Project Report

Prepared by : Nihit Gupta

Introduction

The aim of the project was to implement a Kalman Filter for the purpose of localization for an autonomous mobile robot. The implementation of the filter was done using ROS 2 Galactic and in python language. The localization had to be done using the data collected by sensors such as IMU and encoders. The project utilized a bag file containing all the data published on the topics during the simulation. This bag file thus contained the data from the required sensors which was then utilized by Kalman Filter in order to perform localization of the autonomous mobile robot. In general, Kalman filters are reliable and are used in the industry to estimate the variable of interest as perfectly as possible. Kalman filters reduces the variance of the state estimate error as they can combine information from numerous sensors in the presence of noise to determine the best estimate of states.

Theory & Methodology

The Kalman Filter was implemented to find the estimate of x, y, v_x, v_y and Θ at time interval k. In Kalman filter the variables/parameters of interest are called the system states. The code starts with an initial guessed value of the states. Another matrix P which is the uncertainty of an estimate-covariance matrix of the k-1 state is initialized.

$$\hat{x}_{0|-1} = E[x_o] = mean$$
$$P_{o|-1} = var(x_o)$$

Figure 1: Initializing state matrix and P matrix

The value of all the state members were also chosen to zero initially after auditing the bag file. As there are 5 state members, the P matrix was chosen to be a 5x5 identity matric as having an uncertainty of estimate-covariance as 1 is bad and with the help of Kalman filter it should be minimized.

The prediction algorithm takes the state values at time k-1 as input and outputs the values of the state at time k, here the term time interval is being used as the system is considered as a discrete time system. The prediction part of the algorithm also utilizes the dynamic model/State space model which is the motion model of the robot i.e. matrix A and the sensor measurements at time k-1, in our case the sensor measurements would be from IMU sensor keeping in mind our state members the required sensor data was for acceleration along x and y axis, and angular velocity along z axis however in code a rotation matrix was utilized to convert the IMU acceleration data from robot frame to Odom frame, this matrix was called as u matrix which thus contains the global a_x, a_y and w control input values. Moreover, to transition the sensor data, u matrix to help in finding the estimate a control input transition matrix is used, i.e., matrix B was used to predict the state values at time k.

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} + Bu_{k-1}$$

Figure 2: Prediction equation 1

As the system states were chosen to be of x, y, v_x, v_y and Θ, the matrix A and B were found using the Newton's Laws of motion and are as below –

$$\begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \\ \Theta \end{pmatrix}_k = \begin{pmatrix} 1 & 0 & dt & 0 & 0 \\ 0 & 1 & 0 & dt & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \\ \Theta \end{pmatrix}_{k-1} + \begin{pmatrix} \frac{1}{2}t^2 & 0 & 0 \\ 0 & \frac{1}{2}t^2 & 0 \\ dt & 0 & 0 \\ 0 & dt & 0 \\ 0 & 0 & dt \end{pmatrix} \begin{pmatrix} a_x \\ a_y \\ \dot{\Theta} \end{pmatrix}$$

Figure 3: Prediction equation 1 in matrix form with values of A and B

Similarly, the P matrix also needs to be predicted at time k, which is done using the following equation –

$$P_{k|k-1} = AP_{k-1,k-1}A^T + Q$$

Figure 4: Prediction of P matrix at time k using value from time k-1

Here, the matrices P and A are already known and defined, however matrix Q is the process noise matrix. The process noise matrix was considered to be independent between different states and thus was chosen to be a diagonal matrix. It was a little challenging to fin the value of this matrix to get the best performance of the Kalman Filter so initially it was chosen to be an identity matrix multiplied with a small number i.e. 0.05. It was made more than zero and not too high in order to have Kalman Filter trust the models moderately.

This marked the end of the prediction part of the Kalman Filter and the need is to develop the updating part of the filter. This implies once we have the predicted value, the estimated system state vector at time k could be found using the predicted value at time k, Kalman Gain K, the measurement from the sensors i.e. encoders, but as odom values are driven from encoders and are much simplified it was decided that it was better to just use odom values for the measurement and the matrix is called $y_{k,}$ moreover, this matrix contains the x and y values from pose, v_x and v_y values from Twist and Theta value from the orientation quaternions converted Euler form to get yaw. Furthermore, it also requires the observation matrix C which transforms the predicted state values from the local frame to global frame as in this case the transformation was done in a preliminary step for predicting the global values only, thus making the C matrix to be identity matrix here.

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K(y_k - C\hat{x}_{k,k-1})$$

Figure 5: Update equation for state values to get the estimates

Before calculating the results of the above matrix, the Kalman gain had to be calculated which is calculated using the following equation –

$$K = P_{k|k-1}C^T(CP_{k|k-1}C^T + R)^{-1}$$

Figure 6: Equation to calculate the Kalman Gain

Here, the values of predicted matrix P and matrix C is already known as defined above. However, the matrix R over here is the measurement uncertainty i.e. the measurement noise covariance matrix. The Kalman gain is used to minimize the estimated variance. Moreover, again matrix R is tuned to optimize the performance of the filter and thus the value chosen was to be same as matrix Q. Thus using the Kalman filter it is possible to get the estimates of state values as depicted in figure 5. Now, the matrix P which is the uncertainty (covariance) matrix of the current state estimation must be calculated using the following equation –

$$P_{k,k} = (I - KC)P_{k|k-1}$$

Figure 7: Update equation for matrix P for estimates

All the matrices in the above equation are known and thus matrix P for estimates could be calculated.

This code for filter runs continuously either in a loop or as a timer triggered function in order to continuously estimate the location of the moving robot and thus completing the task of localization of the autonomous mobile robot.

The code was implemented in ROS 2 Galactic. A package was created and inside that package a node was created which subscribed to the topics such as “/odom”, “/imu” and “/tf”, where “/imu” gives the information for the motion model, “/odom” gives the information for measurement and “/tf” gives us the ground truth. Moreover the explained Kalman Filter was obtained and the rms error between the estimates and ground truths were found. Finally, the states were also plotted against time and could be found in appendix with the code.

Results

It was found that Kalman filter was able to minimize the error in the estimate calculation and thus offers as a reliable filter when localizing an autonomous mobile robot. Moreover, it was found that changing the values of Q and R matrices changed the results in different fashions. Thus, it can be said that the matrices Q and R are hyperparameters which need to be tuned properly to make the Kalman filter as efficient as possible. However, with the Q and R the filter was trying average out the motion and measurement model. The Kalman filter uses both motion and measurement model for the purpose of localization and hence it does not blindly just trust one channel and thus it keeps on correcting using the model which is more trustworthy it is able to localize optimally and hence gives reliable results. The obtained results shows that Kalman Filter is an optimal method for localization and tries to minimize the error but was as the world is not ideal it wasn’t able to completely eliminate the error and some small error was observed. The plots could be found in the appendix.
The rms errors were calculated using the estimated value and the ground truth values and were found to be as below –

Calculated RMS error in x position = 3.8671930941575465e-05 units
Calculated RMS error in y position = 2.331766273093574e-05 units
Calculated RMS error in x velocity = 0.00029831262938162067 units
Calculated RMS error in y velocity = 0.00023834956417386968 units
Calculated RMS error in theta = 1.715030445191784e-07 units

Hence, the results indicate that there is some error, but it is negligible and thus for this project Kalman Filter is tuned appropriately and works as desired.

Discussion

The Kalman filter was found to be an optimal filter for the purpose of localization of an autonomous mobile robot. A challenge was to adjust the matrices Q and R for the Kalman Filter in order to obtain the best results. Interestingly, the Kalman Filter uses matrix Q at various stages which is the process noise matrix, and it was observed that the performance of the Kalman filter can be greatly influenced by the process noise variance, matrix Q. Moreover, the value can make the filter have lag error or even just make the filter only trust the measurement values which might cause noisy estimations. Furthermore, it was observed that even the matrix R works in similar fashion. Thus it was observed that making matrix R larger increases the filters trust on the measurement model and increasing the values of matrix Q increases the filters trust on motion model. Thus the matrices Q and R could be referred to as hyperparameters and should be changed with more experimentally. For the scope of this project only moderate values of Q and R were considered and they were taken to be same. However, a possible future improvement should be to tune these hyperparameters using adaptive algorithm like adaptive particle swarm optimization algorithm to get the best values of Q and R in order to make the filter work most optimally and thus get good estimates of the states. It was also observed that the filter was efficient in terms of computational power, however, it might become memory intensive for large number states. But in use cases like this it was both space and time efficient and thus becomes an ideal choice for localizing the robot.

It was observed that Kalman filter was not able to eliminate the error and there existed little to small error due to the lack completely tuning the hyperparameters to the best values, the world is not ideal hence the values are sometimes noisy, finicky and unreliable. But this error was considered to be acceptable for the scope of this project as it was almost neglegible.

Another method that could be implemented in future to improve the performance of the Kalman Filter would be to utilize the dynamic kinetic model of the robot based on the inputs given on “/cmd_vel” topic to generate the idealistic trajectory of the robot and then do the comparison. However, this data would not be the ground truth and would just represent the “ideal” scenario if the robot was perfect.

However, for the scope of this project we just used “/odom” as the measurement as “/odom” provides simplifies values from encoders and the Kalman Filter was found to be giving the optimal estimates and hence successfully localization was obtained.

Conclusion

The conclusion of the project is that the Kalman Filter is an optimal filter used for localization of an autonomous mobile robot as it utilizes the knowledge from both motion and sensor model and is thus able to provide optimal estimates. However, it was not able to eliminate error and does outputs error which can accepted for the scope of this project.

Moreover, it was concluded that the matrices Q i.e. the process noise matrix and matrix R i.e. the measurement noise covariance matrix are crucial to be tuned for the best performance of the Kalman Filters and thus to obtain best localization estimates. Moreover, it was concluded that higher the Q higher is the Kalman Filter’s trust on the motion model and if R is higher then the Kalman Filter’s trust is more on the measurement model. Thus, it must be tuned appropriately to make the filter work in most configuration and obtain reliable results.

As mentioned in discussions it was also concluded that an adaptive algorithm could be implemented in order tune the hyperparameters i.e. the Q and R matrices to get the most optimal results.

Finally, for this project the Kalman Filter was tuned appropriately and was found to be successful is minimizing error and providing reliable localization for the robot.

Appendix

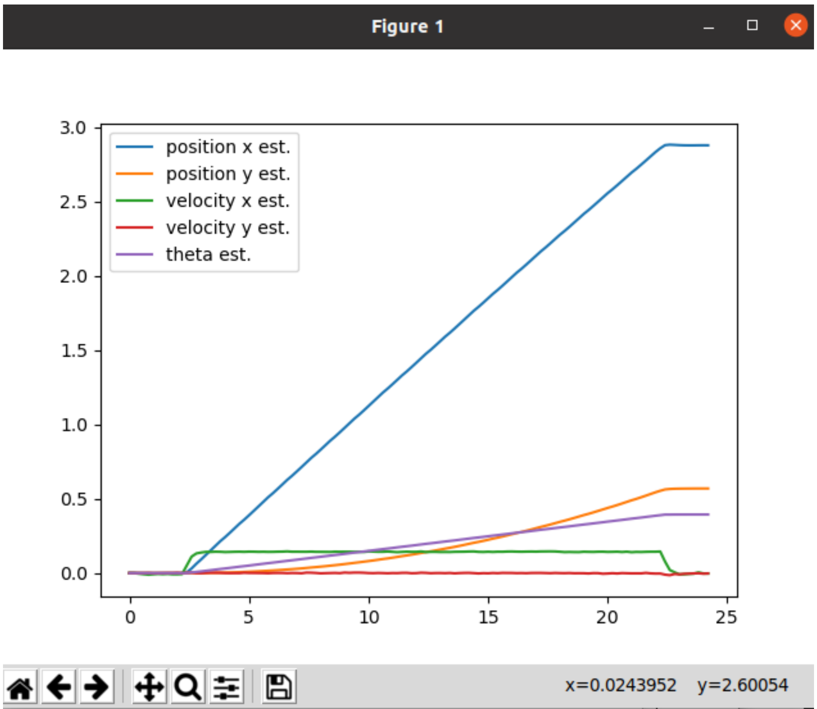

Figure 8: All the estimated states against time [units/time]

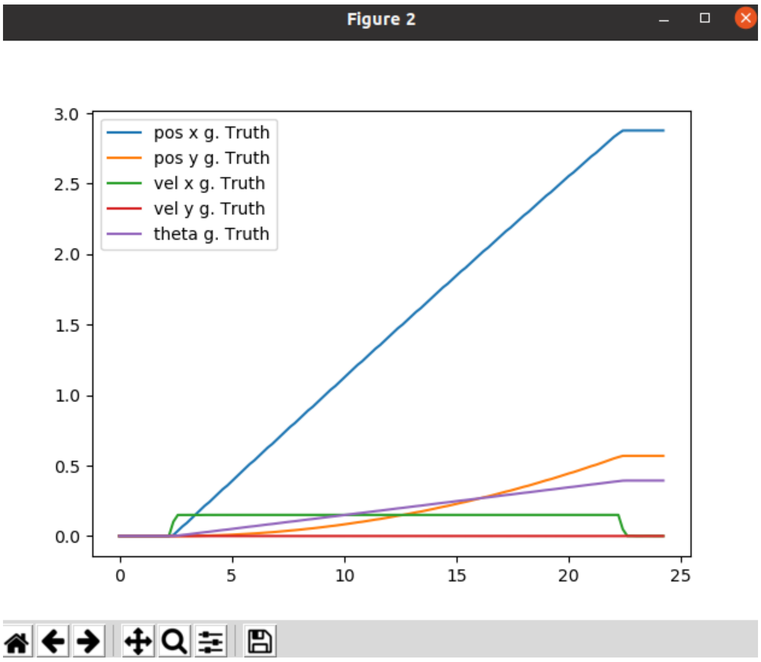

Figure 9: Ground truth values for all states against time [units/time]

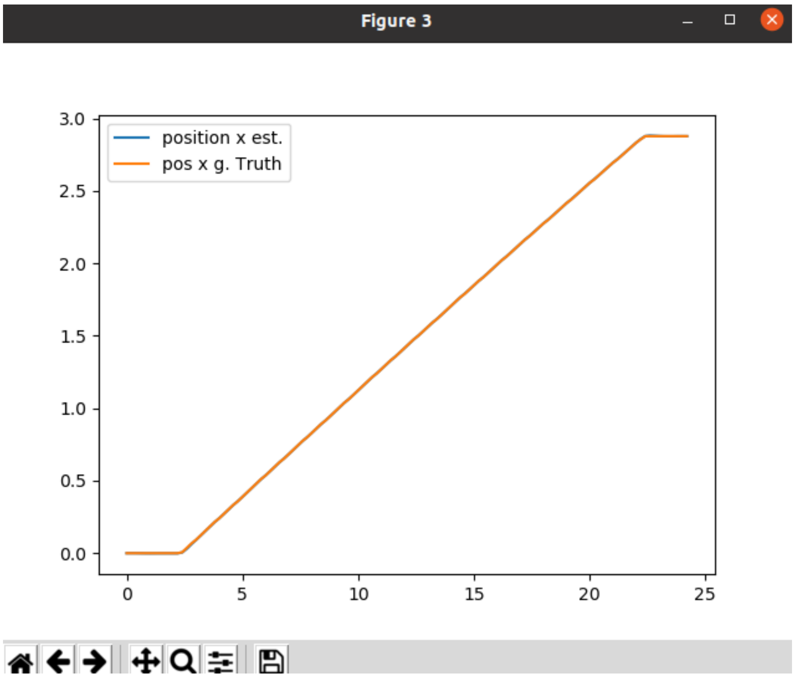

Figure 10 : Estimated X pose and Ground Truth of X pose against time

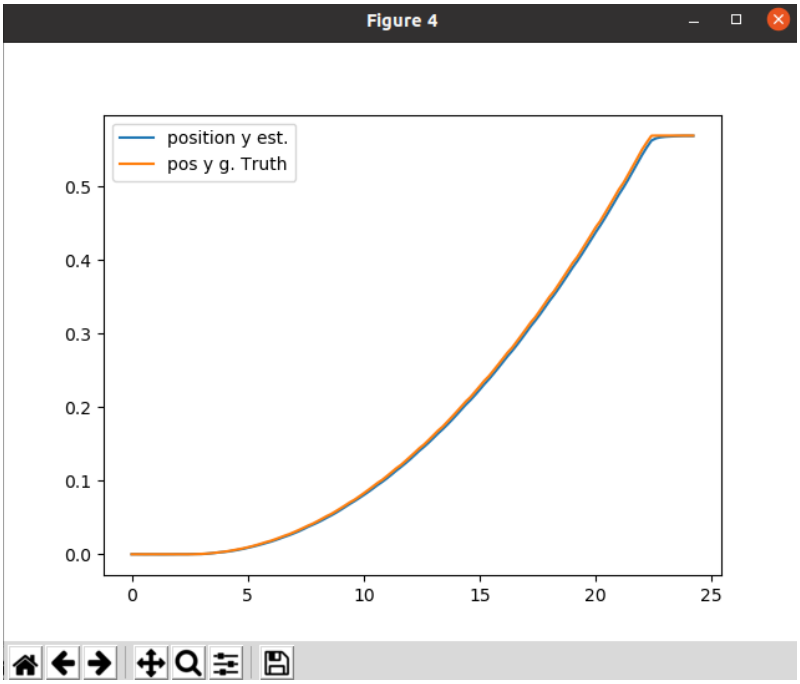

Figure 11 : Estimated Y pose and Ground Truth of Y pose against time

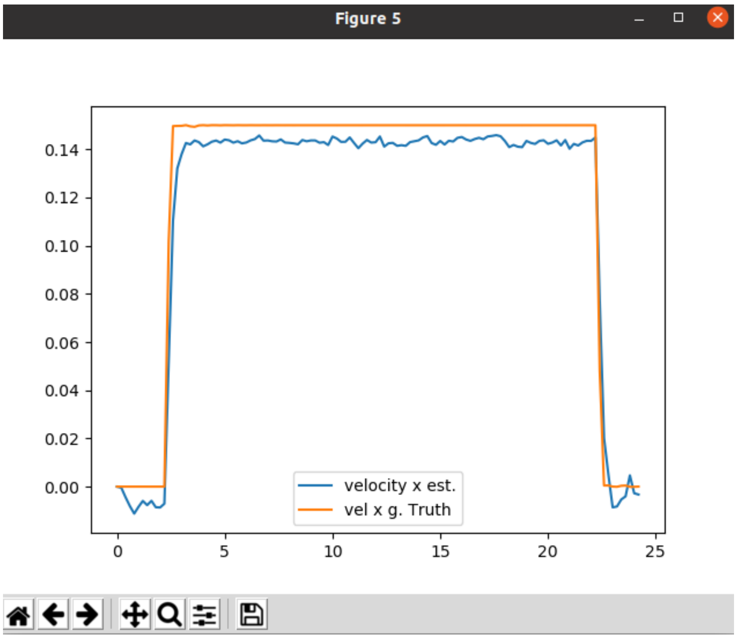

Figure 12 : Estimated velocity in x and Ground Truth of velocity in x against time

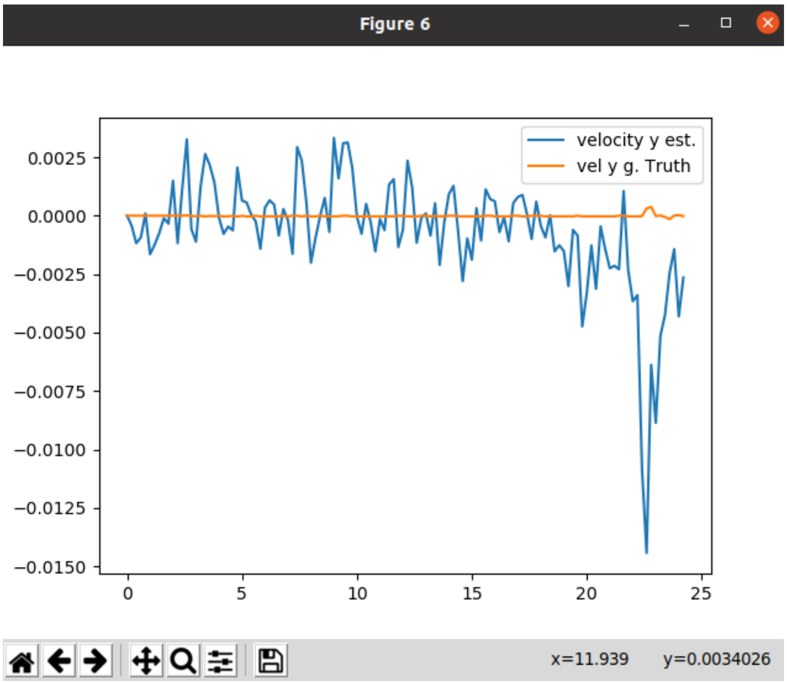

Figure 13: Estimated velocity in y and Ground Truth of velocity in y against time

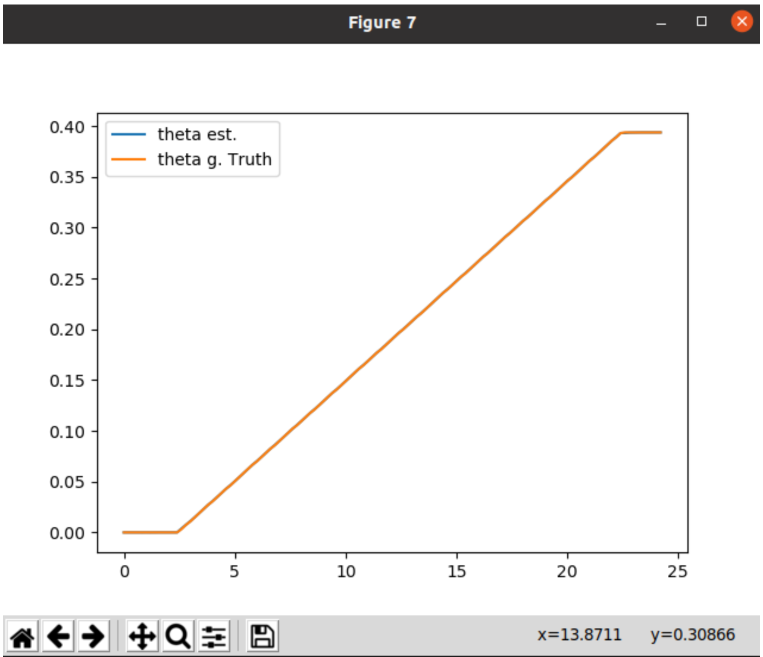

Figure 14: Estimated theta and Ground Truth of theta against time