Where Am I

1 Abstract

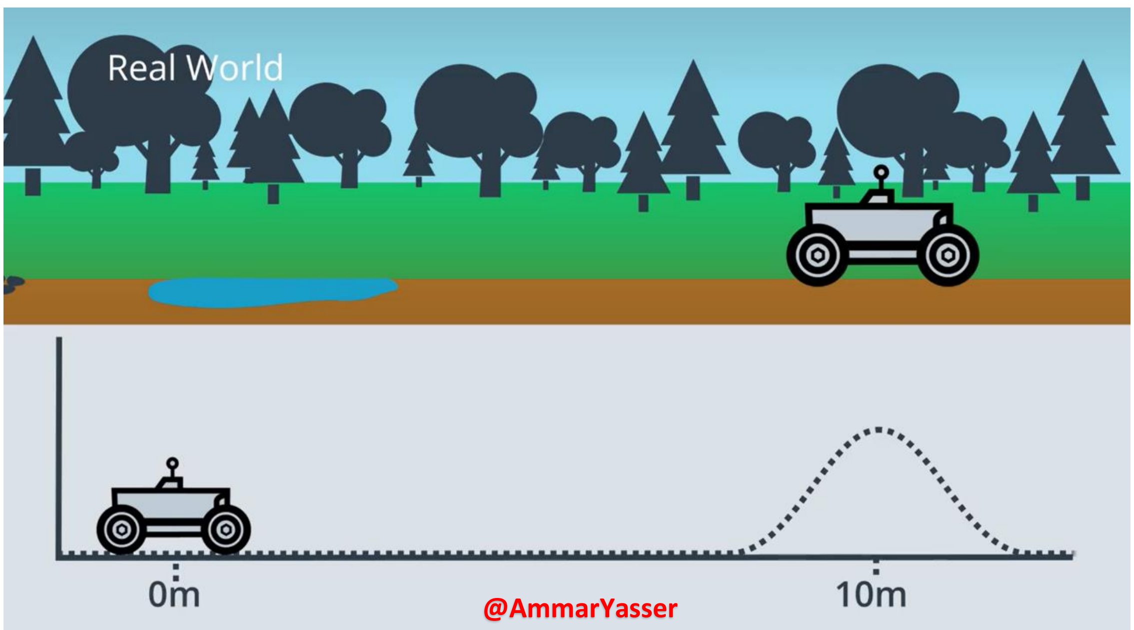
This report is for a Navigation Robotics ND from Udacity. In this project, a robot model is built and used to experiment localization and navigation within given map. Localization is performed by the AMCL, Adaptive Mont Carlo Localization, ROS-package. with parameter's tuning, the robot could successfully perform localization and navigation. Using Gazebo and RViz, tracking the motion of the robot wouldn't be easier.

2 Introduction

Localization is one of the most challenging problems in Robotics. Robots need to localize themselves respectfully with the environment's map to successfully perform motion planning and navigation. In this project, the robot model with laser scanner and a camera will localize and navigate through a given environment using the AMCL package using ROS package.

3 Background

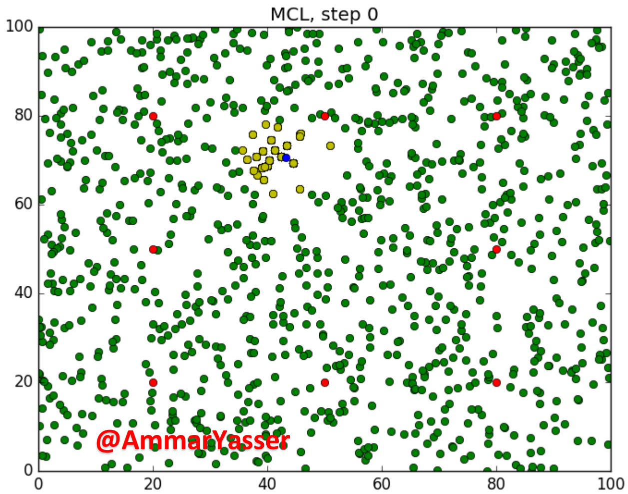
Localization is always a problem about uncertainty because in the real world, the measurements and robot motion always have noise and other factors which affects its certainty about the location.

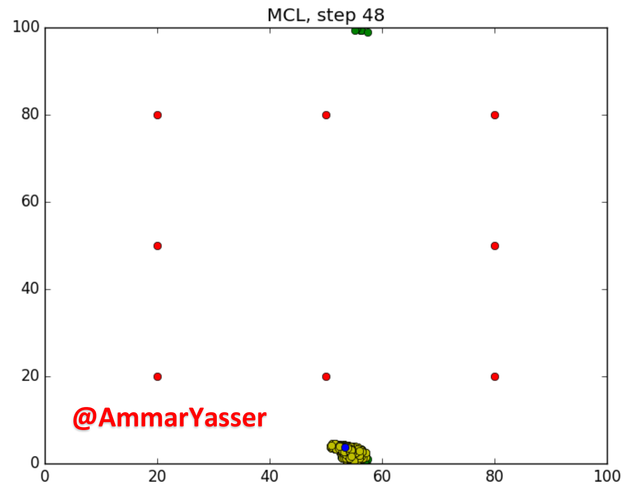


Kalman filter is one of the most popular methods used mainly to remove this uncertainty by sensor fusion. This gives the robot the confidence about its measurements which allows the robot to be more accurate about its location.

Mont Carlo Localization or MCL algorithm is also a popular method to attack the, local and global, localization problem. In this algorithm, many virtual particles are initiated with an estimated location of the robot. Then, the particles are re-sampled and reduced to less amount where the more accurate ones are survived to gives the estimation about the location after every motion and sensor measurements.

The following graphs shows the power of MCL, Particle filter, to estimate the location of the robot.





The following table shows the major comparison between the both algorithms.

	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

In our project, We are using AMCL (Adaptive Mont Carlo Localization) dynamically adjusts the number of particles over a period of time, as the robot navigates around in a map. This adaptive process offers a significant computational advantage over MCL.

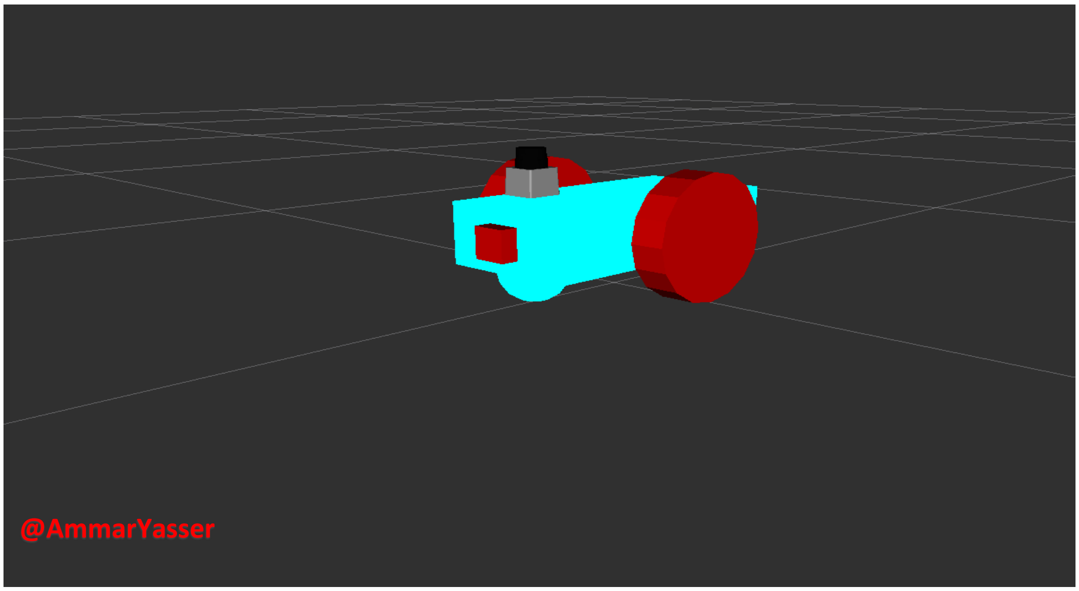
The ROS amcl package implements this variant and you will integrate this package with the robot to localize it inside the provided map.

4 Robot Model

The robot is built in a URDF file and integrated to the gazebo file to simulate it physically. The robot mainly has two differential wheels and a drive roller.

The robot has two integrated sensors:

a. Laser scanner (Hokuyo)
b. Camera



5 Localization

Localization and navigation process in this project depends on 3 main ROS packages:

- amcl : is used to apply the localization using the measurements from the hoyuko sensor and compare it with the given map.
- move_base: mainly used to perform the navigation to a specific position (Goal).
- map_serever: is used to give the simulation map for the given world (environment).

5.1 amcl parameters

Tuning these parameters helps the robot to create more accurate estimation to its position

- min_particles="15"
- max_particles="100"
- recovery_alpha_fast="0.750"
- recovery_alpha_slow="0.01"
- initial_pose="[0 0 0]"
- update_min_d="0.15"
- update_min_a="0.785"
- resample_interval="5"
- laser_model_type="likelihood_field"

- laser_min_range="0.02"
- laser_max_range="15.0"
- laser_z_hit="0.90"
- laser_z_rand="0.250"
- laser_max_beams="20"
- laser_sigma_hit="0.20"
- laser_likelihood_max_dist="3.50"
- odom_model_type="diff-corrected"
- odom_alpha1="0.2"
- odom_alpha2="0.2"
- odom_alpha3="0.2"
- odom_alpha4="0.2"

5.2 move_base

This package has been divided into several configurations files, with different parameters

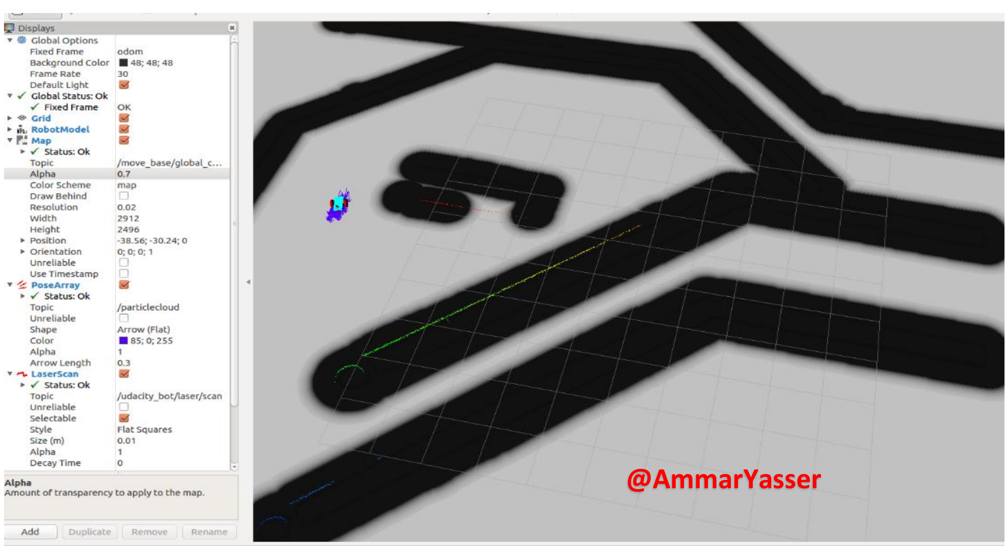
- config/costmap_common_params.yaml
- config/local_costmap_params.yaml
- config/global_costmap_params.yaml
- config/base_local_planner_params.yaml

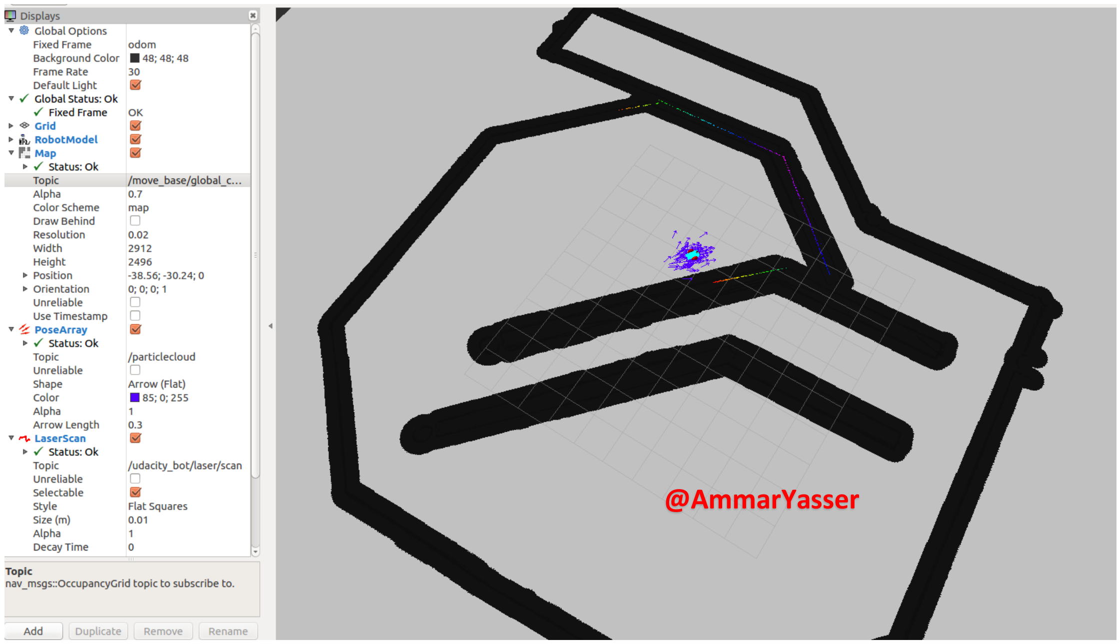
the most important parameters

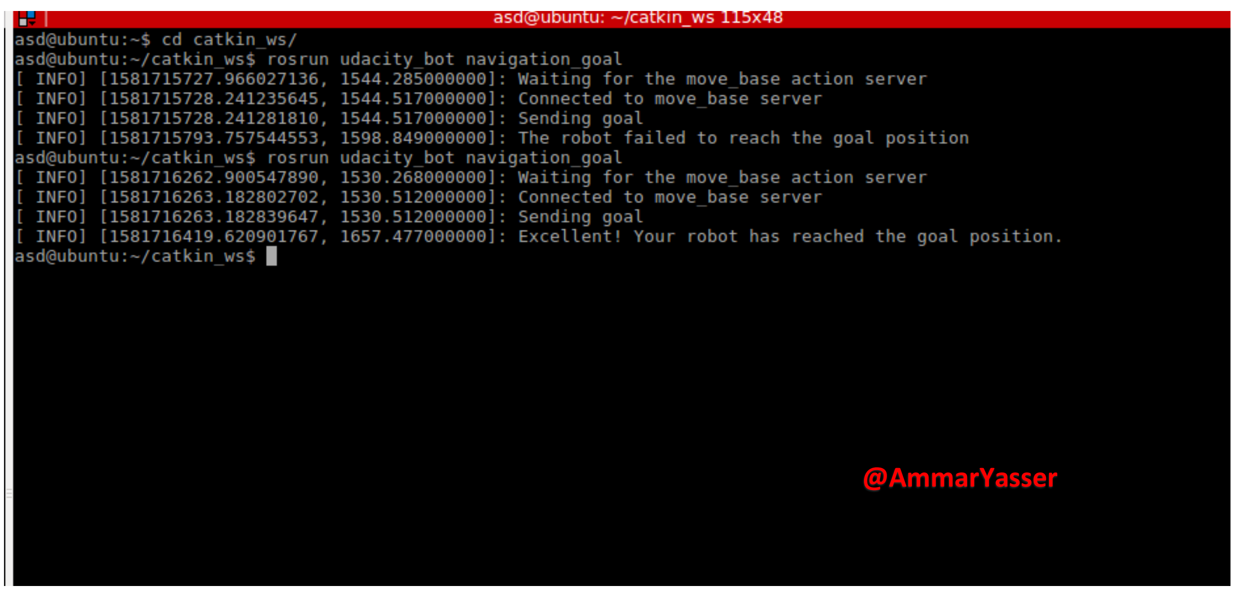
- obstacle_range: 1.0
- raytrace_range: 3.50
- transform_tolerance: 1.0
- inflation_radius: 0.35
- update_frequency: 3.0
- publish_frequency: 3.0

6 Results

The localization accuracy is very good but the parameter tuning takes a lot of time to find the best values.







7 Discussion

The result seems too be very good due to the parameter tuning especially for particles numbers, Reducing them has been a good idea to decrease uncertainty and increase the computation time.Also for obstacle_range,raytrace_range, transform_tolerance, inflation_radius which causes very precise obstacle avoindance and more accurate map according to the measurements from the laser scanner.

AMCL is very good algorithm which attack only the local and global localization problems and It’s not convenient for the Kidnapped one.

8 Future Work

- For more accurate measurement we can use/add more sensor like: GPS, RGP-D camera ..etc , which helps to provide more accurate info. About the location
- Speed Vs. Accuracy is challenging because the relation in not proportional. So, It’s more likely dependent on the application and how much speed / accuracy you want.
- Deploy the project on actual physical robot in real physical environment and try to use SLAM to provide the map of the world and try to optimize the localization parameters.