AMCL Localization Project: Where Am I?

Arnab Paul

Abstract—To meet localization challenges many probabilistic approach is adopted- Kalman filter, Grid-based technique and Particle filters are to name a few. The presented work demonstrates the Adaptive Monte Carlo Localization(AMCL) technique and its complexity for global position estimation through simulation and analysis. Two different robot models are considered for performance evaluation and how the shape of the robot, location and number of sensors affect the localization is understood.

Index Terms—Robot, Kalman, Particle Filter, AMCL, Localization.

1 Introduction

Mobile robot localization encompasses many challenges out of which global position estimation and local position tracking are two fundamental issues. Both of these capabilities are necessary to enable a mobile robot to execute useful tasks, such as following a command to reach a specific room in a house or going back to charging station at the end of the day. By knowing its global position, the robot can make use of the existing maps, which allows it to plan and navigate reliably in complex environments. To resolve these challenges probabilistic approach is the most popular, but current methods still have some drawbacks. Although the Kalman filter can be amended in various ways to cope with some of these difficulties, recent approaches [1, 2, 3] have used richer schemes to represent uncertainty, moving away from the restricted Gaussian density assumption inherent in the Kalman filter. Bugard et al. introduced the grid based Markov localization approach which can represent arbitrarily complex probability densities at fine resolutions[4]. However, the computational burden and memory requirements of this approach are considerable. Recently Monte Carlo Localization(MCL) method has won attention to roboticists that were invented in the seventies [5], and rediscovered independently in the target-tracking [6], statistical [7] and computer vision literature [8,9]. MCL is a particlefilter based implementation of recursive Bayesian filtering for robot localization. In each iteration of MCL, the likelihood function is evaluated at sample points that are randomly distributed according to the posterior estimate of the robot location.It has several key advantages over Kalman filter which has been described in section (2.3).

2 Background

A key problem with particle filter is maintaining the random distribution of particles throughout the state space, which goes out of hand if the problem is high dimensional. In this work Adaptive Monte Carlo Localization(AMCL) method is used which converges much faster and is computationally much more efficient than a basic particle filter[10].The key idea is to bound the error introduced by the sample-based representation of the particle filter. To derive this bound, it is assumed that the true posterior is given by a discrete, piece-wise constant distribution such as a discrete density tree or a multidimensional histogram. For such a representation the number of samples are determined so that the distance between the maximum likelihood estimate (MLE) based on the samples and the true posterior does not exceed a pre-specified threshold. As is finally derived, the number of particles needed is proportional to the inverse of this threshold. Sensor models developed for MCL typically assume that the vehicle location x is known exactly; i.e.one of the sampling points corresponds to the true location of the robot. However, this assumption is only valid in the limit of infinitely many particles. Otherwise, the probability that the pose of a particle exactly corresponds to the true location of the robot is virtually zero. As a consequence, these likelihood functions do not adequately model the uncertainty due to the finite, sample-based representation of the posterior. In his paper[11] Patrick et al. introduced a adaptive sensor model that explicitly takes location uncertainty due to the sample-based representation into account. When evaluating the likelihood function at a sample location, the region a density estimation technique is considered to take into account while estimating the density at that location. The likelihood function applied to the sample was then computed by integrating the point-wise likelihood over the corresponding region.

So even though MCL based particle filter resolves lot of challenges in localization, it still has some limitations. For the above challenges problem domain is an important piece of robotics.

2.1 Kalman Filters

Kalman filter is an optimal estimation algorithm that predicts the parameter of interest such as location, speed and direction in the presence of noisy measurement. It is used to estimate the system state that cannot be measured directly. e.g.In a spacecraft combustion chamber a temperature sensor that is placed outside the chamber is used to find the best estimate of the internal temperature from an indirect measurement. Kalman filters can be used to estimate the state of the system by combining measurements from different sources that may be subject to noise. Common application of Kalman Filters are: guidance, navigation and control systems; computer vision and signal processing. What a Kalman filter does is to multiply the pdf(probability

density function) of predicted state estimate with pdf of the measurement with noise variance. The mean of the resulting pdf gives the optimal estimate of the state which has narrower variance than either of the variances of its inputs. However,if either the motion or measurement model or both are non-linear,then the variance of the resulting pdf is no longer a Gaussian. In fact, the distribution cannot be computed in a closed form(in a finite number of operations). The solution to this problem is using Extended Kalman Filter(EKF) where each small section of non-linear model is approximated to a linear function. This only applies to updating of the co-variance of the non-linear model. The mean is still updated with non-linear function.

2.2 Particle Filters

Particle filter is also known as Monte Carlo Localization (MCL) algorithm. Particle filtering uses a set of particles (also called samples) to represent the posterior distribution of a stochastic process given noisy and/or partial observations. In robotics, it is a virtual element that resembles a robot. The state-space model of the process(robot)can be nonlinear and the initial state and noise distributions can take any form required. The samples from the distribution are represented by a set of particles; each particle has a likelihood weight assigned to it that represents the probability of that particle being sampled from the probability density function. In the re-sampling step, the particles with negligible weights are replaced by new particles in the proximity of the particles with higher weights. Thus particles with larger weights are most likely to survive in the re-sampling process. After several iteration of the algorithm, and after different stages of re-sampling, particles will converge and estimate robot's pose. Particle filters are mainly used to solve robot's global localization challenge where robots initial pose is unknown and robot need to find its pose relative to the ground truth map.

2.3 Comparison / Contrast

With Kalman Filter(EKF) localization algorithm one can estimate the pose of almost any robot with accurate onboard sensors. But Particle filter (MCL) has many advantages over EKF:

- MCL can estimate almost any state space distribution.
- The posterior of MCL are particles which is easier to generate than a gaussion fuction, which is the posterior of EKF.
- MCL is able to handle almost any measurement noise whereas EKF can handle gaussian noise only.
- MCL can be used to resolve global localization problems
- MCL has control over computing memory and resolution by varying the number of particles.

For the above reasons the work presented here will be using only particle filters.

3 Simulations

For the purpose of the project a ROS package was used in Ubuntu OS, that launches a custom robot model in a Gazebo world and utilizes packages like AMCL and the Navigation Stack. These packages would accurately localize a mobile robot inside a provided map in the Gazebo and RViz simulation environments. Then specific parameters corresponding to each package were tuned to achieve the best possible localization results.

environment.png

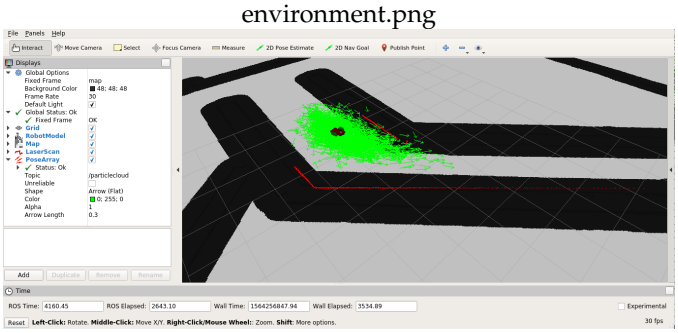

Fig. 1. Simulation

3.1 Achievements

This project demonstrates how AMCL(Advanced Monte Carlo Localization) technique effectively determines the pose of a mobile robot fusing camera and laser sensor data while the robot is in motion. Once robot's current pose is determined, it can effectively plan its path for a new navigation goal. A more detailed description of the robot model that is being used in the project is given below.

3.2 Benchmark Model

3.2.1 Model design

Robot's design considerations are as follows:

- Chassis geometry, mass and color: box(40x20x10 cm^3), 15 kg and Red
- Left Wheel geometry, mass and color: cylinder(L-10cm R-5cm), 5 kg and Green
- Right Wheel geometry, mass and color: cylinder(L-10cm R-5cm), 5 kg and Green
- Camera geometry, mass and color: cylinder(5x5x5 cm^3), 100 g and Blue
- Hokuyo(Laser Sensor) geometry, mass and color: Hokuyo Mesh File, 100 g and Red

TABLE 1
Robot model Joint parameters

Joint Name	Origin(XYZ)	Type	Parent	Child
Robot Footprint	0 0 0	Fixed	Robot Footprint	Chassis
Left Wheel Hinge	0 .15 0	Continuous	Chassis	Left Whe
Right Wheel Hinge	0 -.15 0	Continuous	Chassis	Right Wh
Camera Joint	0 .2 0	Fixed	Chassis	Camera
Hokuyo Joint	.15 0 .1	Fixed	Chassis	Hokuyo

3.2.2 Packages Used

For ROS simulation project Catkin Workspace and Catkin Packages has been used. A unique package called "Udacity_Bot" was built for the purpose of the project, where

xacro package was used to generate the URDF from the xacro file.Following are the topics that the udacity_bot package created for the camera and laser node subscription/publication:

- /udacity_bot/camera1/camera_info
- /udacity_bot/camera1/image_raw
- /udacity_bot/camera1/image_raw/compressed
- /udacity_bot/camera1/image_raw/compressed /parameter_descriptions
- /udacity_bot/camera1/parameter_descriptions
- /udacity_bot/camera1/parameter_updates
- /udacity_bot/laser/scan

For simulation gazebo_ros package spawned the robot model from the URDF that robot_description launch file helped generate. In the robot_description launch file, two nodes are added. One node uses the package: joint_state_publisher that publishes joint state messages for the robot, such as the angles for the nonfixed joints. The second node uses the robot_state_publisher package that publishes the robot's state to tf (transform tree).The project robot model has several frames corresponding to each link/joint. The robot_state_publisher publishes the 3D poses of all of these links.

3.2.3 Parameters

Default Localization parameter values in the AMCL(Adaptive Monte Carlo Localization) node are shown in Figure 2 and the values after modification are shown in Figure 3. AMCL launch file has three nodes: one

parameter list Before1.png

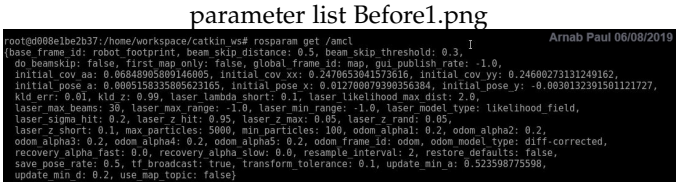

Fig. 2. AMCL default parameter list

parameter list1.png

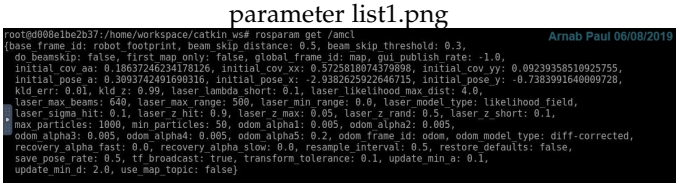

Fig. 3. AMCL tuned parameter list

of which is for the amcl package.The move_base package was used to define a goal position for robot in the map, and the robot will navigate to that goal position.Following parameters listed in the amcl package was used to tune the laser scanner model(measurement) and odometry model (motion):

- 1.”laser_z_hit”: 0.9,,
- 2. ”laser_sigma_hit”: 0.1,
- 3. ”laser_z_rand” :0.5,
- 4. ”laser_likelihood_max_dist”: 4.0

For the odometry model, parameters are tuned so that the algorithm assumes there is low noise in odometry:

- 1 ”kld_err”: 0.01,
- 2 ”kld_z”: 0.99,
- 3 ”odom_alpha1”: 0.005,
- 4 ”odom_alpha2”: 0.005,
- 5 ”odom_alpha3”: 0.005,
- 6 ”odom_alpha4”: 0.005

The following costmap parameters and set values were used to control laser scanner interaction with global/local truth map:

- obstacle_range: 1.2 (The default maximum distance from the robot at which an obstacle will be inserted into the cost map in meters.)
- raytrace_range: 1.4 (The default range in meters at which to raytrace out obstaclesfrom the map using sensor data)
- transform_tolerance: 1.0
- inflation_radius: 0.35 (controls how far away the zero cost point is from the obstacle)
- observation_sources: laser_scan_sensor

3.3 Personal Model

The personal model design, packages used and parameters were similar to that of the bechmark model except the following key differences:

- Chassis geometry, mass and color: box(60x20x10 cm^3), 15 kg and Blue
- Left Wheel geometry, mass and color: cylinder(L-10cm R-5cm), 5 kg and Red
- Right Wheel geometry, mass and color: cylinder(L-10cm R-5cm), 5 kg and Red
- Camera geometry, mass and color: cylinder(5x5x5 cm^3), 100 g and Green
- Hokuyo(Laser Sensor) geometry, mass, color and joint origin: Hokuyo Mesh File, 100 g, Blue and 0 0 0

4 RESULTS

Analysing the localization result(pose) it could be concluded that both the robots were successfully able to plan path and reach the goal pose. However, the time to reach the goal from the starting point varied between 2 types of robot. It took almost 5 minutes for both the robots to reach the goal from the starting point. Sometimes the robots were stuck by the obstacles and the parameters for cost map were tuned to find the optimum result for smooth navigation.

4.1 Localization Results

4.1.1 Benchmark

Figure 4 shows the bechmark robot model have reached the goal successfully.

4.1.2 Student

Figure 5 shows the Student robot model was able to localize itself and through correct path planning have reached the goal successfully.

reached goal12.png

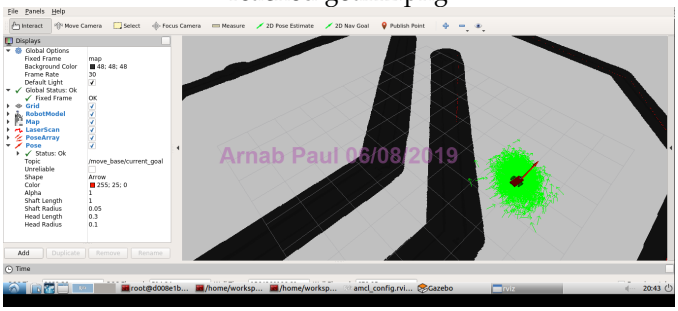

Fig. 4. Bechmark Robot at Destination

reached goal22.png

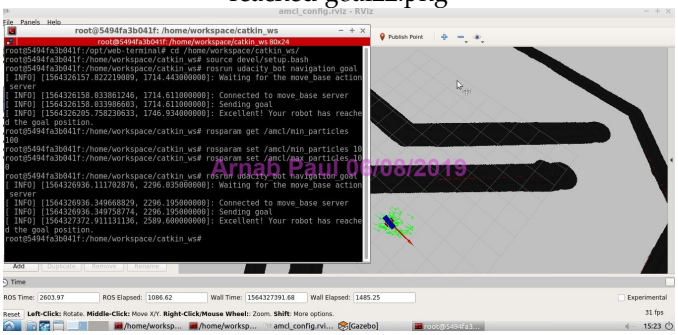

Fig. 5. Student robot at Destination

4.2 Technical Comparison

The presented robot differ to the benchmark robot by color, dimension of the chassis and location of the laser sensor. It is found that the dimension of the chassis and location of the laser sensor contributed to the variation in performance between the two robots. The student robot could not navigate through a narrow passage smoothly with same global cost map parameters since the origin of the sensor is comparatively far from the front surface of the chassis. The following parameters were updated to overcome this problem:

- obstacle_range
- raytrace_range
- inflation_radius

However, the student robot showed increased stability while moving from one point to another as the laser sensor were placed right above the origin and its center of gravity aligned with that of the chassis.

5 Discussion

It can be concluded that the student robot performed better in comparison to the benchmark robot for reason described in the previous section. Also, it was able to reach the goal with less number of particles(figure 6) and most particles posing the same direction which indicates that it was more certain about its pose. The possible reason again could be positioning of the laser sensor at the center of the chassis, therefore, the pose measured by the sensor was equal to the true pose of the robot. The localization approach showed in this work can be useful to resolve 'Kidnapped Robot' problem. The Kidnapped Robot problem in robotics commonly refers to a situation where an autonomous robot cannot locate itself against the map. It can be caused by external force when a robot is carried to an arbitrary place in the map. The solution could be calibrating the robot's pose by amcl node using feature matching technique and thus localizing the robot in the map. MCL/AMCL could be of great use for rescue robot(both ground and flying) when it is trying to navigate through an unstructured environment. It could also be used for home assistant robot where the component of the map may not be static and the robot could still localize and plan path for navigation in the dynamic environment.

6 Conclusion / Future work

The work presented here shows an application of AMCL to localize itself and demonstrate to identify a goal pose and reach to the goal fusing two sensors data: a laser scanner and a camera in a simulated environment. It also shows how the construction of the robot and sensor location on the robot affect the performance of the localization by comparing two different construction of robots. This project could be taken further by experimenting with different laser sensors and determining the key characteristics that provides the best result. Also analyzing how the accuracy of the localization improve by introducing more sensors and fusing those sensors data to determine the robot pose. This project could be advanced further by building a prototype according to the design used in the simulator and deploying it to the hardware. And then comparing the experimantal result with the simulation. This will provide an insight how the system/parameters could be modified in order to improve performance. For real time application, the hardware must contain a high speed GPU to accomodate time to compute the path plan and must be able to interface the camera and laser sensors to the processor, e.g. NVIDIA Jetson Tx2 hardware developement kit could be explored for this purpose.

References

[1] I. Nourbakhsh, R. Powers, and S. Birchfield, DERVISH an officenavigating robot, AI Magazine 16, pp. 5360, Summer 1995.

[2] R. Simmons and S. Koenig, Probabilistic robot navigation in partially observable environments, in Proc. International Joint Conference on Artificial Intelligence, 1995.

[3] L. P. Kaelbling, A. R. Cassandra, and J. A. Kurien, Acting under uncertainty: Discrete Bayesianmodels for mobile-robot navigation, in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 1996.

[4] W. Burgard,D. Fox, D. Hennig, and T. Schmidt, Estimating the absoluteposition of a mobile robot using position probability grids,in Proc. of the Fourteenth National Conference on Artificial Intelligence(AAAI-96), pp. 896901, 1996.

[5] J. E. Handschin, Monte Carlo techniques for prediction and filtering of non-linear stochastic processes, Automatica 6, pp. 555563,1970.

[6] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, Novel approach to nonlinear/non-GaussianBayesian state estimation, IEE Procedings F 140(2), pp. 107113, 1993.

[7] G. Kitagawa, Monte carlo filter and smoother for non-gaussian nonlinear state space models, Journal of Computational and Graphical Statistics 5(1), pp. 125, 1996.

[8] M. Isard and A. Blake, Contour tracking by stochastic propagation of conditional density, in European Conference on Computer Vision, pp.343356, 1996.

[9] M. Isard and A. Blake, Condensation conditional density propagation for visual tracking, International Journal of Computer Vision 29(1), pp. 528, 1998.

[10] Dieter Fox, "KLD Sampling: Adaptive Particle Filters", Department of Computer Science & Engineering,University of Washington,1998.

[11] P. Pfaff, W. Burgard & Dieter Fox, "Robust Monte Carlo Localization using Adaptive Likelihood Models", Department of Computer Science, University of Freiburg, Germany, Department of Computer Science & Engineering,University of Washington, 2006.