Visual-Intertial SLAM

Sanchit Gupta
Department of Electrical and
Computer Engineering
University of California San Diego
sag006@ucsd.edu

***Abstract*—This report discusses about the methodology to implement Visual-Inertial Simultaneous Localization and Mapping (SLAM) using an Extended Kalman Filter (EKF) and the synchronized measurements provided from an inertial measurement unit (IMU) and a stereo camera. The results of the path followed by the robot in dead reckoning state and after implementation of IMU update step with the landmark update step are presented in this report.**

Keywords—Visual Inertial SLAM, Extended Kalman Filter, Dead reckoning, Robot Localization

I. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) is a popular and widely researched topic in the field of robotics, autonomous driving and augmented reality. SLAM is a technique through which the robot estimates its state in the environment it is traversing and simultaneously, constructs the map of the environment according to the feedback received from the sensors. This technique aids the robot in identifying free spaces for locomotion in the unknown environment as well as correcting its own trajectory basis the landmarks identified in the path.

The feedback from the sensors plays an important role in the implementation of SLAM algorithms. In this project, the synchronized measurements from an IMU and a stereo camera are provided. The measurements from IMU will help in predicting the trajectory of the robot. The measurements of landmarks identified in the path using a stereo camera will aid in correcting and updating the robot's trajectory.

Extended Particle Filter (EKF) is one of the popular methods to implement Visual-Inertial SLAM. EKF prediction and update steps are implemented in this project to obtain the dead reckoning trajectory of the robot and to estimate the positions of landmarks observed during the path. The complete Visual-Inertial SLAM algorithm is then implemented combining the IMU prediction step and landmark update step to obtain updated trajectory of the robot. The steps followed to implement EKF and Visual-Inertial SLAM are discussed in detail in Section II and III. The results for the dead reckoning trajectory of the robot, updated positions of the landmarks and updated robot's trajectory after Visual-Inertial SLAM for different number of features are presented in Section IV of this report.

II. PROBLEM FORMULATION

This section will discuss about the implementation of visual-inertial SLAM using EKF in terms of mathematical formulas. The problem formulation of SLAM includes estimation of robot states $x_{0:t}$ and map states $m_{0:t}$ from the sequence of control inputs $u_{0:t}$ and observations $z_{0:t}$. Given the control inputs and observations, the motion model and observation model will be formulated using equation 1.

$$Motion\ model: x_{t+1} = f(x_t, u_t, w_t) \sim p_f(.|x_t, u_t)$$
$$Observation\ model: z_t = h(x_t, v_t) \sim p_h(.|x_t) \quad (1)$$

Following the Bayes' rules, the framework to be used for the update and prediction steps is shown in equation in 2.

$$Prediction: p_{t+1}(x) = \int p_f(x|s, u_t) p_{t|t}(s) ds$$
$$Update: p_{t+1|t+1}(x) = \frac{p_h(z_{t+1}|x)\, p_{t+1|t}(x)}{\int p_h(z_{t+1}|s)\, p_{t+1|t}(s) ds} \quad (2)$$

Subsequently, the framework to update the map is given in equation 3.

$$P(m_t|x_t, z_{(1:t)}, u_{1:t}) = \quad (3)$$
$$\sum_{x_t} \sum_{y_t} P(m_t|x_t, m_{t-1}, z_t, u_{1:t})\ P(m_{t-1}, x_1|z_{1:t-1}, m_{t-1}, u_{1:t}$$

Sensor Dataset: The IMU measurements comprising of linear velocity $v_t \in R^3$ and angular velocity $\omega_t \in R^3$ measured in the body frame of the IMU are given in the project. The time-stamps τ_t in UNIX standards are given. The pixel co-ordinates of the landmarks observed by the stereo camera are also given along-with. The measurements comprising of IMU data and stereo camera data are synchronized with the time-stamps.

The intrinsic calibration inclusive of baseline 'b' and calibration matrix is provided. The extrinsic calibration matrix for transformation from camera to IMU frame is also provided. The data is provided in two datasets namely, 03.npz and 10.npz.

$$K = \begin{bmatrix} fs_u & 0 & c_u \\ 0 & fs_v & c_v \\ 0 & 0 & 1 \end{bmatrix} \quad (4)$$

The problem statement requires the implementation of following tasks:

A) IMU Localization via EKF Prediction: Given linear velocity $v_t \in R^3$ and angular velocity $\omega_t \in R^3$, the pose of the IMU $T_t \in SE(3)$ is to be predicted over time.

B) Landmark mapping via EKF Update: Given the pixel co-ordinates $z_t \in R^{4xM}$ of detected features by stereo camera and assuming the predicted trajectory of IMU to be correct, the positions of landmarks $m \in R^{3xM}$ are to be estimated and updated.

C) Visual Inertial SLAM: Now, combining the prediction step from part (a) and landmark update step from (b), the IMU update step is required to be implemented based on the stereo camera observation model.

III. TECHNICAL APPROACH

The technical approach to solve the three problems defined in Section II using the Extended Kalman Filter approach of SLAM is discussed in this section and is elaborated in different sub-sections below.

A. Extended Kalman Filter Formulation

Kalman filter is a type of Bayes filter with the assumptions:

- The prior pdf $p_{(t|t)}$ is Gaussian
- The motion model is linear in state x_t with Gaussian noise w_t
- The observation model is linear in state x_t with Gaussian noise v_t
- The motion noise w_t and observation noise v_t are independent of each other, of state x_t and across time.

As the Kalman filter requires the motion and observation model to be linear, it cannot be applied if the models are non-linear in state. The non-linear Kalman filter is an extension of Kalman filter which forces the predicted and updated pdfs to be Gaussian via approximation, in-order to solve the non-linear models. Extended Kalman filter (EKF) is one method to implement non-linear Kalman filter. It uses first-order Taylor series approximation to the motion and observation models around the state and noise. After approximation, the motion model along with the associated Jacobians in EKF are given in equation 5.

$$f(x_t, u_t, w_t) \approx f(\mu_{(t|t)}, u_t, 0) + F_t\left(x - \mu_{(t|t)}\right) + Q_t w_t$$
$$F_t = \frac{df}{dx}(\mu_{(t|t)}, u_t, 0) \text{ and } Q_t = \frac{df}{dw}(\mu_{(t|t)}, u_t, 0) \quad (5)$$

The mean and covariance for the predicted step are given in equation 6.

$$\mu_{(t+1|t)} = f(\mu_{(t|t)}, u_t, 0)$$
$$\Sigma_{(t+1|t)} = F_t \Sigma_t F_t^T + Q_t W Q_t^T \quad (6)$$

The observation model after approximation and the associated Jacobians in EKF are given in equation 7.

$$h(x_{t+1}, v_{t+1}) \approx h\left(\mu_{(t+1|t)}, 0\right) + H_{t+1}\left(x_{t+1} - \mu_{(t+1|t)}\right) + R_{t+1} v_{t+1} \quad (7)$$
$$H_{t+1} = \frac{dh}{dx}\left(\mu_{(t+1|t)}, 0\right) \text{ and } R_{t+1} = \frac{dh}{dv}(\mu_{(t+1|t)}, 0)$$

The Kalman gain, mean and covariance for the update step are given in equation 8.

$$\mu_{(t+1|t+1)} = \mu_{(t+1|t)} + K_{(t+1|t)}(z_{t+1} - m_{(t+1|t)})$$
$$\Sigma_{(t+1|t+1)} = \left(I - K_{(t+1|t)} H_{t+1}\right)\Sigma_{(t+1|t)} \quad (8)$$
$$K_{(t+1|t)} = \Sigma_{(t+1|t)} H_{t+1}^T (H_{t+1} \Sigma_{(t+1|t)} H_{t+1}^T + R_{t+1} V R_{t+1}^T)$$

B. IMU Localization via EKF Prediction

The datasets 03.npz and 10.npz comprise of linear velocity $v_t \in R^3$ and angular velocity $\omega_t \in R^3$ measurements from the IMU. These two measurements form the prior control input u_t for the model. The data given in 03.npz is over 1010 seconds whereas in 10.npz, it is over 3026 seconds. The pose of the IMU $T_t \in SE(3)$ is estimated using the EKF model by Visual-Inertial SLAM and perturbation kinematics, as given in equation 9.

$$\boldsymbol{\mu}_{t+1|t} = \boldsymbol{\mu}_{t|t} \exp\left(\tau_t \hat{\mathbf{u}}_t\right)$$
$$\delta\boldsymbol{\mu}_{t+1|t} = \exp\left(-\tau_t \hat{\underset{\curvearrowright}{\mathbf{u}}}_t\right) \delta\boldsymbol{\mu}_{t|t} + \mathbf{w}_t \quad (9)$$
$$\Sigma_{t+1|t} = \mathbb{E}[\delta\boldsymbol{\mu}_{t+1|t} \delta\boldsymbol{\mu}_{t+1|t}^\top] = \exp\left(-\tau \hat{\underset{\curvearrowright}{\mathbf{u}}}_t\right) \Sigma_{t|t} \exp\left(-\tau \hat{\underset{\curvearrowright}{\mathbf{u}}}_t\right)^\top + W$$

where

$$\mathbf{u}_t := \begin{bmatrix} \mathbf{v}_t \\ \boldsymbol{\omega}_t \end{bmatrix} \in \mathbb{R}^6 \quad \hat{\mathbf{u}}_t := \begin{bmatrix} \hat{\boldsymbol{\omega}}_t & \mathbf{v}_t \\ \mathbf{0}^\top & 0 \end{bmatrix} \in \mathbb{R}^{4\times 4} \quad \hat{\underset{\curvearrowright}{\mathbf{u}}}_t := \begin{bmatrix} \hat{\boldsymbol{\omega}}_t & \hat{\mathbf{v}}_t \\ 0 & \hat{\boldsymbol{\omega}}_t \end{bmatrix} \in \mathbb{R}^{6\times 6}$$

In order to obtain a prediction of the pose of the IMU, no noise is considered in this step. The mean $\in R^{4x4xt}$ and covariance $\in R^{6x6xt}$ as obtained in this prediction step are stored in the arrays of appropriate dimensions for use in subsequent steps. The pose trajectory of the IMU is plotted for both the given datasets. Figure 1 show the estimated trajectory for 03.npz and Figure 2 show the estimated trajectory for 10.npz.

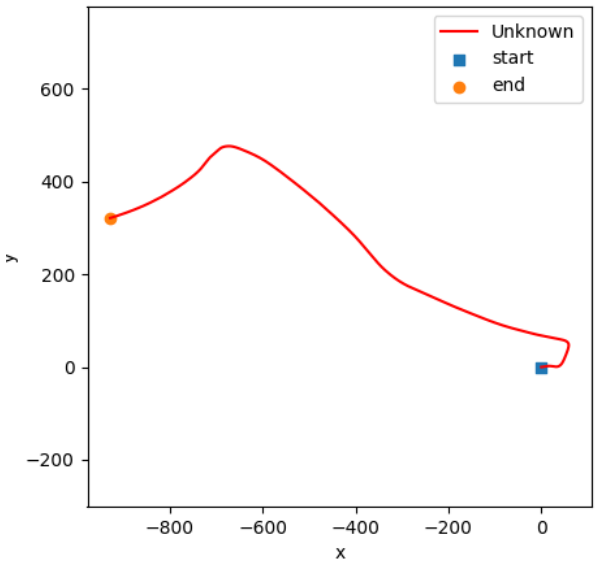

Figure 1: Estimated pose trajectory of IMU after prediction step for 03.npz

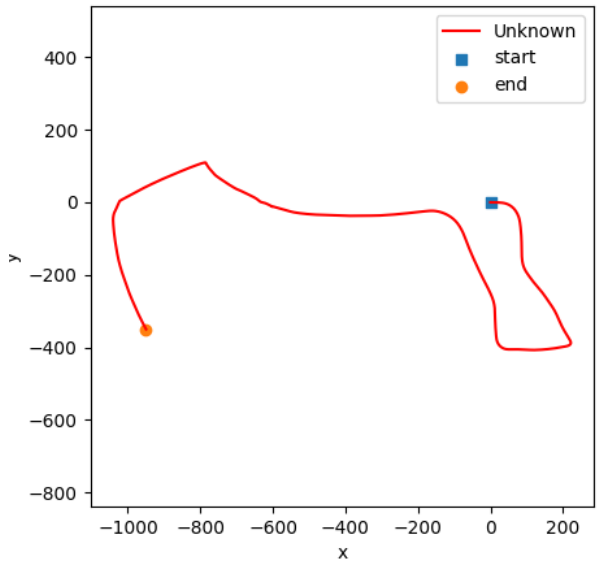

Figure 2: Estimated pose trajectory of IMU after prediction step for 10.npz

B) Landmark mapping via EKF Update: The pixel co-ordinates $z_t \in R^{4xM}$ of detected features by stereo camera are provided in the data sets. The data set 03.npz consists of data for 5105 features and 10.npz consists of 13289 features. The first step here is to convert the pixel co-ordinates z of the features into world frame. The baseline, calibration matrix and pose to transform the coordinates from camera frame to IMU frame are provided in the dataset. The pose obtained in part (a) of the problem statement will transform the coordinates from IMU frame to world frame. The transformation is shown in equation 10.

$$z = [u_l, v_l, u_r, v_r]$$

$$\begin{bmatrix} u_L \\ v_L \\ u_R \\ v_R \end{bmatrix} = \underbrace{\begin{bmatrix} fs_u & 0 & c_u & 0 \\ 0 & fs_v & c_v & 0 \\ fs_u & 0 & c_u & -fs_u b \\ 0 & fs_v & c_v & 0 \end{bmatrix}}_{M} \frac{1}{z} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{10}$$

$$\begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = {}_wT_{imu} \; * \; {}_{imu}T_{cam} * \begin{bmatrix} x_o \\ y_o \\ z_o \\ 1 \end{bmatrix}$$

Since the number of features given is high and it would be computationally very difficult to process the data for all the features, the best features as per a set count have been computed for the update step and plotting. As a particular feature appears in more than one time-stamp, the best features are collected by taking the features which occur at maximum instances across all time-stamps.

Further, the average of the coordinates for a particular feature across all instances of its appearance is taken to formulate the homogenous mean for the update step. This step is carried out for all the features collected using the 'best features' methodology. The EKF update steps for the positions of landmarks $m \in R^{3xM}$ are implemented as shown in equations 11 and 12.

$$\text{Prior: } \boldsymbol{\mu}_t \in \mathbb{R}^{3M} \text{ and } \Sigma_t \in \mathbb{R}^{3M \times 3M}$$

$$\tilde{\mathbf{z}}_{t+1,i} := K_s \pi \left({}_O T_I \, T_{t+1}^{-1} \underline{\boldsymbol{\mu}}_{t,j} \right) \in \mathbb{R}^4 \qquad \text{for } i = 1, \ldots, N_{t+1}$$

$$H_{t+1,i,j} = \begin{cases} K_s \frac{d\pi}{d\mathbf{q}} \left({}_O T_I \, T_{t+1}^{-1} \underline{\boldsymbol{\mu}}_{t,j} \right) {}_O T_I \, T_{t+1}^{-1} P^\top & \text{if } \Delta_t(j) = i, \\ \mathbf{0}, & \text{otherwise} \end{cases} \tag{11}$$

Here, H is the Jacobian of $z - tilda_{t+1,j}$ with respect to m_j evaluated at $\mu_{t,j}$.

EKF update:

$$K_{t+1} = \Sigma_t H_{t+1}^\top \left(H_{t+1} \Sigma_t H_{t+1}^\top + I \otimes V \right)^{-1}$$
$$\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t + K_{t+1} \left(\mathbf{z}_{t+1} - \tilde{\mathbf{z}}_{t+1} \right)$$
$$\Sigma_{t+1} = (I - K_{t+1} H_{t+1}) \Sigma_t$$

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

$$\pi(\mathbf{q}) := \frac{1}{q_3} \mathbf{q} \in \mathbb{R}^4 \qquad \frac{d\pi}{d\mathbf{q}}(\mathbf{q}) = \frac{1}{q_3} \begin{bmatrix} 1 & 0 & -\frac{q_1}{q_3} & 0 \\ 0 & 1 & -\frac{q_2}{q_3} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{q_4}{q_3} & 1 \end{bmatrix} \in \mathbb{R}^{4 \times 4} \tag{12}$$

C. Visual-Inertial SLAM

The IMU pose prediction step from part (a) is combined with the features position update step from part (b) to obtain the complete Visual-Inertial SLAM model based on the stereo camera observation model. The EKF update steps for this part are implemented as per the equations shown in equation 13. The prior mean and covariance for this part are obtained from the ones calculated in the IMU pose prediction step. These parameters are updated after implementation of complete

visual-inertial SLAM model, giving the updated trajectory of the robot.

$$\textbf{Prior: } \boldsymbol{\mu}_{t+1|t} \in SE(3) \text{ and } \Sigma_{t+1|t} \in \mathbb{R}^{6\times 6}$$

$$\tilde{\mathbf{z}}_{t+1,i} := K_s \pi \left({}_O T_I \boldsymbol{\mu}_{t+1|t}^{-1} \underline{\mathbf{m}}_j \right) \qquad \text{for } i = 1, \ldots, N_{t+1} \tag{13}$$

$$H_{t+1,i} = -K_s \frac{d\pi}{d\mathbf{q}} \left({}_O T_I \boldsymbol{\mu}_{t+1|t}^{-1} \underline{\mathbf{m}}_j \right) {}_O T_I \left(\boldsymbol{\mu}_{t+1|t}^{-1} \underline{\mathbf{m}}_j \right)^{\odot} \in \mathbb{R}^{4\times 6}$$

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

Perform the EKF update:

$$K_{t+1} = \Sigma_{t+1|t} H_{t+1}^{\top} \left(H_{t+1} \Sigma_{t+1|t} H_{t+1}^{\top} + I \otimes V \right)^{-1}$$
$$\boldsymbol{\mu}_{t+1|t+1} = \boldsymbol{\mu}_{t+1|t} \exp\left((K_{t+1}(\mathbf{z}_{t+1} - \tilde{\mathbf{z}}_{t+1}))^{\wedge} \right)$$
$$\Sigma_{t+1|t+1} = (I - K_{t+1} H_{t+1}) \Sigma_{t+1|t}$$
$$H_{t+1} = \begin{bmatrix} H_{t+1,1} \\ \vdots \\ H_{t+1,N_{t+1}} \end{bmatrix}$$

IV. Results

This section discusses about the results as obtained in different iterations for both the datasets.

Figure 3 show the estimated trajectory taken by the robot along with its orientation and direction vectors for the dataset 10.npz. Figure 4 show the same for the dataset 03.npz.

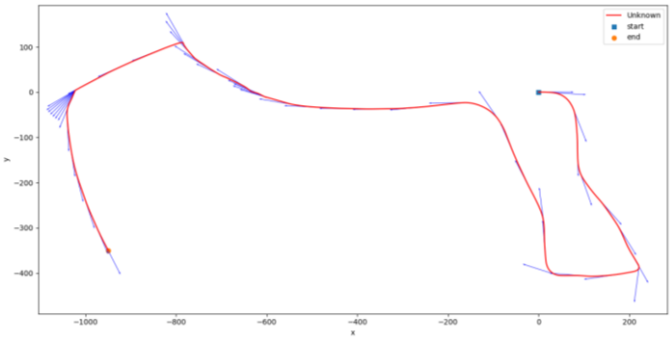

Figure 3: Estimated pose trajectory of IMU after prediction step along with robot orientation and direction vectors for 10.npz

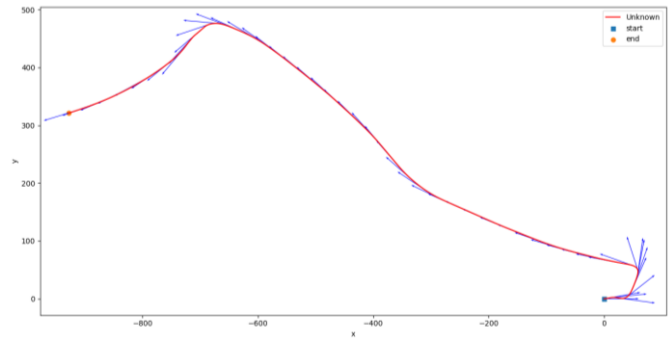

Figure 4: Estimated pose trajectory of IMU after prediction step along with robot orientation and direction vectors for 03.npz

For the part b of the problem statement, the iterations were run at different number of features in the range of 200 to 1000 to identify the best count of features for the update step. Further, the difference in number of features highlighted the associated difference in the plots for this part as well as for part c.

In this case, the variance matrix was initialized as 0.01 times the identity matrix of dimension 3m x 3m, where m is the total number of best features considered for the update step. The observation model noise V of dimension $4n_t$ x 1 was initialized as the identity matrix, meaning the variance was considered as 1. n_t is considered as the total number of features visible at any particular time-stamp. The motion model noise was considered as:

$$w = diag(0.5, 0.5, 0.5, 0.05, 0.05, 0.05)$$

The plots obtained for both the datasets at number of best features 700 and 1000 are shown in Figures 5 to 8. The green dots in all the images represent the locations of landmarks.

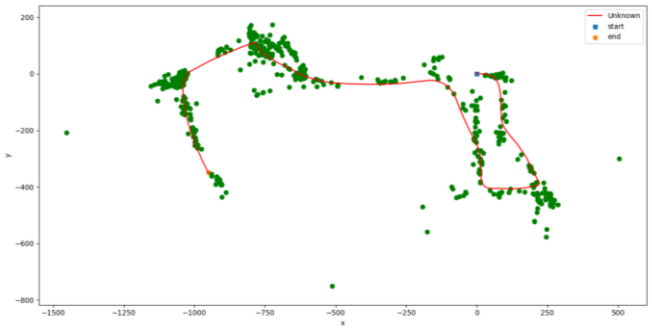

Figure 5: Plot after landmark update step for dataset 10.npz and number of features = 700

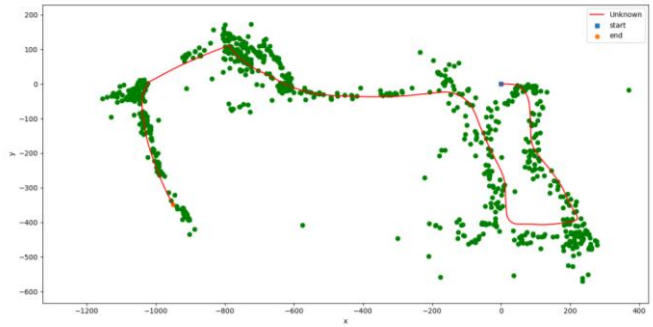

Figure 6: Plot after landmark update step for dataset 10.npz and number of features = 1000

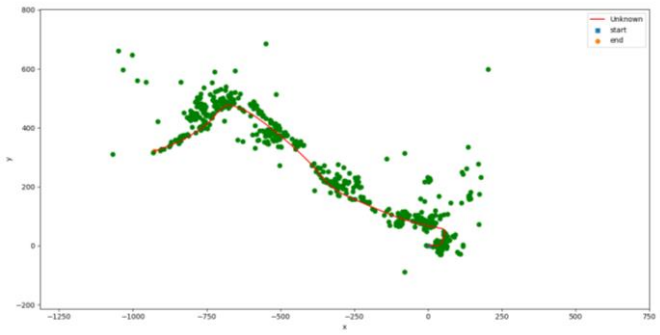

Figure 7: Plot after landmark update step for dataset 03.npz and number of features = 700

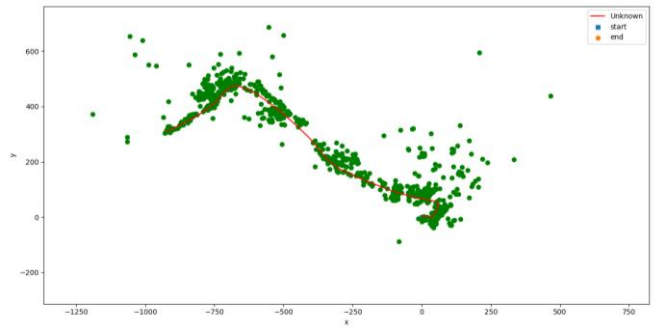

Figure 8: Plot after landmark update step for dataset 03.npz and number of features = 1000

For combining the IMU prediction step and the landmark update step for implementing Visual-Inertial SLAM in part c of the problem statement, the noises were considered in the motion and observation models. While executing the code with low noises, the singularity error on the covariance matrix of the motion model was obtained. Thus, the noise was adjusted iteratively in order to deal with the singularity error as well as to improve the trajectory as much as possible.

While the Visual SLAM algorithm worked perfectly for 03.npz dataset with the features count as 1000, the algorithm still showed singularity error for different particles count and different noise levels in 10.npz dataset. The plot obtained for 03.npz dataset with 1000 features is shown in Figure 9.

The noise in motion model was kept same as before but the noise in observation model was initialized as 15 times the Identity matrix.

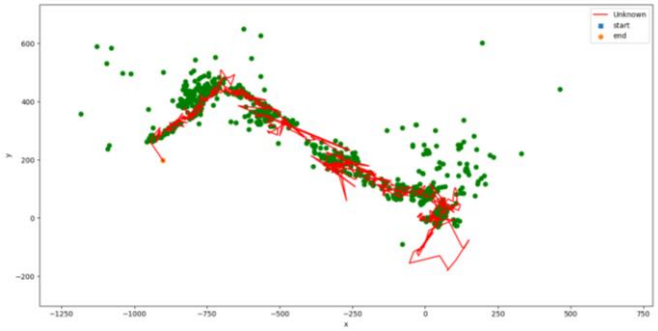

Figure 9: Plot obtained after executing the complete Visual-Inertial SLAM algorithm.

It can be seen from the plots that the trajectory obtained after updating the landmarks is similar to the dead reckoning trajectory. This justifies the correct implementation of the landmark update step. However, the trajectory after implementation of complete Visual-Inertial SLAM algorithm looks similar to dead reckoning trajectory for 03.npz but it is still noisy. This highlights the further work required in the direction of fine-tuning the noise levels and optimizing the overall algorithm.