman Filter become Extended Kalman Filter.

2.2 Particle Filters

Monte Carlo Localization,First it defines particles that predict the robot position then After motion and position of each particle updates, the uncertainty increases but after the measurements from the sensors, And finally another update is done to decrease the uncertainty.

2.3 Comparison / Contrast

Particle Filter can work with any distribution while Kalman Filter only works with a Gaussian distribution. Particle Filter takes raw measurements; however, Kalman Filter requires known places. The posterior is particles for Particle Filter and Gaussian for Kalman Filter. Kalman Filter is more efficient and provides more resolution; on the other hand Particle Filter is easier to implement and more robust. Only Particle Filter has Memory and Resolution control and can provide a solution for Glocal Localization.

3 Simulations

The simulations are Gazebo that provide the environment to simulate the two robots rviz to control over each robot and get visual all the data and both these simulations run on ROS framework, The ROS packages are AMCL, move base, map server, and navigation.

3.1 Achievements

Both my bot and udacity bot reaced the target but udacity bot went wrong first in contrary to my bot which reached the goal faster and without any mistake.

3.2 Benchmark Model

3.2.1 Model design

Most of parts of the robot provided by udacity is the chassis. It has rectangular box with dimensions 0.4 x 0.2 x 0.1. It has two casters to overcome the balancing problem, one at -0.15 0 -0.05 and the other at 0.15 0 -0.05. There are two wheels connected to the chassis at 0 -0.15 0 and 0 +0.15 0. The laser sensor located at position relative to the chassis 0.15 0 0.085. It has a cubic shape with length 0.1. The camera sensor is connected to the chassis at 0.2 0 0, and it has a cubic shape with lenght 0.05. The difference between the my robot and the Udacity robot is the a link. The camera and the laser sensors are connected to this link. The link has the dimensions of 0.5 0.01 0.02. It is connected to the chassis at location 0 0 0.075. The camera is connected to sensor_link at 0.175 0 0 and the laser sensor connected to the at 0 0 0.05.

3.2.2 Packages Used

Packages used for ROS kinetic aree:

- ros-kinetic-navigation
- ros-kinetic-map-server
- ros-kinetic-move-base
- ros-kinetic-amcl

4 Results

4.1 Localization Results

At first i used 10 particles to 30 max and it made the robot go to the wrong path and wander a lot but after that i increased the particles to 100 as min and 1000 as max and that made the robot more efficient.

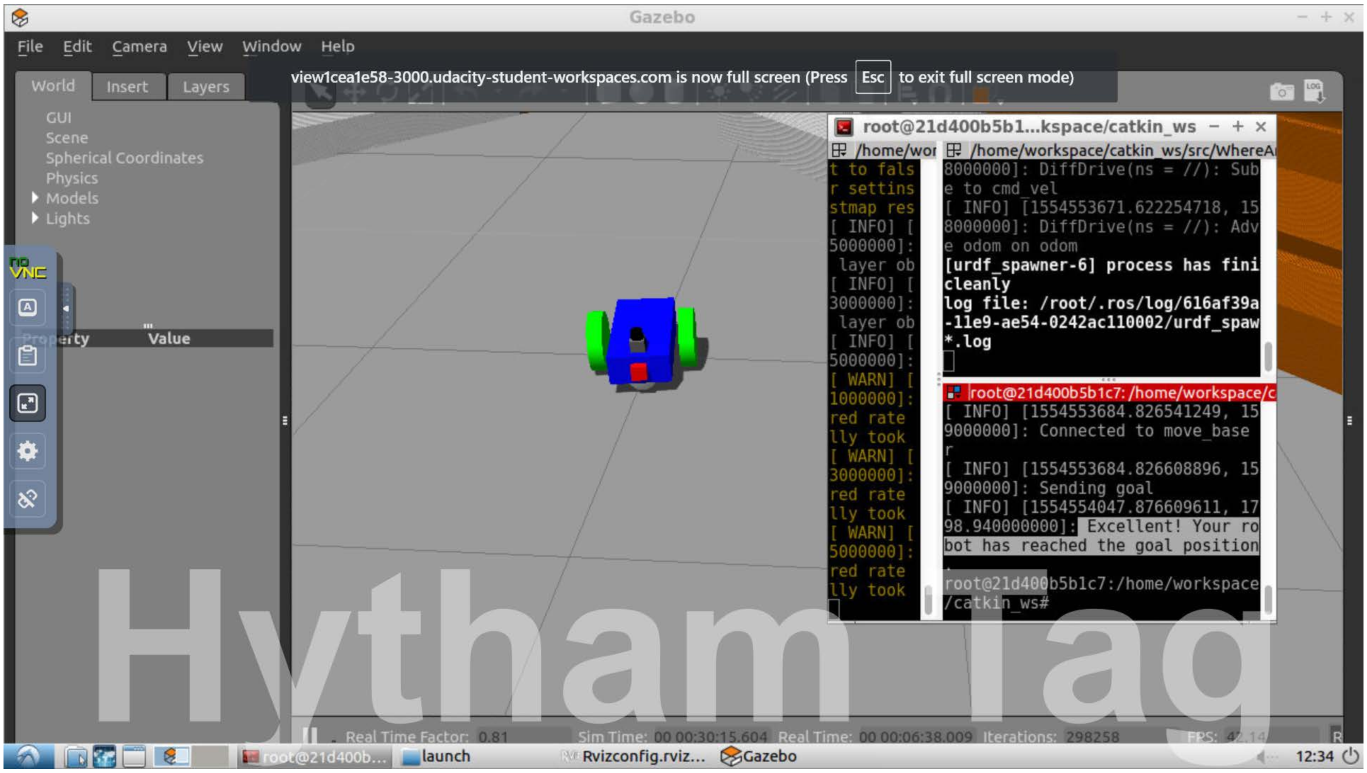

Fig. 1. bot design

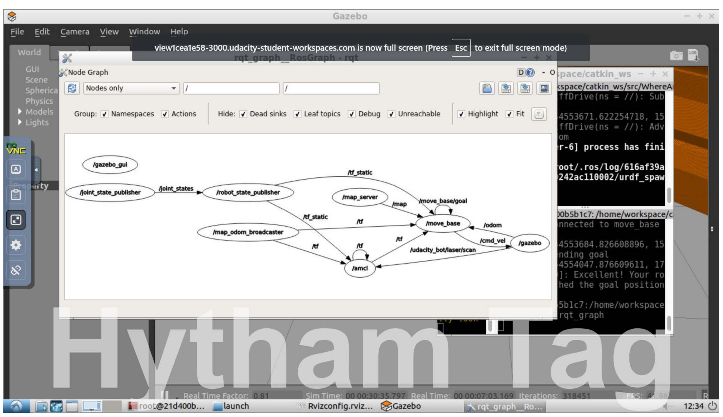

Fig. 2. bot Graph

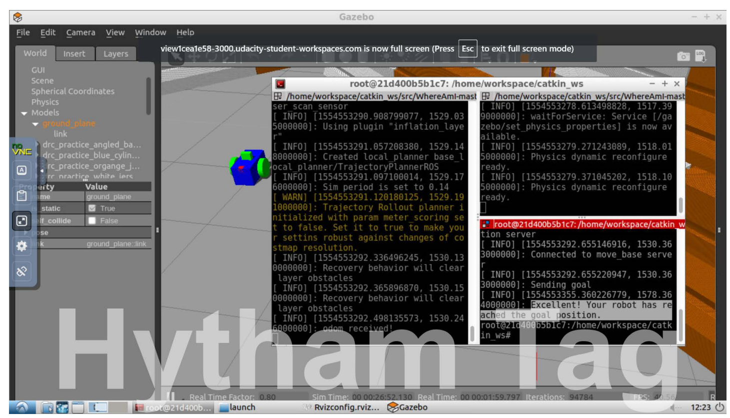

Fig. 3. Bot Goal

TABLE 1
Parameter Differences Between Udacity Bot and bot

File or Pkg	Parameter	Udacity Bot	bot
AMCL	min_particles	15	100
AMCL	max_particles	250	1000
Common	obstacle_range	5.0	6.0
Common	raytrace_range	6.0	9.0
Common	inflation_radius	0.5	1.0
Common	robot_radius	N/A	0.5
Global	update_frequency	5.0	3.0
Global	publish_frequency	5.0	3.0
Local	update_frequency	5.0	1.0
Local	publish_frequency	5.0	1.0
Local	width	40.0	3.0
Local	height	40.0	3.0
Local	resolution	0.05	0.01

4.2 Technical Comparison

The main difference between the two robot models is the link, and the sensors position. There are differences in the selected parameters, which can be seen from the provided table. My bot performed better because of the parameter tuning that was applied to it.

5 Discussion

The two robot models were able to reach the target. The Udacity Bot first went to opposite direction, while bot directly moved to the goal direction. The reason for this difference was the differently set parameters.

The convergence time for both of the robots were quite similar. Which shows that the AMCL package parameters were almost the same. Actually, only the number of particles parameter was different. As a result, these parameters do not have much effect on localization performance.

5.1 Topics

- bot performed better. It took less time to reach the goal position.
- The different costmap parameters let bot to perform better. The bot directly started following the global path.
- The 'Kidnapped Robot' problem is randomly moving the robot to another location. Both robots will fail to overcome this problem maybe if there is an algorithm that was applied to detect the kidnapping and decide a new plan this will solve this problem.
- Localization can be performed in scenarios in which the map is known, and the location of the robot does not randomly change.
- MCL/AMCL algorithms were the best choice in small known environment as the particles are few but in large environment more particles are used and that could lead to a heavy computational crisis!

6 Conclusion / Future work

Both robots could reach their goal the bot robot could reach it faster, however for future work one can see that both robot lacks a commercial goal and it can be fixed if we applied some changes in the body design maybe to add a robot arm, or some sort of a flat box to act like a moving table which can be helpful in warehouses. Also modify the AMCL algorithm to solve the kidnapping problem.

6.1 Modifications for Improvement

Examples:

- Use two motered wheel plus 1 free wheel
- Use two or more laser sensors