Probabilistic Robotics Lab Report
Extended Kalman Filter Localization

Raabid Hussain

April 20, 2016

1 Overview

The objective of this lab was to implement the extended kalman filter for localization of the robot in a given map. The program was tested using turtlebot simulator. A bag file containing the environment was provided along with the robot instructions for moving. We had to predict the current position of the robot using kalman filter. The walls in the map were represented as lines using the code for the previous lab in which split and merge technique was used to define geometry of the walls using straight lines. These lines were used as measurement input to the filter for localization.

2 Methodology

The algorithm implemented in this lab work is based on the kalman filter which is an algorithm that predicts the current state of the robot based on the bayes' approach. It works by estimating a probability distribution over the states for each time lapse. Like the previous filter used, it also updates the state variables according to the motion of the robot and measurements from the environment.

Extended kalman filter (EKF) is an extension of this kalman filter. It can be termed as the non-linear version of the kalman filter. A critical step in the EKF algorithm is the reasonable estimation of the initial state and the model of the robot. If this step is not done correctly, the filter is often seen to diverge quickly. The EKF algorithm can be divided into the following main steps:

2.1 Prediction Step

The first step of the algorithm is to predict the state of the robot based on the motion parameters. In the EKF algorithms, the predict step can be divided into two steps mainly relating to the prediction of the robot position and the prediction of the uncertainty of the robot's position. The uncertainty is represented by an ellipse. When we ran the code without the predict step, the robot did not move. So the first thing was to make the robot move. For this we had the motion in the robot frame and the current state of the robot in the world coordinate system. So, a simple transformation was made to the motion measurement and they were added to the current position to get the new state of the robot.

Since there is always some uncertainty in the measurements, an uncertainty cloud needs to formed which would represent all the possible states the robot can be in, provided a model for uncertainty. For this equations from the lecture notes were used. In the equations, jacobians of the state of the robot and the noise model were used to predict the cloud. Now after writing the prediction step and running the code, it was observed that the robot started moving in the appropriate direction and the uncertainty cloud grew bigger and bigger as the robot moved as the update step had not yet been written and more uncertainty is added in each iteration as the robot moves.

2.2 Data Association Step

The next step is called data association in which the best correspondences are found between the measured lines in the environment with the lines in the given map of the environment. This is the most important step in the algorithm as if carried perfectly, only then the update step will be meaningful.

In order to get the closest match, mahalanobis distance was used as uncertainties in both measurements should be taken into account. So, for each measured line, the distance is measured with all the lines in the known map. The smallest distance among all of them was compared to a set threshold.

Only if this smallest distance was lesser than that threshold, then the line was associated and taken into consideration. For this step equations from the lab instruction manual were used. During this step, an additional function for the jacobian was also created which also took care of the transformations involved. Noise certainties were also added.

For the optional part, the euclidean lengths of all the lines were measured while associating the the data. Like in the previous lab, if the measured line was found to be bigger than the map line, it was ignored as obviously, as it hint that there is some discrepancy in the measured line data.

Ofcourse, as in this step the output of the algorithm is not updated so, no change in the robot movement was observed.

2.3 Update Step

The final part of the algorithm is the update step in which the position and the uncertainty model of the robot are updated. The matches obtained from the data association step are used to do this update. The equations from the lab instruction manual were used for this step to do the update of the current state of the robot. A point to note was that here more than one observation might have to be used to do the update. After implementing the equations and running the code, it was observed that the robot position was updated more correctly now and the uncertainty ellipse, starting decreasing in size as the measurements were associated now. This provided a more reasonable and precise estimate of the robot position.

3 Results

After writing the code for all the stages, it was time to test them. The circle representing the probability distribution of the current state of the robot can be seen to decrease in size as the robot moves, thus giving us a more precise location of the robot. However, at some instances, it grows bigger due to the measurements observed. In general it worked good on my laptop everytime except every first time which proved to be quite irritating at times. Also, the behavior of the distribution ellipse was observed to be different sometimes. this was probably due to established communication channel as the computer is both sending and receiving data. A sample output is shown in the following figure:-

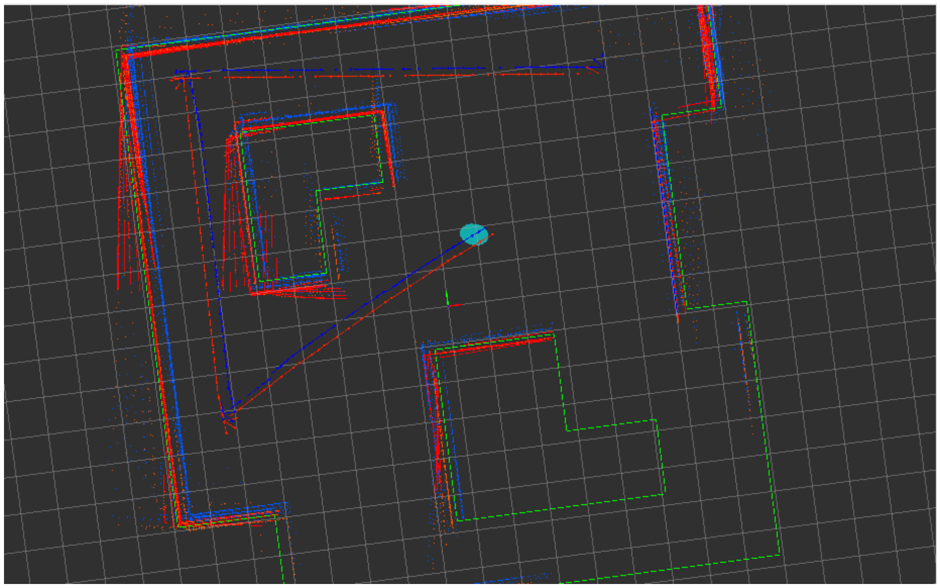

Figure 1: Output of the algorithm at a particular moment in time

4 Conclusion

In this lab work, an extended kalman filtering approach was used for the localization of the robot using ROS environment. The algorithm was divided into three main functions, each representing its main steps. Lecture notes were consulted for the equations used. The state of the robot was initially predicted, the measured data associated with the model and map and the algorithm concluded with the update of the current state. The algorithm was found to behave properly in almost all of the test runs. For this lab more weeks were allocated which proved helpful in completing the tasks.