Visual-Inertial SLAM

Dhruv Talwar
Electrical and Computer Engineering
ECE 276A

I. Introduction

Simultaneous Localization And Mapping (SLAM) is a critical problem in robotics, especially for autonomous systems with motion planning and estimation. The problem involves creating and updating a map of an unknown area while simultaneously keeping track of the agent's localization. To address this problem, various sensors and techniques are used.

In this study, we have implemented a Visual-Inertial Simultaneous Localisation and Mapping (VI-SLAM) system using an extended Kalman filter (EKF). The system utilizes synchronized measurements from an IMU and a stereo camera, as well as camera calibration and the calibration between the two sensors specifying the transformation from the IMU to the left camera frame. This approach enables the prediction of the car's localization, mapping of visual features, and implementation of a visual-inertial SLAM.

Our proposed solution is capable of generating reliable maps and trajectories in a reasonable amount of time. The extended Kalman filter we used can handle the noise and uncertainties in the sensor measurements, allowing for accurate and precise mapping of the environment. This makes it a useful tool for applications such as autonomous driving, UAV navigation, and robotics in general.

In summary, our Visual-Inertial SLAM solution based on an extended Kalman filter is a promising approach for accurately and efficiently mapping unknown environments. The algorithm's versatility enables it to be applied to various applications where precise localization and mapping are necessary, making it a valuable addition to the field of robotics. The report includes a mathematical problem formulation, a technical approach description, and the results obtained.

II. Problem Formulation

The aim of this project is to implement Simultaneous Localization and Mapping (SLAM) using the Extended Kalman Filter. To fully comprehend the SLAM problem, it is crucial to first understand the fundamental principles of SLAM and EKF.

A. SLAM and Bayes Filter

The problem of SLAM involves, given sensor measurements $z_{0:t}$ and inputs $u_{0:t-1}$ estimating the robot state x_t at time t and map state m, under Markov assumptions. The Markov assumptions state that The state x_{t+1} only depends on the previous input u_t and state x_t, and is independent of the history $x_{0:t1}$, $z_{0:t1}$, $u_{0:t1}$ and the observation z_t only depends on the state x_t

The state x_t and map m can be computing the joint probability distribution of the agent state and map state as described by the following equation:

$$p(x_{0:T}, m|z_{0:T}, u_{0:T-1}) \tag{1}$$

To solve this problem, we leverage the motion model and observation model. The motion model describes the state of the agent given the previous state and control input as in the following equation:

$$x_{t+1} = f(x_t, u_t, w_t) \sim p_f(\cdot|x_t, u_t) \tag{2}$$

where f defines the motion model and w_t is the motion noise. The observation model helps model the surroundings based on the observation z_t conditioned on the observation model h and observation noise v_t as in the following equation:

$$z_t = h(x_t, v_t) \sim p_h(\cdot|x_t) \tag{3}$$

To simplify the equation and make it easier for understanding we aim to club the map m and x_t as one and will be calling it x_t from now on. Using Equations (2) and (3) and applying them to Equation (1) under the Markov assumption, we can break down the problem into the following equation:

$$\begin{aligned} p(x_{0:T}, z_{0:T}, u_{0:T-1}) = p(x_0) \prod_{t=0}^{T} p_h(z_t|x_t) \\ \cdot \prod_{t=0}^{T-1} p_f(x_{t+1}|x_t, u_t) \cdot \prod_{t=0}^{T-1} p(u_t|x_t) \end{aligned} \tag{4}$$

Bayes filtering is a statistical technique used to estimate the state x_t of dynamic systems, such as robots, by incorporating evidence from both control inputs and observations. This is achieved by utilizing the Markov assumptions and Bayes rule. To keep track of the probability density functions $p_{t|t}(x_t)$ and $p_{t+1|t}(x_{t+1})$, the Bayes filter relies on two steps.

Prediction step: given a prior probability density function $p_{t|t}(x_t)$ of x_t and control input u_t, use the motion model p_f to compute the predicted probability density function $p_{t+1|t}(x_{t+1})$ of x_{t+1}:

$$p_{t+1|t}(x_{t+1}) = \int p(x_{t+1}|x_t, u_t) \cdot p_{t|t}(x_t) dx_t \tag{5}$$

where $p(x_{t+1}|x_t, u_t)$ is the motion model which describes the probability of the state x_{t+1} given the current state x_t and control input u_t. The integral is taken over all possible values of x_t.

Update step: given a predicted probability density function $p_{t+1|t}(x_{t+1})$ of x_{t+1} and measurement z_{t+1}, use the observation model p_h to obtain the updated probability density function $p_{t+1|t+1}(x_{t+1})$ of x_{t+1}:

$$p_{t+1|t+1}(x_{t+1}) = \frac{p(z_{t+1}|x_{t+1}) \cdot p_{t+1|t}(x_{t+1})}{\int p(z_{t+1}|x_{t+1}) \cdot p_{t+1|t}(x_{t+1}) dx_{t+1}} \tag{6}$$

where $p(z_{t+1}|x_{t+1})$ is the observation model which describes the probability of the measurement z_{t+1} given the state x_{t+1}. The integral is taken over all possible values of x_{t+1}.

B. Extended Kalman Filter

It is not feasible to use the Kalman filter for non-linear cases because it requires the computation of integrals for approximating the mean and covariance in the prediction step, as well as for the measurement mean, measurement covariance, and state-measurement correlation in the update step. These integrals are difficult to compute and can become computationally expensive in non-linear cases. Therefore, non-linear filters such as the Extended Kalman filter or the Unscented Kalman filter are used to approximate the non-linear functions and provide a better estimation for non-linear systems.

In our project we shall implement the EKF. Similar to the Kalman filter the EKF has the following assumptions:

- The prior pdf $p_{t|t}$ is Gaussian,
- The motion model is not linear in the state x_t with Gaussian noise w_t.
- The observation model is not linear in the state x_t with Gaussian noise v_t.
- The motion noise w_t and observation noise v_t are independent of each other, of the state x_t, and across time.
- The predicted and updated pdfs are forced to be Gaussian via approximation.

These assumptions enable the EKF to perform state estimation in non-linear systems by approximating the non-linear functions using a linear Gaussian model.

In EKF the prior pdf $p_{t|t}$ is assumed to be Gaussian with with a mean of $\mu_{t|t}$ and covariance of $\Sigma_{t|t}$. The prior is given as following:

$$x_t | z_{0:t}, u_{0:t-1} \sim \mathcal{N}(\mu_{t|t}, \Sigma_{t|t}) \tag{7}$$

In the Extended Kalman Filter (EKF), the motion model is a non-linear function of both the state variable and the control inputs. The noise in the motion model is assumed to be Gaussian with a zero mean and covariance W. In other words, the EKF represents the non-linear relationship between the state and control inputs in the motion model using a Gaussian distribution with a specific mean and covariance. The motion model is given by:

$$x_{t+1} = f(x_t, u_t, w_t), \quad w_t \sim \mathcal{N}(0, W) \tag{8}$$

To tackle the non-linear functions, we have to linearize them using a first-order Taylor series expansion around the current mean and covariance estimates. This linearization is performed in two steps:

Jacobian matrix: The Jacobian matrix is computed for the non-linear function with respect to the state variable. This matrix captures the partial derivatives of the non-linear function with respect to each state variable.

Linearize the function: Using the Jacobian matrix, the non-linear function is linearized around the current mean and covariance estimates using a first-order Taylor series expansion. This involves approximating the non-linear function as a linear function of the state variable by evaluating the Jacobian matrix at the current mean estimate and adding a correction term to account for the non-linear behavior.

The resulting linearized function will be used in the EKF update step to estimate the posterior mean and covariance. The EKF linearization process allows for the estimation of non-linear systems by approximating the non-linear functions using a linear Gaussian model. The resuling motion motion after the above linearization is given by:

$$f(x_t, u_t, w_t) \approx f(\mu_{t|t}, u_t, 0) + F_t(x_t - \mu_{t|t}) + Q_t w_t \tag{9}$$

The matrices F_t and Q_t) represent the Jacobians of the non-linear motion function f with respect to the state variable x and the motion noise w. These matrices are evaluated at the current estimate of the state variable. The matrix F_t captures the partial derivatives of the motion function with respect to each state variable, while the matrix Q_t captures the partial derivatives of the motion function with respect to the motion noise. Both the matrix are given by:

$$F_t = \frac{df}{dx}(\mu_{t|t}, u_t, 0) \tag{10}$$

$$Q_t = \frac{df}{dw}(\mu_{t|t}, u_t, 0) \tag{11}$$

Using the linearized motion model we can now implement the prediction step. The prediction step for the state variable at time step $t+1$ is defined by a set of equations that involve computing the predicted mean $\mu_{t+1|t}$ and covariance $\Sigma_{t+1|t}$.

These values are computed using the state transition function f, as well as the Jacobian matrices F_t and Q_t.

The predicted $\mu_{t+1|t}$ is given by:

$$\mu_{t+1|t} = f(\mu_{t|t}, u_t, 0) \tag{12}$$

And the predicted $\Sigma_{t+1|t}$ is given as:

$$\Sigma_{t+1|t} = F_t \Sigma_{t|t} F_t^T + Q_t W Q_t^T \tag{13}$$

Similar to the non-linear motion model, the observation model is also non-linear. To incorporate this non-linear relationship into the EKF, the observation model is linearized. The observation model is as follows:

$$z_t = h(x_t, v_t), \quad v_t \sim \mathcal{N}(0, V) \tag{14}$$

where v_t is the gaussian noise with 0 mean and V variance. After we linearize the observation model we get the following equations:

$$h(x_{t+1}, v_{t+1}) \approx h(\mu_{t+1|t}, 0) + H_{t+1}(x_{t+1} - \mu_{t+1|t}) + R_{t+1} v_{t+1} \tag{15}$$

where H_{t+1} and R_{t+1} represent the Jacobians of the non-linear observation function h with respect to the state variable x and the observation noise v. Both the matrix are given by:

$$H_{t+1} = \frac{dh}{dx}(\mu_{t+1|t}, 0) \tag{16}$$

$$R_{t+1} = \frac{dh}{dv}(\mu_{t+1|t}, 0) \tag{17}$$

Using the linearized observation model we can now implement the update step. The updated mean $\mu_{t+1|t+1}$ and covariance $\Sigma_{t+1|t+1}$ are computed using the Kalman gain $K_{t+1|t}$ and the measurement z_{t+1}. The Kalman gain is used to weight the difference between the predicted mean and the actual measurement, while also incorporating the covariance of the predicted state and the measurement noise. This results in an updated estimate of the mean and covariance that reflect the latest observations, improving the accuracy of the EKF's estimation. The higher the V, noise in the observation model, the smaller the kalman gain will become and vice versa. It tells us how much can we trust our observation The Kalman gain is given as follows:

$$K_{t+1|t} := \Sigma_{t+1|t} H_{t+1}^T (H_{t+1} \Sigma_{t+1|t} H_{t+1}^T + R_{t+1} V R_{t+1}^T)^{-1} \tag{18}$$

The update step for the mean and covariance is ggiven as follows:

$$\mu_{t+1|t+1} = \mu_{t+1|t} + K_{t+1|t}(z_{t+1} - h(\mu_{t+1|t}, 0)) \tag{19}$$

$$\Sigma_{t+1|t+1} = (I - K_{t+1|t} H_{t+1}) \Sigma_{t+1|t} \tag{20}$$

C. Data Set Given

IMU measurements: linear velocity $\mathbf{v}_t \in \mathbb{R}^3$ and angular velocity $\boldsymbol{\omega}_t \in \mathbb{R}^3$ of the body with coordinates expressed in the body frame of the IMU.

Visual feature measurements: pixel coordinates $\mathbf{z}_{t,i} \in \mathbb{R}^4$ of detected visual features from M point landmarks with precomputed correspondences between the left and the right camera frames. Landmarks i that were not observable at time t have a measurement of $\mathbf{z}_{t,i} = [-1, -1, -1, -1]^\top$, indicating a missing observation.

Time stamps: t in unix time (seconds since January 1, 1970).

Intrinsic calibration: stereo baseline b in meters and camera calibration matrix:

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

integrating the baseline as well in the matrix we can write the camera calibration matrix as

$$K = \begin{bmatrix} fs_u & 0 & c_u & 0 \\ 0 & fs_v & c_v & 0 \\ fs_u & 0 & c_u & -fs_u b \\ 0 & fs_v & c_v & 0 \end{bmatrix} \tag{21}$$

where f_{su} and f_{sv} are the focal lengths in pixels, and c_u and c_v are the principal points in pixels.

Extrinsic calibration: transformation $\mathbf{T}_{CI} \in SE(3)$ from the left camera to the IMU frame. The IMU frame is oriented as $x =$ forward, $y =$ right, $z =$ down.

We also have the following assumptions for the dataset given Assumption: The data association $\Delta_t : 1, ..., M \to 1, ..., N_t$ stipulating that landmark j corresponds to observation $z_{t,i} \in \mathbb{R}^4$ with $i = \Delta_t(j)$ at time t is known or provided by an external algorithm.

Assumption: The landmarks m are static, i.e., it is not necessary to consider a motion model or a prediction step for m.

III. Technical Approach

1) IMU localization via EKF prediction
: We are given the IMU measurements as u_t as v_t and ω_t as the linear and angular velocity respectively. We assume that there is data association between the observation and the features/landmarks. We know world-frame landmark coordinates $m \in \mathbb{R}^{3M}$ We will use the EKF prediction step to calculate the mean and variance at each step.

$$\text{Prior:} \quad T_t | z_{0:t}, u_{0:t-1} \sim \mathcal{N}(\mu_{t|t}, \Sigma_{t|t}) \tag{22}$$

$$\mu_{t|t} \in \mathrm{SE}(3) \tag{23}$$

$$\Sigma_{t|t} \in \mathbb{R}^{6\times 6} \tag{24}$$

$$\tag{25}$$

Motion model: nominal kinematics of $\mu_{t|t}$ and perturbation kinematics of $\delta\mu_{t|t}$ with time discretization τ_t :

$$\mu_{t+1|t} = \mu_{t|t}\exp(\tau\hat{u}_t) \tag{26}$$
$$\delta\mu_{t+1|t} = \exp(-\tau u^{\wedge})\delta\mu_{t|t} + w_t \tag{27}$$

where τ is the time difference in seconds between two contiguous frames.

$$\begin{aligned}\mu_{t+1|t} &= \mu_{t|t}\exp(\tau_t\hat{u}_t)\\ \Sigma_{t+1|t} &= E[\delta\mu_{t+1|t}\delta\mu_{t+1|t}^T]\\ &= \exp(-\tau u_t^{\wedge})\Sigma_{t|t}\exp(-\tau u_t^{\wedge})^T + W\end{aligned}$$

where we have

$$u_t = \begin{bmatrix} v_t \\ \omega_t \end{bmatrix} \in \mathbb{R}^6$$

$$\hat{u}_t = \begin{bmatrix} \hat{\omega}_t & v_t \\ 0^T & 0 \end{bmatrix} \in \mathbb{R}^{4\times 4}$$

$$u^{\wedge} = \begin{bmatrix} \omega_t^{\wedge} & v_t^{\wedge} \\ 0 & \omega_t^{\wedge} \end{bmatrix} \in \mathbb{R}^{6\times 6}$$

In my implementation I have used the above equations only to get the trajectory overtime. the linear and angular velocity are derived from the data itself.

2) Landmark mapping via EKF update
: The observation model is used for the Landmark mapping procedure. The observation model is assumed with measurement noise as $v_{t,i} \sim \mathcal{N}(0, V)$
We also have the following assumptions for the dataset given The data association $\Delta_t : 1,...,M \to 1,...,N_t$ stipulating that landmark j corresponds to observation $z_{t,i} \in \mathbb{R}^4$ with $i = \Delta_t(j)$ at time t is known or provided by an external algorithm. The landmarks m are static, i.e., it is not necessary to consider a motion model or a prediction step for m. Thus the observation model boils down to

$$z_{t,i} = h(T_t, m_j) + v_{t,i} := K_s\pi({}_oT_tT_t^{-1}m_j) + v_{t,i} \tag{28}$$

Here T_t is the IMU pose which is known from the localization step, K_s is the camera calibration matrix given to us and m_j is the homogeneous coordinate and is of the form m_j = $\begin{bmatrix} m_j \\ 1 \end{bmatrix}$ We are given the observation z_i using equation 28, I was able to find the m_j. Note in our data we are given the transformation from optical camera frame to IMU, thus we used the inverse of that to fit into the equations. The T state was used from the previous motion model calculation

The projection function given by π and its derivative is given by

$$\pi(q) := \frac{1}{q_3}, \quad q \in \mathbb{R}^4 \tag{29}$$

$$\frac{d\pi}{dq}(q) = \frac{1}{q_3}\begin{bmatrix} 1 & 0 & -\frac{q_1}{q_3} & 0 \\ 0 & 1 & -\frac{q_2}{q_3} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{q_4}{q_3} & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4} \tag{30}$$

We can represent all the observations $z_{t,i}$ as a single vector $\mathbf{z}_t \in \mathbb{R}^{4N_t}$ and express the observation model equation as:

$$z_t = K_s\pi({}_oT_tT_t^{-1}m) + v_t \tag{31}$$
$$\mathbf{v}_t \sim \mathcal{N}(0, \mathbf{I}\otimes\mathbf{V}), \tag{32}$$
$$\mathbf{I}\otimes\mathbf{V} := \begin{bmatrix} V & 0 & \cdots & 0 \\ 0 & V & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & V \end{bmatrix} \tag{33}$$

where $\otimes$ is the Kronecker product.

The prior for mapping is given by the following

$$m|z_{0:t} \sim \mathcal{N}(\mu_t, \Sigma_t) \quad \text{with} \quad \mu_t \in \mathbb{R}^{3M}, \Sigma_t \in \mathbb{R}^{3M\times 3M} \tag{34}$$

The EKF Update step can be written as:

$$K_{t+1} = \Sigma_t H_{t+1}^{\top}(H_{t+1}\Sigma_t H_{t+1}^{\top} + \mathbf{I}\otimes\mathbf{V})^{-1} \tag{35}$$
$$\mu_{t+1} = \mu_t + K_{t+1}(z_{t+1} - K_s\pi({}_OT_IT_{t+1}^{-1}\mu_t) \tag{36}$$
$$\Sigma_{t+1} = (I - K_{t+1}H_{t+1})\Sigma_t \tag{37}$$

where H_{t+1} is the Jacobian of the predicted observation with respect to $\mathbf{m}_j$:

$$H_{t,i,j} = \begin{cases} \mathrm{K_s}\frac{\partial\pi}{\partial q}({}_oT_IT_{t+1}^{-1}\mu_{t,j})\,{}_oT_IT_{t+1}^{-1}P^T \\ \quad\text{if obs } i \text{ correspond to landmak } j \text{ time } t \\ \mathbf{0} \quad\quad \text{otherwise} \end{cases}$$
$$H_t \in \mathbb{R}^{4N_t\times 3M} \tag{38}$$
$$P = \begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix} \in \mathbb{R}^{3\times 4} \tag{39}$$

Where P is defined as the projection matrix. Note that $\mathbf{I}$ represents the 3×3 identity matrix and $\mathbf{0}$ represents the 3×1 zero matrix. In our In EKF-based visual mapping project, it is assumed that all the landmarks are static, and hence, prediction steps are not necessary.

3) Visual-Inertial SLAM:
: We now combine the steps from the localization and mapping steps to implement SLAM.

We use the motion model with prior given as follows

$$\text{Prior:} \quad T_t|z_{0:t}, u_{0:t-1} \sim \mathcal{N}(\mu_{t|t}, \Sigma_{t|t}) \tag{40}$$
$$\mu_{t|t} \in \mathrm{SE}(3) \tag{41}$$
$$\Sigma_{t|t} \in \mathbb{R}^{6\times 6} \tag{42}$$
$$\tag{43}$$

and the observational model as

$$z_{t,i} = h(T_t, m_j) + v_{t,i} := K_s\pi({}_oT_tT_t^{-1}m_j) + v_{t,i} \tag{44}$$

Udating the model involves computing the observation model Jacobian H_{t+1} with respect to the IMU pose T_{t+1}, evaluated at the predicted mean $\mu_{t+1|t}$. The resulting Jacobian matrix H_{t+1} is a $R^{4N_{t+1}X6}$ matrix, where each element corresponds to a different observation, allowing for the incorporation of new sensor measurements and improving the accuracy of the EKF's estimation.

We use the m_j calculated from the mapping step and use it to calculate the predicted observation which is given by:

$$z_t^* = K_s\pi(\,{}_oT_tT_t^{-1}m_j) \tag{45}$$

for i values ranging from 1 to N_{t+1} We also calculate the Jacobian as follows

$$H_{t+1} = -K_s\frac{d\pi}{dq}(\,{}_oT_t\mu_{t+1}^{-1}{}_tm_j)\,{}_oT_I(\mu_{t+1}^{-1}m_j)^{\circ}\cdot \tag{46}$$

The H_{t+1} matrix would be a 4 cross 6 matrix. The $\circ\cdot$ operator is the odot function and is defined as follows

$$\bar{s} \mathrel{\ni} \begin{pmatrix} I_3 & -\hat{s} \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{4\times 6}$$

where hat is the standard hat map function which gives the skew symmetric matrix.

Thus the overall matrix would look like this

$$H_{t+1} = \begin{bmatrix} \mathrm{H}_{t+1,1} \\ \vdots \\ H_{t+1,N_{t+1}} \end{bmatrix}$$

Finally the update step equations for the SLAM are as follows:

$$K_{t+1} = \Sigma_{t+1|t}H_{t+1}^{\top}(H_{t+1}\Sigma_{t+1|t}H_{t+1}^{\top} + \mathbf{I}\otimes\mathbf{V})^{-1} \tag{47}$$

$$\mu_{t+1|t+1} = \mu_{t+1|t}exp((K_{t+1}(z_{t+1} - z_{t+1}^*)) \tag{48}$$

$$\Sigma_{t+1|t+1} = (I - K_{t+1}H_{t+1})\Sigma_{t+1|t} \tag{49}$$

IV. Results and Discussions

We were provided with two datasets, dataset 03 and dataset 10 We shall discuss the accuracy of the trajectory, mapping and EKF SLAM. The trajectories were very accurate I used the provided videos as a sanity check to see the turns and the overall trajectory and direction of the car. The landmark mapping was also good and I was able to get several landmark maps with different feature numbers. The details about the plots and results are shown below

A. Trajectory and Motion Model

We were able to plot the trajectory for both the datasets. For this part when we calculate the μ_{t+1} we see it is independant of the σ value, thus independant of the W noise. As in the motion model we are just using the prediction step the W noise is of no effect to the trajectory plotted. The prediction step or the motion model just used the previous state to calculate the new one. Similar to what we did in PR1 and PR2.

As shown in Figure 1(a)(b) we can see the trajectory for both dataset 03 and dataset 10 respectively.

The trajectory was compared with the videos given in the dataset and thus validated.

B. Visual Mapping

We were able to plot the visual mapping results for both data sets successfully. We could see in the Figure that the landmarks are very close to the trajectory thus supporting our mapping formulation. To reduce computing time, we down sampled the features. I used every 50th landmark in the dataset 10 and every 30th landmark in dataset 03 to show the mapping results. Our observation that the landmarks closely follow the car's trajectory suggests that our mapping was successful. To reduce computing time, we down sampled the features.

In figure 1(c)(d) we can see the landmarks mapped in the world frame along with the trajectories obtained from the previous part.

The hyperparemeters used were as follows: Observation noise covariance V was taken to be 10, the motion model noise W was taken as 0.001, and the landmark and μ covariance was taken as 0.001 respectively.

C. Visual SLAM

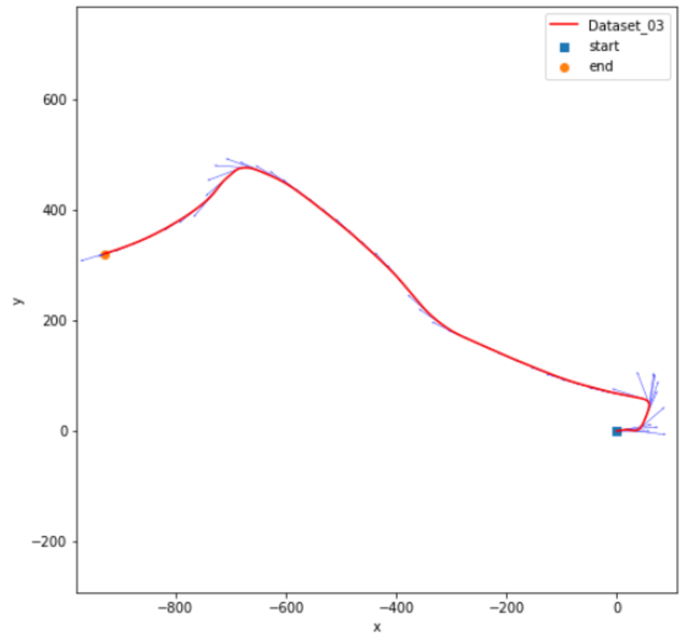
We were successfully able to perform EKF SLAM on our dataset. The new trajectory obtained from EKF is quite different from what we saw in only the "trajectory" generation this is because when we found the trajectory we were only relying on the IMU data and performing the prediction step. While in the EKF slam we use the position of the landmarks as well to update the car position.

When we performed the EKF slam we see that the landmarks position has also changed and now it follows the EKF trajectory because initially we used the erroneous IMU trajectory which was used to calculate the landmark position. Now the landmarks are in the correct position.

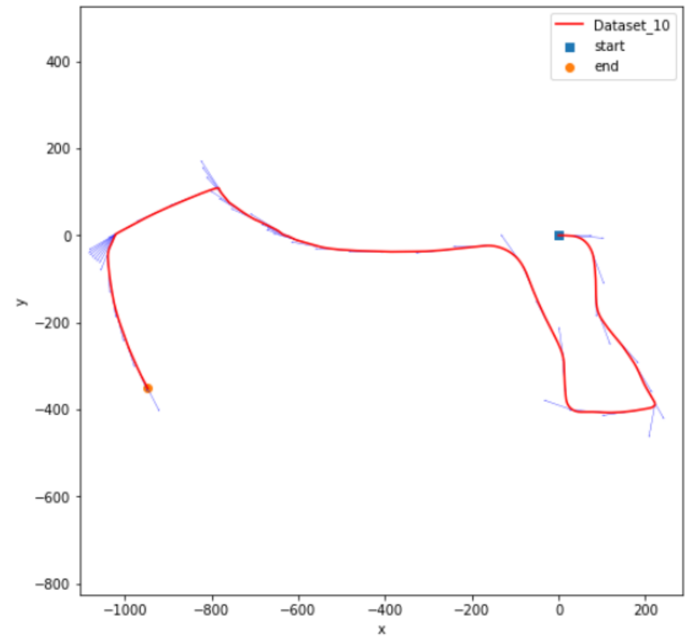
The results for the EKF slam are shown in Figure 2(a)(b) and the comparison between the IMU only trajectory for both the data sets 03 and 10 can be seen in Figure 2(c)(d)

D. Conclusion

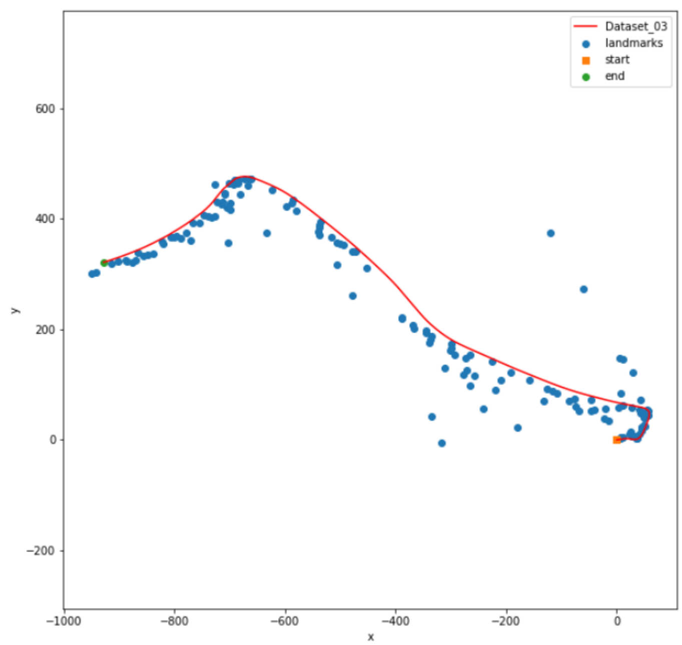
In conclusion, this paper presents a comprehensive approach for simultaneous localization and mapping of a car using the EKF. As shown from the results we can say that for non linear systems EKF is a very handy tool. The combination of IMU, stereo camera provides a robust estimation of the cars position and landmarks.



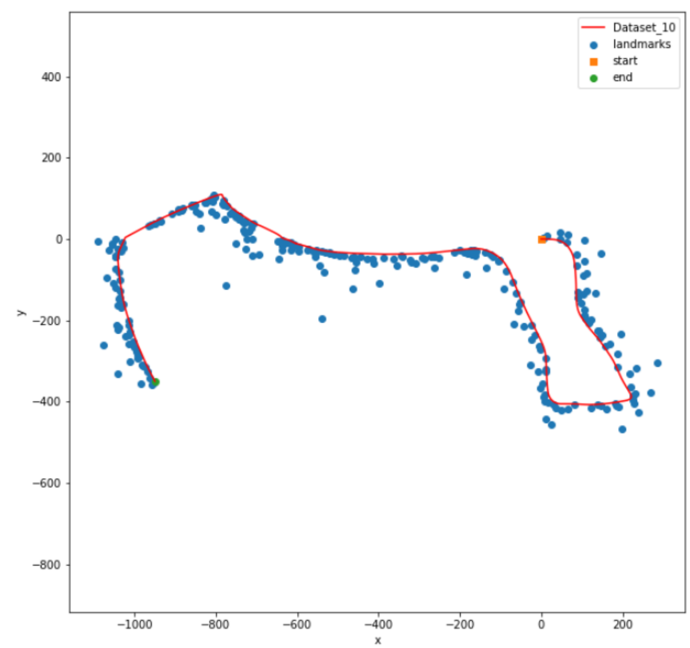
(a) Trajectory for dataset 03



(b) Trajectory for dataset 10

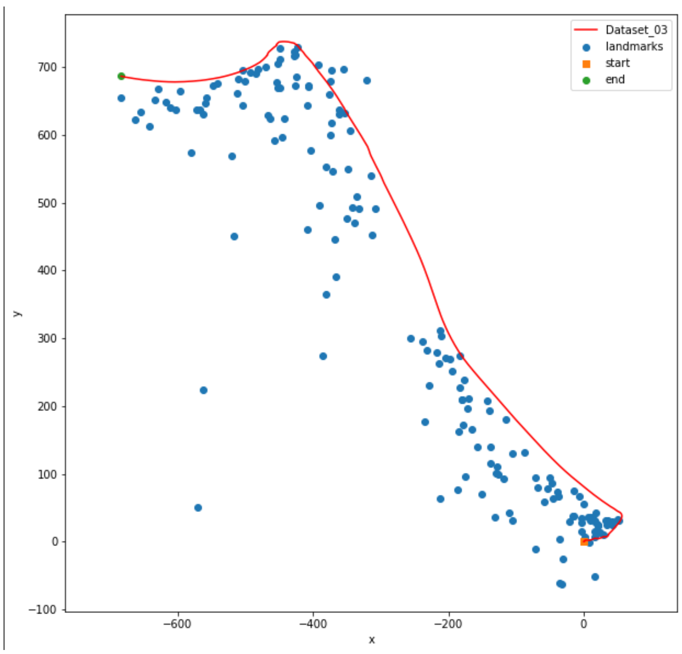


(c) Landmark mapping for dataset 03 (every 30th feature)

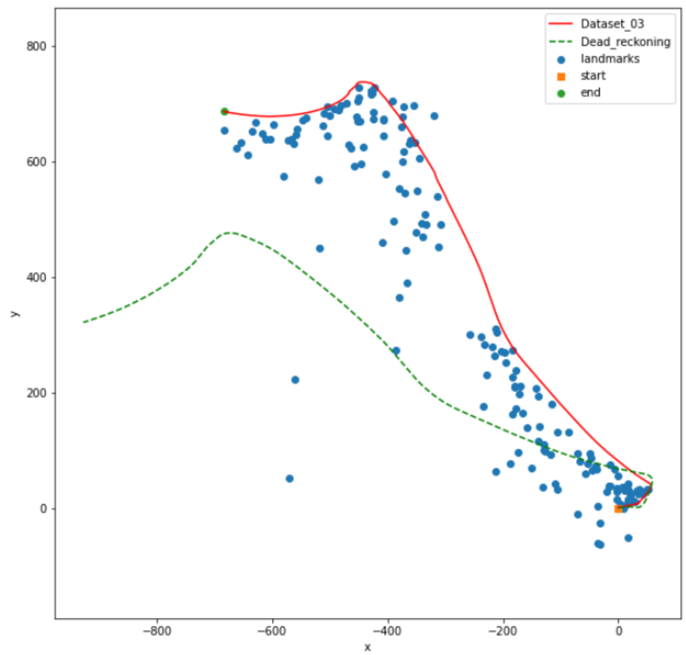


(d) Landmark mapping dataset 10 (every 50th feature)

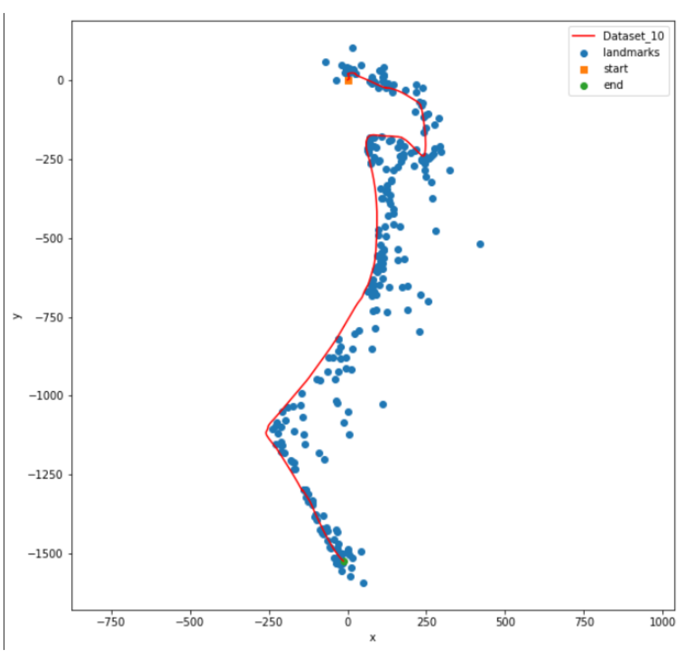
Figure 1: Trajectory and mapping results



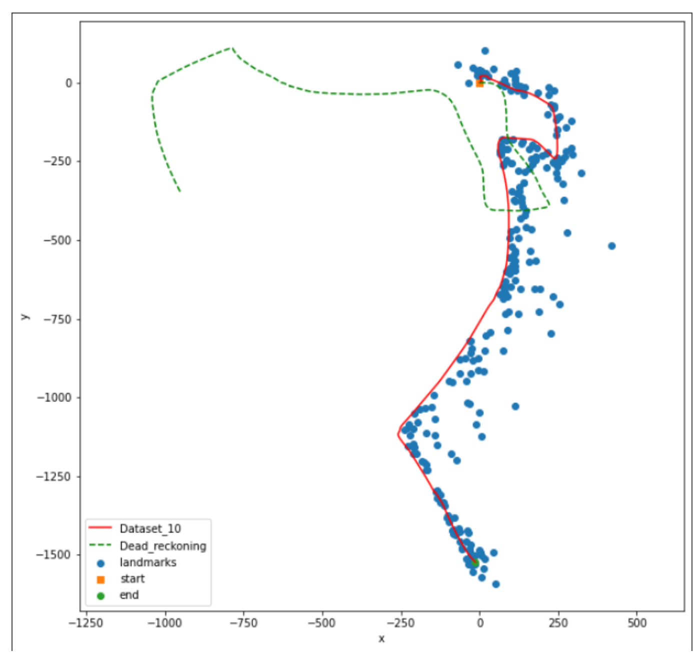
(a) SLAM results for Dataset 03



(b) Comparison between SLAM and dead reckoning trajectory



(c) SLAM results for Dataset 10



(d) Comparison between SLAM and dead reckoning trajectory

Figure 2: SLAM and Dead reckoning trajectories