# Visual-Inertial SLAM

Narayanan Elavathur Ranganatha
*University of California San Diego*
nelavathurranganatha@ucsd.edu

***Abstract*—The following report formulates the problem of *Visual-Inertial Simultaneous Localization and Mapping (SLAM)* using time-synchronised data provided by a Inertial Measurement Unit (IMU) and a stereo camera setup. The algorithm used for the above is the Extended Kalman Filter where the measurements are the stereo camera features and IMU acts as the control input.**

## I. Introduction

### A. Problem Statement

The objective of the problem is to simultaneously create a map of the surroundings by computing the 3D locations of the stereo camera features while also localizing the robot in this map with the help of the IMU , and using the stereo camera observation model and the camera features.

### B. Motivation

*Simultaneous Localization and Mapping (SLAM)* is a fundamental problem in robot autonomy. This provides the basic information needed by any robot to perform any task at hand. A robot has to know where it is (Localization) and what is around it (Mapping). If a map is known apriori, then the task of localization can be treated as a feature matching task based on current observations, but this is usually not the case for real-world robots. Most robots are required to have the ability of exploring the environment and map it for future use, for example, robots used for surveying caves, mines or areas affected by natural disasters. Doing this mapping is not possible without knowing where the robot is in the map at each instant. This also can't be directly computed via control inputs and motion models due to noise induced by real-world actuators and external factors, therefore this process has to be bootstrapped with temporal consistency of observations. Using cameras for such a application is desirable because cameras are cheaper compared to other sensors like LiDAR and they are extremely dense and rich in features. So Visual-Inertial Odometry can be a great alternative in areas where we don't have direct odometry from something like Global Positioning System (GPS).

### C. Approach

We use the Extended Kalman Filter (EKF) to solve the problem of SLAM. The Kalman Filter acts as the optimal estimator in the linear case. But when this linearity assumption is broken, we use the EKF to approximate the non-linear system as a linear system. We have a non-linear motion model and a non-linear observation model. We approximate both of these as linear systems via their first-order Taylor series expansions and then use this approximation in the Non-Linear Bayes Filter equations.

## II. Problem Formulation

### A. IMU Pose Estimation using EKF Prediction

Given IMU measurements, $\mathbf{u}_{0:t} = \begin{bmatrix}\mathbf{v}_{0:t} & \boldsymbol{\omega}_{0:t}\end{bmatrix}$ where $\mathbf{u}_t \in \mathbb{R}^6$ is the Twist. $\mathbf{v}_t$ is the linear velocity and $\boldsymbol{\omega}_t$ is the angular velocity measure obtained from the IMU in body-frame. We want to predict the current pose of the robot (same as IMU), $T_{0:t}$. The update is performed via a motion model which uses the current pose of the robot $T_t \in SE(3)$, Twist measured via the IMU $\mathbf{u}_t$ and Gaussian noise $w_t$ given as:

$$T_{t+1|t} = f(T_t, \mathbf{u}_{0:t}, w_t) \tag{1}$$

where $f(.)$ is the motion model.

### B. Landmark Mapping via EKF Update

The objective here is to estimate static landmark positions, $\mathbf{m} \in \mathbb{R}^{3\times M}$ using the visual features observed in images. The image observations are marked as $\mathbf{z}_t \in \mathbb{R}^4 \times M$. Given image observations $\mathbf{z}_{0:t}$ and IMU poses $T_{0:t}$, the objective is to estimate the $3D$ locations of the landmarks in the world $\mathbf{m}$. The image observations are the image pixel coordinates in the left and right cameras of the stereo camera setup. Therefore they can be written as:

$$\mathbf{z}_t = \begin{bmatrix}u_L, v_L, u_R, v_R\end{bmatrix} \tag{2}$$

where $(u, v)$ represent the pixel coordinates of the feature. At any timestep, $\mathbf{z}_t \in \mathbb{R}^{4\times N_t}$, where $N_t$ represents the landmarks visible at the current timestep.

Based on our current estimate of the IMU location, we can also get the predicted measurements of the landmarks as:

$$\tilde{\mathbf{z}}_t = h(T_t, \mathbf{m}, v_t) \tag{3}$$

where $v_t$ is the Gaussian noise associated with the observation model.

The function $h(.)$ is the observation model and makes use of the stereo camera calibration matrix defined as:

$$K_s = \begin{bmatrix} fs_u & 0 & c_u & 0 \\ 0 & fs_v & c_v & 0 \\ fs_u & 0 & c_u & -fs_u b \\ 0 & fs_v & c_u & 0 \end{bmatrix} \tag{4}$$

where $(fs_u, fs_v, c_u, c_v)$ define the camera intrinsics and $b$ is the baseline of the stereo setup.

The function $h(.)$ also uses ${}_IT_C$ which is the transformation to go from the camera to IMU frame, or equivalently it is the pose of the Camera in the IMU frame.

*C. Visual-Inertial Simultaneous Localization And Mapping (SLAM)*

At any time $t$, we have the robot pose $T_t$, control command at this time are given by $\mathbf{u}_t$ and the observation is given by $\mathbf{z}_t$. Our aim is to find the map denoted as $\mathbf{m}$ and the trajectory of poses of the robot denoted as $T_{0:T} = T_0, T_1, \ldots, T_T$, given the control commands $\mathbf{u}_{0:T} = \mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{T-1}$ and the observations $\mathbf{z}_{0:T} = \mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_{T-1}$. This can be written mathematically as estimation the below probability distribution:

$$p(T_{0:T}, \mathbf{m}|\mathbf{z}_{0:T-1}, \mathbf{u}_{0:T-1}) \tag{5}$$

We solve this problem while maintaining causal consistency, so any time $t$, only the data till $t-1$ is available. We also only estimate out pose at the current timestep $t$ rather than the entire trajectory at once which is also known as the Online SLAM problem. So at any time $t$, we estimate:

$$p(T_t, \mathbf{m}|\mathbf{z}_{0:t-1}, \mathbf{u}_{0:t-1}) \tag{6}$$

We also follow the regular assumptions associated with a extended Kalman Filter, i.e.

- The prior pdf $p_{t|t}$ is Gaussian
- The motion noise $w_t$ and observation noise $v_t$ are independent of each other, of the state, and across time.

## III. Technical Approach

Our goal is to estimate the distribution, $p(T_t, \mathbf{m}|\mathbf{z}_{0:t-1}, \mathbf{u}_{0:t-1})$. We can do this via a Bayes Filter.

*A. Bayes Filter*

The Bayes filter is a probabilistic inference technique for estimating the state $\mathbf{x}_t$ by combining the effects of control input $\mathbf{u}_t$ and observations $\mathbf{z}_t$ while exploiting the Markov Assumptions, conditional probability, total probability and Bayes rule. It keeps track of two two Probability Distribution Functions (pdfs):

$$\textbf{Predicted pdf :}\ p_{t+1|t}(\mathbf{x}_{t+1}) = p(\mathbf{x}_{t+1}|\mathbf{z}_{0:t}, \mathbf{u}_{0:t}) \tag{7}$$

$$\textbf{Updated pdf :}\ p_{t+1|t+1}(\mathbf{x}_{t+1}) = p(\mathbf{x}_{t+1}|\mathbf{z}_{0:t+1}, \mathbf{u}_{0:t}) \tag{8}$$

*1) Markov Assumptions:*

- The state $\mathbf{x}_{t+1}$ only depends on previous input $\mathbf{u}_t$ and state $\mathbf{x}_t$, i.e., $\mathbf{x}_{t+1}$ given $\mathbf{u}_t$, $\mathbf{x}_t$ is independant of the history $\mathbf{x}_{0:t-1}, \mathbf{z}_{0:t-1}, \mathbf{u}_{0:t-1}$
- The observation $\mathbf{z}_t$ is only dependant on the state $\mathbf{x}_t$

*2) Prediction Step:* Given a prior belief for the state $\mathbf{x}_t$ as $p_{t|t}$ and control input $\mathbf{u}_t$, we can use the probabilistic motion model $p_f$ to compute the predicted pdf $p_{t+1|t}$ of $x_{t+1}$:

$$p_{t+1|t}(\mathbf{x}) = \int p_f(\mathbf{x}|\mathbf{s}, \mathbf{u}_t) p_{t|t}(\mathbf{s}) d\mathbf{s} \tag{9}$$

*3) Update Step:* Given a predicted pdf $p_{t+1|t}$ of $\mathbf{x}_{t+1}$ and measurement $\mathbf{z}_{t+1}$, we can use the probabilistic observation model $p_h$ to obtain the updated pdf $p_{t+1|t+1}$ of $x_{t+1}$:

$$p_{t+1|t+1}(\mathbf{x}) = \frac{p_h(\mathbf{z}_{t+1}|\mathbf{x}) p_{t+1|t}(\mathbf{x})}{\int p_h(\mathbf{z}_{t+1}|\mathbf{s}) p_{t+1|t}(\mathbf{s})} \tag{10}$$

*B. Extended Kalman Filter*

The Extended Kalman Filter is derived from the Bayes Filter by imposing some assumptions on it, specifically:

- The prior pdf $p_{t|t}$ is Gaussian.
- The motion noise $w_t$ and observation noise $v_t$ are independent of each other, of the state, and across time.
- The predicted and updated pdfs are forced to be Gaussian via approximation.
- The motion and observation models are linearized via first order Taylor series expansions.

Once these assumptions are imposed and the equations are solved, we get gaussian distributions at the end of the predict and update steps and that is desirable as that makes the distributions easy to store and propagate.

The motion and the observation noise are hyperparameters that quantify how much you trust the predict and update step respectively and are zero-mean gaussians.

*1) Linearization of Motion and Observation models:* We use a first-order Taylor series expansion to approximate the models. For the motion model, we write it as:

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t, w_t) \approx f(\mathbf{x}_t, \mathbf{u}_t, 0) + F_t(\mathbf{x}_t - \mu_{t|t}) + Q_t w_t \tag{11}$$

where:

$$F_t = \frac{df}{dx}(\mathbf{x}_t, \mathbf{u}_t, w_t)\Big|_{\mu_{t|t}, \mathbf{u}_t, 0} \tag{12}$$

$$Q_t = \frac{df}{dw}(\mathbf{x}_t, \mathbf{u}_t, w_t)\Big|_{\mu_{t|t}, \mathbf{u}_t, 0} \tag{13}$$

Thus the expected mean and covariance of the predicted pdf become:

$$\mu_{t+1|t} \approx f(\mu_{t|t}, \mathbf{u}_t, 0) \tag{14}$$

$$\Sigma_{t+1|t} \approx F_t \Sigma|t|t F_t^T + Q_t W Q_t^T \tag{15}$$

For the observation model we can do:

$$h(\mathbf{x}_{t+1}, v_{t+1}) \approx h(\mu_{t+1|t}, 0) + H_{t+1}(\mathbf{x}_{t+1} - \mu_{t+1|t}) + R_{t+1} v_{t+1} \tag{16}$$

where:

$$H_{t+1} = \frac{dh}{dx}(\mathbf{x}_{t+1}, v_{t+1})\Big|_{\mu_{t+1|t}, 0} \tag{17}$$

$$R_{t+1} = \frac{dh}{dv}(\mathbf{x}_{t+1}, v_{t+1})\Big|_{\mu_{t+1|t}, 0} \tag{18}$$

*2) Motion Model:* The Motion model computed the predicted pose of the robot based on the control command and the current pose of the robot. We use the IMU measurements as the control commands and the. The IMU gives us $\mathbf{u}_t = \begin{bmatrix} \mathbf{v}_t & \boldsymbol{\omega}_t \end{bmatrix}$ where $\mathbf{u}_t \in \mathbb{R}^6$ is the Twist. $\mathbf{v}_t$ is the linear velocity and $\boldsymbol{\omega}_t$ is the angular velocity. The current estimated pose of the robot at time $t$ is represented as $\mu_{t|t} \in SE(3)$ and the covariance associated with this pose is $\Sigma_{t|t} \in \mathbb{R}^{6\times6}$. Thuss given $\mathbf{u}_t$, we can compute the predicted pose $\mu_{t+1|t}$ as:

$$\mu_{t+1|t} = \mu_{t|t} exp(\tau \hat{\mathbf{u}}_t) \tag{19}$$

where $exp(.)$ is the exponential map that maps any element in $se(3)$ to $SE(3)$. $\tau$ is the time discretization and $\hat{\mathbf{u}}_t \in \mathbb{R}^{4\times4}$ is matrix given by:

$$\hat{\mathbf{u}}_t = \begin{bmatrix} \hat{\boldsymbol{\omega}}_t & \mathbf{v}_t \\ \mathbf{0} & 0 \end{bmatrix} \tag{20}$$

where $\hat{\boldsymbol{\omega}}_t$ is the skew-symmetric matrix of angular velocity $\boldsymbol{\omega}_t$.

The covariance of the predicted pdf is computed as:

$$\Sigma_{t+1|t} = exp(-\tau \overset{\curlywedge}{\mathbf{u}}_t)\Sigma_{t|t} exp(-\tau \overset{\curlywedge}{\mathbf{u}}_t)^T + W \tag{21}$$

where $W \in \mathbb{R}^{6\times6}$ is the motion model noise and and $\overset{\curlywedge}{\mathbf{u}}_t$ is defined as:

$$\overset{\curlywedge}{\mathbf{u}}_t = \begin{bmatrix} \hat{\boldsymbol{\omega}}_t & \hat{\mathbf{v}}_t \\ \mathbf{0} & \hat{\boldsymbol{\omega}}_t \end{bmatrix} \tag{22}$$

where $\hat{\mathbf{v}}_t$ is the skew-symmetric matrix of $\mathbf{v}_t$

*3) Observation Model:* At any time $t$, given the current pose $T_{t+1|t}$ and the map $\mathbf{m}$, the observation model maps this to an observation. In this project, specifically given a map point $m_j$, that corresponds to the $i^{th}$ observation, we have:

$$\tilde{\mathbf{z}}_t = K_s \pi({}_I T_C T_{t+1|t}^{-1} \underline{m}_j) \tag{23}$$

where $\pi(\mathbf{x}) = \frac{1}{x_3}\mathbf{x}$ is the projection function, ${}_C T_I$ is the transformation to go from the IMU to the camera frame, $\underline{m}_j$ is the map point in homogeneous coordinates.

### C. Inverse Observation Model

This is used to initialize a new landmark that is seen for the first time and maps a stereo camera observation to a 3D location. essentially let's say the mean of the landmark $\mu = \begin{bmatrix} x & y & z \end{bmatrix}^T$ and we have a observation $z = \begin{bmatrix} u_L & v_L & u_R & v_R \end{bmatrix}^T$, we have the equations as:

$$d = u_L - u_R = \tfrac{1}{z} f s_u b \tag{24}$$

$$z = \tfrac{f s_u b}{d} \tag{25}$$

$$x = \tfrac{(u_L - c_u)z}{f s_u} \tag{26}$$

$$y = \tfrac{(v_L - c_v)z}{f s_v} \tag{27}$$

$$\mu_W = T_t \mu \tag{28}$$

*1) Update Step for Landmarks:* The objective is to apply the update step of the EKF using the observations obtained from a stereo camera setup. The landmarks already in the map each have mean associated with them, for the $j^{th}$ landmark, the mean is $\mu_j \in \mathbb{R}^3$. Let $\mu \in \mathbb{R}^{M\times3}$ be an array with means of all $M$ landmark. Let $\Sigma \in \mathbb{R}^{3M\times3M}$ be the covariance associated with all these landmarks. If a landmark is seen for the first time, it is initialized using the inverse observation model and the covariance is intialized. When the $j^{th}$ point is already in the map and later observed, we apply the EKF update equations to update it. For this we need the $H_{t+1}$ matrix, which are defined as follows:

$$H_{t+1,i,j} = \begin{Bmatrix} \frac{\partial h(T_{t+1}, \mathbf{m}_j)}{\partial \mathbf{m}_j}, & \Delta_t(j) = i \\ 0, & otherwise \end{Bmatrix} \tag{29}$$

$$\tag{30}$$

where $\Delta_t(.)$ is the data association function and $\Delta_t(j) = i$ if the $j^{th}$ landmark in the map corresponds to the $i^{th}$ observation. $H_{t+1} \in \mathbb{R}^{4N_t\times3M}$, where $N_t$ is the number of landmarks currently visible and have observations and $M$ is the total number of landmarks in the map. So for $i^{th}$ observation corresponding to the $j^t h$ landmark in the map, we fill the block corresponding to the rows $4i$ to $4i+4$ and the columns $3j$ to $3j+3$. The derivative is as follows:

$$\frac{\partial}{\partial m_j} h(T_{t+1}, \mathbf{m}) = K_s \frac{\partial \pi}{\partial m_j}({}_I T_C T_{t+1|t}^{-1} \underline{m}_j)\dot{}_I T_C T_{t+1|t}^{-1} P^T \tag{31}$$

where $P = \begin{bmatrix} I & 0 \end{bmatrix}$ and $\frac{\partial \pi}{\partial q}(q)$ is the jacobian of projection function given by:

$$\frac{\partial \pi}{\partial q}(q) = \begin{bmatrix} 1 & 0 & -\frac{q_1}{q_3} & 0 \\ 0 & 1 & -\frac{q_2}{q_3} & 0 \\ 0 & 0 & 0 & 0 \\ 0- & 0 & -\frac{q_4}{q_3} & 1 \end{bmatrix} \tag{32}$$

Once the H matrix is calculated, we can apply the EKF equations and what we get is as follows:

$$K_{t+1|t} = \Sigma_{t+1|t} H_{t+1}(H_{t+1}\Sigma_{t+1|t} H_{t+1} + I \bigotimes V)^{-1} \tag{33}$$

$$\mu_{t+1|t+1} = \mu_{t+1|t} + K_{t+1|t}(\mathbf{z}_t - \tilde{\mathbf{z}}_t) \tag{34}$$

$$\Sigma_{t+1|t+1} = (\mathbf{I} - K_{t+1|t} H_{t+1})\Sigma_{t+1|t} \tag{35}$$

*2) Update Step for Pose and Landmarks:* This includes updating both the Pose of the robot and the landmarks. We first have to define what the $H_{t+1}$ matrix looks like. Let $\mu^p_{t+1|t} \in SE(3)$ denote the pose mean and $\mu^l_{t+1|t} \in \mathbb{R}^{3M\times3}$ denote landmark means. $\Sigma_{t+1|t} \in \mathbb{R}^{(3M+6)\times(3M+6)}$ is the covariance of the state. Now $H_{t+1} \in \mathbb{R}^{4N_t\times(3M+6)}$ with the first 6 columns now corresponding to the pose. The matrix after these 6 columns is filled exactly as stated in the previous section. The first 6 columns are filled as follows:

$$H_{t+1,i,j} = \begin{Bmatrix} \frac{\partial h(T_{t+1}, \mathbf{m}_j)}{\partial \mathbf{T_{t+1}}}, & \Delta_t(j) = i \\ 0, & otherwise \end{Bmatrix} \tag{36}$$

So for $i^{th}$ observation corresponding to the $j^t h$ landmark in the map, we fill the block corresponding to the rows $4i$ to $4i+4$ and the columns $1$ to $6$.

This derivative when calculated is:

$$\frac{\partial h(T_{t+1}, \mathbf{m}_j)}{\partial \mathbf{T_{t+1}}} = -K_s \frac{\partial \pi}{\partial T_{t+1}}({}_I T_C (\mu^p_{t+1|t})^{-1} \underline{m}_j)\dot{}_I T_C ((\mu^p_{t+1|t})^{-1} \underline{m}_j)^{\odot} \tag{37}$$

where for a $s \in \mathbb{R}^3$, $\odot$ is defined as:

$$\begin{bmatrix} s \\ 1 \end{bmatrix}^{\odot} = \begin{bmatrix} \mathbf{I} & -\hat{s} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{38}$$

Now we can calculate the Kalman gain using this new $H_{t+1}$ as in Eq 33. This will give us $K_{t=1|t} \in \mathbb{R}^{(3M+6)\times4N_t}$ and then the update is done as follows:

$$\mu^p_{t+1|t+1} = \mu_{t+1|t} exp(((K_{t+1|t}(\mathbf{z}_t - \tilde{\mathbf{z}}_t))^{\wedge}) \tag{39}$$

$$\mu^l_{t+1|t+1} = \mu_{t+1|t} + K_{t+1|t}(\mathbf{z}_t - \tilde{\mathbf{z}}_t) \tag{40}$$

$$\Sigma_{t+1|t+1} = (\mathbf{I} - K_{t+1|t} H_{t+1})\Sigma_{t+1|t} \tag{41}$$

**Algorithm 1** Visual-Inertial EKF SLAM

Intialize $\mu^p_{0|0} = I^{4\times4}$ and $\Sigma_{0|0} = \mathbf{0} \in \mathbb{R}^{6\times6}$
**for** timestep $t$ **do**
  Predict Step
  Update pose $\mu^p_{t|t-1} \leftarrow \mu_{t-1|t-1} exp(\tau \hat{\mathbf{u}}_{t-1})$
  Update Covariance of Pose
  $\Sigma_{t|t-1} \leftarrow exp(-\tau \overset{\curlywedge}{\mathbf{u}}_{t-1}) \Sigma_{t-1|t-1} exp(-\tau \overset{\curlywedge}{\mathbf{u}}_{t-1})^T + W$
  Update Step
  **for** Observation $i$ **do**
    If never seen before, Initialize using Eq 24
    If seen before, Then populate H Matrix
    **for** landmark $j$ for which $\Delta_j = i$ **do**
      Fill $H$ matrix rows $4i$ to $4i+4$ columns 0 to 6 for Pose using Eq 37
      Fill $H$ matrix rows $4i$ to $4i+4$ columns $3j+6$ to $3j+6+3$ for landmark 31
    **end for**
    Update using Eq 39
  **end for**
**end for**

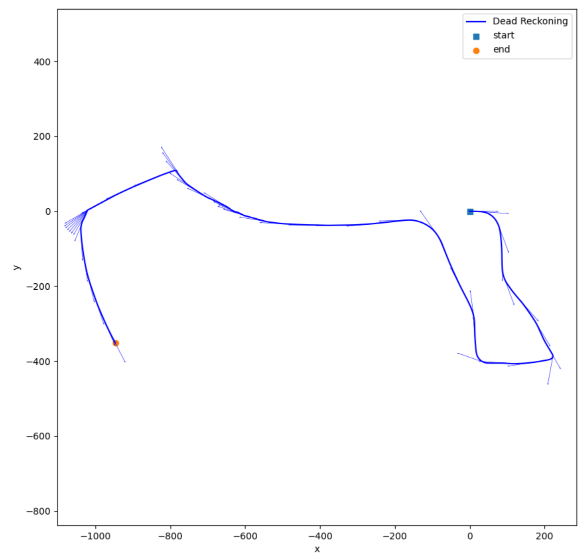


Fig. 2. Dataset 10 - Dead Reckoning

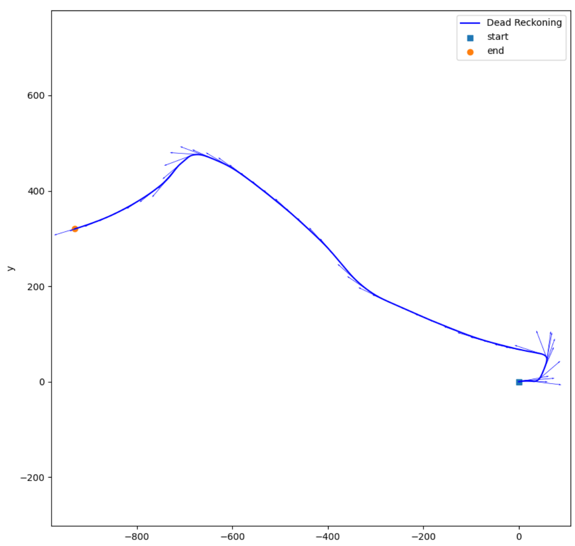


Fig. 1. Dataset 3 - Dead Reckoning

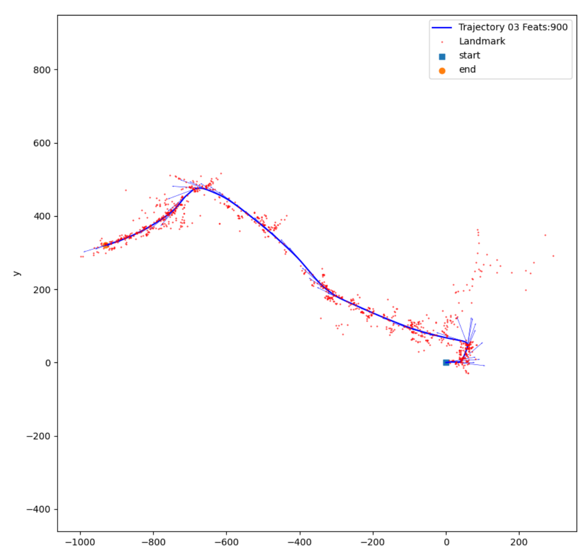


Fig. 3. Dataset 3 - Mapping EKF - 900 Features

The algorithm is summarized as follows:

Features are skipped to reduce compute time, For eg, Every $9^{th}$ feature is used, every $11^{th}$ feature is used, etc.

## IV. Results

### A. IMU Pose Estimation using EKF Prediction

Figure 1 and 2 show the trajectory of the robot if only the update step is applied according to Eq 19 and Eq 21. This is also known as the Dead Reckoning trajectory as there are no corrections from observations.

### B. Landmark Mapping via EKF Update

Figure 3 and Figure 4 show the mapping output using the trajectory from the previous section and using the Eq 33 for updating the landmark location for 900 features.

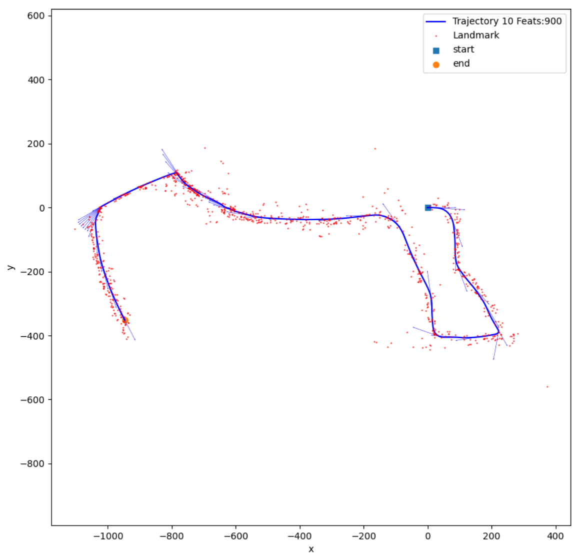


Fig. 4. Dataset 10 - Mapping EKF - 900 Features

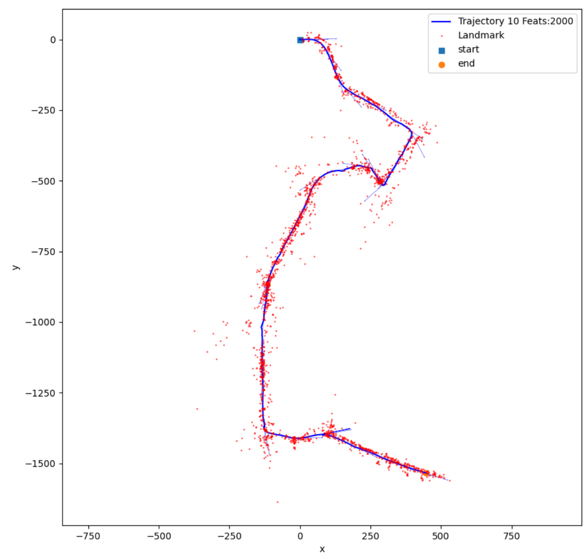


Fig. 6. Dataset 10 - VI SLAM Map with 2000 Features


Fig. 5. Dataset 3 - VI SLAM Map with 2000 Features

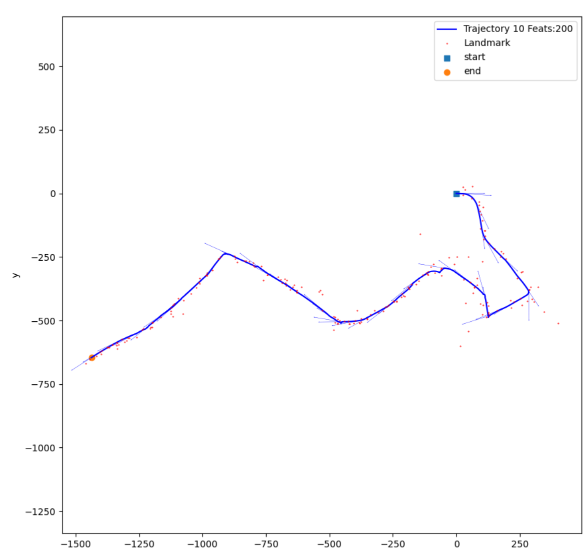


Fig. 7. Dataset 10 - VI SLAM Map with 200 Features

### C. *Visual-Inertial Simultaneous Localization And Mapping (SLAM)*

Figure 5 and Figure 6 show the VI SLAM maps with 2000 landmarks used.

### D. *Discussion*

*1) Changing Number of Landmarks:* As it can be seen from Figure 7 - 19, the map changes quite significantly as more and more features are added. The algorithm is also very sensitive to which feature is skipped and which is not.

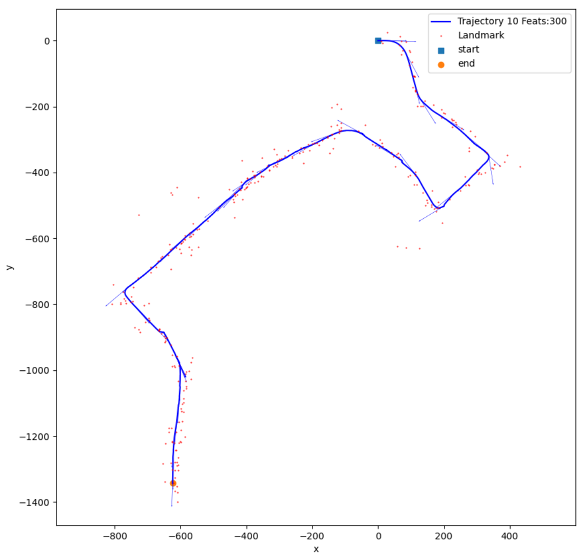


Fig. 8. Dataset 10 - VI SLAM Map with 300 Features

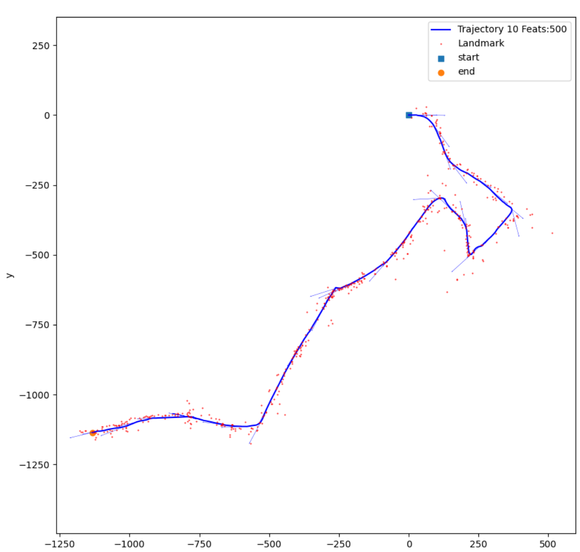


Fig. 9. Dataset 10 - VI SLAM Map with 500 Features

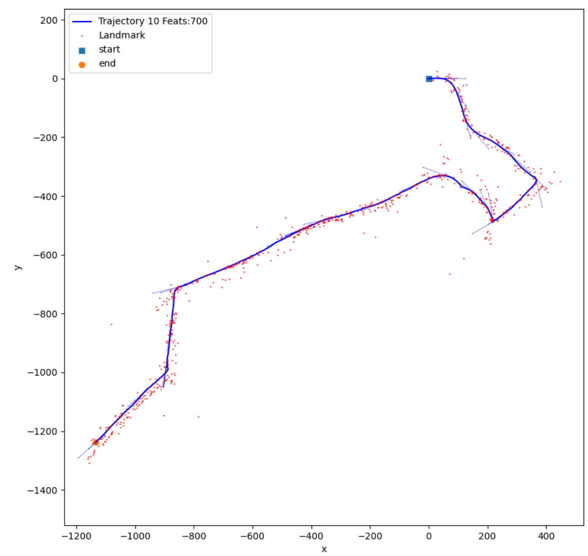


Fig. 10. Dataset 10 - VI SLAM Map with 700 Features

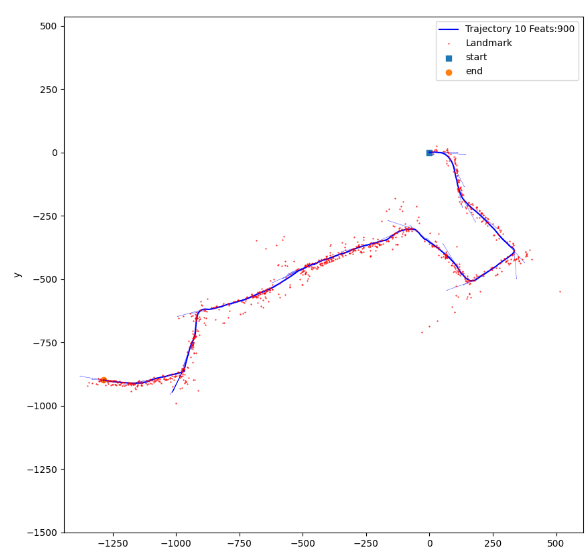


Fig. 11. Dataset 10 - VI SLAM Map with 900 Features

*2) Motion and Observation Model noise:* The final values used for this are:

$$W = \begin{bmatrix} 0.2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.02 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.02 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.02 \end{bmatrix} \tag{42}$$

$$V = \begin{bmatrix} 25 & 0 & 0 & 0 \\ 0 & 25 & 0 & 0 \\ 0 & 0 & 25 & 0 \\ 0 & 0 & 0 & 25 \end{bmatrix} \tag{43}$$

The motion model noise had to be low for a good result and the observation model noise higher. The algorithm is also

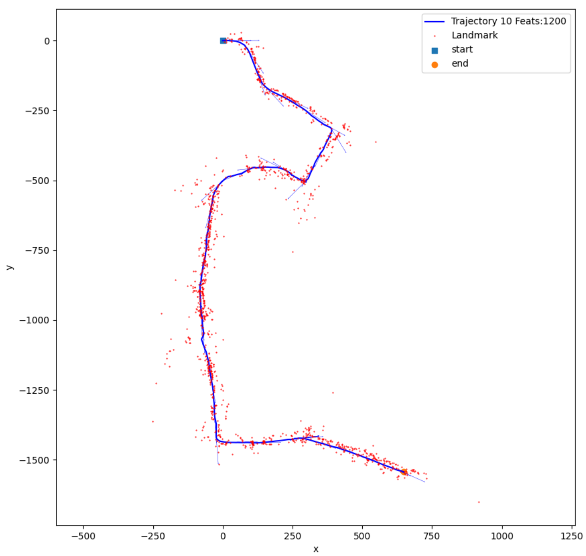


Fig. 12. Dataset 10 - VI SLAM Map with 1200 Features

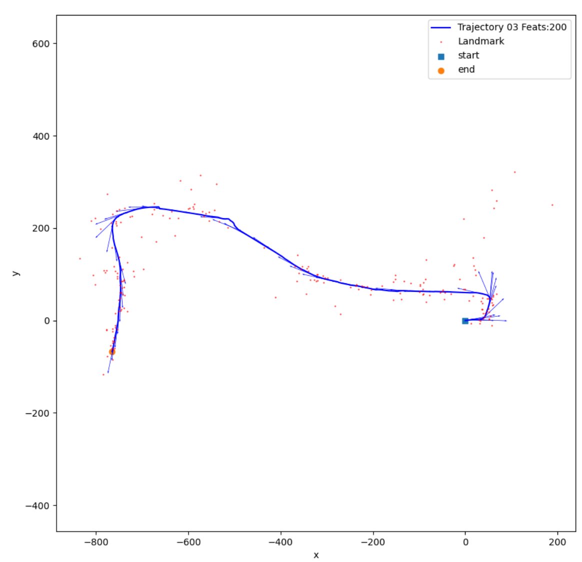


Fig. 13. Dataset 3 - VI SLAM Map with 200 Features

extremely sensitive to these values and gives very different plots for different values of the noise.

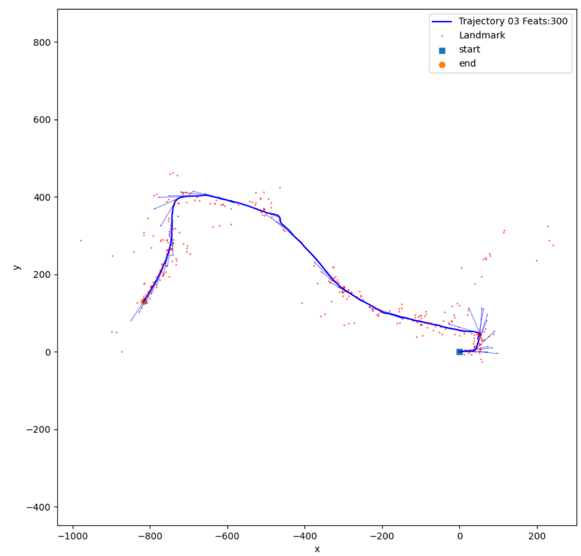


Fig. 14. Dataset 3 - VI SLAM Map with 300 Features

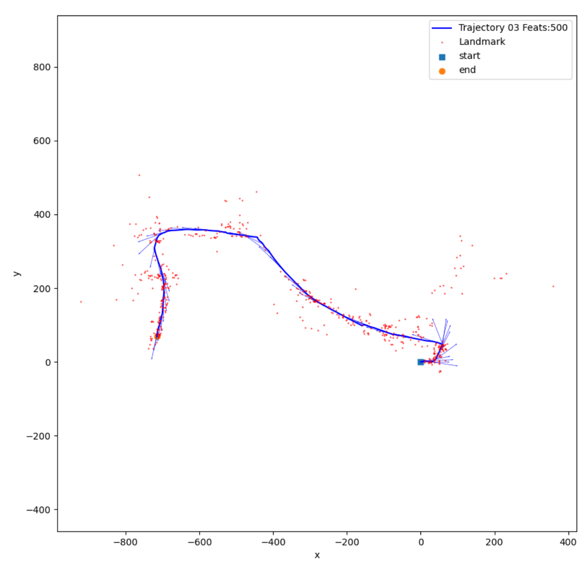


Fig. 15. Dataset 3 - VI SLAM Map with 500 Features

*3) Initial Covariance:* The initial covariance that worked best for the pose is:

$$\Sigma = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{44}$$

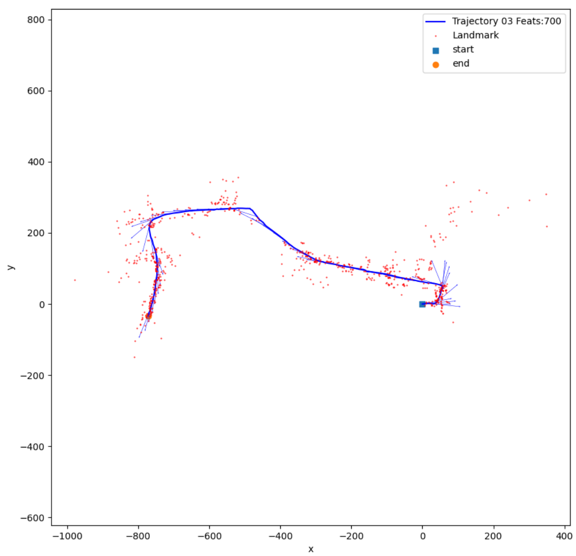


Fig. 16. Dataset 3 - VI SLAM Map with 700 Features

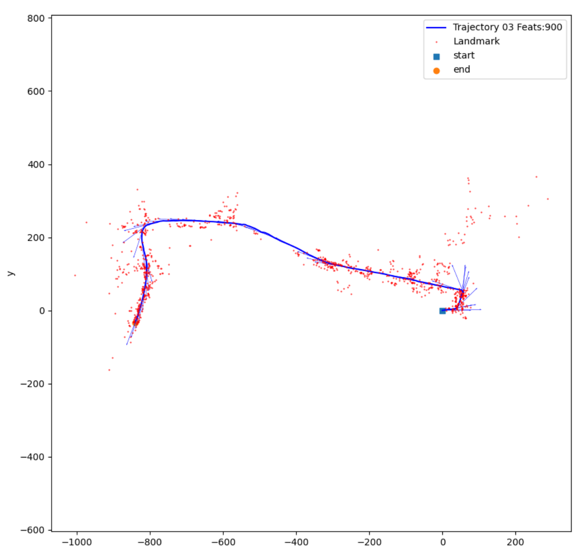


Fig. 17. Dataset 3 - VI SLAM Map with 900 Features

and the best for the landmark was:

$$\Sigma = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix} \tag{45}$$

I also tried to assign the robot location covariance when the landmark was detected for the first time as the initial landmark covariance but that gave worse results.

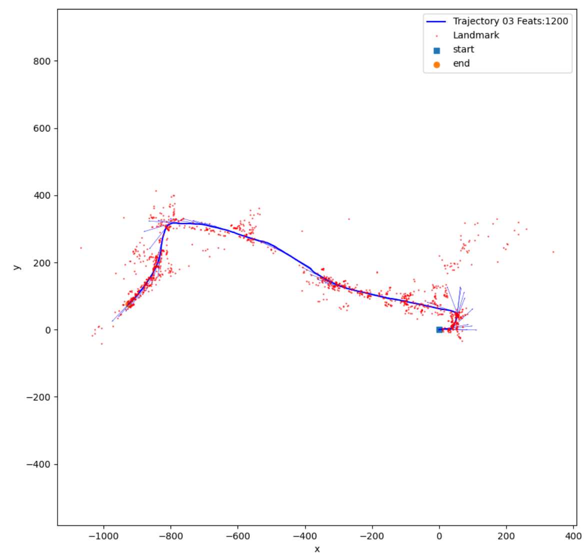


Fig. 18. Dataset 3 - VI SLAM Map with 1200 Features

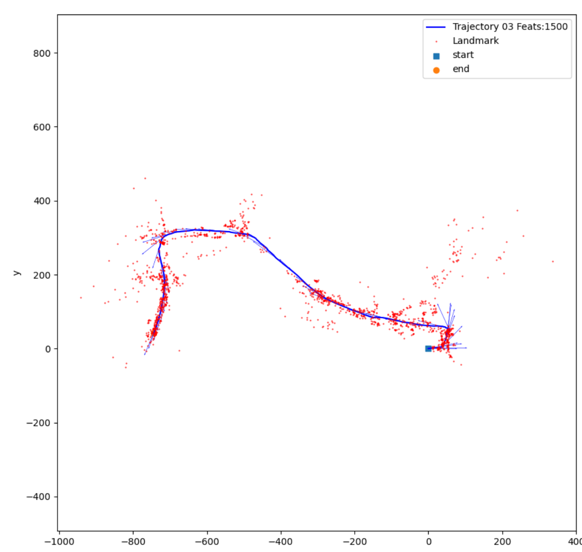


Fig. 19. Dataset 3 - VI SLAM Map with 1500 Features

## V. Conclusion and Future Work

- Come up with a better way to initialize the landmark covariance based on the current robot covariance.
- Make the code faster and process more features.
- The implementation is very sensitive to the noise matrices, reducing this sensitivity.
- Adding Bayes Smoother and Loop Closure capabilities

to the algorithm to improve the final map generated.
- Use a better algorithm for skipping features than just skipping equally. This could be something like features at a turn are more important than features on straights etc.

## VI. Acknowledgements

Thank you for this wonderful course. It is one of the best courses I have ever taken. The content was insightful and beautiful and the way it was taught and internalized through the classes and the assignments was perfect. Thank you Nikolay, Shrey, Sambharan and Yigit.