Visual-Inertial SLAM using Extended Kalman Filter

Shrey Kansal
Mechanical and Aerospace Engineering
University of California San Diego
skansal@ucsd.edu

I. Introduction

There has been rapid development in the field of self-driving cars, automated logistics, drones, and manufacturing techniques. All these areas can be justifiably grouped into what we call Robotics. Regardless of the area of application, sensing and estimation has been at the core of research and development for robotics. As the name suggests, sensing and estimation is essential for a robot to function with as much dexterity and perception as humans. The latter of the two, Estimation comprises of tracking the movement of the robot using sensors like encoders, IMU and gyroscope, and estimating its environment using LiDAR scans and stereo camera. It is crucial to understand how the pose of various sensors change in accordance with robot movement in order to accurately model them.

From an analytical point of view, SLAM (Simultaneous Localization and Mapping) is the computational problem of constructing or updating the map of an unknown environment while simultaneously keeping track of the robot's position and orientation within it. In more naïve terms, it can be considered as a chicken-and-egg problem as both the map of environment and the robot's pose are interdependent. While there exists algorithms such as Particle Filter and learning based approaches, in this paper we implement a Extended Kalman Filter based algorithm. We use IMU for the robot's pose, and stereo camera images data for visual mapping. We implement the SLAM algorithm on two different data sets.

II. Problem Formulation

Our problem statement of Simultaneous Localization and Mapping (SLAM) comprises of estimating the robot's pose ($\boldsymbol{x_t}$) at each time step with respect to its changing environment ($\boldsymbol{m_t}$) and vice-versa. Hence, we define our problem in two parts: predicting the current position of the car, updating it using inputs from stereo camera images feature data.

We have two data sets (03.npz and 10.npz). It is observed that data and images from all sensors in both data sets is synchronous and does not require pre-processing.

A. IMU Localization

Motion model is a nonlinear function f or equivalently a probability density function p_f that describes the motion of the robot to a new state $\boldsymbol{x_{t+1}}$ after applying control input $\boldsymbol{u_t}$ at state $\boldsymbol{x_t}$.

$$\boldsymbol{x_{t+1}} = f(\boldsymbol{x_t}, \boldsymbol{u_t}, \boldsymbol{w_t}) \sim p_f(\cdot|\boldsymbol{x_t}, \boldsymbol{u_t})$$
$$\boldsymbol{w_t}\ is\ motion\ noise\ (Gaussian)$$

$$\boldsymbol{\mu_{p,t+1|t}} = \boldsymbol{\mu_{p,t|t}} \exp(\tau_t \boldsymbol{\hat{u}_t})$$

Motion model simply describes the movement of the robot, given the data collected from various sensors. We implement it in the form of SE(3) pose, which propagates with time.

B. Landmark Mapping

Observation model is a nonlinear function h or equivalently a probability density function p_h that describes the observation $\boldsymbol{z_t}$ of the robot depending on state $\boldsymbol{x_t}$ and map $\boldsymbol{m_t}$.

$$\boldsymbol{z_t} = h(\boldsymbol{x_t}, \boldsymbol{m_t}, \boldsymbol{v_t}) \sim p_h(\cdot|\boldsymbol{x_t}, \boldsymbol{m_t})$$
$$\boldsymbol{v_t}\ is\ motion\ noise$$

$$\boldsymbol{z_t} = h(\boldsymbol{\mu_p}, \boldsymbol{m}) + \boldsymbol{v_t} := M\pi(\boldsymbol{\mu_p}^{-1} T_{imu\_cam} \underline{\boldsymbol{m}}) + \boldsymbol{v_t}$$

C. Extended Kalman Filter SLAM

Combining the Motion and Observation Models, we get a joint distribution as follows:

$$p(x_{0:t}, m, z_{0:t}, u_{0:t-1})$$
$$= p_0(x_0, m) \prod_{t=0}^{t} p_h(z_i|x_t, m) \prod_{t=1}^{t} p_f(x_t|x_{t-1}, u_{t-1})$$

Extended Kalman Filter is essentially a Bayes Filter with following assumptions:

1. The prior pdf $p_{t|t}$ is Gaussian.
2. The motion model is non-linear with Gaussian noise $\boldsymbol{w}_t$.
3. The observation model is non-linear with Gaussian noise $\boldsymbol{v}_t$.
4. The motion noise $\boldsymbol{w}_t$ and observation noise $\boldsymbol{v}_t$ are independent of each other, of the state $\boldsymbol{x}_t$, and across time.
5. The predicted and updated pdfs are forced to be Gaussian via first-order Taylor series approximation.

In case of EKF SLAM, we update the position estimates ($\boldsymbol{\mu}$), Kalman Gain ($\boldsymbol{K}$) and the joint Covariance ($\boldsymbol{\Sigma}$) of the system simultaneously.

III. Technical Approach

SLAM can be described as parameter estimation problem for state, $\boldsymbol{x}_{0:T}$ and map $\boldsymbol{m}$ given a dataset of the robot inputs, $\boldsymbol{u}_{0:T-1}$ and observations $\boldsymbol{z}_{0:T}$. In the given case, we implement Extended Kalman Filter SLAM with following steps:

A. IMU Localization via EKF Prediction

Time Discretization:

We have 1010 timestamps in UNIX standard seconds-since-the-epoch January 1, 1970. Since we are interested in time intervals instead of the actual time passed, we take

$$\tau_k = t_{k+1} - t_k$$

Motion Model:

We have two sets of synchronous IMU data consisting of angular ($\boldsymbol{\omega}$) and linear velocities ($\boldsymbol{v}$). We assume prior $\mu_{0|0} = I \in SE(3)$ and $\Sigma_{0|0} = I \in \mathbb{R}^{6\times 6}$. The motion model for the nominal kinematics of $\boldsymbol{\mu}_{p,t|t}$ with time discretization τ_t:

$$\boldsymbol{\mu}_{p,t+1|t} = \boldsymbol{\mu}_{p,t|t}\exp(\tau_t \hat{\boldsymbol{u}}_t)$$

EKF Predict:

The EKF Prediction Step with $\boldsymbol{w}_t \sim \mathcal{N}(0, W_p)$:

$$\boldsymbol{\mu}_{p,t+1|t} = \boldsymbol{\mu}_{p,t|t}\exp(\tau_t \hat{\boldsymbol{u}}_t)$$

$$\begin{aligned}\Sigma_{p,t+1|t} &= F_p \Sigma_{t|t} F_p^T \\ &= \exp(-\tau \breve{\boldsymbol{u}}_t)\, \Sigma_{p,t|t} \exp(-\tau \breve{\boldsymbol{u}}_t)^T \\ &\quad + W_p\end{aligned}$$

where,

$$\boldsymbol{u}_t := \begin{bmatrix} \boldsymbol{v}_t \\ \boldsymbol{\omega}_t \end{bmatrix} \in \mathbb{R}^6$$

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

$$\breve{\boldsymbol{u}}_{t_t} := \begin{bmatrix} \hat{\boldsymbol{\omega}}_t & \hat{\boldsymbol{v}}_t \\ \boldsymbol{0} & \hat{\boldsymbol{\omega}}_t \end{bmatrix} \in \mathbb{R}^{6\times 6}$$

where W_p is motion noise, initialized as 0.001*I. The IMU covariance $\Sigma_{p,t|t}$ was initialized as 0.001*I along the diagonals.

Perturbations were ignored in the case of IMU Localization as it did not yield the desired results.

These equations were implemented in Python with the aid of NumPy and SciPy. The nominal kinematics $\boldsymbol{\mu}_{p,t+1|t}$ were stored in a global array with respect to time and was plotted using Matplotlib. The trajectory for both data sets can be visualized in Fig.1 and Fig.2 respectively.

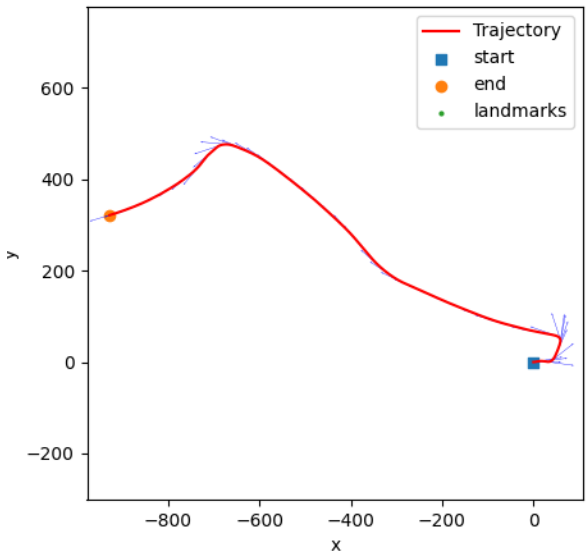

Fig. 1 IMU Localization for 03.npz file

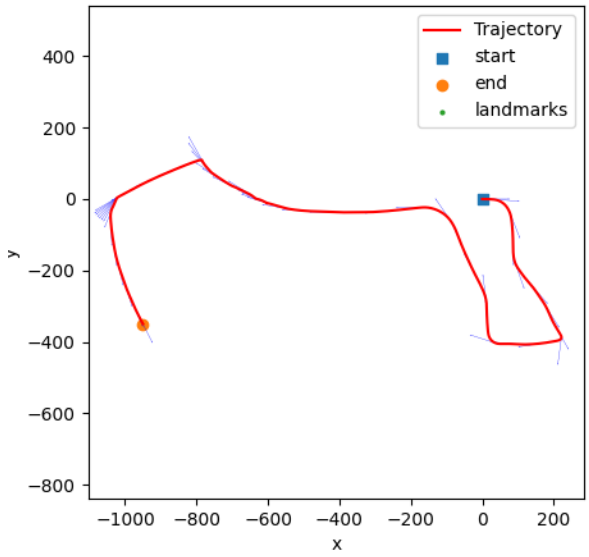

Figure 2 IMU Localization for 10.npz file

B. Landmark Mapping via EKF Update

Transformation of feature values:

We are given feature sets in the shape of (4, m, T), where m represents the total number of features for a given data set. These feature values are in the form of pixel coordinates (u_L, v_L, u_R, v_R). At any given timestep, if the feature is not observed, a value of -1 is returned for all 4 data points. We maintain an index array from the start indicating all the points where $u_L \neq -1$. The T_{imu_cam} is rotated around x-axis by 180deg.

The observation model with measurement noise $\boldsymbol{v}_{t,i} \sim \mathcal{N}(0, V)$ is given by:

$$\boldsymbol{z}_t = h(\boldsymbol{\mu}_p, \boldsymbol{m}) + \boldsymbol{v_t} := M\pi(\boldsymbol{\mu}_p{}^{-1} T_{imu\_cam} \underline{\boldsymbol{m}}) + \boldsymbol{v_t}$$

where, $\boldsymbol{m}$ are the world frame coordinates of landmarks.

First, we add Gaussian noise to the given feature data. Next, we transform these pixel coordinates to optical coordinates (x, y, z) with the aid of following equations:

$$z = \frac{f s_u b}{u_L - u_R}$$

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

Next, we transform these optical coordinates to homogenous world coordinates with the aid of following equation:

$$\boldsymbol{m} = \boldsymbol{\mu}_p T_{imu\_cam} \boldsymbol{x}$$

where, $\boldsymbol{x}$ is the vector containing optical coordinates.

EKF Update:

Next, our aim is to initialize the nominal world coordinates and a non-zero covariance for each point that is visible for the first time. We devise a short algorithm to get such data point and initialize its nominal position in world coordinates from the above equation. Post this, we extract the world coordinates as an array:

$$\underline{\boldsymbol{\mu}}_{m,t} \in \mathbb{R}^{4m}$$

At each timestep, we get different number (N_t) of visible features; hence, we maintain a loop over N_t to compute the predicted observations $(\tilde{\boldsymbol{z}})$ based on $\underline{\boldsymbol{\mu}}_t$, and known correspondences. Simultaneously, the Jacobian (H) of $\tilde{\boldsymbol{z}}$ is computed with respect to the world coordinates evaluated at $\underline{\boldsymbol{\mu}}_{m,t}$.

$$\tilde{\boldsymbol{z}}_t = M\pi\left(T_{cam\_imu}\, \boldsymbol{\mu}_p{}^{-1} \underline{\boldsymbol{\mu}}_{m,t}\right)$$

$$H_m = \begin{cases} M \frac{d\pi}{dq}\left(T_{cam\_imu}\, \boldsymbol{\mu}_p{}^{-1} \underline{\boldsymbol{\mu}}_{m,t}\right)\left(T_{cam\_imu}\, \boldsymbol{\mu}_p{}^{-1} P^T\right); \Delta_t = i \\ \mathbf{0}; otherwise \end{cases}$$

Now, we compute the Kalman Gain (K) and update nominal positions $(\boldsymbol{\mu}_{t+1})$ of each visible feature and the covariance $(\Sigma_{m,t+1})$ too.

$$\boldsymbol{\mu}_{m,t+1|t} + K_{t+1|t}(\boldsymbol{z}_t - \tilde{\boldsymbol{z}}_t)$$

C. Visual-Inertial SLAM

For the Visual-Inertial SLAM, the predict and update steps from part A (IMU Localization) and part B (Visual Mapping) are merged. However, there is a difference in how we maintain and update the system variables (state $\boldsymbol{\mu}$ and covariance Σ).

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_m \\ \boldsymbol{\mu}_p \end{bmatrix} \in \mathbb{R}^{3M+6}$$

$$\Sigma \in \mathbb{R}^{(3M+6)\times(3M+6)}$$

where, $\boldsymbol{\mu}_m$ is the estimated landmark position, and $\boldsymbol{\mu}_p$ is the estimated inverse IMU pose consisting of 6 degrees of freedom.

Since all landmarks are assumed to be static, we do not predict covariance during landmark mapping; hence it can be derived from IMU Localization step. Moreover, the states can be individually predicted as follows:

$$\boldsymbol{\mu}_{t+1|t} = \begin{bmatrix}\boldsymbol{\mu}_{m,t+1|t}\\ \boldsymbol{\mu}_{p,t+1|t}\end{bmatrix} = \begin{bmatrix}\boldsymbol{\mu}_{m,t+1|t}\\ \boldsymbol{\mu}_{p,t|t}\exp\left(-\tau\widehat{\boldsymbol{u}}_{\boldsymbol{t}}\right)\end{bmatrix}$$

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

$$F_t = \begin{bmatrix}\boldsymbol{I} & \boldsymbol{0}\\ \boldsymbol{0} & \exp\left(-\tau\widehat{\boldsymbol{u}}_{\boldsymbol{t}}\right)\end{bmatrix} \in \mathbb{R}^{(3M+6)\times(3M+6)}$$

$$W = \begin{bmatrix}0 & 0\\ 0 & W_p\end{bmatrix} \in \mathbb{R}^{(3M+6)\times(3M+6)}$$

where, W_p is the process noise from IMU.

The update step for EKF SLAM is formed by merging the update steps of both IMU Localization and Visual Mapping. We set out to concatenate the two Jacobians, compute combined Kalman Gain and Covariance for the complete system.

$$H_{t+1|t} = \begin{bmatrix}H_{m,t+1|t} & H_{p,t+1|t}\end{bmatrix} \in \mathbb{R}^{4N_t\times(3M+6)}$$

where $H_{m,t+1|t}$ and $H_{p,t+1|t}$ is the Jacobian of $\tilde{\boldsymbol{z}}_{t+1}$ with respect to $\boldsymbol{m}$ and $\boldsymbol{\mu}_{p,t+1}$ evaluated at $\boldsymbol{\mu}_{m,t}$ and $\boldsymbol{\mu}_{p,t+1}$ respectively.

$$H_{p,t+1|t} = -M\frac{d\pi}{dq}\left(T_{cam\_imu}\,{\boldsymbol{\mu}_p}^{-1}\underline{\boldsymbol{\mu}_{m,t}}\right)T_{cam\_imu}\,({\boldsymbol{\mu}_p}^{-1}\underline{\boldsymbol{\mu}_{m,t}})^{\odot}$$

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

$$\boldsymbol{\mu}_{t+1|t+1} = \begin{bmatrix}\boldsymbol{\mu}_{m,t+1|t+1}\\ \boldsymbol{\mu}_{p,t+1|t+1}\end{bmatrix}$$

$$= \begin{bmatrix}\boldsymbol{\mu}_{m,t+1|t} + \boldsymbol{K}_{t+1|t}(\boldsymbol{z}_t - \tilde{\boldsymbol{z}}_t)\\ \boldsymbol{\mu}_{p,t+1|t}\exp\left((\boldsymbol{K}_{t+1|t}(\boldsymbol{z}_{t+1} - \tilde{\boldsymbol{z}}_{t+1}))\hat{}\,\tau\widehat{\boldsymbol{u}}_{p,t+1|t}\right)\end{bmatrix}$$

$$\Sigma_{t+1|t+1} = \left(I - K_{t+1|t}H_{t+1|t}\right)\Sigma_{t+1|t}\left(I - K_{t+1|t}H_{t+1|t}\right)^T + K_{t+1|t}VK_{t+1|t}^T$$

where, V is the observation noise covariance.

The hyperparameters for this project are listed in Table1.

Table 1 Hyperparameters for EKF SLAM

S.No.	Hyperparameter	03.npz	10.npz
1	Motion noise W	0.001	0.001
2	Observation noise V	0.01	0.01
3	IMU Covariance Σ_p	0.01	0.01
4	Landmarks Covariance Σ_m	0.001	0.001
5	Features skipped	6	12

IV. RESULTS

The EKF SLAM is tested on 2 data sets (03.npz and 10.npz), each containing data for about 1000 and 3000 timesteps respectively. Consequently, the number of observable landmarks also increase. It is observed that the computation time for EKF SLAM $\sim O(n^2)$. Hence, we skipped features for both datasets (6 and 12 features were skipped).

We compare the results of EKF SLAM with IMU based Landmark Mapping without the merged update steps and with the series of images provided. While the IMU based landmark mapping only estimates the position of the landmarks, the EKF SLAM predicts and updates both the pose of the vehicle and the landmark positions simultaneously.

From the comparisons, it is apparent that, the trajectory of robot changes substantially, while still maintaining the essential characteristics such as turns, and total distance travelled. The results can be visualized in Fig.3 and Fig.4.

It is discernible that the results from part a, i.e., IMU Localization and part-b, Landmark Mapping are similar as there is no update step for the trajectory in both cases. However, as we update both trajectory and landmarks in part-c, i.e., EKF SLAM, we see there is respectable amount of difference in the two trajectories as well as the landmark points plotted in the world frame.

V. FUTURE WORK

As part of future work, we would like to quantitatively compare our output of SLAM algorithm with some real-world data. It is often seen that GPS is used in conjunction with the IMU to collect actual and precise data regarding the position of the robot in world frame.

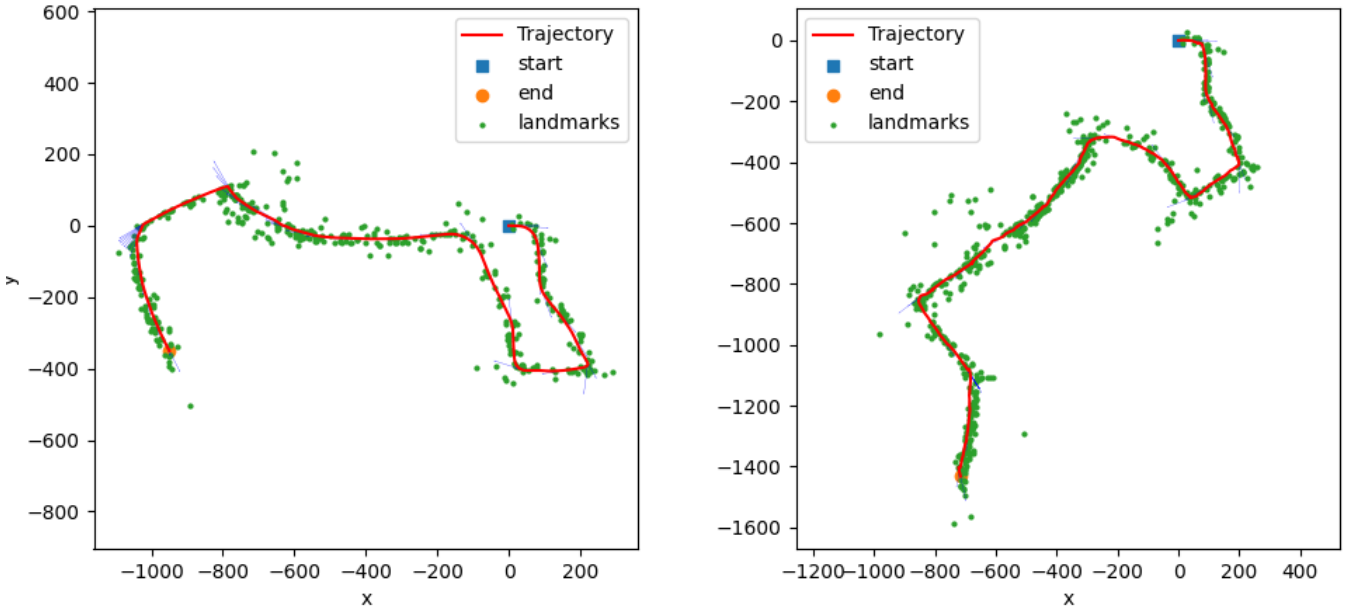

Figure 3 Comparison between Landmark Mapping and EKF SLAM for 10.npz data set

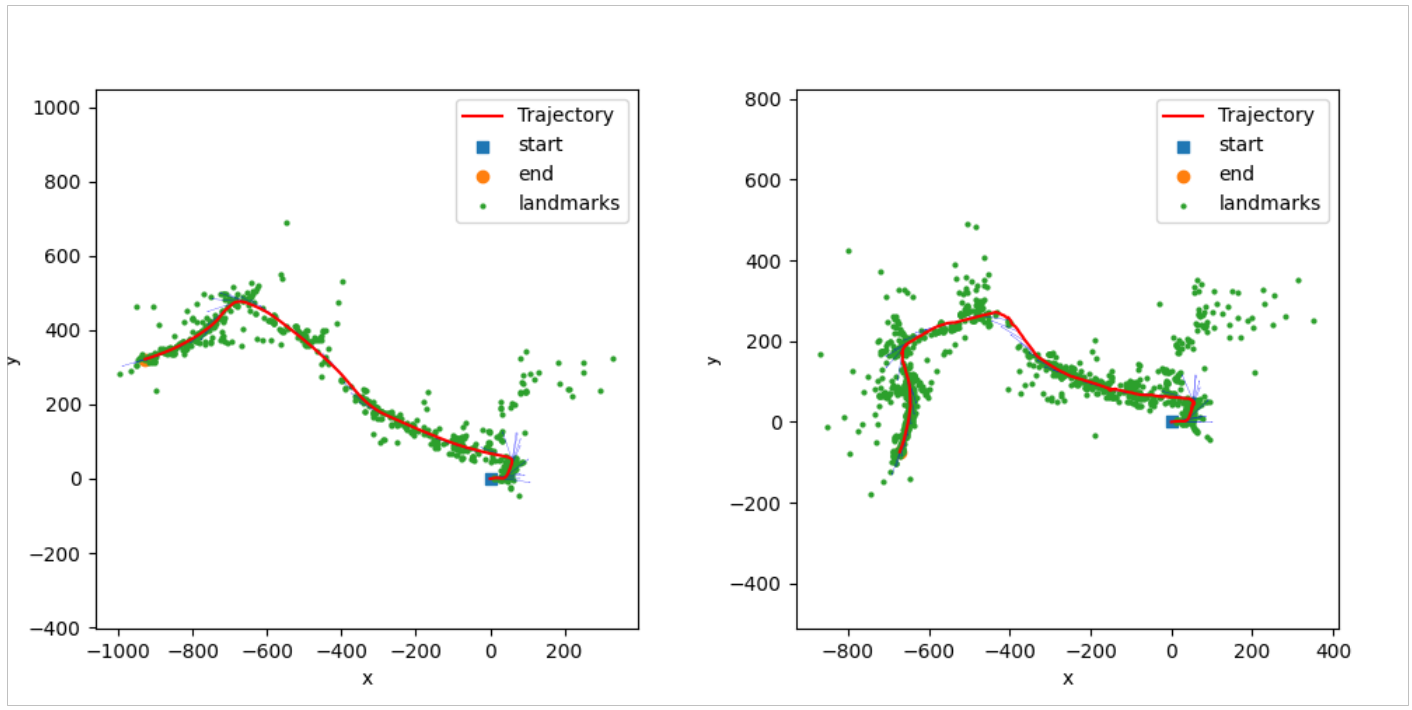

Figure 4 Comparison between Landmark Mapping and EKF SLAM for 03.npz data set