ECE276A Project-3
Visual Inertial SLAM using Extended Kalman Filter

Saqib Azim
UC San Diego
sazim@ucsd.edu

Abstract

In this project, we implemented a visual-inertial SLAM system to localize and map the environment simultaneously using an extended kalman filter. The system uses linear and angular acceleration data from an IMU to localize a car moving on a road and stereo RGB cameras fixed to the robot front to map the environment. The project is divided into three parts - in the first part, we localize the robot using linear and angular velocity data from IMU and use Extended kalman filter (EKF) algorithm to estimate the SE(3) pose of the car at each time step. Then, in the second part, we use the estimated poses from the first step and the provided stereo observations of landmark positions $m \in R^{3\times M}$ to estimate the landmark position in world frame using EKF update step on the set of landmarks. In the third step, we combine the localization and mapping together and add the EKF update step for correcting the predicted poses in order to build a VISLAM system.

Index Terms: *SLAM, Kalman filter (KF), Extended kalman filter (EKF), robot trajectory, mapping, environment.*

1. Introduction

SLAM is a widely used technique in robotics and it has seen significant improvements over the past decades. It enables robots to navigate and map their surroundings using visual and inertial sensors. This technique combines the information from a camera and an IMU (Inertial Measurement Unit) to estimate the robot's motion and the 3D structure of the environment. The importance of visual-inertial SLAM (VISLAM) lies in its ability to provide robots with accurate and real-time information about their environment, which is essential for various robotic applications such as autonomous navigation, indoor or outdoor localization, augmented reality, and 3D mapping. Visual-inertial SLAM is an essential technology for robotics, enabling robots to navigate, interact with their environment.

1.1. Brief Overview of our approach

In our project, the data is taken from the KITTI dataset where a car is moving on a road and collecting IMU data and visual observations using a stereo camera. For this project, we are provided with the linear and angular velocity of the car at different time instants and stereo observations of the visible landmarks. These measurements have noise and hence without any estimation technique, the dead-reckoning trajectory would have noise, accumulated drift and inaccurate. Therefore, EKF SLAM alleviates these issues as it uses an approximated Bayes theorem to combine observations from multiple sensors.

Our approach to VISLAM can be broadly divided into these steps:

- First, we estimate the dead-reckon SE(3) poses of the IMU using the linear and angular velocity measurements obtained from IMU and perform EKF prediction based on SE(3) kinematics equations to estimate pose $T_{t+1|t} \in SE(3)$ over time t.
- Next, we process the features data to reduce the number of landmarks and also remove potential outliers.
- Then, we estimate the landmarks using the predicted IMU pose trajectory via the EKF update step.
- Finally, we combine the IMU prediction and landmark update step and add an additional IMU update step to simultaneously perform localization and mapping.

2. Problem Formulation

In the first part of this project, we consider the localization-only problem assuming the world-frame landmark coordinates $m = [m_1^T, m_2^T, \dots, m_M^T]^T \in R^{3M}$ are known (M is the total landmarks count observed for the entire car trajectory). Given the linear and angular velocity measurements from IMU $u_{0:T}$ where $u_t = [v_t^T, \omega_t^T]^T \in R^6$

which is attached to the car, we need to estimate the IMU poses $T_t =_W T_{I,t} \in SE(3)$. Assuming a gaussian prior IMU poses $T_t|z_{0:t}, u_{0:t-1} \sim N(\mu_{t|t}, \Sigma_{t|t})$ are known at any time t where the mean $\mu_{t|t} \in SE(3)$ and covariance $\Sigma_{t|t} \in R^{6\times 6}$, we are required to predict the IMU poses $T_{t+1}|z_{0:t}, u_{0:t}$ at time $t+1$ using EKF prediction step based on the $SE(3)$ kinematics equation described in the next section.

In the second part, we consider the mapping-only problem and assume the predicted IMU pose trajectory computed in the first part is correct. Given the observations $z_t = [z^T_{t,1}, \dots, z^T_{t,N_t}]^T \in R^{4N_t}$ stacked together at time t, we need to estimate the landmark coordinates $m = [m^T_1, \dots, m^T_M]^T \in R^{3M}$ assuming these are static. The data association $\Delta_t : \{1, 2, \dots, M\} \to \{1, \dots, N_t\}$ specifying that landmark $m_j \in R^3$ corresponds to observation $z_{t,i} \in R^4$ with $i = \Delta_t(j)$ at time t is precomputed and provided to us. Here M denotes the total landmarks count and N_t denotes number of observed landmarks at time t. In this sub-problem, we need to estimate the unknown landmark coordinates $m \in R^{3M}$ using an EKF and update the gaussian mean and covariance of the visible landmarks at each time step t.

In the final part, we combine the previous two subparts to build a VISLAM algorithm that estimates the IMU pose T_t at each time step t and simultaneously estimates the landmark coordinates in world frame $m \in R^{3M}$ given the linear and angular velocity measurements from IMU $u_{0:T}$ and matching feature observations $z_{0:T}$ from the stereo camera. Given the prior IMU pose $T_t|z_{0:t}, u_{0:t-1} \sim N(\mu_{t|t}, \Sigma_{t|t})$, we need to predict the pose at the next time step $T_{t+1}|z_{0:t}, u_{0:t} \sim N(\mu_{t+1|t}, \Sigma_{t+1|t})$ using SE(3) kinematic equations. We approximate the predicted PDF to be gaussian even though it is not exactly gaussian. After that, using the current landmark estimates $m_t|z_{0:t} \sim N(\mu^m_t, \Sigma^m_t)$ (where superscript m denotes map/landmarks) and using the predicted pose $T_{t+1|t}$, we need to use EKF to update the mean and covariance of the IMU pose $(\mu_{t+1|t+1}, \Sigma_{t+1|t+1})$ and mean and covariance of visible landmarks $(\mu^m_{t+1}, \Sigma^m_{t+1})$ in a combined fashion. Here also we approximate the updated PDF to be gaussian distribution.

3. Technical Approach

In this section, we describe in detail our approach to solving the EKF-VISLAM problem.

3.1. Data Processing

In this project, we are provided with a features matrix containing observations for each landmark at each time instant t. To reduce the computations, we are not using all features as suggested in the problem statement. We uniformly pick every k (6 or 10) landmark out of the total number of landmarks provided. Thus, our total landmarks count is reduced by a factor of k. In addition, for the remaining landmarks, we iterate through the dead-reckoning IMU trajectory and remove any landmarks whose distance from the IMU at that time step is greater than some distance threshold (hyperparameter). This way we ensure to remove any outliers that may degrade the filtering quality.

3.2. IMU Pose Prediction

Given the prior IMU poses $T_t|z_{0:t}, u_{0:t-1} \sim N(\mu_{t|t}, \Sigma_{t|t})$ which is assumed to be gaussian distributed with mean $\mu_{t|t} \in SE(3)$ and covariance $\Sigma_{t|t} \in R^{6\times 6}$, we use the SE(3) kinematic equations to predict the IMU pose for next time step $T_{t+1}|z_{0:t}, u_{0:t} \sim N(\mu_{t+1|t}, \Sigma_{t+1|t})$. Given the linear and angular velocity measurements $u_t = [v^T_t, \omega^T_t]^T \in R^6$ at time t, the SE(3) kinematic motion model for continuous time IMU pose $T(t)$ with noise $w(t)$ is given by -

$$\dot{T}(t) = T(t)(\hat{u} + \hat{w}) \tag{1}$$

In discrete time with time discretization τ_t, the pose kinematics can be split into nominal and perturbation kinematics:

$$\mu_{t+1|t} = \mu_{t|t} \exp(\tau_t \hat{u}_t) \tag{2}$$

$$\delta\mu_{t+1|t} = \exp(-\tau_t \tilde{u}_t)\delta\mu_{t|t} + w_t \tag{3}$$

Here $\hat{u}_t = \begin{bmatrix} \hat{\omega}_t & v_t \\ 0^T & 0 \end{bmatrix} \in R^{4\times 4}$ and $\tilde{u}_t = \begin{bmatrix} \hat{\omega}_t & \hat{v}_t \\ 0 & \hat{\omega}_t \end{bmatrix} \in R^{6\times 6}$ $\tau_t = t_{k+1} - t_k$. We assume motion noise to be gaussian distributed $w_t \sim N(0, W)$. The covariance prediction is given by -

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

In this way, we predict the IMU poses for the next time step $(t+1)$ given prior IMU pose using kinematic equations.

3.3. Pose and Landmark Update

For the visual mapping problem, we assume the predicted IMU pose $T_{t+1|t} \in SE(3)$ to be correct pose. Given the gaussian prior landmark estimates $m|z_{0:t} \sim N(\mu^m_t, \Sigma^m_t)$ where $\mu_t \in R^{3M}$ (linearized) are the landmark mean and $\Sigma_t \in R^{3M\times 3M}$ is the covariance matrix at time t. The observation model of the stereo camera is given by -

$$\begin{aligned} \hat{z}_{t+1,i} &= h(T_{t+1}, \mu^m_{t+1|t,j}) + v_{t+1,i} \\ &= K_s \pi({}_oT_I T^{-1}_{t+1} \mu^m_{t+1|t,j}) + v_{t+1,i} \end{aligned}$$

where $v_{t,i} \sim N(0, V)$, K_s is the stereo calibration matrix. We are provided the calibration matrix K of each individual perspective camera and the baseline distance between them using which the stereo calibration matrix K_s can be computed.

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

The above observation model simply transforms each landmark's world frame coordinates m_j observed at time $(t+1)$ to pixel coordinates $z_{t+1,i} \in R^4$. We use the predicted IMU pose $T_{t+1|t}$ to convert from world to IMU frame and then from IMU to left stereo optical frame and apply perspective projection $\pi(q) = \frac{q}{q_3}$ to convert to normalized image coordinates and finally multiply by calibration matrix K_s to get left and right camera pixel coordinates. These represent predicted observations obtained by combining information from the landmarks and the predicted IMU pose.

The EKF approximates the non-linear motion and observation models using Taylor series expansion instead of using the exact form of linear models used in KF. To approximate, we need jacobian $H^m_{t+1} \in R^{4N_t \times 3M}$ of the observation model evaluated at the current mean of the landmarks μ^m_t where each block element $H^m_{t+1,i,j} \in R^{4\times 3}$ is given by -

$$\begin{aligned} H^m_{t+1,i,j} &= \frac{d}{dm_j} h(T_{t+1|t}, m_j)\Big|_{m_j=\mu_{t,j}} \quad \text{if} \quad \Delta_t(j) = i \\ &= K_s \frac{d\pi}{dq}({}_oT_I T^{-1}_{t+1|t}\mu^m_{t,j})\, {}_oT_I T^{-1}_{t+1|t} P^T \end{aligned}$$

Using the jacobian computed above, we estimate the Kalman gain using the below expression -

$$K^m_{t+1} = \Sigma^m_t (H^m_{t+1})^T (H^m_{t+1}\Sigma^m_t (H^m_{t+1})^T + I \times V)^{-1}$$

and finally update the mean and covariance matrix of the M landmarks as follows -

$$\mu^m_{t+1} = \mu^m_t + K^m_{t+1}(z_{t+1} - \hat{z}_{t+1})$$

Here $r_{t+1} = z_{t+1} - \hat{z}_{t+1}$ represents innovation and measures the error between the actual observation and predicted observation and corrects the mean in the update step.

$$\Sigma^m_{t+1} = \Sigma^m_t - K^m_{t+1} H^m_{t+1} \Sigma^m_t$$

3.4. EKF Combined Pose and Landmark Update

For combined update of the IMU pose and the landmarks, we create a combined covariance matrix for the IMU pose and the landmarks and we maintain and update this matrix so as to account for any cross-correlations between the IMU pose and the landmarks.

$$\Sigma_t = \begin{bmatrix} \Sigma^m_t & \Sigma^{m,imu}_t \\ \Sigma^{imu,m}_t & \Sigma^{imu}_t \end{bmatrix} \tag{4}$$

We perform the predict step which gives the predicted IMU pose $T_{t+1|t} \sim N(\mu_{t+1|t}, \Sigma_{t+1|t})$. One difference here is while predicting the IMU pose covariance matrix, we need to update the cross-correlations between the IMU poses and landmarks. If we denote $F = \exp(-\tau_t \tilde{u}_t) \in R^{6\times 6}$, then the predicted covariance matrix is given by -

$$\Sigma_{t+1|t} = \begin{bmatrix} \Sigma^m_{t|t} & \Sigma^{m,imu}_{t|t} F^T \\ F\Sigma^{imu,m}_{t|t} & F\Sigma^{imu}_{t|t} F^T + W \end{bmatrix} \in \mathbb{R}^{(3M+6)\times(3M+6)}$$

Next we compute the jacobian of the observation model with respect to the landmarks as done in above subsection. Next, we estimate the jacobian of the observation model with respect to the IMU predicted pose $T_{t+1|t}$ is given by -

$$\begin{aligned} H^{imu}_{t+1,i,j} &= \frac{d}{dT_{t+1|t}} h(T_{t+1|t}, m_j)\Big|_{T_{t+1|t}=\mu_{t+1|t}} \quad \text{if} \quad \Delta_t(j) = i \\ &= -K_s \frac{d\pi}{dq}({}_oT_I \mu^{-1}_{t+1|t} m_j)\, {}_oT_I \left(\mu^{-1}_{t+1|t} m_j\right)^{\odot} \end{aligned}$$

Here $\begin{bmatrix} s & 1 \end{bmatrix}^{\odot} = \begin{bmatrix} I & -\hat{s} \\ 0 & 0 \end{bmatrix} \in R^{4\times 6}$ Then, we combine the jacobians computed wrt the landmarks and wrt the IMU pose to get a combined jacobian matrix -

$$H_{t+1} = [H^m_{t+1} H^{imu}_{t+1}] \in R^{4N_t \times (3M+6)} \tag{5}$$

where $H^m_{t+1} \in R^{4N_t \times 3M}$ and $H^{imu}_{t+1} \in R^{4N_t \times 6}$. Using this we compute the Kalman gain matrix -

$$K_{t+1} = \Sigma_{t+1|t} H^T_{t+1} \Big(H_{t+1} \Sigma_{t+1|t} H^T_{t+1} + I \otimes V \Big)^{-1} \tag{6}$$

In our implementation, we have used numpy library 'solve' method instead of trying to invert the matrix because of its size. The kalman gain K_{t+1} scales the observation error by its covariance and determines how much to trust this correction to the mean in the mean update step. After this, we separate the Kalman gain such that the first $3M$ rows represents the Kalman gain for the landmarks whereas the last 6 rows represents the kalman gain for the robot pose.

$$K_{t+1} = \begin{bmatrix} K^m_{t+1} \in R^{3M \times 4N_t} \\ \\ K^{imu}_{t+1} \in R^{6 \times 4N_t} \end{bmatrix}_{(3M+6)\times 4N_t} \tag{7}$$

We update the means of the IMU and landmarks separately, and use the kalman gain component corresponding to the imu and landmarks respectively for updating each mean.

$$\mu^{imu}_{t+1|t+1} = \mu^{imu}_{t+1|t} \exp\left[\left(K^{imu}_{t+1}(z_{t+1} - \hat{z}_{t+1}) \right)^{\wedge} \right]$$

$$\mu^m_{t+1} = \mu^m_t + K^m_{t+1}(z_{t+1} - \hat{z}_{t+1})$$

We update the covariance matrix together since we want to update not only the covariance matrix of the landmarks and IMU poses but also the cross-covariances between the landmarks and IMU poses introduced during the process. To update the full covariance matrix -

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

4. Conclusion

- In most experiments, I took every 10 landmark features to reduce the computation time.
- To remove outlier landmarks, I used a distance threshold of 200 in most experiments.
- We observed that the algorithm is very sensitive to noise values and therefore required significant tweaking for these set of hyperparameters: $\Sigma_{0|0}$ (IMU pose initial covariance), Σ_{LM} (landmark initial covariance matrix), W (motion model noise covariance) and V (observation model noise covariance).
- We observed that some innovations values are high and due to which there is big deviation in the IMU pose. We have corrected this by clipping the innovation values which have very high magnitude.
- To increase the speed of computation, we are computing the full kalman gain of size $3M + 6 \times 4N_t$ but after that for mean and covariance update, we extract only those rows from the initial $3M$ rows for which there is a valid observation at that time step.
- Time taken to run VISLAM on dataset-3 is roughly 7-8 minutes whereas on dataset 10, it is roughly 12-13 mins.
- I have implemented another version where we compute kalman gain of size $(3N_t + 6) \times 4N_t$ instead of $(3M+6) \times 4N_t$. This is considering the fact that we do not use all the rows of kalman gain when updating the mean or covariance. Hence, it might be better to compute only for those rows for which there is valid observation. This version is a bit faster and takes roughly half the time as compared to the full version.
- Feature skipping helps to save a lot of time. I wonder how can we make it faster so that these methods can be used for larger datasets with much more points than our dataset.

5. Acknowledgments

Thanks to the instructor, and the TAs for such a great course and projects. For this project, I had minor discussions with Chinmay, and Narayan and of course, got help from Piazza. Thanks to everyone mentioned above.

6. Results - Dataset 3

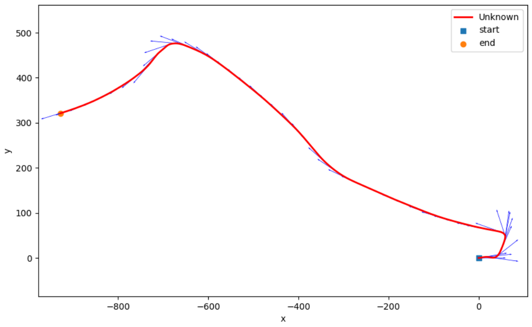

Figure 1. Figure showing dead reckoning IMU localization via EKF Prediction based on SE(3) kinematics equations

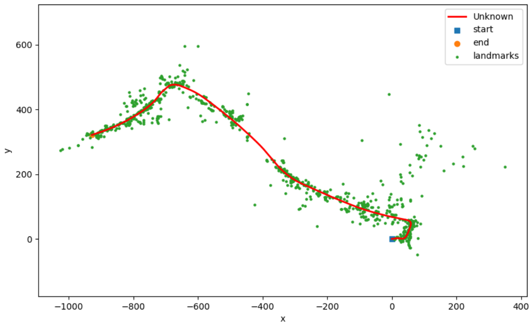

Figure 2. Landmark mapping with dead reckoning IMU trajectory $\Sigma_w = 0.1I_6, \Sigma_v = 10.0I_4, \Sigma_{0|0} = 0.0001I_6, \Sigma_{LM} = 1.0I_{3M}$

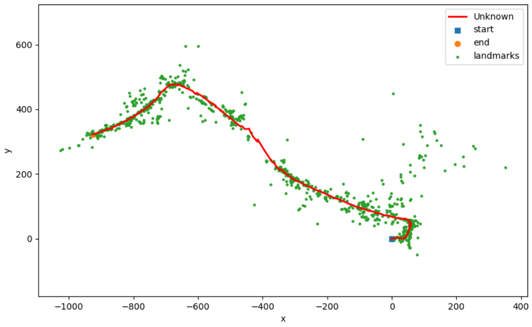

Figure 3. VISLAM results with $\Sigma_w = 0.1I_6, \Sigma_v = 10.0I_4, \Sigma_{0|0} = 0.0001I_6, \Sigma_{LM} = 1.0I_{3M}$

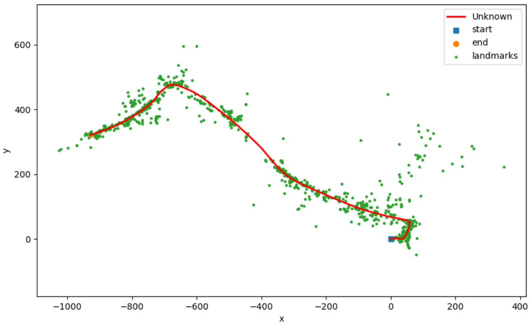

Figure 4. Landmark mapping with dead reckoning IMU trajectory $\Sigma_w = 0.1I_6, \Sigma_v = 10.0I_4, \Sigma_{0|0} = 0.001I_6, \Sigma_{LM} = 1.0I_{3M}$

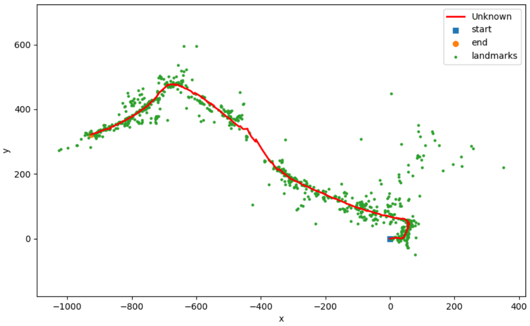

Figure 5. VISLAM results with $\Sigma_w = 0.1I_6, \Sigma_v = 10.0I_4, \Sigma_{0|0} = 0.001I_6, \Sigma_{LM} = 1.0I_{3M}$

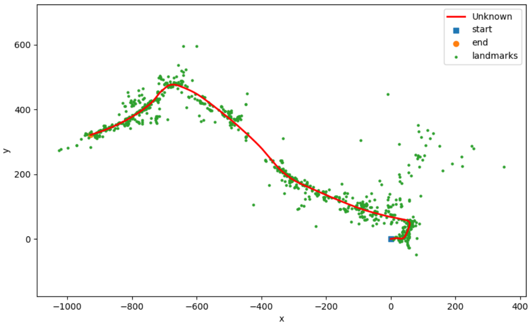

Figure 6. Landmark mapping with dead reckoning IMU trajectory $\Sigma_w = 0.1I_6, \Sigma_v = 10.0I_4, \Sigma_{0|0} = 0.001I_6, \Sigma_{LM} = 1.0I_{3M}$

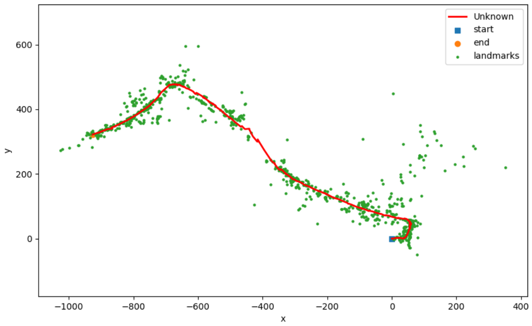

Figure 7. VISLAM results with $\Sigma_w = 0.1I_6, \Sigma_v = 10.0I_4, \Sigma_{0|0} = 0.001I_6, \Sigma_{LM} = 1.0I_{3M}$

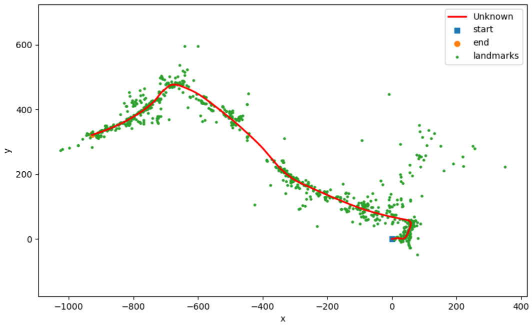

Figure 8. Landmark mapping with dead reckoning IMU trajectory $\Sigma_w = 0.1I_6, \Sigma_v = 10.0I_4, \Sigma_{0|0} = 0.01I_6, \Sigma_{LM} = 1.0I_{3M}$

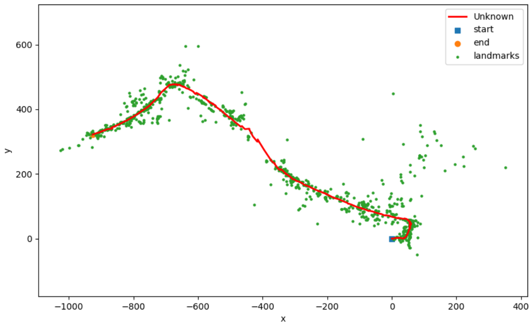

Figure 9. VISLAM results with $\Sigma_w = 0.1I_6, \Sigma_v = 10.0I_4, \Sigma_{0|0} = 0.01I_6, \Sigma_{LM} = 1.0I_{3M}$

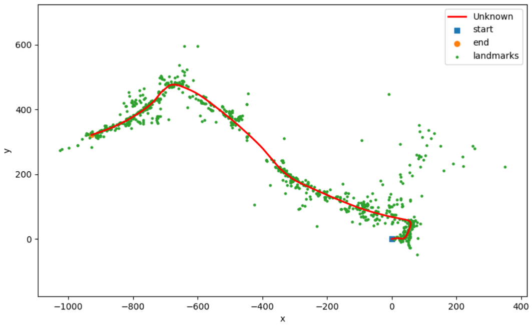

Figure 10. Landmark mapping with dead reckoning IMU trajectory $\Sigma_w = 0.1I_6, \Sigma_v = 10.0I_4, \Sigma_{0|0} = 0.1I_6, \Sigma_{LM} = 1.0I_{3M}$

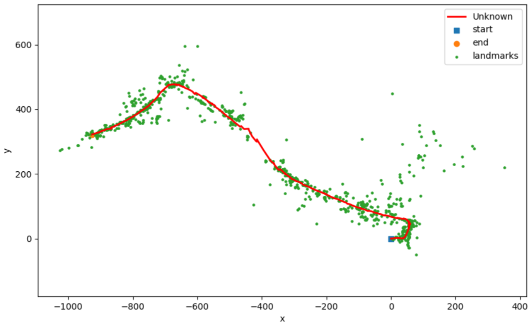

Figure 11. VISLAM results with $\Sigma_w = 0.1I_6, \Sigma_v = 10.0I_4, \Sigma_{0|0} = 0.1I_6, \Sigma_{LM} = 1.0I_{3M}$

7. Results - Dataset 10

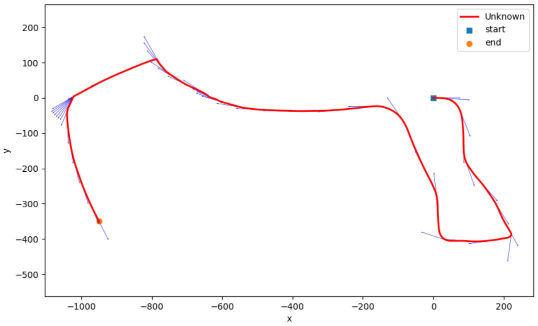

Figure 12. Figure showing dead reckoning IMU localization via EKF Prediction based on SE(3) kinematics equations

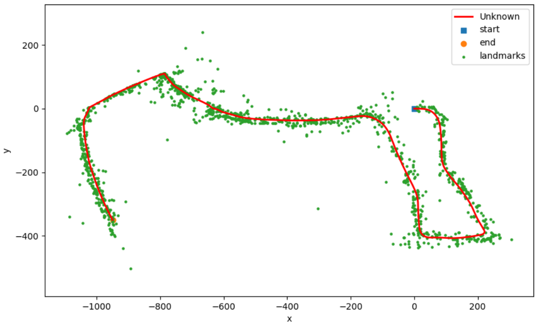

Figure 13. Landmark mapping with dead reckoning IMU trajectory $\Sigma_w = 0.1I_6, \Sigma_v = 10.0I_4, \Sigma_{0|0} = 0.0001I_6, \Sigma_{LM} = 1.0I_{3M}$

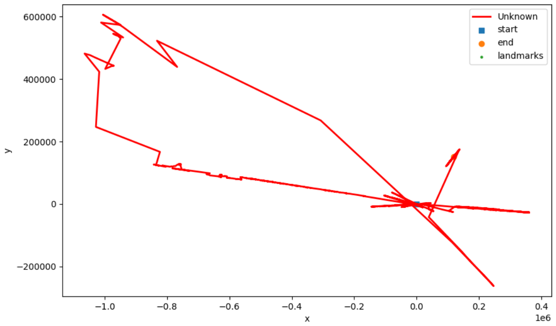

Figure 14. VISLAM results with $\Sigma_w = 0.1I_6, \Sigma_v = 10.0I_4, \Sigma_{0|0} = 0.0001I_6, \Sigma_{LM} = 1.0I_{3M}$

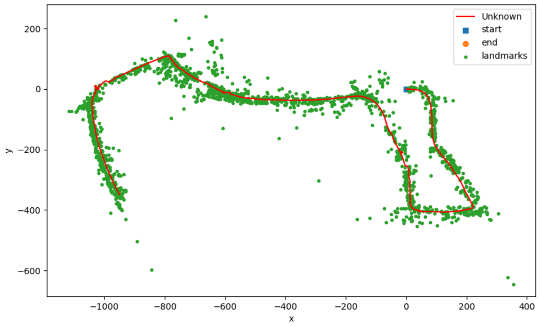

Figure 15. VISLAM results with $\Sigma_w = 0.01I_6, \Sigma_v = 10.0I_4, \Sigma_{0|0} = 0.0001I_6, \Sigma_{LM} = 1.0I_{3M}$

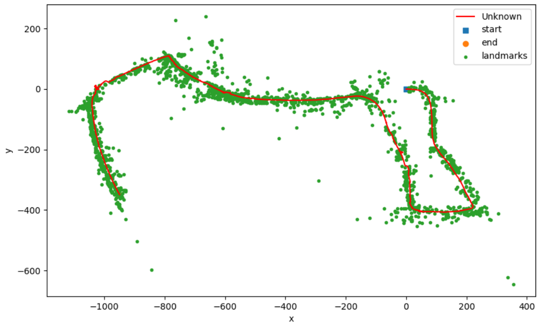

Figure 16. VISLAM results with $\Sigma_w = 0.0001I_6, \Sigma_v = 10.0I_4, \Sigma_{0|0} = 0.0001I_6, \Sigma_{LM} = 1.0I_{3M}$