Visual EKF SLAM

Awies Mohammad Mulla

March 24, 2023

1 Introduction

Our goal in this project is to implement Visual-Inertial SLAM algorithm using Extended Kalman Filter (EKF) to estimate the trajectory of robot and the position of landmarks in the given datasets. We have assumed that position of landmarks and the pose of the robot each have a Gaussian distribution. we obtained the linear and angular velocities of the robot from the IMU. From the camera, we get the left and right image pixels of the landmarks. We are also given the stereo camera calibration matrix and the transformation from IMU to the camera optical frame. Our goal is to estimate the world-frame IMU pose over time and world-frame coordinates of the point landmarks that generated the visual features. This is an important problem in the SLAM as in present day, we are trying to navigate the robots using only camera and focusing on eliminating the need of LiDAR. One of the main reasons is beacause it is far more expensive compared to a stereo camera. Other reason is that it might give different errors in changing weather conditions (for example, the point cloud obtained during a rainy or even foggy day could be very noisy as the one compared to sunny day).

We are solving the above mentioned problem using a bayes filter known as Extended Kalman Filter, which consists of a prediction and an update step. In prediction step, the robot pose is predicted using the motion model. In the update step, the robot pose and map is corrected using the observation model. The observation model is used to update the robot pose and the landmark positions simultaneously. This is the reason why we are solving the problem of Simultaneous Localization and Mapping (SLAM). Since, both robot pose and landmark position are gaussian, the product of two gaussian distribution is also gaussian. Hence, we can probabilistically estimate the robot pose and landmark position using the EKF.

2 Problem Formulation

Our goal is to estimate the trajectory of robot and the position of landmarks in the given datasets. We will be implementing Visual-Inertial SLAM algorithm using Extended Kalman Filter (EKF) to solve the mentioned problem. We have assumed that position of landmarks and the pose of the robot each have a Gaussian distribution.

2.1 Motion Model

As we know, the landamrks are fixed in the environment. So, we only need motion model for pose of the robot. The distribution of the pose is given by $T_{t|t} \sim N(\mu_{t|t}, \Sigma_{t|t})$. The motion model in terms of nominal kinematics of $\mu_{t|t}$ and perturbation kinematics of $\delta\mu_{t|t}$ with time discretization τ_t is given by:

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

$$\delta\mu_{t+1|t} = \exp(-\tau_t \overset{\curlywedge}{\mathbf{u}}_t)\ \delta\mu_{t|t} + \mathbf{w}_t \tag{2}$$

From above equations, we get the pose of the robot as $T_{t|t} = \mu_{t|t} \exp(\delta \hat{\mathbf{u}}_{t|t})$ with $\delta_{t|t} \sim N(0, \Sigma_{t|t})$.

2.2 Observation Model

Using the predicted pose of the robot (from motion model), and intrinsic camera parameters we get the following observation model:

$$\begin{aligned} \tilde{\mathbf{z}}_{t+1,i} &= h(T_{t+1|t}, \mathbf{m}_i) + \mathbf{v}_{t+1,i} && (3) \\ &= K_s \pi(T_{O,I} T_{t+1|t}^{-1} \underline{\mathbf{m}}_j) + \mathbf{v}_{t+1,i} && (4) \end{aligned}$$

where $T_{O,I}$ is the transformation from the camera frame to the inertial frame, $\underline{\mathbf{m}}_j$ is the position of the landmark in the inertial frame, K_s is the camera calibration matrix, $\mathbf{v}_{t+1,i}$ is the noise in the observation, and $\tilde{\mathbf{z}}_{t+1,i}$ is the observation of the landmark i at time $t+1$.

2.3 Extended Kalman Filter

Using the above mentioned models we aim to estimate the pose of the robot and the positon of the landmarks. The filter is divided into two parts - prediction and update. The prediction step is given by:

$$T_{t+1|t} \sim N(\mu_{t+1|t}, \Sigma_{t+1|t}) = f(T_{t|t}, \mathbf{u}_t, \mathbf{w}_t) \tag{5}$$

where f is the motion model. The update step is given by:

$$T_{t+1|t+1} \sim N(\mu_{t+1|t+1}, \Sigma_{t+1|t+1}) = h(T_{t+1|t}, \mathbf{z}_{t+1,i}, \mathbf{v}_{t+1,i}) \tag{6}$$

$$m_{j|t+1} \sim N(\mu_{j|t+1}, \Sigma_{t+|t+1}) = h(T_{t+1|t}, \mathbf{z}_{t+1,i}, \mathbf{v}_{t+1,i}) \tag{7}$$

where h is the observation model. The above mentioned steps are feasible because of the fact that the distribution of the quantities to be estimated are both gaussian and product of two gaussian distribution is also gaussian. The EKF is an iterative algorithm that estimates the state of the system by using the measurements and the motion model. This is one the Bayes filter. As we can see, both landmarks and robot pose aer simultaneously being estimated, hence we are essentially solving the problem of Simultaneous Localization and Mapping (SLAM).

3 Technical Approach

As we know, we obtain the linear ($\mathbf{u}_t$) and angular velocity(
$mathbf\omega_t$) of the robot from the IMU. From the camera, we get the left and right image pixels of the landmarks($\mathbf{z}_{t,i} \in \mathbb{R}^4$). We are also given the stereo camera calibration matrix K_s and the transformation $T_{O,I} \in SE(3)$, from IMU to the camera optical frame. Our goal is to estimate the world-frame IMU pose $T_{W,I} \in SE(3)$ over time and world-frame coordinates $\mathbf{m}_j \in \mathbb{R}^3 \ \forall \ j \in \{1, 2, 3, \cdots, M\}$ of the point landmarks that generated the visual features, $\mathbf{z}_{t,i}$.

3.1 Localization via EKF prediction

As we know, the landmarks are fixed in the environment. So, we need motion model only for robot pose.

3.1.1 Motion Model

In $SE(3)$, we can define a Gaussian distribution over a pose matrix using a perturbation ϵ on the Lie algebra,

$$T \ = \ \mu \exp(\hat{\epsilon}) \tag{8}$$

where, $\mu \in SE(3)$ and $\epsilon \sim N(0, \Sigma)$. The discrete-time and continuous-time pose kinematics with constant $\mathbf{u}(t) = \begin{bmatrix} \mathbf{v}(t) \\ \omega(t) \end{bmatrix}$, for $\tau_k \in [t_k, \ t_{k+1}]$ and t are respectively given by:

$$T_{k+1} = T_k \exp(\tau_k \hat{\mathbf{u}}_k) \tag{9}$$

$$\dot{T} = T(\hat{\mathbf{u}}_t + \hat{\mathbf{w}}_t) \tag{10}$$

where $\tau_k = t_{k+1} - t_k$ is the time step, noise $\mathbf{w}(t)$ and $\hat{\mathbf{u}}_t = \begin{bmatrix} \hat{\omega}_t & \mathbf{v}_t \\ \mathbf{0}^T & 0 \end{bmatrix}$. To consider the Gaussian distribution over T, express it as a nominal pose $\mu \in SE(3)$ with small perturbation $\hat{\delta\mu} \in se(3)$,

$$T = \mu \exp(\hat{\mu}) \approx (\mu)(I + \hat{\mu}) \tag{11}$$

Substituting the nominal + perturbed pose into the continuous-time pose kinematics, we get:

$$\dot{\mu}(I + \hat{\delta\mu}) \ + \ \mu(\hat{\dot{\delta\mu}}) = \mu(I + \hat{\delta\mu})(\hat{\mathbf{u}} + \hat{\mathbf{w}}) \tag{12}$$

$$\dot{\mu} = \mu\hat{\mathbf{u}} \quad \hat{\dot{\delta\mu}} = (-\overset{\curlywedge}{\hat{\mathbf{u}}}\delta\mu)^{\wedge} + \hat{\mathbf{w}} \tag{13}$$

On solving the above differential equation we get our motion model. The motion model in terms of nominal kinematics of $\mu_{t|t}$ and perturbation kinematics of $\delta\mu_{t|t}$ with time discretization τ_t is given by:

$$\mu_{t+1|t} = \mu_{t|t} \exp(\tau_t \hat{\mathbf{u}}_t) \tag{14}$$

$$\delta\mu_{t+1|t} = \exp(-\tau_t \overset{\curlywedge}{\mathbf{u}}_t) \ \delta\mu_{t|t} + \mathbf{w}_t \tag{15}$$

where $\overset{\curlywedge}{\mathbf{u_t}} = \begin{bmatrix} \hat{\omega}_t & \hat{\mathbf{v}}_t \\ 0 & \hat{\omega}_t \end{bmatrix} \in \mathbb{R}^{6\times 6}$. The distribution of pose of the robot is given by $T_t|\mathbf{z}_{0:t}, \mathbf{u}_{0:t} \sim N(\mu_{t|t}, \Sigma_{t|t})$ with $\mu_{t|t} \in SE(3)$ and $\Sigma_{t|t} \in \mathbb{R}^{6\times 6}$. From above equations, we get the pose of the robot as $T_{t+1|t} = \mu_{t+1|t} \exp(\delta \hat{\mathbf{u}_{t+1|t}})$ with $\delta_{t+1|t} \sim N(0, \Sigma_{t+1|t})$.

3.1.2 EKF prediction

The EKF prediction step with noise $\mathbf{w}_t \sim N(0, \mathbf{W})$ and above mentioned motion model is given by:

$$\mu_{t+1|t} = \mu_{t|t} \exp(\tau_t \hat{\mathbf{u}}_t) \tag{16}$$

$$\Sigma_{t+1|t} = \mathbb{E}[\delta\mu_{t+1|t}\delta\mu_{t+1|t}^T] \tag{17}$$

$$\Sigma_{t+1|t} = \exp(-\tau_t \overset{\curlywedge}{\mathbf{u}}_t)\ \Sigma_{t|t} \exp(-\tau_t \overset{\curlywedge}{\mathbf{u}}_t)^T + \mathbf{W} \tag{18}$$

where all the notations are same as in (14). For initial analysis, we plot the dead reckoning trajectory without considering the uncertainity in the motion model, i.e. we only update the mean using given control input $\mathbf{u}_t$.

As we know, the covarinace matrix(Σ) is joint covariance of robot pose and landmark, the prediction of covariance is given by:

$$\Sigma_{t+1|t} = \begin{bmatrix} F\Sigma_{RR}F^T + \mathbf{W} & F\Sigma_{RL} \\ \Sigma_{LR}F^T & \Sigma_{LL} \end{bmatrix} \tag{19}$$

where $F = \exp(-\tau_t \overset{\curlywedge}{\mathbf{u}}_t)$ is the Jacobian of the motion model w.r.t. robot pose, Σ_{RR} is the covariance of robot pose, Σ_{RL} is the covariance of robot pose and landmark, Σ_{LR} is the covariance of landmark and robot pose and Σ_{LL} is the covariance of landmark.

3.2 Mapping via EKF Update Step

Here we assume that robot pose obtained in previous step is accurate and accordingly proceed to estimate the position of the landmarks using this fact. We are given the observations $\mathbf{z}_t := \begin{bmatrix} \mathbf{z}_{t,1}^T & \mathbf{z}_{t,2}^T & \cdots & \mathbf{z}_{t,n}^T \end{bmatrix}^T \in \mathbb{R}^{4n}$ of the landmarks at time t and the corresponding landmark positions $\mathbf{m}_j \in \mathbb{R}^{3n}$ in the world frame.

Since, we have access to complete dataset, we could use the data in a manner such that $\mathbf{z}_{t,i}$ is corresponding observation of $\mathbf{m}_i$. If we were to implement the algorithm in real-time, we would have to design an external algorithm for data association. Also, since the landmarks are strationary we don't need a motion model to predict the position of the landmarks.

To reduce the runtime of the algorithm we implemented two approaches. Firstly, we sampled the observations at the beginning and will only process them throughout our algorithm. The effects of this approach are mentioned in the Results of the report. Secondly, at time t we are only considering the observations which are visible at that interval.

3.2.1 Observation Model

Given the predicted pose of the robot, $T_{t+1|t}$, and the observation of the landmark i at time $t+1$, $\tilde{\mathbf{z}}_{t+1,i}$, we get the following observation model:

$$\tilde{\mathbf{z}}_{t+1,i} = h(T_{t+1|t}, \mathbf{m}_j) + \mathbf{v}_{t+1,i} := K_s \pi(T_{O,I} T_{t+1|t}^{-1} \underline{\mathbf{m}}_j) + \mathbf{v}_{t+1,i} \tag{20}$$

where $\mathbf{v}_{t+1,i} \sim N(0, \mathbf{V})$ is the measurement noise, and $\tilde{\mathbf{z}}_{t+1,i}$ is the estimated observation of the landmark i at time $t+1$ using the predicted pose. $\underline{\mathbf{m}}_j = \begin{bmatrix} \mathbf{m}_j \\ 1 \end{bmatrix}$ is the homogenous coordinate of landmark j. The projection function ($\pi(.)$) is the used to extract features in the image plane of the camera. The projection function and its derivative is given by:

$$\pi(\mathbf{q}) = \frac{1}{q_3}\mathbf{q} \in \mathbb{R}^4 \tag{21}$$

$$\frac{d\pi}{d\mathbf{q}}(\mathbf{q}) = \frac{1}{q_3} \begin{bmatrix} 1 & 0 & \frac{-q_1}{q_3} & 0 \\ 0 & 1 & \frac{-q_2}{q_3} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-q_4}{q_3} & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4}. \tag{22}$$

The observations stacked as a $4n$ vector, at time t with with notation abuse:

$$\mathbf{z}_t = K_s \pi(T_{O,I} T_{t|t}^{-1} \underline{\mathbf{m}}) + \mathbf{v}_t \tag{23}$$

$$\mathbf{v}_t \sim N(0, I \otimes \mathbf{V}) \tag{24}$$

$$I \otimes \mathbf{V} := \begin{bmatrix} V & & \\ & \ddots & \\ & & V \end{bmatrix} \tag{25}$$

where V is the covariance matrix of the measurement noise.

3.2.2 EKF update

Since, we assunmed the position of the landmarks follows Gaussian distribution, we model the landmark positions as $\mathbf{m}|\mathbf{z}_{0:t} \sim N(\mu_t, \mathbf{\Sigma}_t)$ where, $\mu_t \in \mathbb{R}^{3n}$ and $\mathbf{\Sigma}_t \in \mathbb{R}^{3n\times 3n}$. The EKF update step for landmarks is given by:

$$K_{t+1} = \Sigma_{t+1|t} H_t^T (H_t \Sigma_{t+1|t} H_t^T + I \otimes \mathbf{V})^{-1} \tag{26}$$

$$\mu_{t+1} = \mu_{t+1|t} + K_{t+1}(\mathbf{z}_{t+1} - \tilde{\mathbf{z}}_{t+1}) \tag{27}$$

$$\Sigma_{t+1} = \Sigma_{t+1|t} - K_{t+1} H_t \Sigma_{t+1|t} \tag{28}$$

where H_t is the Jacobian of the observation model with respect to the landmark positions, K_t is the Kalman gain and $\tilde{\mathbf{z}}_{t+1}$ is the estimated observation of the landmarks at time $t+1$ using the predicted pose $T_{t+1|t}$.

3.2.3 Stereo Camera Jacobian

According to the observation model, $\tilde{\mathbf{z}}_{t+1,i} := h(T_{t+1|t}, \mathbf{m}_j) = K_s\pi(T_{O,I}T^{-1}_{t+1|t}\underline{\mathbf{m}}_j)$. Now calculating the jacobian using chain rule:

$$\frac{\partial}{\partial \mathbf{m}_j}\tilde{\mathbf{z}}_{t+1,i} = \frac{\partial}{\partial \mathbf{m}_j}K_s\pi(T_{O,I}T^{-1}_{t+1|t}\underline{\mathbf{m}}_j) \tag{29}$$

$$= K_s\frac{\partial}{\partial \mathbf{q}}\pi(T_{O,I}T^{-1}_{t+1|t}\underline{\mathbf{m}}_j)T_{O,I}T^{-1}_{t+1|t}\frac{\partial \underline{\mathbf{m}}_j}{\partial \mathbf{m}_j} \tag{30}$$

$$= K_s\frac{\partial}{\partial \mathbf{q}}\pi(T_{O,I}T^{-1}_{t+1|t}\underline{\mathbf{m}}_j)T_{O,I}T^{-1}_{t+1|t}P^T \tag{31}$$

where $P = [I\ 0] \in \mathbb{R}^{3\times 4}$. So, the observation model jacobian ($H_{t+1} \in \mathbb{R}^{4n\times 3n}$) evaluated at μ_t with block elements $H_{t+1,i,j}$ is given by:

$$H_{t+1,i,j} = \begin{cases} \frac{\partial}{\partial \mathbf{m}_j}h(T_{t+1|t}, \mathbf{m}_j)|_{\mathbf{m}_j=\mu_{t,j}} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{32}$$

3.2.4 Initialization of Landmarks

The initializations of mean positions of landmarks is done by calculating the positions of the landmarks from the gievn observations. Since, we know the paramters of the camera, we can calculate the position of landmark j using the following equation:

$$\begin{bmatrix} u_L \\ v_L \\ u_R \\ v_R \end{bmatrix} = K_s\frac{1}{z}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{33}$$

$$K_s = \begin{bmatrix} fs_u & 0 & c_u & 0 \\ 0 & fs_v & c_v & 0 \\ fs_u & 0 & c_u & -fs_ub \\ 0 & fs_v & c_v & 0 \end{bmatrix} \tag{34}$$

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_{camera} = \begin{bmatrix} (\frac{fs_ub}{fs_u})(\frac{v_L-c_u}{u_L-u_R}) \\ (\frac{fs_ub}{fs_v})(\frac{v_L-c_v}{u_L-u_R}) \\ \frac{fs_ub}{u_L-u_R} \\ 1 \end{bmatrix} \tag{35}$$

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_{world} = T_tT^{-1}_{O,I}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_{camera} \tag{36}$$

$$\tag{37}$$

where u_L and v_L are the pixel coordinates of landmark in the left image and u_R and v_R are the pixel coordinates of landmark in the right image, b is the baseline of the stereo camera which is provided. T_t is the pose of robot (IMU) at time t, and $T_{O,I}$ is the transformation from the camera frame to the IMU frame. Hence, we initialize the values obtained to the mean of the landmark positions. The initializations of covariance will be discussed in later sections.

3.3 Visual-Inertial SLAM

As we have discussed the EKF prediction step and the EKF update step for landmarks in the previous sections. We will be using identical notations and equations from previous sections to describe the Visual-Inertial SLAM algorithm. We have estimated the predicted pose $T_{t+1|t}$ using the EKF prediction step. Now, we will use the observations of the landmarks to update the predicted pose and landmark positions. In the update step mentioned the jacobian and the covarinace matrices are fused for both (landmarks and robot pose), i.e. $H_{t+1} \in \mathbb{R}^{4n\times(3n+6)}$ and $\Sigma_{t+1} \in \mathbb{R}^{3n+6\times 3n+6}$. The EKF update step is described below.

3.3.1 EKF Update Step

Given $T_{t+1|t} \sim N(\mu_{t+1|t}, \Sigma_{t+1|t})$ and $\tilde{\mathbf{z}}_{t+1}$, the EKF update step is given by:

$$K_{t+1} = \Sigma_{t+1|t} H_{t+1}^T (H_{t+1} \Sigma_{t+1|t} H_{t+1}^T + I \otimes \mathbf{V})^{-1} \tag{38}$$

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

$$\Sigma_{t+1|t+1} = \Sigma_{t+1|t} - K_{t+1} H_t \Sigma_{t+1|t} \tag{40}$$

where H_{t+1} is the jacobian of $\tilde{\mathbf{z}}_{t+1}$ with respect to the landmark positions and the robot pose. K_{t+1} is the Kalman gain and $\tilde{\mathbf{z}}_{t+1}$ is the estimated observation of the landmarks at time $t+1$ using the predicted pose $T_{t+1|t}$. The update equaitons of landmark positions distributions' mean is mentioned above. But the covariance of the system will be updated altogether as described below. We have calculated the jacobian with respect to the landmarks. Now we will calculate the jacobian with respct to the robot pose.

3.3.2 Jacobian of Observation Model with Respect to Robot Pose

Let the elements of $H_{t+1} \in \mathbb{R}^{4n\times 6}$corresponding to different observations i be $H_{t+1,i} \in \mathbb{R}^{4\times 6}$. The jacobian of the observation model with respect to the robot pose is calculated by Taylor series approximation of the observation i at time $t+1$ using an IMU pose perturbation $\delta\mu$ is:

$$\mathbf{z}_{t+1,i} = K_s \pi(T_{O,I}(\mu_{t+1|t} exp(\hat{\delta\mu}))^{-1} \underline{\mathbf{m}}_j) + \mathbf{v}_{t+1,i} \tag{41}$$

$$\approx K_s \pi(T_{O,I}(I - \hat{\delta\mu})\mu_{t+1|t}^{-1} \underline{\mathbf{m}}_j) + \mathbf{v}_{t+1,i} \tag{42}$$

$$\approx K_s \pi(T_{O,I}\mu_{t+1|t}^{-1} \underline{\mathbf{m}}_j) - K_s \frac{d\pi}{d\mathbf{q}}(T_{O,I}\mu_{t+1|t}^{-1} \underline{\mathbf{m}}_j) T_{O,I} (\mu_{t+1|t}^{-1} \underline{\mathbf{m}}_j)^{\odot} \delta\mu + \mathbf{v}_{t+1,i} \tag{43}$$

$$= \tilde{\mathbf{z}}_{t+1,i} + H_{t+1,i}\delta\mu + \mathbf{v}_{t+1,i} \tag{44}$$

where for homogenous coordinates $\underline{\mathbf{s}} \in \mathbb{R}^4$ and $\hat{\mathbf{u}} \in se(3)$:

$$\hat{\mathbf{u}}\underline{\mathbf{s}} = \underline{\mathbf{s}}^{\odot}\mathbf{u} \tag{45}$$

$$\begin{bmatrix} \mathbf{s} \\ 1 \end{bmatrix}^{\odot} = \begin{bmatrix} I & -\hat{\mathbf{s}} \\ 0 & 0 \end{bmatrix} \tag{46}$$

$$\tag{47}$$

3.3.3 Implementation Details

As shown above, the update eqautions for landmarks and poses would be different we will split the Kalman gain to two parts. First 6 rows (in our implementation) will be used to update the pose using above equations. The other rows would be used to update the landmarks as described in the Mapping update step.

For covariance, as there would be correlations between robot pose and landmark positions and also between different landmark positions, we would update it cumulatively for both. As we mentioned in above sections that we will only be using landmarks observed in current timestep for update. We will be sampling (for both mean of landmarks and the covariances) the required indices accordingly (keeping in mind the correlations) and update the covariance matrix.

Initialization of covariance can be done according to the dataset. For example, here we are driving a real-sized vehicle, so we choose the initial covariance matrix as $5I \in \mathbb{R}^{6+4n \times 6+4n}$ where n is the number of landmarks.

4 Results

- The filter is quite robust to different values of covariance initializations. We have tested it with different values of covariance matrix and the results are quite similar upto $100I$.

- The filter was also tested for different noise levels in both IMU and camera. The results were similar when the noise covariance ranged between I and $50I$. We also observed that it was mandatory filter to be given certain level to avoid running of the filter into singularity.

- We also observed that on sampling different number of landmarks, the performance of the filter differed. As we sampled more landmarks the performance of the filter was improved.

- The algorithm was visibly reducing the innovation term before and after the update step, which was as expected. The covariance of system was also reducing after the update step.

- We also tried to test the algorithm with different initializations of landmarks at the beginning. The results did not change much when we initialized the landmarks a little far from obstained graoundtruth location from camera (about the same distance till when covariance initialization was reasonable). But certainly it wasn't able to converge when we initialized all the landmarks to initial robot pose.

- Above mentioned results are valid for both trajectories.

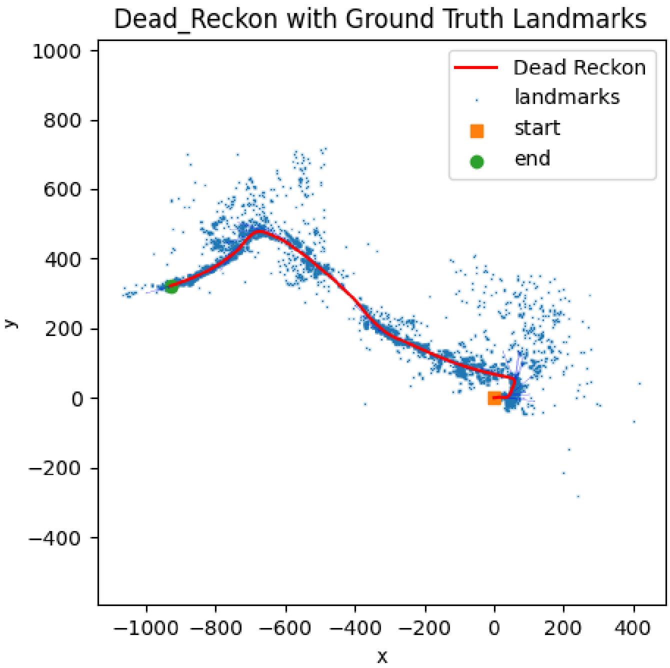

Figure 1: Deadreckoning trajectory 03

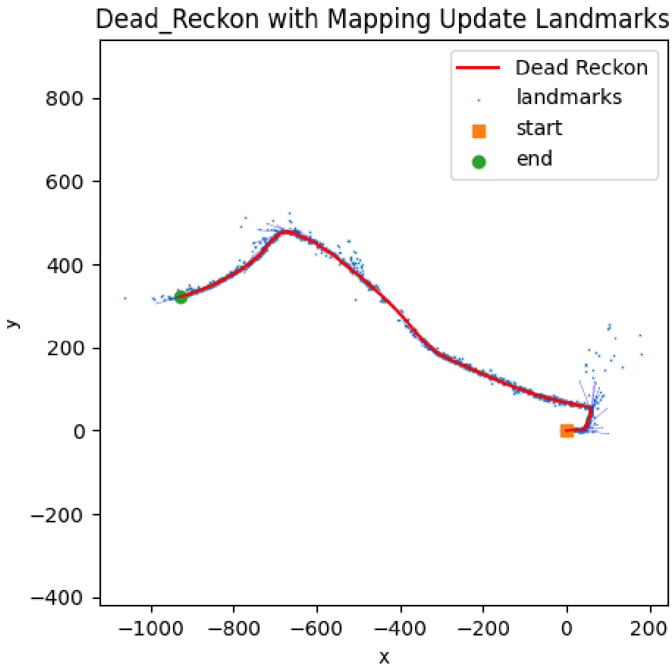

Figure 2: Deadreckoning trajectory with mapping update 03

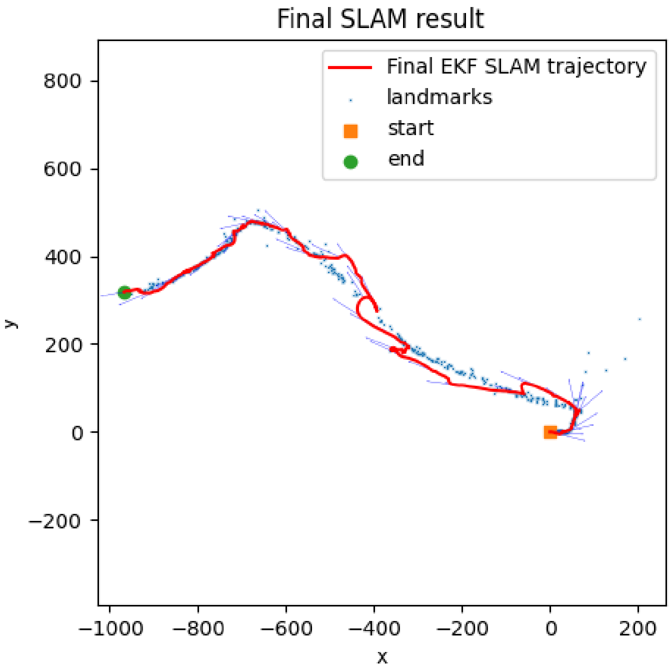

Figure 3: EKF 03

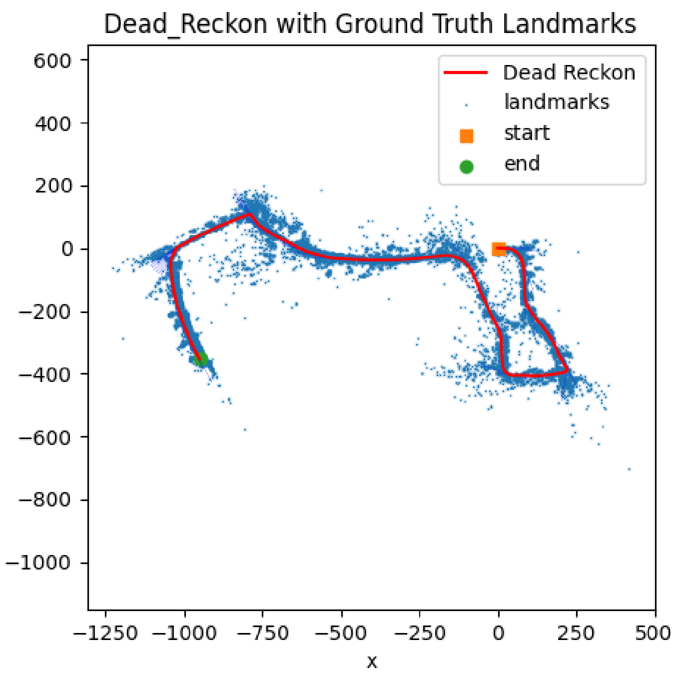

Figure 4: Deadreckoning trajectory 10

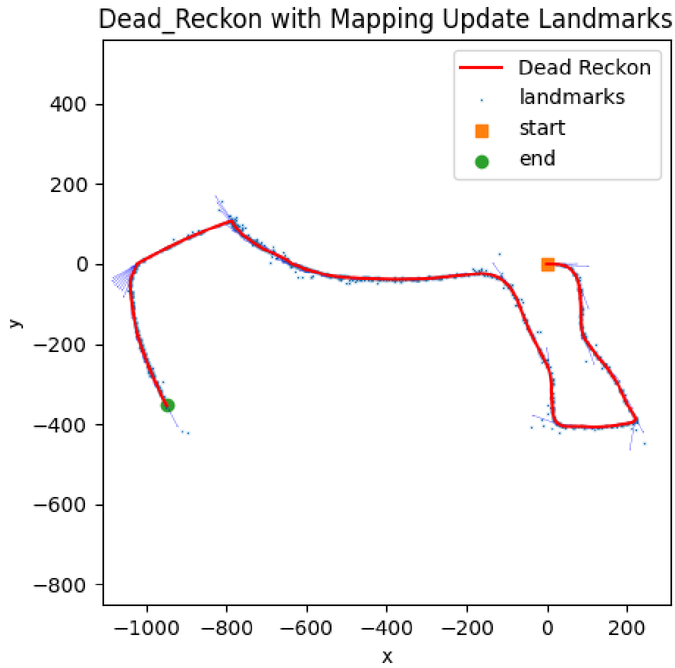

Figure 5: Deadreckoning trajectory with mapping update 10

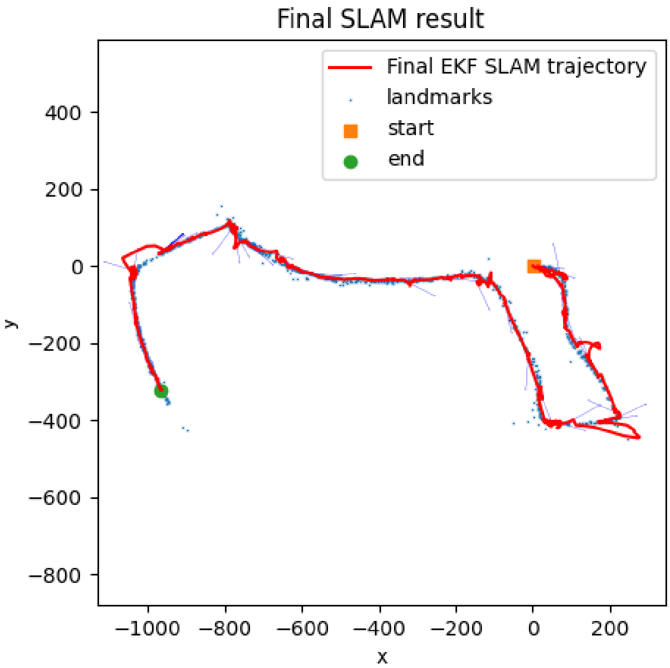

Figure 6: EKF 10