A Mobile Robot Localization Using Extended Kalman Filter

MD Furkanul Islam
19th May, 2022

The goal is to calculate the current position of a mobile robot in a known area using Extended Kalman Filter.

Introduction

In robotics, it's important to localize a robot in its environment. The most commonly and optimally used approach is the kalman filter. A mobile robot kinematic model is nonlinear and linearized by jacobian. So In this project, The nonlinear version of kalman filter called Extended Kalman Filter(EKF) was used to localize a mobile robot in a known static environment.The sensor fusion been done by collecting data from Odometry and inertial measurement unit (imu) sensors.

Background

In this section, The project relates tools such as softwares, sensors and approaches are described. Also discuss the general idea and how they work.

2.1 ROS

Robotics Operating System (ROS or ros)[1] is a robotics middleware and collection of software frameworks.

2.2 Stage

Stage is a robot simulator. It provides a virtual world with objects, mobile robots and sensors. The world can be modified as user requirements. Stage has different sensor and actuator models such as sonar or infrared rangers, scanning laser rangefinder, color-blob tracking, fiducial tracking, bumpers, grippers and mobile robot bases with odometric or global localization[2]. It is very light and robust to simulate large populations. There are various ways to use a stage simulator. In this project, I used the Stage ROS package which provides a Stage 2-D multi-robot simulator. The details of the mobile robot are described in the .world file. By

running a ros command in the terminal, we can launch the stage simulation. Once the simulation is running, Stage ROS publishes the robot models and sensor data in the ROS node. Stage ROS also subscribed to some ROS topics to receive the velocity commands to drive the position model of the robot.

2.3 LiDAR

LiDAR is a sensing technology that measures distance to a target object by throwing up laser light. It also has a sensor to receive the reflected light. By using laser light traveling time and wavelength, produce a 2-D/3-D representation of the target object.

2.4 Mapping

Map is constructed by the help of a robot using its laser scanner data. it holds the information of the world and we can point our model where it is located in the map. In this project, The map is generated by using a ros package called "gmapping"[3].

2.5 Kalman Filter

The Kalman filter produces estimates of hidden variables by giving incorrect and uncertain measurements[4]. It can assume the future state of a system using past state data. By running recursively, it provides more accurate estimation. The Extended Kalman filter is a nonlinear version of the Kalman filter.

2.6 IMU

IMU stands for Inertial Measurement Unit. It's a small device which can measure gravity and angular rate of an attached object[5]. Typically an IMU is formed up with gyroscopes which measure the angular rate and accelerometer which report the force/acceleration.

3. Motion Model

The Motion Model is a mathematical equation which describes how the state of an object will change to another state once the control inputs are performed. The state describes position, orientation and other information of the object and controls v

and omega. In this project, the state variable contains the position of x and y, theta, v and omega. The control inputs are linear velocity v and angular velocity omega. W_{t-1} is the input noise at t-1 time frame. The motion model is, State = $x_t = A_{t-1}x_{t-1} + B_{t-1}u_{t-1} + W_{t-1}$. x_t is current state vector and x_{t-1} is the previous state vector, W_{t-1} is process noise, u_{t-1} is a vector of control input (linear and angular velocity), A_{t-1} is a matrix which describes how the state will change if there is no control input and B_{t-1} is a matrix which describes how the system will change from one state to another state when the control input is applied. In this project a differential drive robot is used. After placing the calculating of A and B matrices, calculate a jacobian to linearize the nonlinear equations. All together the equation is

$$\begin{bmatrix} x_t \\ y_t \\ theta_t \\ v_t \\ omega_t \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ theta_{t-1} \\ v_{t-1} \\ omega_{t-1} \end{bmatrix} + \begin{bmatrix} cos(theta_{t-1})T & 0 \\ sin(theta_{t-1})T & 0 \\ 0 & T \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_{t-1} \\ omega_{t-1} \end{bmatrix} + \begin{bmatrix} W_{t-1} \\ W_{t-1} \\ W_{t-1} \\ W_{t-1} \\ W_{t-1} \end{bmatrix}$$

4. Observation Model

The observation model estimates the sensor measurement using the state data with the help of a mathematical equation. It's also known as a sensor or measurement model. The equation is, $y_t = Hx_t + W_t$. y is the estimated sensor observation at time t, H is the measurement matrix, x is the state of robot at time t and the nose of sensors each observations W_t .

$$\begin{bmatrix} y_1^t \\ y_2^t \\ y_3^t \\ y_4^t \\ y_5^t \\ y_6^t \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_t \\ y_t \\ theta_t \\ v_t \\ omega_t \end{bmatrix} + \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \\ W_5 \\ W_6 \end{bmatrix}$$

5. EKF Algorithm

5.1 Initialization

In the first ekf iteration, time step k assumed and time step k-1 initialized the previous state is equal to zero and the control inputs are also equal to zero. Also initialized, Variance of process noise sigma_v = 0.1, sigma_omega = 0.1 and covariance matrix $P_{size(x)\ x\ size(x)}$ is equal to zero.

$$State = \hat{x}_{k-1|k-1} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ theta_{k-1} \\ v_{k-1} \\ omega_{k-1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, u_{k-1} = \begin{bmatrix} v_{k-1} \\ omega_{k-1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

5.2 Prediction

5.2.1 State Estimation

The first step of the ekf algorithm is predicted state estimation which is the same as the motion model. Here time step k instead of time step t.

$$State = \hat{x}_{k|k-1} = f\left(\hat{x}_{k-1|k-1}, u_k\right)$$

Here is the new equation,

$$State = \hat{x}_{k|k-1} = A_{k-1}x_{k-1} + B_{k-1}u_{k-1} + v_{k-1}$$

In motion model (chapter 3), have been discussed more details about the equation and values of all the matrices.

5.2.2 State Covariance Estimation

The state covariance estimated $\widehat{P}$ at given time step k. The equation is,

$$\widehat{P}_{k|k-1} = F_k P_{k-1|k-1} F_k^T + V_k Q V_k^T$$

Using motion model (chapter 3), we can write

$$\widehat{x} = \begin{bmatrix} x + v * T * cos(theta + omega * T) \\ y + v * T * sin(theta + omega * T) \\ theta + omega * T \\ v \\ omega \end{bmatrix}$$

F is the jacobian of f(x,u) for state x

$$F = \begin{bmatrix} 1 & 0 & -T * v * sin(theta + omega * T) & T * cos(theta + omega * T) & -T^2 * v * sin(theta + omega * T) \\ 0 & 1 & T * v * cos(theta + omega * T) & T * sin(theta + omega * T) & T^2 * v * cos(theta + omega * T) \\ 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

F_K^T is the transpose of the F matrix. V is jacobian of f(x,u) for control u,

$$V = \begin{bmatrix} T * cos(theta + omega * T) & -T^2 * v * sin(theta + omega * T) \\ T * sin(theta + omega * T) & T *^2 * cos(theta + omega * T) \\ 0 & T \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

V_K^T is the transpose of the V matrix. Q is the noise of control inputs.

$$Q = \begin{bmatrix} sigma\_v^2 & 0 \\ 0 & sigma\_omega^2 \end{bmatrix}$$

Now all the values are defined to calculate the state covariance estimated $\widehat{P}$.

5.3 Kalman Update

5.3.1 Measurement Residual

In this phase, compute the difference between actual sensor observations and estimated sensor observations. Innovation or measurement residual,

$\overline{y_k}$ = z_k - h($\widehat{x_{k|k-1}}$) . z_k is the actual observation vector at time step k,

z_k = [odom_x odom_y odom_theta odom_v imu_theta imu_omega]'

h($\widehat{x_{k|k-1}}$) is our observation model which predicts the sensor observation at time step k. In chapter 4, there are more details about our observation model.

5.3.2 Innovation Covariance

The Innovation or residual covariance $S_k = H_k P_{k|k-1} H_k^T + R_k$. The value of $P_{k|k-1}$ is calculated in chapter 5.2.2 and measurement matrix H_k is defined in chapter 4. In this step, the value of R_k is need to defined for computing S_k. R_k is the sensor measurement noise covariance matrix.

$$R_k = \begin{bmatrix} odom\_covPos & 0 & 0 & 0 & 0 & 0 \\ 0 & odom\_covPos & 0 & 0 & 0 & 0 \\ 0 & 0 & odom\_covTh & 0 & 0 & 0 \\ 0 & 0 & 0 & odom\_covV & 0 & 0 \\ 0 & 0 & 0 & 0 & imu\_covOri & 0 \\ 0 & 0 & 0 & 0 & 0 & imu\_covOm \end{bmatrix}$$

5.3.1 Kalman Gain

Now it's time to calculate the optimal kalman gain $K_k = P_{k|k-1} H_K^T S_K^{-1}$. All the variables are defined in the previous chapter. (chapter 4, chapter 5.22 and chapter 5.3.2). The correction of the state and covariance of state are dependent on the value of kalman gain k. If the kalman gain value moves towards 0 then the sensor measurements are ignored because of large measurement noise. On the other hand, k approaches 1 when the prediction noise is large and the sensor measurement is dominated.

5.4 Estimation

5.4.1 Update State

The important step of the kalman filter is to update the state based on all the previous data. The state update estimated equation is

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k$$

Predicted state, optimal kalman gain and residual measurement, all the values are defined in previous steps. Using those values, the ultimate robot current state will be estimated at current time step k.

5.4.2 Update Covariance

In this step, Updating the state covariance using below mathematical equation,

$$\hat{P}_{k|k} = (I - K_k H_k) P_{k|k-1}$$

Based on our previous step values, the updated or corrected covariance matrix of the state is computed.

6. Simulation Results

Stage simulation is used to simulate the project. A two-wheel, differential drive robot with equipped lidar and odometry. Also tested in gazebo with a turtlebot package. Due to computer performance, I moved to stage simulation.

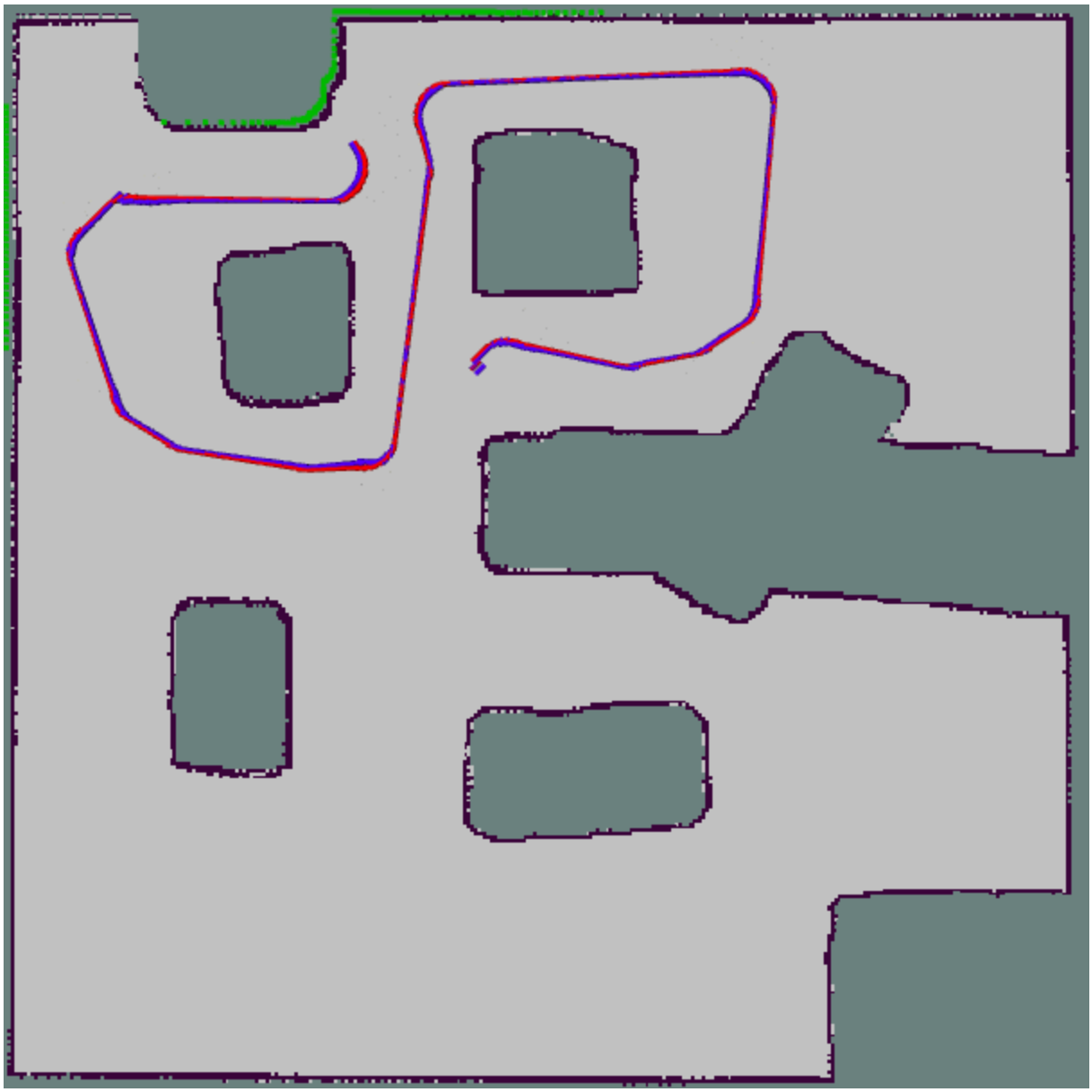

Figure 6.1: In rviz, robot actual position(red line) and EKF pose (Blue Line)

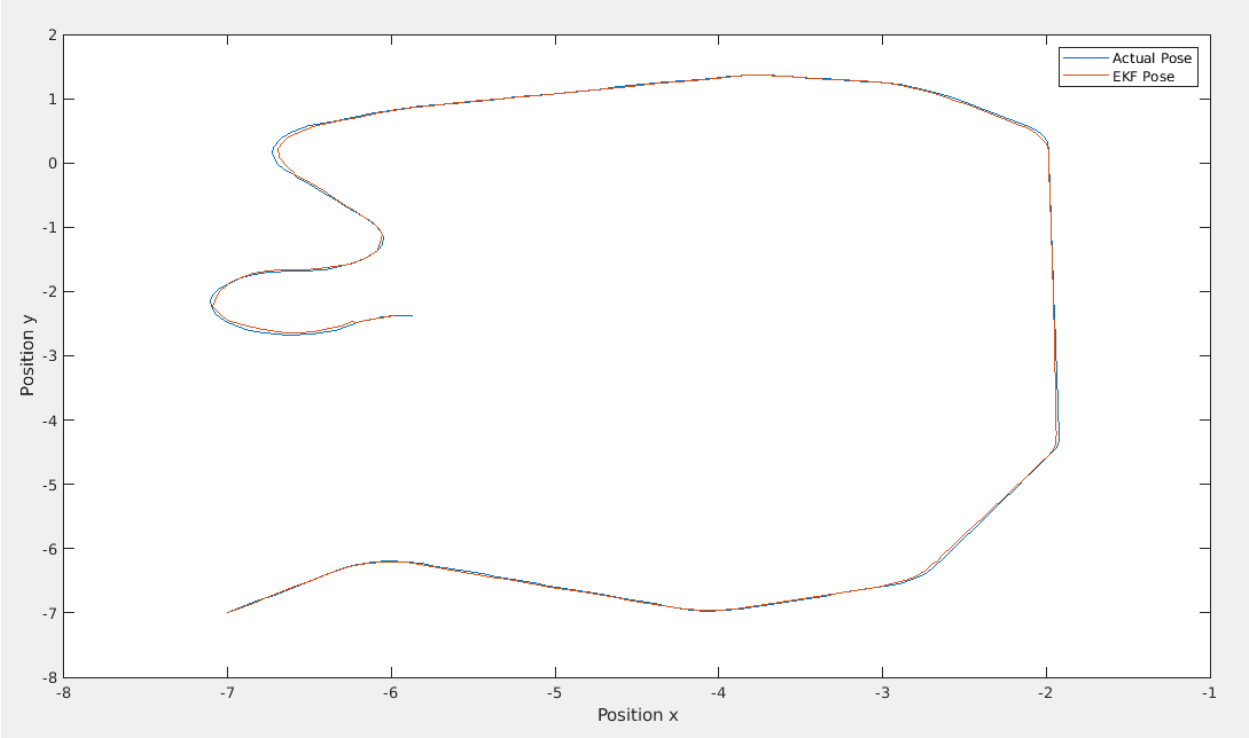

Figure 6.2: Plot of robot actual position and EKF pose

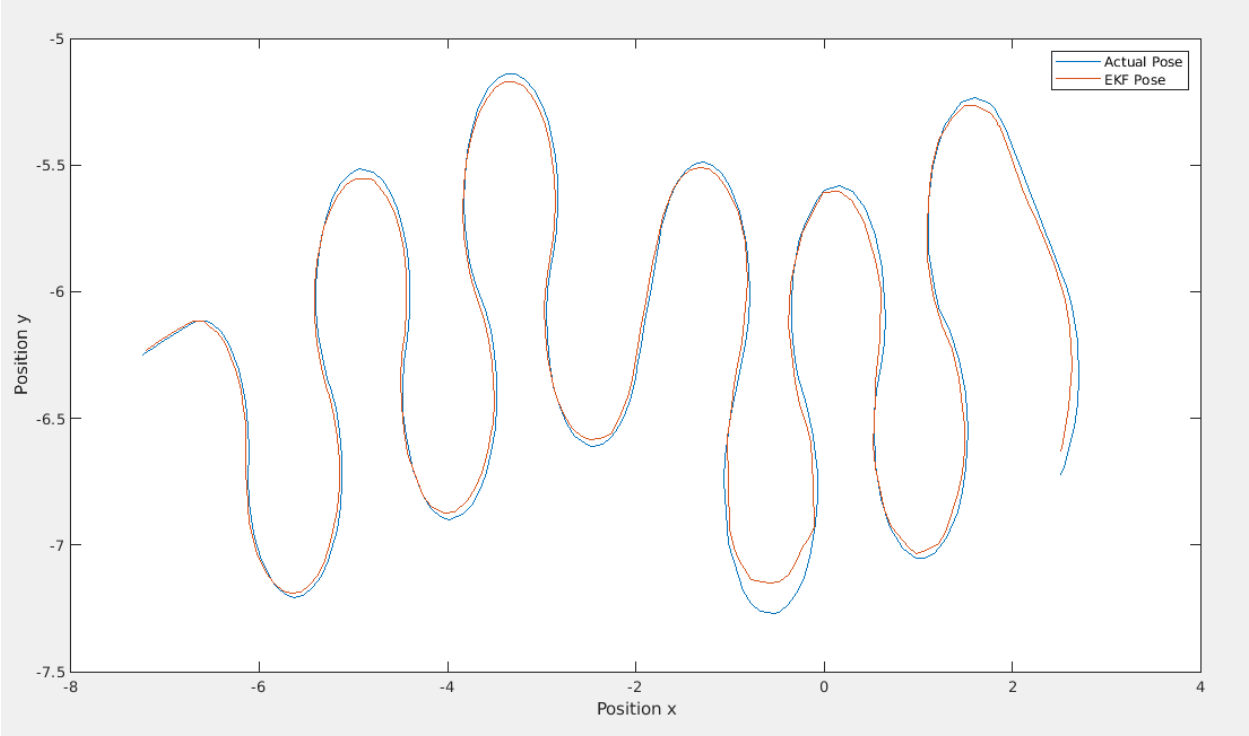

Figure 6.3: Plot of robot actual position and EKF pose

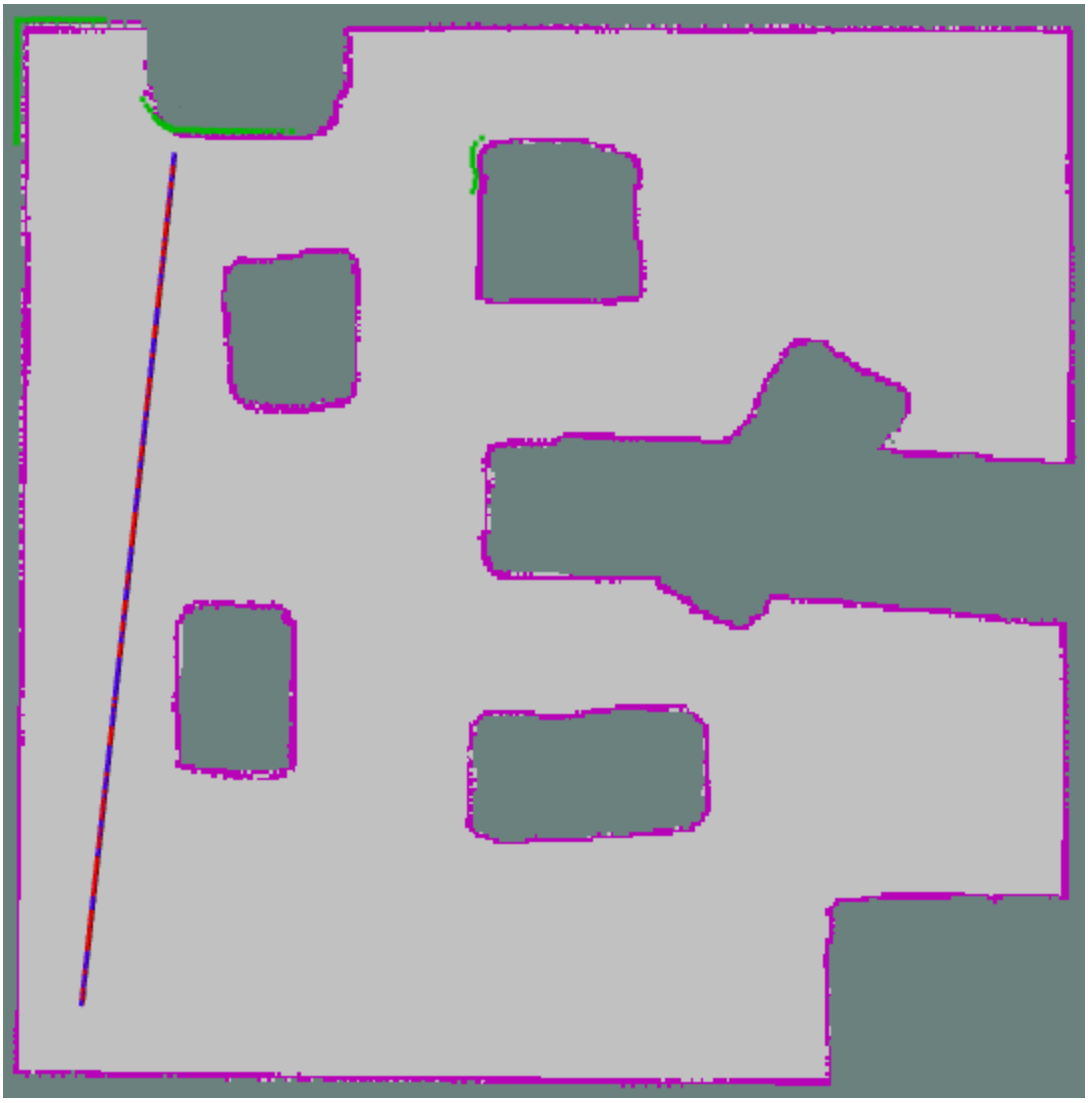

Figure 6.4: In rviz, robot actual position(red line) and EKF pose (Blue Line)

All the simulation experiments are taken in Stage simulation.

Experiment No.	Description of Path	RMSE
1- (Figure 6.1)	Rectangular path	0.0891
2- (Figure 6.2)	Zigzag and straight path	0.0558
3- (Figure 6.3)	zigzag path	0.0659
4- (Figure 6.4)	Straight line	0.0118

Table 6.1: The simulation result's RMSE

By observing the results, Once the angular velocity applied rmse is increased. The reason is that in stage there is no option to add an imu sensor. Which helps to take the angular velocity measurement. A simple simulation demo video is uploaded on youtube[6].

Conclusion

The extended kalman filter result is impressively good when you have a good state motion and sensor observation model. It will be really interesting to run this project in a real robot and observe the performance. Although there are countable errors, the result is quite satisfactory. There are options to change the noise value and modify the project to bring the expected results. A good extension of this project would be to add more sensors and real environment experiments.

References

1. ROS Introduction, Amanda Dattalo, 2018, http://wiki.ros.org/ROS/Introduction
2. The Stage Simulator, Richard Vaughan, 1998-2009, http://playerstage.sourceforge.net/doc/Stage-3.2.1/
3. Gmapping, ROS Wiki, 2019, http://wiki.ros.org/gmapping
4. Kalman Filter, Alex Becker, 2020, https://www.kalmanfilter.net/default.aspx
5. Inertial Measurement Unit, 2022 https://www.vectornav.com/resources/inertial-navigation-articles/what-is-an-inertial-measurement-unit-imu
6. Probabilistic Project -Demo, MD Furkanul Islam, 2022, https://www.youtube.com/watch?v=jYac8x5YMng