Particle Filter-Based Localization for Mobile Robots

Jonathan Atiene - p2839161@my365.dmu.ac.uk

1. Introduction

Localization enables a robot to estimate its position and orientation (pose) within a known map. This report details the implementation of a particle filter-based localization system for a simulated Pioneer robot using ROS. The particle filter algorithm integrates odometry data and laser scan data to estimate the robot's pose probabilistically. The system was tested in a simulated environment using Gazebo and RViz.

2. Software Implementation

The particle filter algorithm estimates the robot's pose using a Monte Carlo approach, it operates as follows;

2.1 Prediction Step

In the prediction step, particles are propagated based on odometry data, which provides information about the robot's movement. To account for uncertainty in the motion model, Gaussian noise is added to the particle positions and orientations. The noise parameters (`ODOM_ROTATION_NOISE`, `ODOM_TRANSLATION_NOISE`, and `ODOM_DRIFT_NOISE`) control the spread of particles.

2.2 Prediction and Update

The `update_particle_cloud` method updates the particle weights based on the likelihood of matching the observed laser scan data. The sensor model computes the weight for each particle using the `get_weight` function. Particles with higher weights are retained during resampling, while low-weight particles are discarded.

Resampling is performed using roulette-wheel selection, ensuring that high-probability particles dominate the new particle cloud. To maintain diversity, Gaussian noise is added to the resampled particles.

2.3 Pose Estimation

The `estimate_pose` method computes the robot's estimated pose by averaging the positions and orientations of the particles. Quaternions are used to calculate the average orientation, avoiding issues with angle wrapping.

The resulting quaternion represents the average orientation of the particle cloud, providing a robust estimate of the robot's heading.

3. Testing and Results

The system was tested in a simulated environment using Gazebo and RViz. The robot was controlled using keyboard teleoperation to generate odometry and laser scan data. Observations include: the particles were able to converge and were able to get the estimated location of the robot this is shown in Figure 3.

Convergence and Effect of parameters

As the robot moved, the particle cloud converged toward the true pose. High-weight particles dominated after resampling, improving the accuracy of the pose estimate. The estimated pose aligned closely with the robot's actual position on the map. The table below shows the effect of parameters on the convergence rate of the particles.

Table 1: Parameter Tuning for Localization Modes

Mode	**Rot. Noise**	**Trans. Noise**	**Drift Noise**
Fast Init.	0.08	0.12	0.05
Stable Track	0.04	0.07	0.03
Balanced	0.05	0.10	0.05

Accuracy

The figure 5 shows the particle tracking system demonstrates initial stability in both X and Y directions for the first few seconds, followed by increased spread variability reaching peaks over 1.0 standard deviation. the shows the influence of noise and the effect of movement on the particle positioning. The final particle distribution reveals distinct clustering patterns across the coordinate space, with major clusters forming around the robot, suggesting physical constraints in the system.

5. Conclusions

The implemented particle filter successfully localizes the robot within a map, leveraging odometry and laser scan data. Key strengths include robustness to sensor noise, adaptability to dynamic environments, and seamless integration with ROS. Limitations include sensitivity to parameter tuning and reduced accuracy in featureless areas. Future work could explore finetuing to ensure quicker convergence.

6. Bibliography

- Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic Robotics.* MIT Press.
- ROS Wiki: Particle Filter Localization. Retrieved from `http://wiki.ros.org`.
- Quigley, M., et al. (2009). "ROS: An Open-Source Robot Operating System." *ICRA Workshop on Open Source Software.*
- Fox, D., Burgard, W., & Thrun, S. (1999). "Markov Localization for Mobile Robots in Dynamic Environments." *Journal of Artificial Intelligence Research.*

1 Appendix

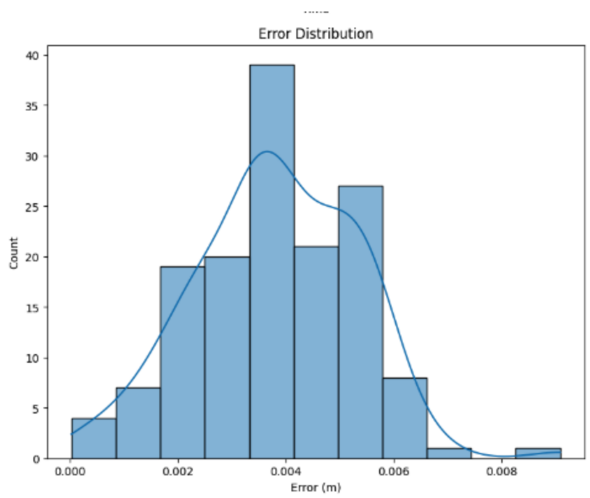

Figure 1: Error distribution between actual position and particle position over time.

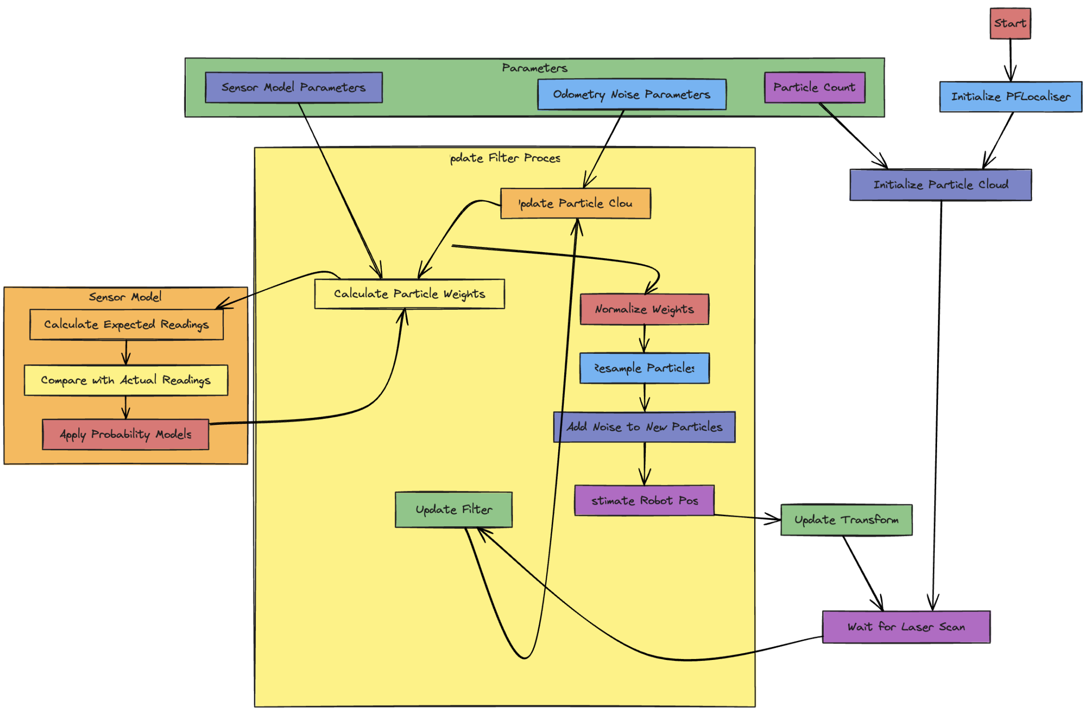

Figure 2: Overview of the particle filter localization process.

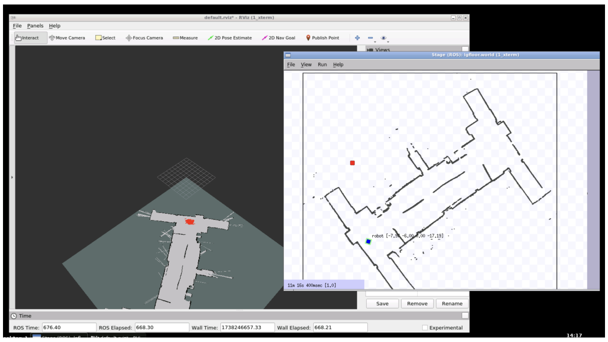

Figure 3: Particle cloud and estimated pose of robot in RViz.

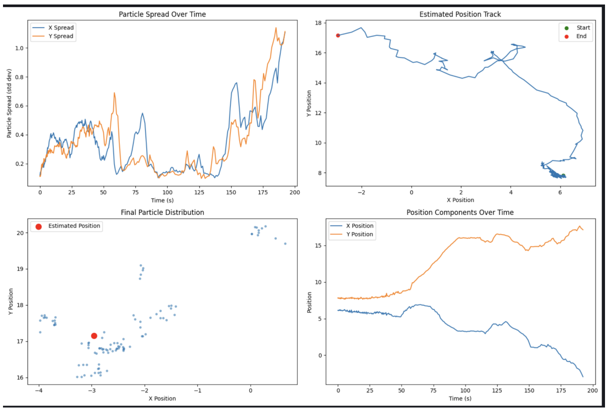

Figure 4: Overview of performance and accuracy before convergence.

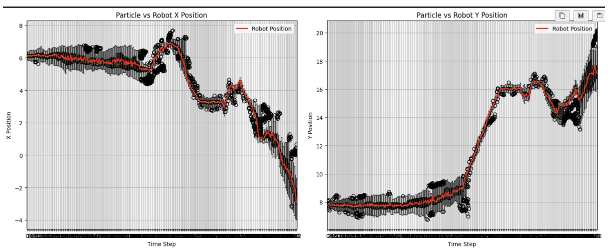

Figure 5: Showing the accuracy of the particles in locating the robots, this plot visualizes the path of the robot and position of particles in bot x and y axis