Visual Inertial SLAM Using EKF

Surya Prakash Thoguluva Kumaran Babu
Department of Mechanical and Aerospace Engineering
University of California San Diego
stk222@ucsd.edu

***Abstract*—The objective of this paper is to implement visual inertial simultaneous localization and mapping (SLAM) by applying Extended Kalman Filter (EKF) on readings from stereo camera and IMU sensor measurements. The trajectory of the robot and the landmark points are plotted and visualized.**

***Index Terms*—Extended Kalman Filter, Visual Inertial SLAM.**

I. Introduction

The ability to accurately know a robot's position in an unknown environment using the sensors is of absolute importance to a mobile robot. In real world, robots face this exact challenge of knowing it's own location in an unknown environment. Several methods and algorithms have been developed to address this issue over the past few decades. SLAM is a general term that is given to the solutions that addresses this problem. This is an active area of research as there is no one standard algorithm which will work for every mobile robot. For example, a moon rover might implement a sophisticated SLAM algorithm using highly accurate sensors, whereas implementing those algorithms for a home cleaning mobile robot may not be feasible. Hence there is a need to explore several variants of SLAM like particle filter SLAM, Extended Kalman Filter (EKF).

In this paper, we have discussed an implementation of Extended Kalman Filter (EKF) SLAM for a car that is equipped with stereo camera and IMU sensors. Motion model using IMU measurements was used to localise the robot and the image features collected from stereo camera was used to map the environment..

II. Problem Formulation For EKF SLAM

A. Notations

The following notations are used in the problem formulation.

- Motion model function f and its probability density function p_f that describes the states x_{t+1} resulting from applying input u_t at state x_t with a Gaussian noise of zero mean and W standard deviation is given as follows:

$$x_{t+1} = f(x_t, u_t, w_t) \sim p_f(.|x_t, u_t) \quad w_t \sim N(0, W)$$

- Motion model jacobian: The jacobian of non linear motion model with respect to state evaluated at noise and state mean is given as:

$$F_t := \frac{\mathrm{df}}{\mathrm{d}x}(\mu_{t|t}, u_t, 0)$$

The jacobian of non linear motion model with respect to noise evaluated at noise and state mean is given as:

$$Q_t := \frac{\mathrm{df}}{\mathrm{d}w}(\mu_{t|t}, u_t, 0)$$

- Observation Model function h and its probability density function p_h that describes the observation z_t depending on x_t with Gaussian zero mean and V standard deviation is given as

$$z_t = h(x_t, v_t) \sim p_h(.|x_t) \quad v_t \sim N(0, V)$$

- Observation model jacobian: The jacobian of non observation motion model with respect to state evaluated at noise and predicted state mean is given as:

$$H_t := \frac{\mathrm{dh}}{\mathrm{d}x}(\mu_{t+1|t}, 0)$$

The jacobian of non linear observation model with respect to noise evaluated at noise and predicted state mean is given as:

$$R_t := \frac{\mathrm{dh}}{\mathrm{d}v}(\mu_{t+1|t}, 0)$$

B. Assumptions For Extended Kalman Filter

The EKF implemented in this projects are based on the following Markov assumptions as well as additional assumptions that are listed below:

- The prior pdf $p_{t|t}$ is Gaussian
- The motion model is assumed to be Gaussian with Gaussian noise $w_t \sim N(0, W)$
- The observation model is assumed to be Gaussian with Gaussian noise $v_t \sim N(0, V)$
- The motion noise w_t and observation noise v_t are independent of each other, of the state x_t and across time.

C. Extended Kalman Filter

The EKF is a special case of Bayes Filter where the nonlinear motion model probability density function and observation model probability density function is assumed to be gaussian.

D. EKF Prediction Step

The predict step of the EKF is as given below

$$\mu_{t+1|t} = f(\mu_{t|t}, u_t, 0)$$
$$\Sigma_{t+1|t} = F_t \Sigma_{t|t} F_t^T + Q_t W Q_t^T$$

where $\mu_{t+1|t}, \Sigma_{t+1|t}$ are the respective predicted mean and predicted covariance of the state x_{t+1}

E. EKF Update Step

The update step of the EKF is as given below

$$K_{t+1|t} := \Sigma_{t+1|t} H_{t+1}^T (H_{t+1} \Sigma_{t+1|t} H_{t+1}^T + R_{t+1} V R_{t+1}^T)^{-1}$$

$$\mu_{t+1|t+1} = \mu_{t+1|t} + K_{t+1|t}(z_{t+1} - h(\mu_{t+1|t}, 0))$$

$$\Sigma_{t+1|t+1} = (I - K_{t+1|t} H_{t+1}))\Sigma_{t+1|t}$$

where $K_{t+1|t}$ is the Kalman gain, $\mu_{t+1|t+1}, \Sigma_{t+1|t+1}$ are the respective updated mean and updated covariance of the state $x_{t+1|t+1}$ The above formulation of the EKF needs be adapted for the visual inertial SLAMM.

F. Visual Inertial Localization

The objective is that given the IMU measurements $u_{0:T}$ with $u_t := [v_t^T, \omega_t^T]^T \in \mathbb{R}^6$ and feature observations $z_{0:T}$, estimate the IMU poses $T_t :=_W T_I \in SE(3)$. The prior is a transformation matrix given by $T_t|z_{0:t}, u_{0:t-1} \in N(\mu_{t|t}, \Sigma_{t|t})$ with $\mu_{t||t} \in SE(3)$ and $\Sigma_{t|t} \in \mathbb{R}^{6\times6}$.

G. Visual Inertial Mapping

The objective is, given the observations $z_t := [z_{t,1}^T, ..., z_{t,N_t}^T]^T \in \mathbb{R}^{4N_t}$ for time $t = 0, 1, ..., T$, estimate the coordinates $m := [m_1^T, ..., m_M^T]^T \in \mathbb{R}^{3M}$ for the landmarks that generated them using the known IMU pose $T_T :=_W T_{I,t} \in SE(3)$. The prior is the landmark coordinates with mean $\mu_t \in \mathbb{R}^{3M}$ and covariance $\Sigma_t \in \mathbb{R}^{3M\times3M}$

H. Visual Inertial SLAM

Given the linear velocity $v_t \in \mathbb{R}^3$ and rotational velocity $\omega_t \in \mathbb{R}^3$ and camera features $z_{t,i} \in \mathbb{R}^4$ for $i = 1, 2, ..., N_t$ determine the world frame IMU pose ${}_W T_I \in SE(3)$ over time and world frame coordinates $m_j \in \mathbb{R}^3$ for the $j = 1, ..., M$ point landmarks that generated the visual features $z_{t,i} \in \mathbb{R}^4$.

III. Technical Approach For EKF SLAM Implementation

The implementation of EKF SLAM on a high level is mentioned below:

- Checking the input data
- Data cleaning of the IMU and feature maps
- Develop algorithm for Visual Inertial Localization
- Develop algorithm for Visual Inertial Mapping
- Develop algorithm for Visual Inertial SLAM
- EKF Prediction Implementation using IMU Data
- EKF Update Implementation for mapping
- Implement visual inertial SLAM by combining the output from the above two steps

A. Given Input

We have 2 sets of data obtained from a mobile robot. All the data was time synced beforehand. Each sets contain the following data:

- Inertial Measurement Unit: A $3 \times N$ matrix containing linear velocity and a $3 \times N$ matrix containing angular velocity measurements from a IMU sensor is provided. There are N timesteps.
- Stereo Camera Features: A $4 \times M \times N$ matrix where M is the number of features that was observed throughout the entire timesteps and N is the number of timesteps that are given to us. If a particular feature was not observed in that timestep, then the matrix entry will have -1 in them.
- Intrinsic calibration: The stereo camera baseline b and a 3×3 matrix containing the calibration values of the camera is provided below:

$$K = \begin{bmatrix} fs_u & 0 & c_u \\ 0 & fs_v & c_v \\ 0 & 0 & 1 \end{bmatrix}$$

- Extrinsic calibration: The transformation from camera optical frame to IMU frame is given as input and is also mentioned below:

$${}_I T_C = \begin{bmatrix} 0.03717 & -0.09861 & 0.99443 & 1.57526 \\ 0.99926 & -0.00535 & -0.03789 & 0.00439 \\ 0.00906 & 0.99511 & 0.09834 & -0.65 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Timestamps is given as a N vector in unix timestamp format.

B. Data Cleaning

This step is necessarily important if the sensors were not time synced, if there was not data association between features across timesteps. This problem was already solved and the input data that we received was time synced and the feature association was already matched. To optimize the code for run time, features were down-sampled. Every 20th feature in the timestep was only considered while implementing this project.

C. Algorithm for Visual Inertial Localization

Following the notations used in formulating the localization problem in the previous section the prediction step with noise $w_t \in N(0, W)$ is given as

$$\mu_{t+1|t} = \mu_{t|t} \exp(\tau \hat{u}_t)$$

$$\Sigma_{t+1|t} = \exp(-\tau \overset{\curlywedge}{u}_t) \Sigma_{t|t} \exp(-\tau \overset{\curlywedge}{u}_t)^T + W$$

where $\mu_{t+1|t}$ and $\Sigma_{t+1|t}$ are the predicted mean and predicted covariance of the state T_{t+1}

Using the known stereo calibration matrix K_s, extrinsic ${}_c T_I \in SE(3)$, landmark positions $m \in \mathbb{R}^{3M}$ and new observations $z_{t+1} \in \mathbb{R}^{4N_{t+1}}$ the predicted observation using camera observation model and predicted mean of the pose, is given as:

$$\tilde{z}_{t+1,i} = K_s \pi({}_C T_I \mu_{t+1|t}^{-1} \underline{m}_j) \in \mathbb{R}^4$$

The jacobian of $\tilde{z}_{t+1,i}$ with respect to

$$H_{t+1,i} = -K_s \frac{d\pi}{dq}(cT_I \mu_{t+1|t}^{-1} \underline{m}_j) cT_I (\mu_{t+1|t}^{-1} \underline{m}_j)^{\odot} \in \mathbb{R}^{4\times6}$$

The EKF update step is then given as

$$K_{t+1} = \Sigma_{t+1|t} H_{t+1}^T (H_{t+1} \Sigma_{t+1|t} H_{t+1}^T + I \otimes V)^{-1}$$

$$\mu_{t+1|t+1} = \mu_{t+1|t} exp((K_{t+1}(z_{t+1} - \tilde{z}_{t+1})^{\wedge}))$$
$$\Sigma_{t+1|t+1} = (I - K_{t+1}H_{t+1})\Sigma_{t+1|t}$$

where $\mu_{t+1|t+1}$ and $\Sigma_{t+1|t+1}$ are the updated mean and covariance for the state $T_{t+1} \in SE(3)$

D. Algorithm for Visual Inertial Mapping

Following the notations used in formulating the mapping problem in the previous section the prediction observation based on μ_t is

$$\tilde{z}_{t+1,i} = K_s \pi({}_C T_I T_{t+1}^{-1} \underline{\mu}_{t,j}) \in \mathbb{R}^4$$

The jacobian of $\tilde{z}_{t+1,i}$ with respect to m_j evaluated at $\mu_{t,j}$ is given as

$$H_{t+1,i,j} = K_s \frac{d\pi}{dq}({}_c T_I T_{t+1}^{-1} \underline{\mu}_{t,j}) {}_c T_I T_{t+1}^{-1} P^T$$

where $P = [I, 0] \in \mathbb{R}^{3\times4}$. The kalman gain and the update step for the landmark mean and covariance are given below:

$$K_{t+1} = \Sigma_t H_{t+1}^T (H_{t+1}\Sigma_t H_{t+1}^T + I \otimes V)^{-1}$$
$$\mu_{t+1} = \mu_t + K_{t+1}(z_{t+1} - \tilde{z}_{t+1})$$
$$\Sigma_{t+1} = (I - K_{t+1}H_{t+1})\Sigma_t$$

where μ_{t+1} and Σ_{t+1} are the updated mean and covariance for the landmark $m \in \mathbb{R}^{3M}$

E. Algorithm for Visual Inertial SLAM

Visual inertial SLAM can be done by necessarily combining the above two prediction and update step of the localization and mapping problem. As the mean of the robot pose and landmark are of different dimensions, they need to be handled separately, while the covariance can be one large matrix. Let $\mu_{t|t,robot} \in SE(3)$ be the robot's prior, $\mu_{t|t,landmarks} \in \mathbb{R}^{3M}$ be the landmark prior, $\Sigma_{t|t,robot} \in \mathbb{R}^{3\times3}$ be robot's covariance and $\Sigma_{t,landmark} \in \mathbb{R}^{3M\times3M}$ be landmark covariance. The predict step is given as:

$$\mu_{t+1|t,robot} = \mu_{t|t,robot} \exp(\tau \hat{u}_t)$$
$$\Sigma_{t+1|t,robot} = \exp(-\tau \overset{\curlywedge}{u}_t)\Sigma_{t|t,robot} \exp(-\tau \overset{\curlywedge}{u}_t)^T + W$$

There is no predict step for the landmarks. However the landmark prior is initialised for the first time using inverse observation model if it was first detected. If a feature was not visible at all, the the prior mean and covariance is set to 0. The jacobians of the motion model and observation model are stacked as mentioned below:

$$\Sigma_{t+1|t,SLAM} = \begin{bmatrix} \Sigma_{t|landmarks} & 0 \\ 0 & \Sigma_{t+1|t,robot} \end{bmatrix}$$
$$H_{t+1,SLAM} = [H_{t+1,landmarks}, H_{t+1,robot}]$$

where $\Sigma_{t+1|t,SLAM} \in \mathbb{R}^{3M+6)\times(3M+6)}$ and $H_{t+1,SLAM} \in \mathbb{R}^{4N_t\times(3M+6)}$ where N_t is the number of features visible at this instant of time. The combined kalman gain is then computed as follows:

$$K_{t+1,SLAM} = \Sigma_t H_{t+1}^T (H_{t+1}\Sigma_t H_{t+1}^T + I \otimes V)^{-1}$$

where H_{t+1} and $\Sigma_{t+1|t}$ is the combined jacobian and covariance. The obtained $K_{t+1|t,SLAM} \in \mathbb{R}^{(3m+6)\times4N)}$ can be decomposed as:

$$K_{t+1,SLAM} = \begin{bmatrix} K_{t+1,landmarks} \\ K_{t+1,robot} \end{bmatrix}$$

where $K_{t+!,landmarks} \in \mathbb{R}^{3M\times4N}$ and $K_{t+1,robot} \in \mathbb{R}^{6\times4N}$. Then the corresponding means and covariance are updated as mentioned in the previous subsection.

$$\mu_{t+1|t+1,robot} = \mu_{t+1|t,robot} exp((K_{t+1,robot}(z_{t+1} - \tilde{z}_{t+1})^{\wedge}))$$

$$\Sigma_{t+1|t+1,robot} = (I - K_{t+1,robot}H_{t+1,robot})\Sigma_{t+1|t,robot}$$

$$\mu_{t+1,landmarks} = \mu_t + K_{t+1,landmark}(z_{t+1} - \tilde{z}_{t+1})$$

$$\Sigma_{t+1,landmark} = (I - K_{t+1,landmark}H_{t+1,landmark})\Sigma_t$$

$$\Sigma_{t+1|t+1,SLAM} = \begin{bmatrix} \Sigma_{t+1|landmarks} & 0 \\ 0 & \Sigma_{t+1|t+1,robot} \end{bmatrix}$$

F. EKF Prediction Implementation using IMU Data

This step corresponds to dead reckoning with EKF prediction step. The prior at time step 1 was initialized to identity and the motion noise had zero mean and the covariance was initialised to 0.001 for linear velocity portion and 0.00001 for the angular velocity. The results obtained by varying these parameters are discussed in the results section. The figure 1 and 2 show the dead reckon trajectory for dataset 10 and 3 respectively by implementing EKF prediction step only.

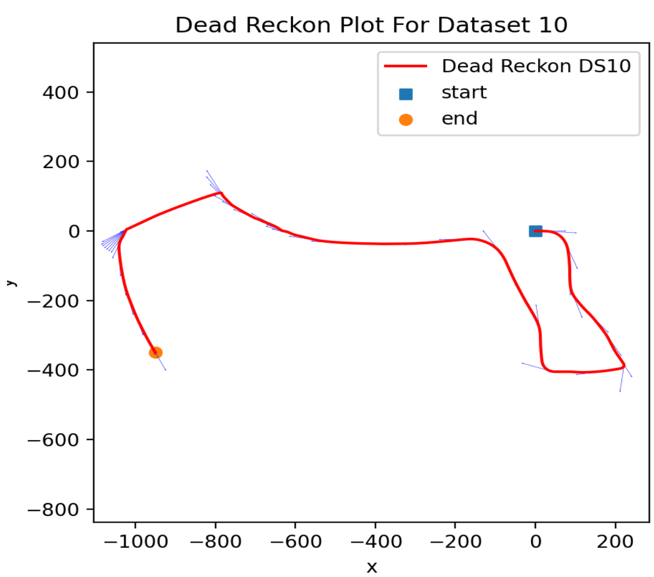

Fig. 1. IMU Localization Plot for Dataset 10.

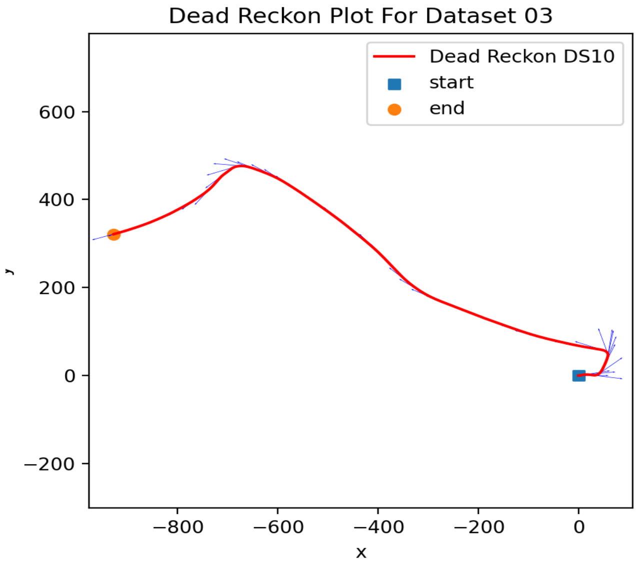

Fig. 2. IMU Localization Plot for Dataset 03.

G. EKF Update Implementation for Mapping

This step is implementing visual inertial mapping assuming the obtained trajectory from the above step is true. This is done as a sanity check our implementation of mapping step only. The algorithm followed is described below:

- For the first timestep, initialise the prior for each landmark using the inverse stereo camera model
- From second timestep, classify the currently visible landmarks to either needs to be initialized or needs to be updated. If a feature is visible for the first time, then it needs to be initialized using the inverse stereo camera model and then the rest of the visible features are considered for the EKF update step.
- The prior corresponding to these features are used to predict the feature observation value at this timestamp using the corresponding pose matrix and the observation model.
- Innovation is calculated based on the predicted observation and the actual observation model.
- The jacobians for the observation is evaluated the prior mean for the landmarks that are to be updated.
- Kalman gain, updated mean and updated covariance for the landmark is computed using the equations mentioned in the problem formulation section.

The intrinsic parameters are calculated using the given k matrix as below:

$$K_s = \begin{bmatrix} fs_u & 0 & c_u & 0 \\ 0 & fs_v & c_v & 0 \\ fs_u & 0 & c_u & -fs_u b \\ 0 & fs_v & c_v & 0 \end{bmatrix}$$

where f, s_u, s_v, c_u, c_v, b are the focal length, pixel scaling in u, pixel scaling in v, principle points in u, principle points in v, stereo baseline respectively. The prior is initialized in the following way. given u_L, u_R from the feature matrix, find the disparity $d = u_L - u_R$. Using the formula $d = fs_u b/z$ find z coordinates of the landmarks. Then x, y can be obtained by soling the below equation:

$$\begin{bmatrix} u_L \\ v_L \\ 1 \end{bmatrix} = \begin{bmatrix} fs_u & 0 & c_u \\ 0 & fs_v & c_v \\ 0 & 0 & 1 \end{bmatrix} \frac{1}{z} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

where x, y, z are the coordinates of landmark in body frame which needs to be converted to world frame using the pose transformation. Given the landmark means in the camera frame, the camera features are extracted as below:

$$\begin{bmatrix} u_L \\ v_L \\ u_R \\ v_R \end{bmatrix} = \begin{bmatrix} fs_u & 0 & c_u & 0 \\ 0 & fs_v & c_v & 0 \\ fs_u & 0 & c_u & -fs_u b \\ 0 & fs_v & c_v & 0 \end{bmatrix} \frac{1}{z} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

where x, y, z are obtained by transforming from world frame to body frame. The mapping of landmarks using IMU localization predict step is shown in figure 3 and figure 4.

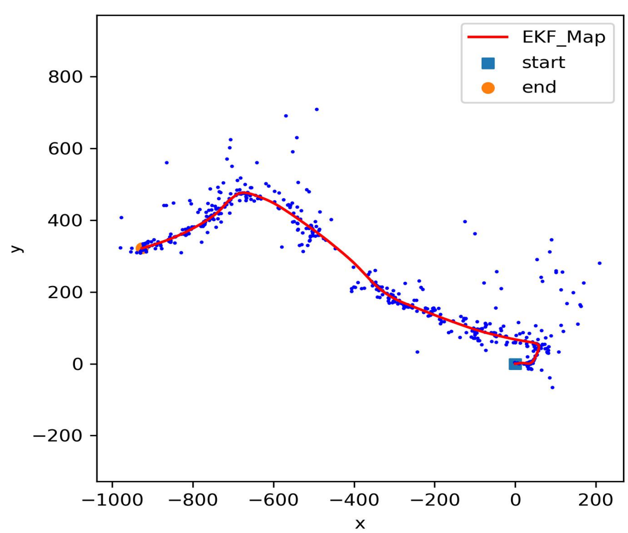

Fig. 3. EKF Mapping Only Plot for Dataset 10.

H. Visual Inertial SLAM Implementation

This step is to integrate the prediction step from IMU localization and combine the update step for the landmark and robot. The method followed for implementing this is outlined below:

- At the first timestep the robot is assumed to be at the identity pose and all the visible features at this timestep is initialized as it was done in the mapping step mentioned above.
- From the second timestep, the prediction step is done for the robot to obtain predicted mean and predicted

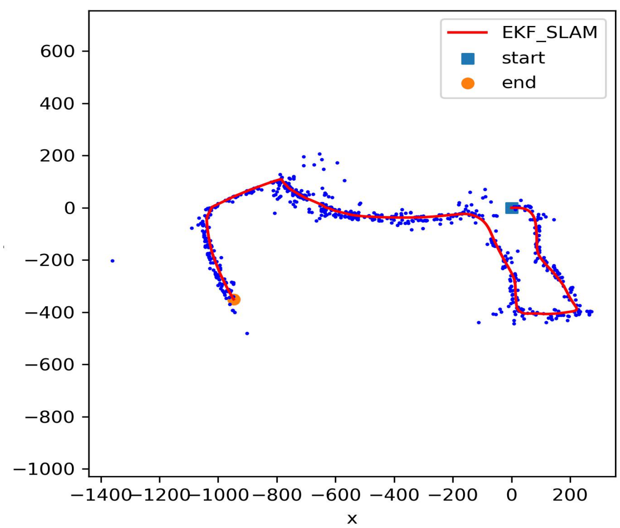

Fig. 4. EKF Mapping Only Plot for Dataset 03.

covariance for the robot using the IMU measurements. This is stored as $\mu_{t+1|t,robot}$ and $\Sigma_{t+1|t,robot}$.

- The prediction step for the landmark using the prior landmark mean $\mu_{t,landmark}$, predicted robot mean $\mu_{t+1|t,robot}$ to find the predicted observation z. Find innovation by $\tilde{z} - z$.
- Use the predicted covariance of the robot and the landmark to form the combined covariance matrix
- Obtained the combined kalman gain, jacobian from the equations mentioned in the problem formulation.
- The compute the predicted covariance $\Sigma_{t+1|t+1,SLAM}$ using the combined kalman gain and innovation.
- Split the kalmain gain corresponding to kalman gain robot and kalman gain landmark and perform the mean update step.
- Repeat this for all the timesteps

At every iteration, the covariance matrix was verified to be positive definite by checking the eigenvalues. The residual of the observation error is checked at each timestep to verify if the gain is obtained properly. A gain check is implemented to see of the computed gain is above a specific threshold.

IV. Discussions

A. Effect on Number of Feature Selection

There are about 13000 features in the total runtime. Applying all the features into EKF SLAM resulted in system crash. This maybe due to the fact that the time complexity of matrix inversion is of the order > 2 the system was not able to handle it. Hence, every 20th feature in each timestamp was used for the SLAM implementation. Also tried using every 10th feature in each timestamp. The result output between them was almost the same with the latter taking 3x more time than the former.

B. Effect of Motion and Observation Model Noise

I experimented with various motion and observation model noise and the findings are listed below

- Varying the landmark covariance matrix initialization value from 1 to 5 did not have a significant impact on the final output. However, the output plot was slightly more towards following the dead reckon trajectory if a high covariance initialization was higher. A value of 2 was used for covariance matrix initialization of the prior.
- The robot covariance initialization was set to be 0.001 for the linear velocity and 0.0001 for the angular velocity. Even a small change in this value casued a huge output in the robot trajectory. This maybe due to the fact that the robot in consideration is a car and even a small change in angular velocity and linear velocity will induce a significant error in the position estimation. Hence it is appropriate to set low values for robot covariance initialization.
- The EKF Slam was very sensitive to the motion model noise. A high noise of order 10^{-2} in the linear velocity and 10^{-4} in the angular velocity caused the trajectory to drift a lot away from the dead reckon trajectory. However a small noise of 10^{-8} in linear velocity and and angular velocity caused it to follow exactly the same path as dead reckon. This is shown in Figure 5,6,7,8. A value of 10^{-5} for linear velocity and 10^{-6} for angular velocity was ideal for this SLAM implementation
- Increasing or deceasing the observation model noise affected the innovation at each time step. A high observation noise of 5 produced a larger innovation and in turn caused the landmark and trajectory to drift away. A value of 2 for the observation noise

In conclusion, the noise parameters can be further tuned to get much better and optimized results.

C. Effect of Combined EKF Update Step

Initially, i tried implementing the EKF SLAM with separate update steps for the robot and the landmark. The observations are listed below:

- There was not much of a difference between using a combined update vs performing individual update step for the robot and the landmark.

D. Results Plot Discussion

A few observations from the plots are listed below.

- With low motion model noise the EKF SLAM follows almost the same path as dead reckon
- With higher motion model noise, the trajectory pattern is preserved while there is a drift induced in the path compared to the dead reckon trajectory.
- There is no significant deviation in the dataset 3 plots. This maybe due to the fact there are not much turns and curvature in the path traversed bu dataset 10 and even a small error in the angle estimation may lead to a different trajectory altogether.

V. RESULTS

The EKF SLAM was implemented. The plots below are shown for different motion noise. Figure 5,6,7,8 shows the EKF SLAM for 1000,2000,3000, Last timestep with motion model noise of 10^{-8} for linear velocity and 10^{-8} for angular velocity for dataset 10. Figure 9 shows the EKF SLAM for Last timestep with motion model noise of 10^{-4} for linear velocity and 10^{-6} for angular velocity for dataset 10. Figure 10 shows the EKF SLAM implemented on dataset 3 with motion model noise of 10^{-5} for linear velocity and 10^{-6} for angular velocity. For all the result plots, the covariance initialization value for the landmark was 2 and that of the robot was 0.001 for linear velocity component and 0.00001 for angular velocity component. Observation noise V for all the plots was set to be 2.

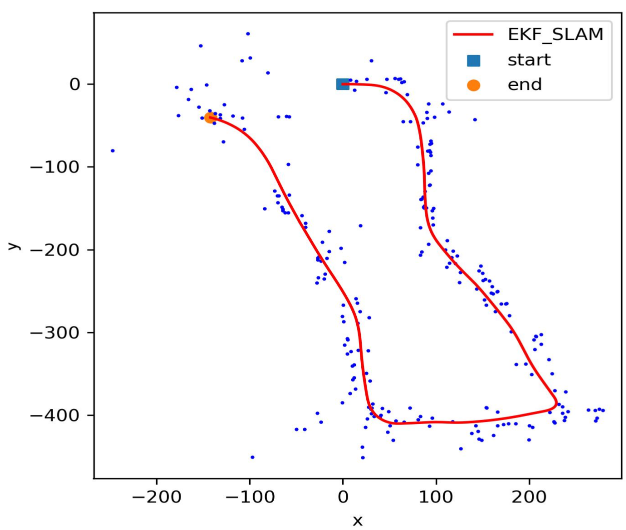

Fig. 5. EKF SLAM For Dataset 10 at 1000 Timestep Low Noise.

VI. ACKNOWLEDGEMENT

I would like to acknowledge that I have discussed concepts related to this project on a high level with Ritika Kishore Kumar.

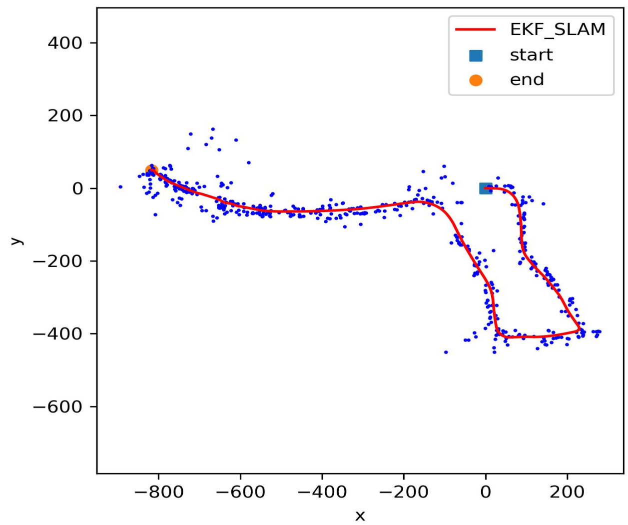

Fig. 6. EKF SLAM For Dataset 10 at 2000 Timestep Low Noise.

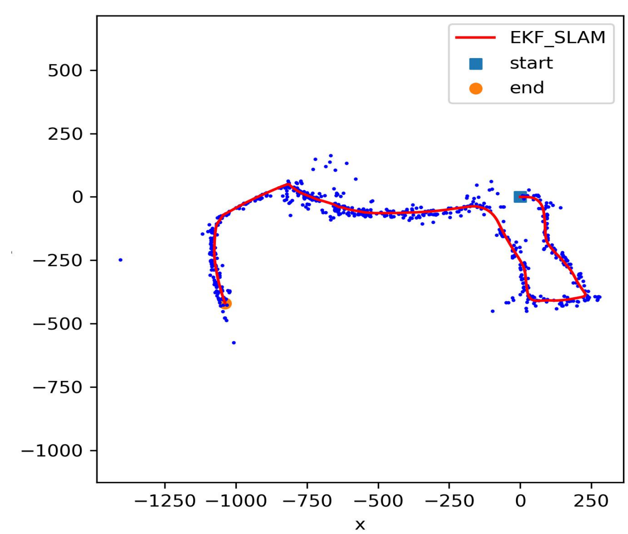

Fig. 7. EKF SLAM For Dataset 10 at 3000 Timestep Low Noise.

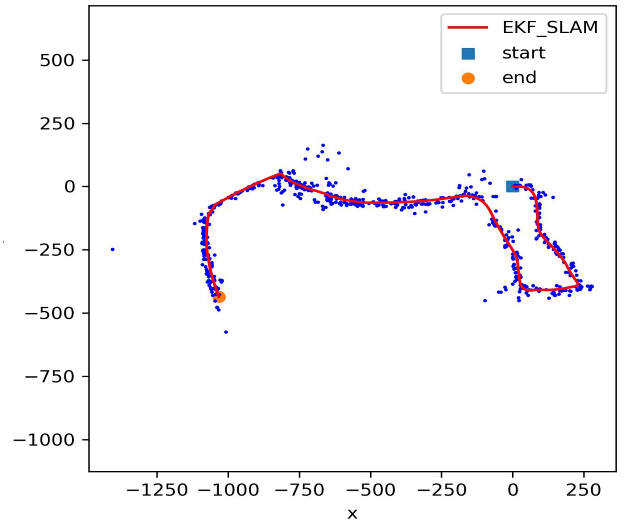

Fig. 8. EKF SLAM For Dataset 10 at Last Timestep Low Noise.

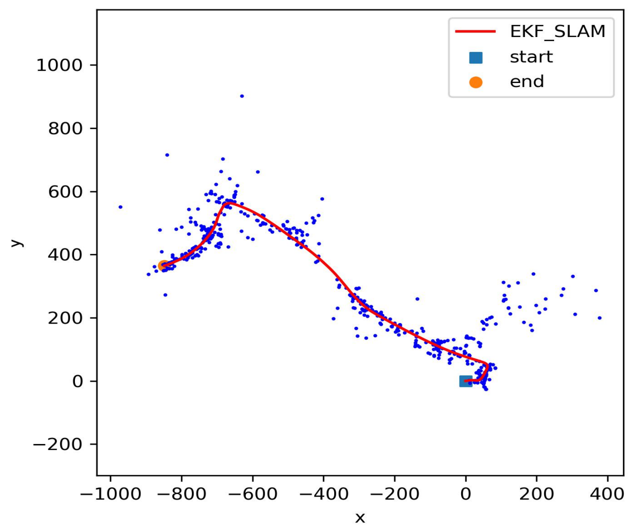

Fig. 10. EKF SLAM For Dataset 3 at Last Timestep.

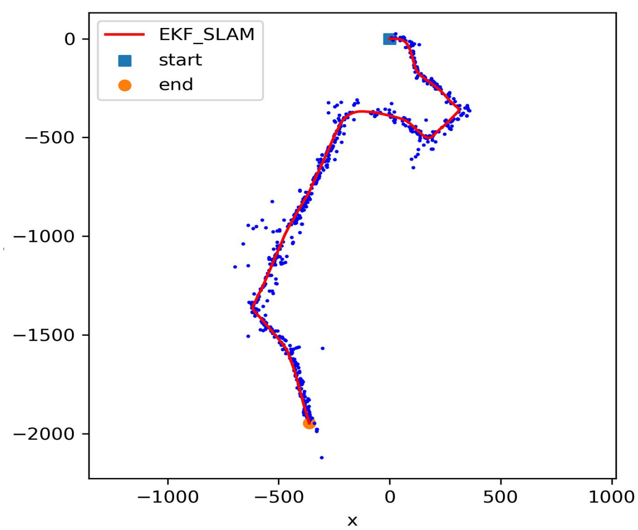

Fig. 9. EKF SLAM For Dataset 10 at Last Timestep High Noise.