Where am i

Abstract:

This project aims to solve one of the most challenging aspects of robotics (localization), using two different robot models.

1-Introduction

Localization is a problem that robots face when introduced to an environment, where the robot must figure out its location as well as be able to navigate through the environment, monte carlo algorthim was used to demonstrate how it can overcome this problem.

2-Background

In order for a robot to be able to know its location within an environment it will have to use its sensors to collect data , these sensors are subject to external noise from the environment itself 2 techniques can be used to increase accuracy even with noise in sensor readings.

2.1- Kalman filter:

Consists of 2 steps:

1. predict: calculates the position after a movement is done.

2. update: updates the location of the robot using the measurements obtained from predict step.

However kalman filter is only suitable in linear actions due to its use of Gaussian distribution.

To overcome this extended kalman filter is used , by using taylor’s series for linear approximation we can still use linear equations in a non-linear model , which our robot happens to be categorized as.

2.2- Monte carlo:

Particles are generated around the robot which predicts the robot’s position and after movement

The particles update their position also using measurement from the sensors, with each iteration of this process uncertainty decreases and the robot is able to localize itself accurately.

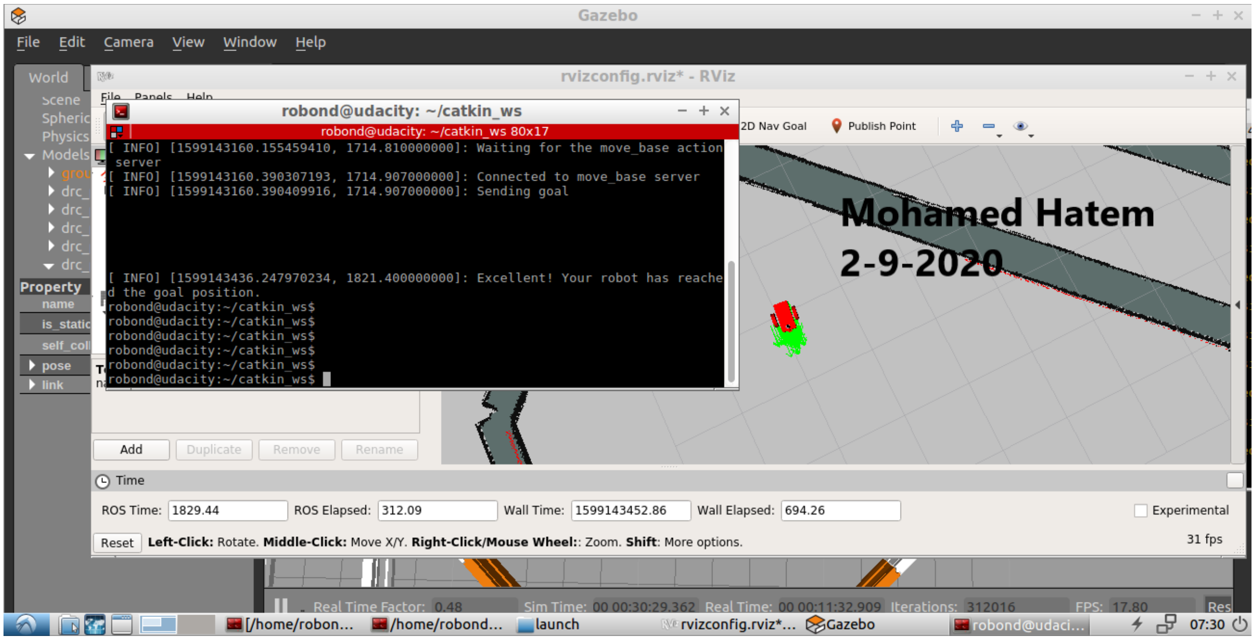
Monte carlo can work with any distribution unlike kalman filter so it’s a better approach to solving our problem in the given environment.

However since monte carlo relies on raw measurement and kalman relies on well know places in the environment kalman is more efficient in localization, on the other hand monte carlo is more robust and can solve global localization problems.

3-Results

2 robots where used for this project and both reached the target goal successfully,

Udacity provided robot:



```
[ INFO] [1599143160.155459410, 1714.810000000]: Waiting for the move_base action
 server
[ INFO] [1599143160.390307193, 1714.907000000]: Connected to move_base server
[ INFO] [1599143160.390409916, 1714.907000000]: Sending goal




[ INFO] [1599143436.247970234, 1821.400000000]: Excellent! Your robot has reache
d the goal position.
robond@udacity:~/catkin_ws$
robond@udacity:~/catkin_ws$
robond@udacity:~/catkin_ws$
robond@udacity:~/catkin_ws$
robond@udacity:~/catkin_ws$
robond@udacity:~/catkin_ws$
```

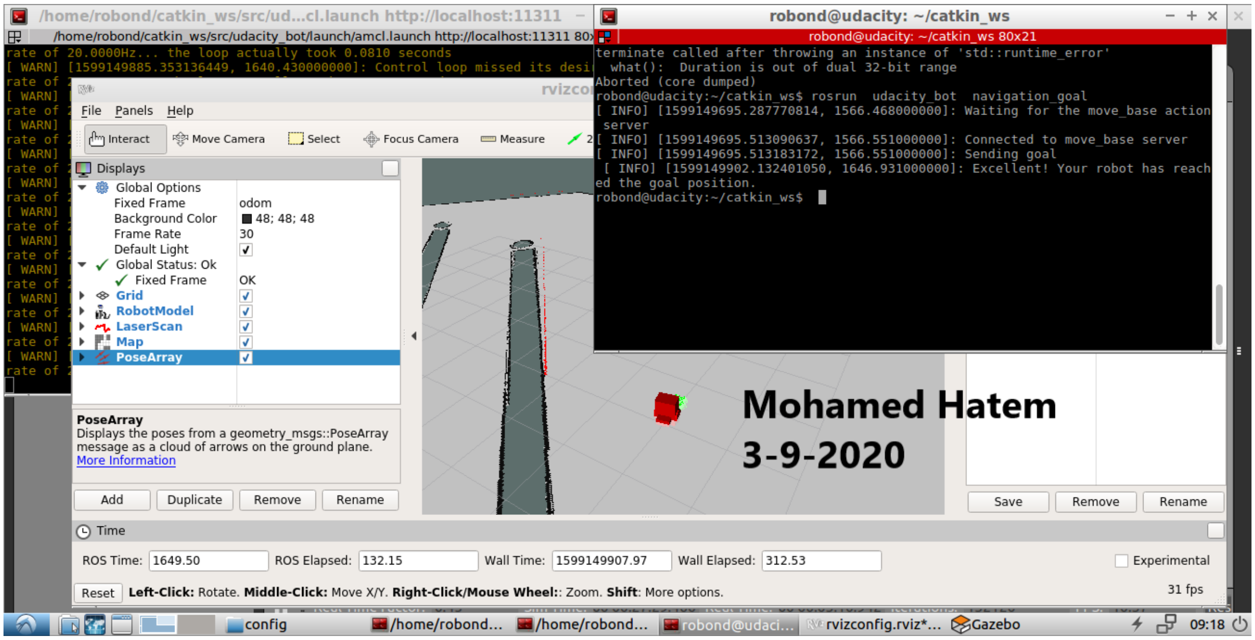
Mohamed Hatem
2-9-2020

ROS Time: 1829.44 ROS Elapsed: 312.09 Wall Time: 1599143452.86 Wall Elapsed: 694.26

Reset **Left-Click:** Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift**: More options. 31 fps

Real Time Factor: 0.48 Sim Time: 00 00:30:29.362 Real Time: 00 00:11:32.909 Iterations: 312016 FPS: 17.80

Custom created robot for this paper:



/home/robond/catkin_ws/src/udacity_bot/launch/amcl.launch http://localhost:11311 80x

```
rate of 20.0000Hz... the loop actually took 0.0810 seconds
[ WARN] [1599149885.353136449, 1640.430000000]: Control loop missed its desi
rate of 2
[ WARN]
rate of 2
[ WARN]
rate of 2
[ WARN]
rate of 2
[ WARN]
rate of 2
[ WARN]
rate of 2
[ WARN]
rate of 2
[ WARN]
rate of 2
[ WARN]
rate of 2
[ WARN]
rate of 2
```

Displays

- Global Options
  - Fixed Frame: odom
  - Background Color: ■ 48; 48; 48
  - Frame Rate: 30
  - Default Light: ☑
- ✓ Global Status: Ok
  - ✓ Fixed Frame: OK
- Grid ☑
- RobotModel ☑
- LaserScan ☑
- Map ☑
- PoseArray ☑

PoseArray
Displays the poses from a geometry_msgs::PoseArray message as a cloud of arrows on the ground plane.
More Information

Time

ROS Time: 1649.50 ROS Elapsed: 132.15 Wall Time: 1599149907.97 Wall Elapsed: 312.53 □ Experimental

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift**: More options. 31 fps

robond@udacity: ~/catkin_ws 80x21

```
terminate called after throwing an instance of 'std::runtime_error'
  what():  Duration is out of dual 32-bit range
Aborted (core dumped)
robond@udacity:~/catkin_ws$ rosrun  udacity_bot  navigation_goal
[ INFO] [1599149695.287770814, 1566.468000000]: Waiting for the move_base action
 server
[ INFO] [1599149695.513090637, 1566.551000000]: Connected to move_base server
[ INFO] [1599149695.513183172, 1566.551000000]: Sending goal
 [ INFO] [1599149902.132401050, 1646.931000000]: Excellent! Your robot has reach
ed the goal position.
robond@udacity:~/catkin_ws$ 
```

Mohamed Hatem
3-9-2020

4- Configuration

Robot Model: The model is the same as the udacity supplied one with an added link that connects the chassis with the camera using 2 joints.

Parameters:

Using a small range of particles (min 20 max 100) resulted in the robot going in circles and not be able to make any progress at all increasing this range had a good impact on the process.

Changes were more for both local and global coverage with the new coverage being (7x7) and

(19x19) respectively, higher value would only increase computation without an acceptable increase in accuracy.

Update and publish frequency is increased cause the default value wasn’t fast enough.

Obstacle_range and ray_range were increased to correctly identify obstacles in the environment which helps during navigation process.

5- Discussion:

During early stages of testing both robots went into the opposite side, and drove around in circles wish indicated that a small amount of particle as discussed earlier , both robots took similar times to reach the goal once appropriate config values were set.

However monte carlo wouldn’t work well if we introduced the robot to the kidnapped scenario , we would need an algorithm to detect the kidnapping and create a new plan.

6-Conclusion/future work:

Both robots successfully reached the target goal.

For future work we could add a sweeper to the robot in order to clean rooms inside a hotel or even houses.

Improvments to the robot model would be to add another laser sensor and use 4 wheels instead of just 2.