Report for Project three ECE276A

Amirhosein Javadi

March 23, 2024

1 Introduction

The objective of this project is to implement a visual-inertial simultaneous localization and mapping (SLAM) algorithm using an extended Kalman filter (EKF). This approach aims to enable accurate and robust estimation of the trajectory of a vehicle equipped with synchronized sensors, specifically an inertial measurement unit (IMU) and a stereo camera system. By fusing data from these sensors, the algorithm can simultaneously localize the vehicle in its environment while creating a map of the surrounding landmarks. The IMU provides measurements of linear velocity and angular velocity of the vehicle, while the stereo camera captures visual feature measurements in the form of pixel coordinates of detected landmarks. The problem entails implementing a visual-inertial SLAM algorithm using an EKF, which involves predicting the vehicle's pose based on IMU measurements and updating the landmark positions based on visual feature observations. The primary challenge lies in effectively fusing data from multiple sensors while handling uncertainties and computational complexities inherent in real-world scenarios. Visual-inertial SLAM has significant relevance and applications in various fields such as robotics, autonomous navigation, augmented reality, and virtual reality.

2 Problem Formulation

The problem of visual-inertial SLAM can be formally stated as follows: Given synchronized measurements from an IMU and a stereo camera system, along with intrinsic and extrinsic calibration parameters, the objective is to estimate the trajectory of the vehicle T_t and the positions of landmarks m_i in the environment. This is achieved by iteratively predicting the vehicle's pose using IMU measurements, updating landmark positions using visual feature measurements, and jointly optimizing the vehicle trajectory and landmark positions to minimize the error between predicted and observed measurements. The problem can be divided into three interconnected tasks: localization, mapping, and simultaneous localization and mapping (SLAM). Let's define each problem and its associated mathematical framework:

1. **Localization**: Localization aims to estimate the pose of the vehicle (its position and orientation) at each time step based on sensor measurements. In the case of visual-inertial SLAM, the pose is represented by a transformation matrix T_t belonging to the Special Euclidean group $SE(3)$. At each time step t, the vehicle's pose T_t is estimated using IMU measurements of linear velocity v_t and angular velocity ω_t in the body frame:

$$T_{k+1} = T_k \exp(\tau_k \hat{\zeta}_k) \quad \hat{\zeta} = \begin{bmatrix} w \\ v \end{bmatrix}^{\wedge} = \begin{bmatrix} \hat{w} & v \\ 0 & 0 \end{bmatrix}$$

2. **Mapping**: Mapping involves estimating the positions of landmarks in the environment based on visual feature measurements obtained from the stereo camera system. Each landmark position

is represented as a vector m_i in 3D space. For each landmark i, the mapping problem aims to estimate its position m_i using visual feature measurements z_t from the stereo camera:

$$m_i = \text{triangulate}(z_t, T_t, K)$$

where triangulate$(\cdot)$ is the triangulation function, described in Technical Approach section completely, T_t is the estimated pose of the vehicle at time t, and K is the camera calibration matrix.

3. **Simultaneous Localization and Mapping (SLAM)**: SLAM integrates localization and mapping to jointly estimate the vehicle's trajectory and the positions of landmarks in the environment. This is achieved by simultaneously updating the vehicle's pose and landmark positions using sensor measurements. The SLAM problem involves optimizing the vehicle's trajectory T_t and landmark positions m_i by minimizing the error between predicted and observed measurements from both the IMU and stereo camera system:

$$\min_{T_t, m_i} \sum_t \|z_t - h(T_t, m_i)\|^2 + \|f(T_{t+1}, T_t, u_t)\|^2$$

where $h(\cdot)$ is the observation model that predicts the expected visual feature measurements based on the current vehicle pose and landmark positions, and $f(\cdot)$ is the loss associated to motion mode.

3 Technical Approach

3.1 IMU localization via EKF prediction

The continuous-time pose of a rigid body $T(t)$ belonging to the Special Euclidean group $SE(3)$ evolves over time according to the generalized velocity $\zeta(t)$. The generalized velocity $\zeta(t) = [\omega(t), v(t)]^T$ consists of angular velocity $\omega(t)$ and linear velocity $v(t)$ in the body-frame coordinates. The derivative of the pose with respect to time $\dot{T}(t)$ is given by:

$$\dot{T}(t) = T(t)\hat{\zeta}(t)$$

where $\hat{\zeta}(t)$ is the skew-symmetric matrix (also known as the hat operator) associated with the twist $\zeta(t)$. The twist $\zeta(t)$ is a combination of angular and linear velocities, represented in a 6-dimensional vector $\zeta(t) = [\omega(t), v(t)]^T$. In discrete-time, if the generalized velocity $\zeta(t)$ is constant over the time interval $[t_k, t_{k+1})$, and denoted as ζ_k where t_k is the start time of the interval, T_k represents the pose of the rigid body at time t_k, and $\tau_k = t_{k+1} - t_k$ is the time duration of the interval, then the pose at time t_{k+1} is obtained by applying the matrix exponential of the skew-symmetric matrix $\tau_k\hat{\zeta}_k$ to the pose at time t_k:

$$T_{k+1} = T_k \exp(\tau_k \hat{\zeta}_k)$$

Here, $\exp(X)$ denotes the matrix exponential function, which is computed as a series expansion:

$$\exp(X) = \sum_{n=0}^{\infty} \frac{1}{n!} X^n$$

By utilizing the provided equations, you can create an initial trajectory for the car.

3.2 Landmark mapping via EKF update

3.2.1 Bearing Measurement Triangulation for Point Localization

The goal of this section is to explain the process of bearing measurement triangulation for determining the coordinates of a point observed by two cameras in the reference frame of the first camera. This method is crucial for accurately localizing points in 3D space based on their projections onto the image planes of multiple cameras.
We are provided with pixel coordinates z_1 and z_2 obtained from two calibrated cameras, along with the relative transformation p and R of camera 2 in the frame of camera 1. The observed point m in 3D space is related to its projections on the image planes of both cameras through bearing measurement equations.

- $\lambda_1 z_1 = m$, where $\lambda_1 = e_3^\top m$ represents the unknown depth of point m from camera 1.
- $\lambda_2 z_2 = R^\top(m - p)$, where $\lambda_2 = e_3^\top R^\top(m - p)$ is the unknown depth of point m from camera 2. Here $R = I$.
- The expression for λ_1 is derived as:
$$\lambda_1 = \frac{a^\top b}{a^\top a}$$
where $a = R^\top z_1 - e_3^\top R^\top z_1 z_2$ and $b = R^\top p - e_3^\top R^\top p z_2$
- The coordinates of point m in the reference frame of camera 1 are obtained as:
$$m = \lambda_1 z_1$$

The pixel coordinates $z_t \in \mathbb{R}^{4\times M}$ represent detected visual features from M point landmarks with precomputed correspondences between the left and right camera frames. Landmarks i that were not observable at time t have a measurement of $z_{t,i} = [-1, -1, -1, -1]^\top$, indicating a missing observation. To establish a prior for the Extended Kalman Filter (EKF) update equations, I identified the first timestamps at which each landmark is observed. Their coordinates are then obtained and set as the prior values for the EKF algorithm.

3.2.2 Extended Kalman Filter (EKF)

The Extended Kalman Filter (EKF) utilizes a first-order Taylor series approximation to linearize the nonlinear motion and observation models around the state and noise means. This approximation allows for the prediction of the mean and covariance of the state at the next time step, given the current state estimate and control input.

- **Motion Model Approximation**: The motion model $f(x_t, u_t, w_t)$ is approximated as:
$$f(x_t, u_t, w_t) \approx f(\mu_{t|t}, u_t, 0) + F_t(x_t - \mu_{t|t}) + Q_t w_t$$
Where: $f(\mu_{t|t}, u_t, 0)$ is the motion model evaluated at the current state mean $\mu_{t|t}$ and zero noise. $F_t := \frac{\partial}{\partial x} f(\mu_{t|t}, u_t, 0)$ is the Jacobian matrix of the motion model with respect to the state evaluated at the current mean. $Q_t := \frac{\partial}{\partial w} f(\mu_{t|t}, u_t, 0)$ is the Jacobian matrix of the motion model with respect to the noise evaluated at the current mean.
- **Observation Model Approximation**: Similarly, the observation model $h(x_{t+1}, v_{t+1})$ is approximated as:
$$h(x_{t+1}, v_{t+1}) \approx h(\mu_{t+1|t}, 0) + H_t(x_{t+1} - \mu_{t+1|t}) + R_t v_{t+1}$$

Where: $h(\mu_{t+1|t}, 0)$ is the observation model evaluated at the predicted state mean $\mu_{t+1|t}$ and zero noise. $H_t := \frac{\partial}{\partial x} h(\mu_{t+1|t}, 0)$ is the Jacobian matrix of the observation model with respect to the state evaluated at the predicted mean. $R_t := \frac{\partial}{\partial v} h(\mu_{t+1|t}, 0)$ is the Jacobian matrix of the observation model with respect to the noise evaluated at the predicted mean.

- **Predicted Mean**:

$$\mu_{t+1|t} \approx f(\mu_{t|t}, u_t, 0)$$

- **Predicted Covariance**:

$$\Sigma_{t+1|t} \approx F_t \Sigma_{t|t} F_t^\top + Q_t W Q_t^\top$$

Where W is the covariance matrix of the process noise.

In the context of visual mapping using EKF, the observation model relates the observed pixel coordinates $z_{t,i}$ to the landmark positions m_j and the IMU pose T_t. This model incorporates measurement noise $v_{t,i}$ following a Gaussian distribution with zero mean and covariance matrix V.

$$z_{t,i} = h(T_t, m_j) + v_{t,i} = K_s \pi({}_O T_I T_{t-1} m_j) + v_{t,i}$$

where: m_j is represented in homogeneous coordinates: $m_j := [m_{1j}, m_{2j}, m_{3j}, 1]^\top$. K_s is the stereo calibration matrix. $\pi(q)$ is the projection function. ${}_O T_I$ is the transformation matrix from the left camera to the IMU frame.

$$\begin{bmatrix} u_L \\ v_L \\ u_R \\ v_R \end{bmatrix} = \begin{bmatrix} f_{su} & 0 & c_u & 0 \\ 0 & f_{sv} & c_v & 0 \\ f_{su} & 0 & c_u & -f_{su}b \\ 0 & f_{sv} & c_v & 0 \end{bmatrix} \frac{1}{z} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \qquad \begin{bmatrix} x \\ y \\ z \end{bmatrix} = R^T(m - p)$$

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

- **EKF Update Step**: Given a new observation z_{t+1} at time $t+1$, the EKF update step involves computing the Kalman gain K_{t+1} and updating the mean μ_{t+1} and covariance Σ_{t+1} of the state estimate.

$$K_{t+1} = \Sigma_t H_{t+1}^\top (H_{t+1} \Sigma_t H_{t+1}^\top + I \otimes V)^{-1}$$
$$\mu_{t+1} = \mu_t + K_{t+1}(z_{t+1} - \tilde{z}_{t+1})$$
$$\Sigma_{t+1} = (I - K_{t+1} H_{t+1}) \Sigma_t$$

where: $\tilde{z}_{t+1}$ is the predicted observation based on the landmark position estimates μ_t at time t. H_{t+1} is the Jacobian matrix of the observation model with respect to the landmark positions, evaluated at μ_t.

- **Jacobian Computation**: To compute the Jacobian H_{t+1}, we need to evaluate the derivative of the observation model with respect to the landmark positions m_j at the current mean μ_t.

$$H_{t+1,i,j} = \begin{cases} K_s \frac{\partial \pi}{\partial q}({}_O T_I T_{t-1} \mu_{t,j}) {}_O T_I T_{t-1} P^\top, & \text{if } \Delta t(j) = i, \\ 0, & \text{otherwise.} \end{cases}$$

Where $P = [I, 0]$ and $\Delta t(j)$ represents the correspondence between observations at time t and landmarks.

At each timestamp, I identify the observed landmarks, compute the H and K matrices using the pose calculated in the previous step, and update the landmark mean and variance. I initialize the landmark variance as a 2× the identity matrix with a size equal to $3M$.

3.3 Visual-inertial SLAM

3.3.1 Visual-Inertial Odometry

In SE(3), a Gaussian distribution over a pose matrix T can be defined by introducing a perturbation ϵ on the Lie algebra, where $\mu \in SE(3)$ is the deterministic mean and $\epsilon \in \mathbb{R}^6$ is a zero-mean Gaussian random vector corresponding to the 6 degrees of freedom of T.

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

Here, $\hat{\epsilon}$ represents the skew-symmetric matrix associated with ϵ, and Σ denotes the covariance matrix of ϵ, reflecting the uncertainty in the pose estimation.
The derivative of a function $f(T)$ in SE(3) can be obtained using a first-order Taylor series approximation with perturbation $\delta\psi$ in $\mathbb{R}^6$.

$$f(T\exp(\delta\hat{\psi})) \approx f(T) + \left[\frac{\partial}{\partial T} f(T)\right] \delta\psi$$

In this expression, $\frac{\partial}{\partial T} f(T)$ represents the derivative of $f(T)$ with respect to T evaluated at T.
The motion model for the continuous-time IMU pose $T(t)$ with noise $w(t)$ involves both the nominal pose μ and a small perturbation $\delta\mu$.

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

Here, μ represents the nominal pose, and $\delta\mu$ is the perturbation.
In the EKF prediction step, the prior distribution of the IMU pose T given previous observations and control inputs is represented as $T_t | z_{0:t}, u_{0:t-1} \sim \mathcal{N}(\mu_{t|t}, \Sigma_{t|t})$, where $\mu_{t|t} \in SE(3)$ and $\Sigma_{t|t} \in \mathbb{R}^{6\times 6}$. This implies that $T_t = \mu_{t|t}\exp(\delta\hat{\mu}_{t|t})$, where $\delta\mu_{t|t}$ follows a Gaussian distribution with zero mean and covariance $\Sigma_{t|t}$. The motion model for the prediction step involves both the nominal kinematics of $\mu_{t|t}$ and the perturbation kinematics of $\delta\mu_{t|t}$ with time discretization τ_t:

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

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

where

$$u_t = \begin{bmatrix} v_t \\ w_t \end{bmatrix} \in \mathbb{R}^6, \quad \hat{u}_t = \begin{bmatrix} \hat{w}_t & v_t \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{4\times 4}, \quad \overset{\curlywedge}{\mathbf{u}}_t = \begin{bmatrix} \hat{w}_t & \hat{v}_t \\ 0 & \hat{w}_t \end{bmatrix} \in \mathbb{R}^{6\times 6}$$

At time $t+1$, given the current state estimate $\mu_{t+1|t}$ and covariance $\Sigma_{t+1|t}$, where $\mu_{t+1|t}$ is in $SE(3)$ and $\Sigma_{t+1|t}$ is a 6×6 covariance matrix. The observation model relates the observed pixel coordinates $z_{t+1,i}$ to the IMU pose T_{t+1} and landmark positions m_j, incorporating measurement noise $v_{t+1,i}$.

$$z_{t+1,i} = h(T_{t+1}, m_j) + v_{t+1,i} = K_s \pi({}_O T_I T_{t+1}^{-1} m_j) + v_{t+1,i}$$

The observation model Jacobian H_{t+1} with respect to the IMU pose T_{t+1} is needed for the update step. We approximate the observation model at time $t+1$ using a first-order Taylor series expansion with respect to the IMU pose perturbation $\delta\mu$.

$$z_{t+1,i} \approx \tilde{z}_{t+1,i} - K_s \frac{\partial \pi}{\partial q}({}_O T_I \mu_{t+1|t}^{-1} m_j) {}_O T_I (\mu_{t+1|t}^{-1} m_j)^{\odot} \delta\mu + v_{t+1,i}$$

Where: $\tilde{z}_{t+1,i} = K_s \pi({}_O T_I \mu_{t+1|t}^{-1} m_j)$ is the predicted observation based on the current state estimate and for homogeneous coordinates $s \in \mathbb{R}^4$ and $\hat{\xi} \in \text{se}(3)$:

$$\hat{\xi} s = s^{\odot}\xi \qquad \begin{bmatrix} s \\ 1 \end{bmatrix}^{\odot} := \begin{bmatrix} I & -\hat{s} \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{4\times 6}$$

3.3.2 Visual-inertial SLAM Steps

Here's a detailed description of the Visual-inertial SLAM process:

1. **Initialization**: The process begins with initializing the full covariance matrix $\Sigma_{t,\text{SLAM}} = [\Sigma_{LL}\ \Sigma_{RL}\ \Sigma_{LR}\ \Sigma_{RR}] \in \mathbb{R}^{(3M+6)\times(3M+6)}$, where Σ_{LL} represents the covariance among landmark positions, Σ_{RL} and Σ_{LR} represent the covariance between landmark positions and robot positions, and Σ_{RR} represents the covariance among robot state.

2. **EKF Prediction Step**: The covariance matrix $\Sigma_{t,\text{SLAM}}$ is updated using the EKF prediction step:
$$\Sigma_{t,\text{SLAM}} = \begin{bmatrix} \Sigma_{LL}F^\top & \Sigma_{RL} \\ \Sigma_{LR}F & \Sigma_{RR} \end{bmatrix} \begin{bmatrix} F & 0 \\ 0 & F^\top \end{bmatrix} + W$$
where F represents the motion model, and W represents the noise covariance.
$$F = \exp\left(-\tau \overset{\curlywedge}{\mathbf{u}}_t\right) \tag{1}$$

3. **SLAM Jacobian**: The resulting SLAM Jacobian, denoted as $H_{t+1,\text{SLAM}} = [H_{t+1,\text{landmarks}}\ H_{t+1,\text{robot}}] \in \mathbb{R}^{4N_t\times(3M+6)}$, contains derivatives of the observation model with respect to landmark positions and robot pose.

4. **Calculate Kalman Gain**: The Kalman Gain matrix $K_{t+1,\text{SLAM}}$ is computed using $\Sigma_{t,\text{SLAM}}$ and $H_{t+1,\text{SLAM}}$. The entire update step for the SLAM problem can be performed in a single step.
$$K_{t+1} = \Sigma_t H_{t+1}^\top (H_{t+1}\Sigma_t H_{t+1}^\top + I \otimes V)^{-1}$$

5. **Compute Mean and Covariance Change**: Finally, the mean change and covariance change are computed directly using the whole Kalman Gain matrix.
$$\delta\mu = K_{t+1}(z_{t+1} - \tilde{z}_{t+1})$$
$$\mu_{t+1} = \mu_{t+1|t} + \delta\mu[:3M] \qquad \text{(Landmark update)}$$
$$m_{t+1|t+1} = m_{t+1|t} exp\left(\widehat{\delta\mu[3M:]}\right) \text{(Car pose update)}$$
$$\Sigma_{t+1|t+1} = (I - K_{t+1}H_{t+1})\Sigma_{t+1|t}$$

3.3.3 Details of implementations

I found that adjusting the sampling strategy helped mitigate memory issues and significantly improved processing speed. Specifically, for dataset 3, I opted to sample every 10 landmarks, while for dataset 10, I sampled every 20 landmarks. These adjustments yielded the best results in terms of both computational efficiency and performance.
During the implementation of Visual-Inertial SLAM (VI SLAM), I encountered significant instability and poor trajectory results, particularly in Part 3 of the project. I identified instability in the covariance values as the primary culprit behind the erratic behavior of the trajectory. EKF is highly sensitive to initial covariances ($\Sigma_{0|0}$) and noise parameters (W and V). I initialized W with very small values to stabilize the covariance matrix. I adjusted V to have slightly larger values to better account for measurement noise, typically in pixels. By meticulously tuning hyperparameters, particularly W and V, and ensuring appropriate initialization of covariance matrices, I succeeded in significantly enhancing the stability and accuracy of the VI SLAM implementation. I attained optimal results through a combination of specific initialization and hyperparameters.

1. $\Sigma_{LL} = 2I_{3M\times 3M}$
2. $\Sigma_{RR} = diag([10^{-3}, 10^{-3}, 10^{-3}, 10^{-4}, 10^{-4}, 10^{-4}])$
3. $V_t = 15I_{4N_t\times 4N_t}$
4. $W = diag([10^{-4}, 10^{-4}, 10^{-4}, 10^{-7}, 10^{-7}, 10^{7}])$

4 Result

4.1 Dataset 3

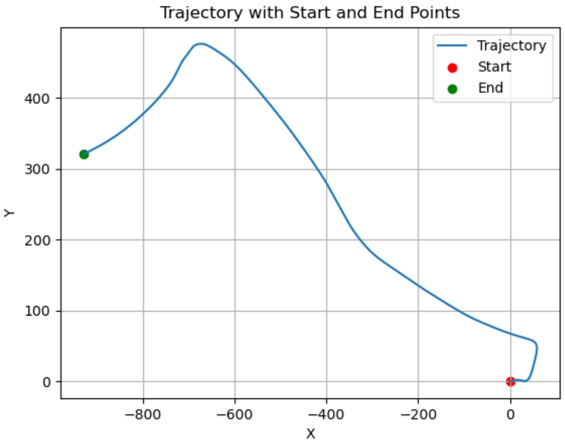

Figure 1: Initial Trajectory based on IMU

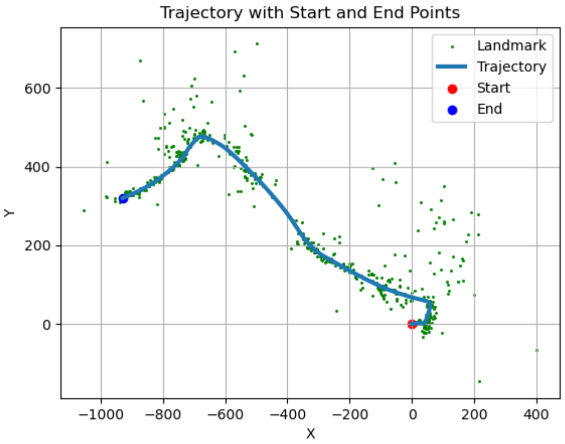

Figure 2: Initial Landmark mapping

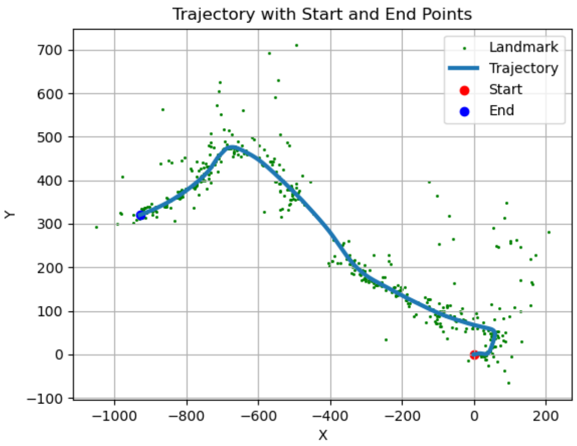

Figure 3: Optimized Landmark mapping via EKF update

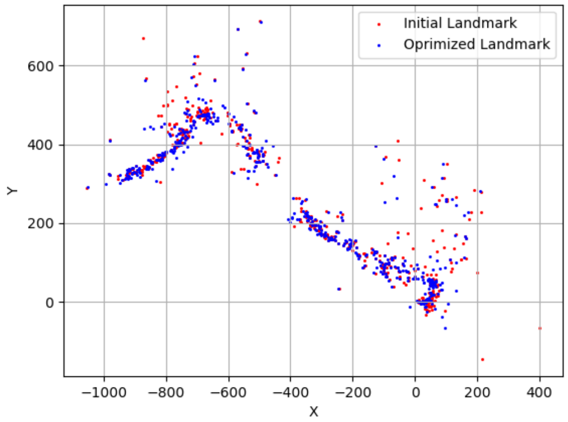

Figure 4: Initial vs. EKF Optimized Landmark mapping

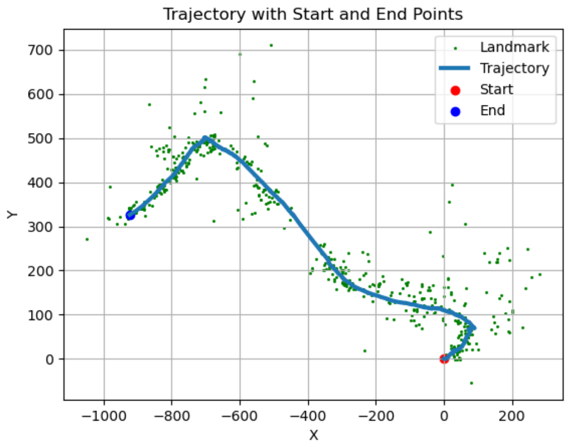

Figure 5: Optimized Landmark mapping via Visual-inertial SLAM

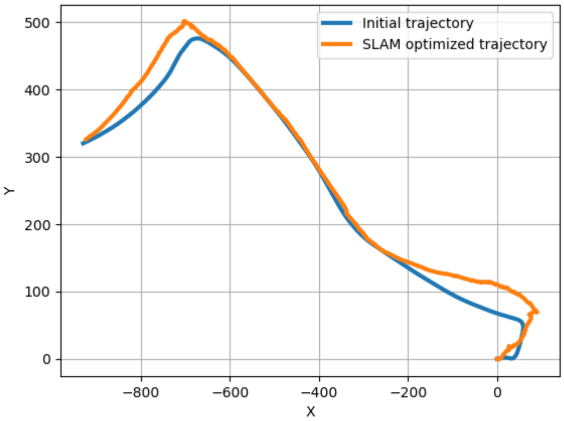

Figure 6: Initial vs. Optimized Trajectory

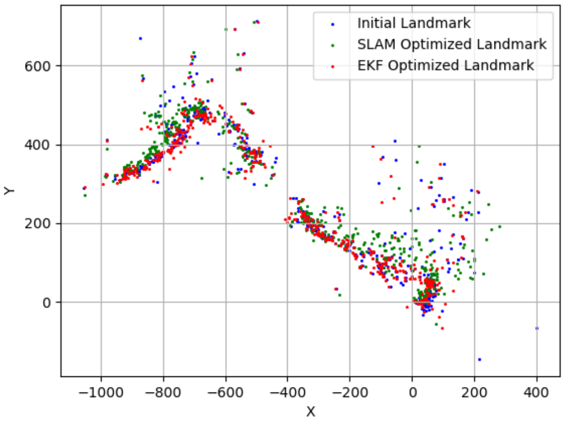

Figure 7: Initial vs. EKF Optimized vs. SLAM Optimized Landmark mapping

4.2 Dataset 10

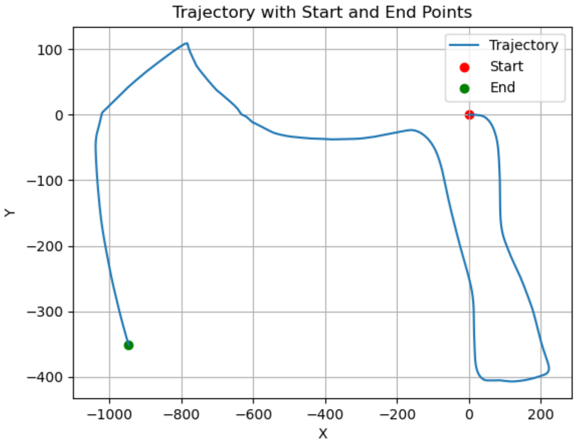

Figure 8: Initial Trajectory based on IMU

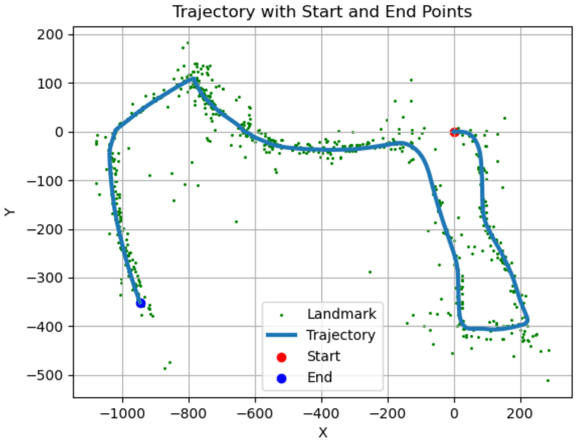

Figure 9: Initial Landmark mapping

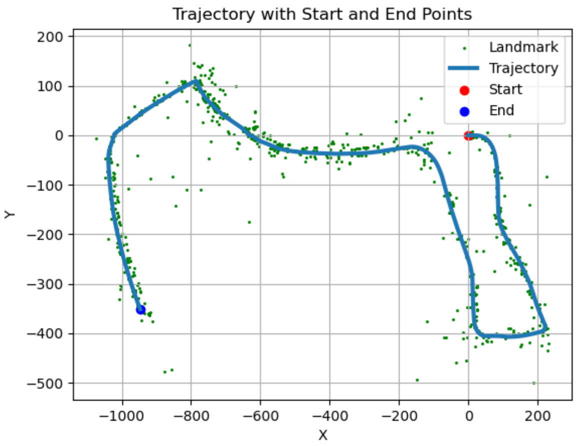

Figure 10: Optimized Landmark mapping via EKF update

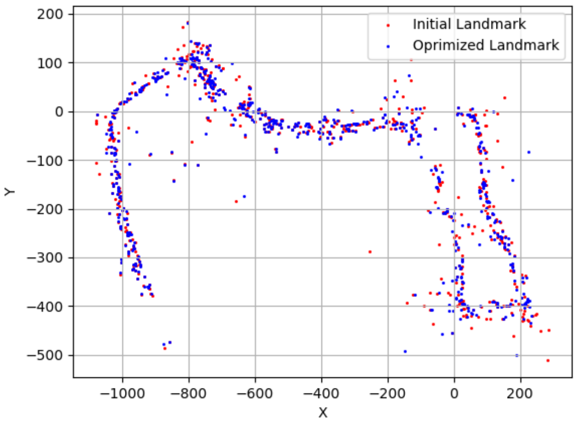

Figure 11: Initial vs. EKF Optimized Landmark mapping

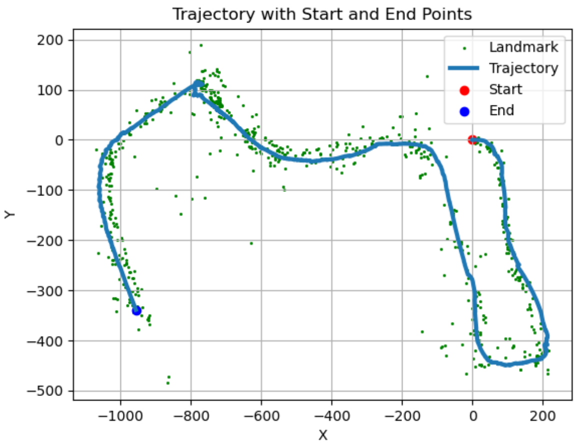

Figure 12: Optimized Landmark mapping via Visual-inertial SLAM

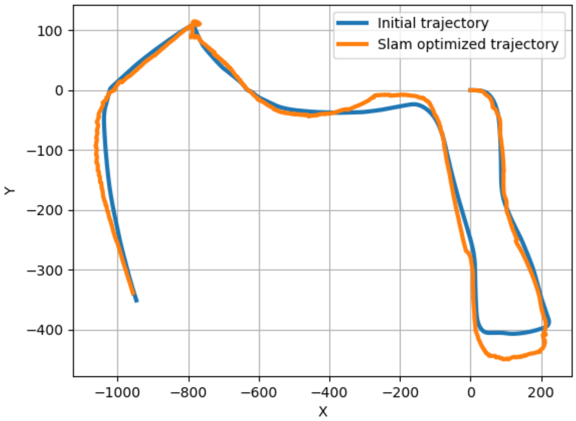

Figure 13: Initial vs. Optimized Trajectory

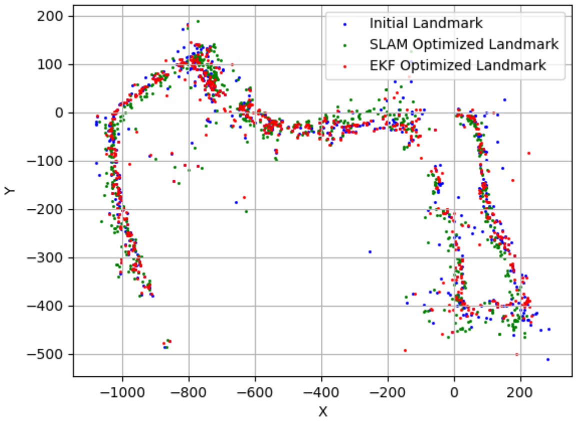

Figure 14: Initial vs. EKF Optimized vs. SLAM Optimized Landmark mapping

4.3 Suboptimal results

Suboptimal results were observed due to insufficient fine-tuning of hyperparameters for Visual-inertial SLAM.

4.3.1 Dataset 3

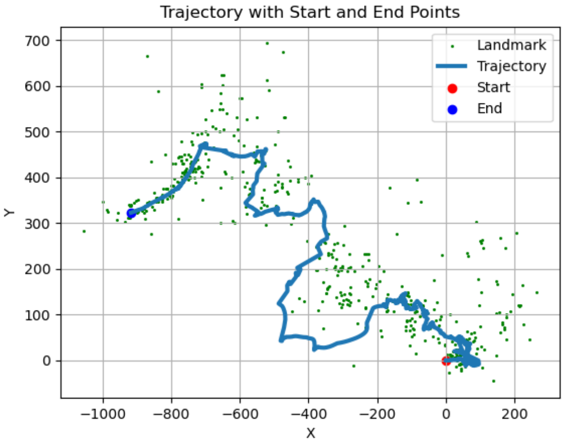

Figure 15: Optimized Landmark mapping via Visual-inertial SLAM

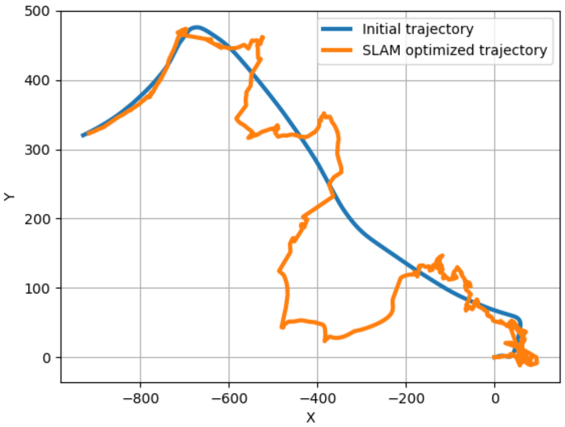

Figure 16: Initial vs. Optimized Trajectory

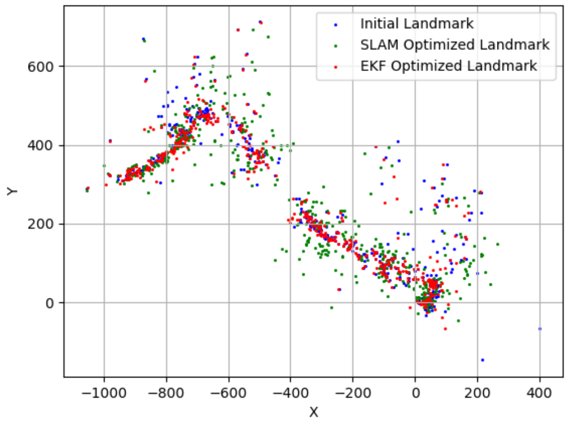

Figure 17: Initial vs. EKF Optimized vs. SLAM Optimized Landmark mapping

4.3.2 Dataset 10

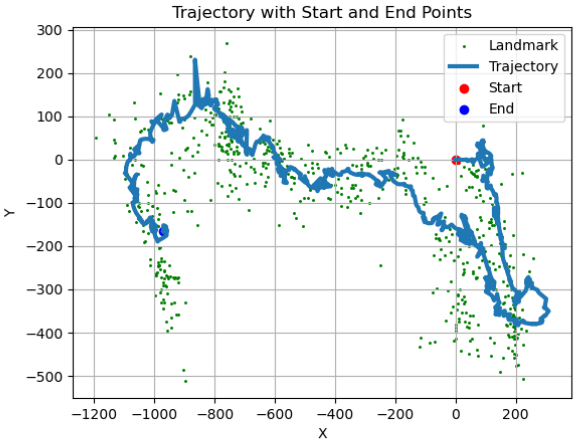

Figure 18: Optimized Landmark mapping via Visual-inertial SLAM

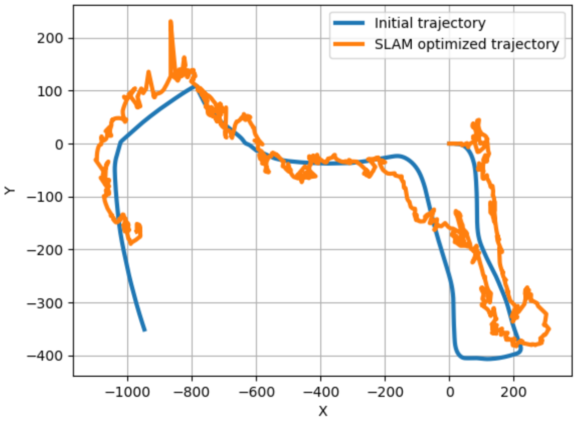

Figure 19: Initial vs. Optimized Trajectory

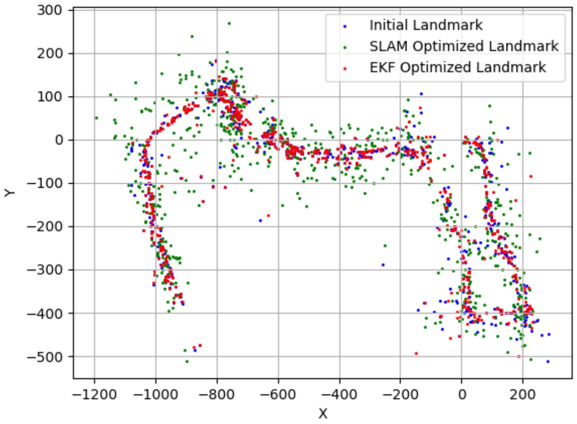

Figure 20: Initial vs. EKF Optimized vs. SLAM Optimized Landmark mapping

5 Conclution

The project aimed to implement visual-inertial simultaneous localization and mapping (SLAM) using an extended Kalman filter (EKF). Synchronized measurements from an inertial measurement unit (IMU) and a stereo camera were utilized, alongside intrinsic and extrinsic sensor calibrations. The EKF prediction step for IMU localization and landmark mapping demonstrated the algorithm's ability to estimate poses and landmark positions accurately. Challenges included sensitivity to parameter settings and computational complexity, highlighting the need for robust parameter tuning methods and efficient algorithms for real-time applications. Future work could focus on further optimizing parameters, exploring advanced algorithms, and enhancing sensor fusion techniques to improve SLAM system performance.