Probabilistic Robotics Lab Report
EKF based SLAM

Raabid Hussain

May 16, 2016

1 Overview

The objective of this lab was to implement an extended kalman filter based simultaneous localization and mapping (SLAM) algorithm The program was tested using the turtlebot simulator. A bag file containing the observations and the robot motion was provided to us and we were to build a map of the environment. The features in the map and the current robot position were approximated using the extended kalman filter. For measurements, robot encoders and kinect module were used. The obstacles in the map were represented using infinite lines.

2 Methodology

Simultaneous localization and mapping (SLAM) is the process of constructing the map of an unknown environment and simultaneously keeping track of the robot location within the created map as it is being updated. Although this a tedious problem, many solutions have been proposed over time. One of the most widely used solutions is the use of extended kalman filters (EKF). There are two types of EKF based SLAM algorithm: one that uses observed features to localize the robot while the other uses previous robot motions to localize.

We were to implement the feature based SLAM algorithm. It involves approximating the new robot position by using the odometry (with the help of encoders) values. The new positions of the already sensed features are also predicted in this step. Next the new observed features from the map are saved and their association with the already stored features is found to see if there is a match. After this the associated features are used to update the feature and robot positions. The unassociated feature are then added to the feature/state vector.

A similar approach as the one we followed during the EKF implementation in the previous lab was followed here. All the equations used were taken form the lab instruction manual and the lecture notes. The algorithm can be divided into four main steps:-

2.1 Prediction Step

The first step in the algorithm is to predict the robot position and the features already present in the state vector. The state vector consists of the robot position parameters followed by the observed features positions. Without writing this step, it was observed that nothing was happening because the robot position is not being updated. The prediction step, first compounds the robot position according to motion measures and then computes the uncertainty of the position based on a gaussian window and the covariance(using jacobians) of the features with each other and the robot. After coding the prediction step, it was observed that the robot starting moving and the uncertainty grew as the robot moved on.

2.2 Data Association Step

Features are observed through the kinect sensor and the next step is to determine which of the newly observed features is already in the state vector and which ones need to be added. For this, the observed lines are first transformed into polar coordinates. Since we are dealing with gaussian uncertainties in both robot position, already observed features and the new observed features, here mahalabonis distance was used to calculate the distance between all the already observed and newly observed lines. If the distance was found to be less than a threshold the new line was associated with its closest line in the state vector whereas the other lines were passed onto the state augmentation step. The main problems faced in this step were the tricky jacobians and the transformations between the world and the robot frames. The observed features were in robot frame and the features stored were in world frame however, during comparisons for different matrices different coordinate systems had to used.

2.3 Update Step

After associating the observed with with the already stored features, it was time to update the robot position and the positions of the features in the state vector. Matrix equations from the lecture notes were used. A point to note was that all observations that are already in th estate vector are used to update not only the robot position but also the features themselves. 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. Also the lines representing the features started to align themselves with the actual location of the obstacles.

2.4 State Augmentation Step

The final step was to augment the state vector with the newly observed but unassociated features. For the state vector, it was a simple case of stacking the new features at the bottom, however, for the uncertainty, the co-variance with the robot position had to be found before updating. Since all the lines are independent of each other so their co-variance was forced to zero. Since, the data provided in the bag file is of real circumstances, few fake lines are observed. In order to omit them from adding to the map, a count was kept for each observed lines. If the same line is observed for more than a certain number of times, only then it is added to the state vector.

3 Conclusion

In this lab work, A feature based EKF implementation of the SLAM algorithm was accomplished. Different functions, each dedicated to a particular step/operation of the algorithm were written. Measurements for the robot encoders and the kinect module were used to build the map of the environment and localize the robot in it. The obstacles or features in the environment were represented as lines in the map.