# A Simple Implementation of Feature-Based FastSlam

*Zipei Wei*

***Abstract*** This paper is a simple implementation of the FastSlam algorithm based on the thesis paper published by Michael Montemerlo. As by the definition of SLAM, Simultaneous Localization and Mapping is one of the most important concepts in mobile robotics. The difficulties of implementing a SLAM algorithm are not only limited by the algorithm itself, but also the limited resources, semantic terms that are not friendly to beginners. Thus, this paper does not only demonstrate the Fastslam algorithm implementation using Gazebo and ROS but also utilizes the HCI principles to provide more user-friendly guidelines for new beginners who want to study robotics.

## I.INTRODUCTION

By the definition of SLAM, SLAM means Simultaneous Localization and Mapping. SLAM is one of the most difficult topics robotics area due to the facts: the computation needed for implementation of a SLAM algorithm and the uncertainty in every move could cause the wrong prediction in results. Furthermore, due to the fact most robots don't have visual aid like human's eyes, it is hard to link different landmarks between different time frames. In this project, I will use a pioneer-3dx robot in Gazebo for simulation and implement the FastSlam 1.0 algorithm with unknown data association. The detection of the landmarks and the localization process will be visualized in Rviz using Markers. After the implementation of the basic FastSlam algorithm with unknown data association, the results of the localization and mapping will be discussed in the discussion section and a conclusion what could be the future work.

## II.RELATED WORK

In 2002, Sebastian Thrun published his paper "FastSlam: A Factored Solution to the Simultaneous Localization and Mapping". In his paper, he proposed by estimating the locations of the landmarks to draw the feature-based map and using a 2 by 2 Extended Kalman Filter for localization (Sebastian Thrun, 2002). The great part about FastSlam is: it can grow as the world grows before the computer run out of memory. However, since this paper is published for the Robotics experts, many of the terms such as Extended Kalman Filter are not provided with enough details and reasoning. In 2003, one of his Ph.D. students published his thesis paper "FastSlam: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association" which gives much more great detail about how FastSlam works that are 123 pages long (Michael Montemerlo, 2003). This paper reveals why the reasons for using a 2 by 2 Extended Kalman Filter for localization and arguments about how to solve the unknown data association.

The ideas of Kalman Filter was first published by R.E. Kalman in 1960 "A New Approach to Linear Filtering and Prediction Problem" (R.E.Kalman, 1960). In a concise summary, Kalman Filter is a technique for estimating a certain state using joint probability distribution when not all observations were given. Extended Kalman

Filter is just a nonlinear version of Kalman Filter. Thus, to really understand how FastSlam, one not only needs to understand the high-level concept of FastSlam but also need to research on the concepts of Extended Kalman Filter, Maximum Likelihood Estimation and range and bearing calculation using sensor data.

III.PROJECT DESCRIPTION

In Industrial Engineering, there are a group of engineers that specialize in workspace design that the job itself required high-concentration and zero tolerance of mistakes (Such as Nuclear Power Plant, NASA mission control center). The thumbs of rules in terms of study and learning rate are: knowledge present in a concise format is easier for human brains to memorize and the learning rate can grow exponentially with the repeat of study. Thus, it is easier for new beginners to acquire the knowledge of FastSlam in a concise concept and later on, to dive down into different sections to evaluate the algorithm itself as well as to gain a deeper understanding of each concept through iteration of studying. In this project, the FastSlam algorithm is implemented using 20 particles with a linear motion on one direction in space.

The framework of FastSlam contains four major steps (Figure 1). Started from the high-level summary, each step will be discussed with details and how it can be achieved.

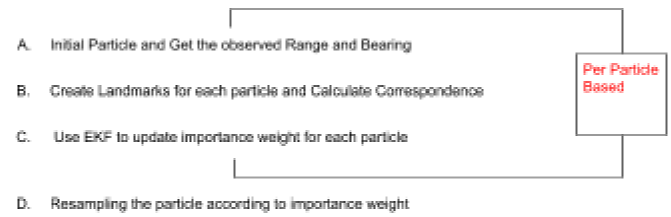



**Figure 1. State Diagram**

*A. Initial Particle and Get the observed Range and Bearing*

The first part is to initiate the particle cloud and obtained the observed range and bearing from the sensor. FastSlam uses a similar data structure like Monte Carlo Particle Filter where in the beginning, a group of particles will be random sampling across the world. In this project, 20 particles will be sampled with random x and y coordinate. Then as the robot moves, each particle location will be updated pose with a Gaussian distribution of velocity (cmd). The next step is to obtain the range and bearing for observed landmarks. This project was used the /frontscan from the Pioneer-3dx that contains 520 laser reading of each time step. As not all sensor reading observed the landmark, a sensor filter applied to only recording the readings that have the real reading (When sensor reading equals “inf”, it means nothing on the other side of the sensor) and calculate the corresponding bearing. A tan function is used to estimate the landmarks’ locations according to the sensor’s range and bearing (Figure 2)

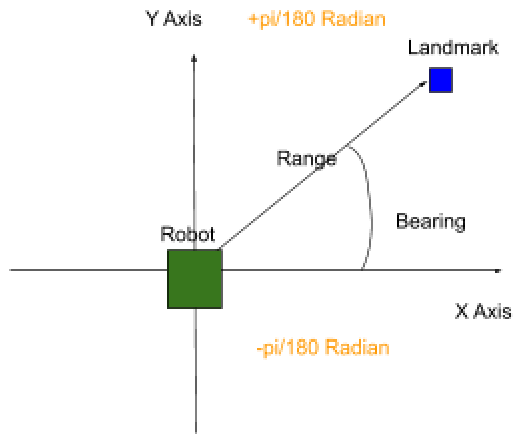



**Figure 2. Find Range and Bearing**

B. *Create Landmarks for each Particle and Calculate Correspondence*

After the observed landmarks had been obtained from the sensor and the initialization of particle cloud. Each observed landmark will be uploaded to each particle where each particle contains: x and y coordinates, particle weight and a list of particles. The next thing we need to find is the correspondence/data association between each time step. By the meaning of correspondence, it refers to the correspondence of landmark's location at each time step. For instance, at time = 0, 5 landmarks labeled z = [z1, z2, z3, z4, z5]. As the robot moves time = 1, only 3 landmarks observed z = [z4, z5, z6]. Maximum Likelihood estimator is used to solving the data association problem.

The basic idea of Maximum Likelihood is given a data point X with mean and standard deviation, using a Gaussian distribution to locate the probability of data point X. Thus, finding the data association is pretty straightforward: At time = 0, each particle is associated with each own landmark lists. As at time = 1, a list of new observed landmarks was generated. For each of the new observed landmark, compared with the current landmark list on the particle, generate a likelihood if the current observed landmark is one of the landmarks appended in the particle. If the likelihood is greater than a threshold (0.001) which means it is likely this is the existed landmark, then update the mean and covariance matrix of the associated landmark on the particle. If the likelihood is smaller than the threshold, then it should be a new landmark and the new landmark is added to the particle.

C. *Use EKF to update importance weight for each particle*

As part B solved the data association problem, part C will use the Kalman Filter to calculate the importance weight. From the definition of Kalman Filter, an importance weight illustrate how confidence this particle location is relevant to the real robot pose. By this means, a higher importance weight is indicating the particle's pose is more likely to be the real robot pose. Initialize, each particle is assigned to an estimate weight p0. As each particle was iterate through the observed landmark list, weight is updated using the covariance matrix that was calculated by Kalman Filter. The Kalman filter actually reduces the uncertainty of movements due to the fact is a joint probability product. In Figure 3, a high-level summary showed by Kalman Filter works in general cases. As a quick summary, since we cannot directly measure the real pose, we will use the indirect measurement ( landmarks' location) to update our belief of how far are we to the real pose.

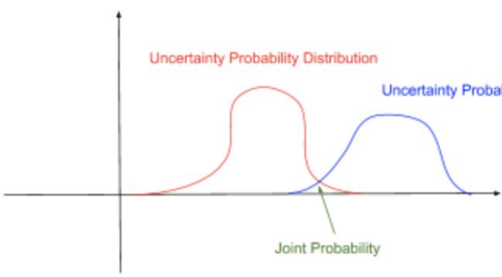



**Figure 3. Kalman Filter**

D. *Resampling the particle according to the importance weight*

As part C and part B are executed, the particle cloud is constructed with a list of particles associated with each belief. Then the particle cloud is resampling according to particle weight. The resampling process is exactly the same as the one we learned in class, which is calculated probability distribution, normalized the probability distribution, calculate the cumulative probability distribution and then use a random probability to assign the particle according to the cumulative probability distribution. By doing this, a particle with a smaller weight will be replaced with the particle with a larger weight.

IV. ANALYSIS OF RESULTS

After constructed the FastSlam algorithm, the FastSlam is being tested using a virtual world created by Gazebo. The virtual world is contained with different objects such as fire hydrant, trees, water fountains and bookshelf (Figure 4). Rviz is used to visualize the results of FastSlam, where the green cube is the real robot pose, red spheres are the particles' trajectory and

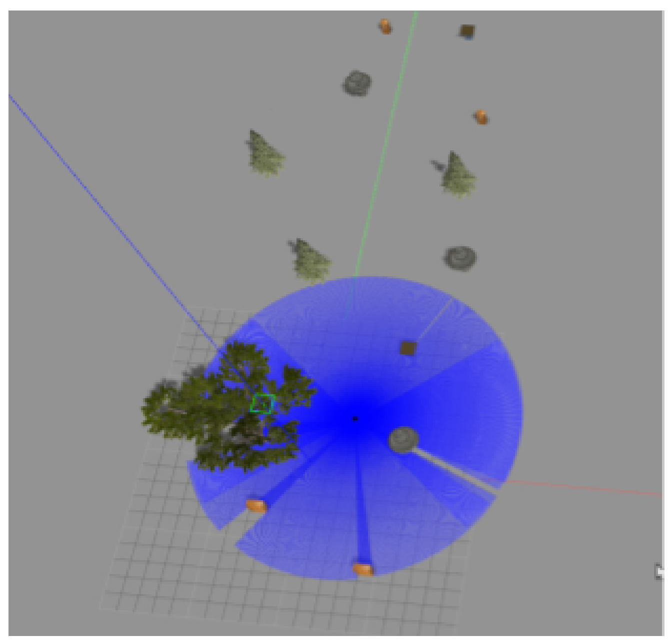
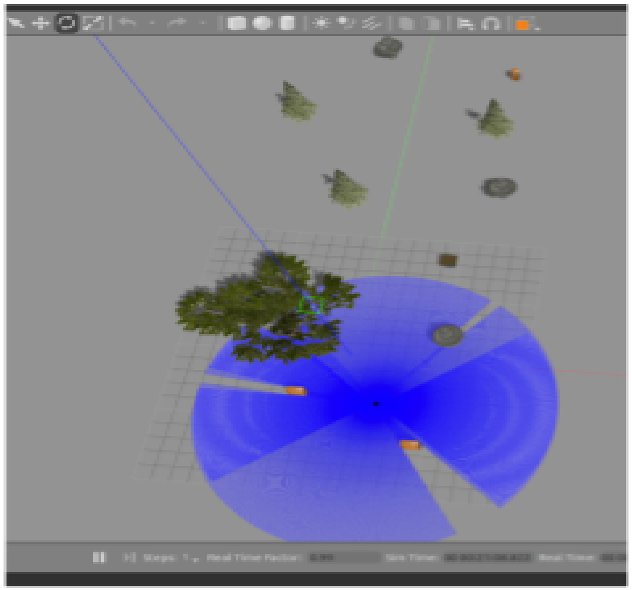


**Figure 4. World created in Gazebo**

blue spheres are marked as landmarks. As the robot moves in one direction with a constant speed (0.05), we can clearly see the initialization of the particles clouds followed by the resampling procedure, where the particles merged into one robot will be read in as well which will increase computation cost. Besides that,



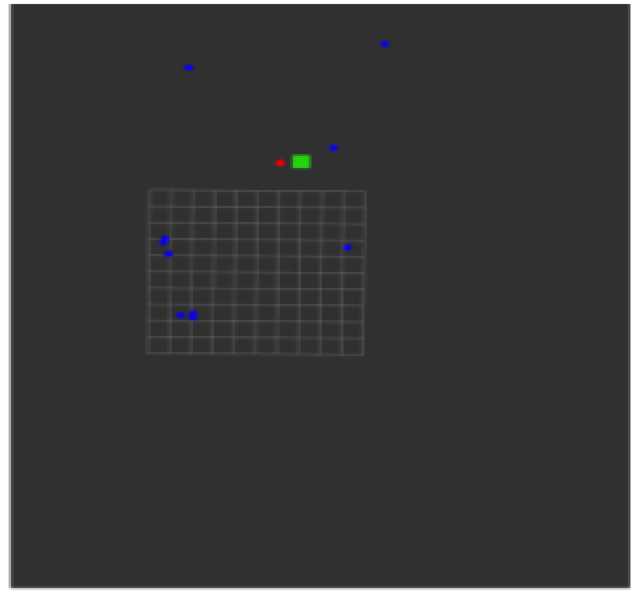


**Figure 5. Comparison Chart a) World in Gazebo. b) Feature-Based map in Rviz**

trajectory. As time increase, the feature-based FastSlam generates the landmarks map similar to the world simulated in Gazebo. Figure 5 is the comparison chart of the world and the feature-based map in Rviz.

## V. DISCUSSION

As the result shown above, we can clearly see that FastSlam generated a great feature-based map without any visual aids. In terms of the computation efficiency, FastSlam achieves the great result by eliminating the useless sensor readings ("Inf") as well as by improving the Kalman filter into 2 by 2 (Solving problems in a smaller subset). Another aspect to consider is the FastSlam is better used as in outdoor environment due to the fact of the sensor filter processing. If FastSlam was used indoor, all the walls on the two sides of the one thing to notice is, in this project, the resampling process is exactly the same process as the functions we used in ps 5, wherein the real world, it is always a good idea to only sample partial particle clouds to prevent sampling impoverished. Since this project is a much simpler solution that doesn't encounter loop closure and nonlinear movement, a FastSlam 1.0 with Unknown data association is sufficient enough.

As FastSlam is a 2D mapping problem, one can easily extend the scenario into 3D mapping with a camera and the point cloud, but one will soon to find the most obstacle in robotics is the computation expensive. As more data points were read in, the matrix of calculation can grow exponential.

## I.CONCLUSION

Since this project is implemented FastSlam in a relatively simple way, one can later improve the FastSlam algorithm to FastSlam 2.0 for computation efficiency, besides that the world can set to be a more complicated scenario by adding addition landmarks. As for the resampling process, one can resample the particles but by keeping a small set of particles during the mapping process to prevent sampling impoverished. In the same time, the motion model can be set to nonlinear with loop closure to evaluate the result. In summary, it is a great and fun project to help one to learn the SLAM algorithm in a relatively simple way.

As a conclusion, SLAM is still a crown jewel in the robotics area that needs more exploration. If you dive deep into the SLAM algorithm, you would find each algorithm is trying to improve the computation efficiency from one to the other by using different mathematical models in machine learning and mathematics. People who have the data science background may have a leading start of the project due to the understanding of the concepts, but to really finish the project, one will need more than that: such as understanding ROS platform, the main structure of the algorithm and certain levels of programming experience. Through a systematic study, everyone can do SLAM.

## REFERENCES


1. Michael Montemerlo, Sebastian Thrun, Daphne Koller, and Ben Wegbreit. 2002. FastSLAM: a factored solution to the simultaneous localization and mapping problem. In Eighteenth national conference on Artificial intelligence, Rina Dechter, Michael Kearns, and Rich Sutton (Eds.). American Association for Artificial Intelligence, Menlo Park, CA, USA, 593-598.
2. Michael Steven Montemerlo. 2003. Fastslam: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association. Ph.D. Dissertation. Carnegie Mellon Univ., Pittsburgh, PA, USA. AAI3102462.
3. R. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.