ECE276A Project 3: Visual-Inertial SLAM

Richard Oliveira
Department of Electrical and Computer Engineering
University of California, San Diego
La Jolla, U.S.A

I. Introduction

In the context of autonomy, more specifically in robotics, as the scope of many problems in this field increase, it is necessary to accurately track and map the motion of a robot's trajectory and also it's environment, through the use of sensors, such as LiDAR, cameras, wheel encoders and other types of sensors. The problem of estimating the robot's position and environment, is a common problem in robotics and it is known as the *Simultaneous Localization and Mapping* (SLAM) problem[3]. In this project we present a way of solving the SLAM problem by using the *Extended Kalman Filter* (EKF). Initially, we simply use the *Prediction Step* of the EKF to track the position of the robot over time (*Localization*). Then, in order to estimate the location of landmark positions measured by a stereo camera mounted on the robot, we only use the *Update Step* of the EKF (*Mapping*). Lastly, the *Prediction Step* and the *Update Step* of the EKF are combined such that we perform *Localization* and *Mapping* simultaneously, hence the SLAM problem, in this context, takes the name of *Visual-Inertial SLAM* [4]. We then compare our trajectories with ground truth data and demonstrate that the EKF approach yields fairly good results.

II. Problem Formulation

Consider a vehicle, moving in the (x, y) plane with an orientation θ_t at time t. The position of the vehicle and it's orientation at time t is represented by a pose T_t:

$$T_t = \begin{pmatrix} R(\theta_t) & \boldsymbol{x}_t \\ \mathbf{0}^T & 1 \end{pmatrix} \tag{1}$$

Here $\boldsymbol{x}_t$ represents the position of the vehicle at time t. $R(\theta_t)$ represents a *Rotation matrix*, yielded by the angle θ_t at some time t.
We now introduce the *Motion model* of the vehicle, through a probabilistic lens. Via a series of inputs $\boldsymbol{u}_t$ at some time t, such as linear velocity $\boldsymbol{v}_t$ and angular velocity $\boldsymbol{\omega}_t$, the *Motion model*, also known as the *Discrete-Time Probabilistic Kinematic model*, can be represented as [1]:

$$\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{w}_t) \sim p_f(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{u}_t) \tag{2}$$

Here, $\boldsymbol{w}_t$ is the noise associated to the *Motion model*, and it is distributed according to a *Multi-Variate Gaussian distribution*, $\boldsymbol{w}_t \sim \mathcal{N}(\boldsymbol{\mu}_w, \Sigma_w)$. According to the so called *Markov Assumptions*, given a series of random variables, $\{X_t\}_{t=0}^T$ indexed in time t, the following is true:

$$p(X_t|X_{t-1}, \dots, X_0) = p(X_t|X_{t-1}), \ \forall t \in \{1, \dots, T\} \tag{3}$$

Therefore, since (3) holds, our formulation in (2) allows us to "predict" the position of the robot at a next time step $t+1$, simply by knowing what it's position was at t. Additionally, we also have a series of measurements $\boldsymbol{z}_t$, captured by the on-board stereo camera, thus yielding an *Observation model*, which is also formulated probabilistically according to [1]:

$$\boldsymbol{z}_t = h(\boldsymbol{x}_t, \boldsymbol{v}_t) \sim p_h(\boldsymbol{z}_t|\boldsymbol{x}_t) \tag{4}$$

Here, $\boldsymbol{v}_t$ is the noise associated to the *Observation model*. Our objective is to estimate $\boldsymbol{x}_t$ and $\boldsymbol{z}_t$ simultaneously. In order to solve this problem we turn our attention to SLAM, which involves solving the following optimization problem [3]:

$$\min_{\boldsymbol{x}_{1:K}, \boldsymbol{m}} \sum_{t=1}^{K} ||\boldsymbol{z}_t - h(\boldsymbol{x}_t, \boldsymbol{m}_t)||_2^2 + \sum_{t=0}^{K-1} ||\boldsymbol{x}_{t+1} - f(\boldsymbol{x}_t, \boldsymbol{u}_t)||_2^2 \tag{5}$$

Here, h and f represent the *Observation* and the *Motion* model defined in (2) and (4), respectively. $\boldsymbol{m}_t$ represents the position of landmarks observed in the *World frame* $\{W\}$ at time t. As stated in the *Introduction*, we initially solve the *Localization* and *Mapping* part separately, which involves solving the first and second term of the cost function in (5) separately. We approach the SLAM problem by using the EKF, as stated in the *Introduction*.

Consider the general *Discrete-Time Non-Linear System* corrupted by *Gaussian* noise:

$$\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{w}_t), \ \boldsymbol{w}_t \sim \mathcal{N}(\mathbf{0}, W) \tag{6}$$

$$\boldsymbol{z}_t = h(\boldsymbol{x}_t, \boldsymbol{v}_t), \ \boldsymbol{v}_t \sim \mathcal{N}(\mathbf{0}, V) \tag{7}$$

In (6) and (7), f and h are non-linear functions in the state $\boldsymbol{x}_t$, but possibly linear in the input $\boldsymbol{u}_t$, and in the *Gaussian* noise $\boldsymbol{w}_t$, $\boldsymbol{v}_t$. W and V correspond to the covariance matrices of $\boldsymbol{w}_t$ and $\boldsymbol{v}_t$ respectively. Given a *prior* $\boldsymbol{x}_t|\boldsymbol{z}_{0:t}, \boldsymbol{u}_{0:t-1} \sim \mathcal{N}(\boldsymbol{\mu}_{t|t}, \Sigma_{t|t})$ which reflects an initial guess of the state $\boldsymbol{x}_0$ at time $t = 0$. The *Prediction* step of the EKF is given by [4]:

$$\boldsymbol{\mu}_{t+1|t} = f(\boldsymbol{\mu}_{t|t}, \boldsymbol{u}_t, \mathbf{0}) \tag{8}$$

$$\Sigma_{t+1|t} = F_t \Sigma_{t|t} F_t^T + Q_t W Q_t^T \tag{9}$$

Where $F_t = \frac{df}{d\boldsymbol{x}}(\boldsymbol{\mu}_{t|t}, \boldsymbol{u}_t, \boldsymbol{0})$ and $Q_t = \frac{df}{d\boldsymbol{w}}(\boldsymbol{\mu}_{t|t}, \boldsymbol{u}_t, \boldsymbol{0})$ are Jacobians. The *Update* step on the other hand, is given by [4]:

$$\boldsymbol{\mu}_{t+1|t+1} = \boldsymbol{\mu}_{t+1|t} + K_{t+1|t}(\boldsymbol{z}_{t+1} - h(\boldsymbol{\mu}_{t+1|t}, \boldsymbol{0})) \tag{10}$$

$$\Sigma_{t+1|t+1} = (I - K_{t+1|t}H_{t+1})\Sigma_{t+1|t} \tag{11}$$

$$K_{t+1|t} = \Sigma_{t+1|t}H_{t+1}^T(H_{t+1}\Sigma_{t+1|t}H_{t+1}^T + R_{t+1}VR_{t+1}^T)^{-1} \tag{12}$$

Where $H_{t+1} = \frac{dh}{d\boldsymbol{x}}(\boldsymbol{\mu}_{t+1|t}, \boldsymbol{0})$ and $R_{t+1} = \frac{dh}{d\boldsymbol{v}}(\boldsymbol{\mu}_{t+1|t}, \boldsymbol{0})$ are also Jacobians. (12) represents the *Kalman gain*, which tells us by how much our update changed over time. Our objective is to use the *Prediction* and the *Update* step of the EKF to perform SLAM.

III. Technical Approach

We approach the SLAM problem in the following manner:

- *Data Pre-Processing*
- *IMU localization via EKF prediction*
- *Landmark mapping via EKF update*
  - Landmark map Initialization
  - EKF Update Step
- *Visual-Inertial SLAM*

A. Data Pre-Processing

In this project, minimal data pre-processing was performed. We note that for sections involving the features collected by the stereo camera, we simply down-sample the amount of features to deal with, since the covariance matrix of the landmarks is of size $3M \times 3M$, where M represents the number of landmarks. We will only deal with every 20th observation, thus reducing the amount of computational workload. Additionally, since the data provided includes un-observed landmarks of the form $\boldsymbol{z}_{t,i} = [-1,\ -1,\ -1,\ -1]^T$ at some time t for an observation i, we exclude these points thus reducing computational workload even more.

B. IMU localization via EKF prediction

In this section we use the linear and angular velocity provided by the IMU data to estimate a pose T_t over T time stamps. In order to solve this *Localization* problem we use equations (8) and (9) of the EKF *Prediction* step.
Given a set of measurements for linear and angular velocity $\{\boldsymbol{v}_t,\ \boldsymbol{\omega}_t\}_{t=0}^T$, we wish to find a pose $T_t \in SE(3)$, where the set $SE(3)$ is defined as follows:

$$SE(3) = \{T = \begin{pmatrix} R(\theta) & \boldsymbol{x} \\ \boldsymbol{0}^T & 1 \end{pmatrix} \in \mathbb{R}^{4\times 4} | R \in SO(3),\ \boldsymbol{x} \in \mathbb{R}^3\}$$

By considering the set of inputs $\boldsymbol{u}_t = [\boldsymbol{v}_t,\ \boldsymbol{\omega}_t]^T$ over T time stamps and a prior of the form $T_t | \boldsymbol{z}_{0:t}, \boldsymbol{u}_{0:t-1} \sim \mathcal{N}(\boldsymbol{\mu}_{t|t}, \Sigma_{t|t})$, where $\boldsymbol{\mu}_{t|t} \in SE(3)$ and $\Sigma_{t|t} \in \mathbb{R}^{6\times 6}$, and after a series of theoretical considerations involving computing derivatives in the *lie-algebra* spaces, the EKF *Prediction* step, defined in (8) and (9) with $\boldsymbol{w}_t \sim \mathcal{N}(\boldsymbol{0}, W)$ become [2]:

$$\boldsymbol{\mu}_{t+1|t} = \boldsymbol{\mu}_{t|t} exp(\tau_t \hat{\boldsymbol{u}}_t) \tag{13}$$

$$\Sigma_{t+1|t} = exp(-\tau \overset{\curlywedge}{\boldsymbol{u}}_t)\Sigma_{t|t} exp(-\tau \overset{\curlywedge}{\boldsymbol{u}}_t)^T + W \tag{14}$$

Where $\hat{\boldsymbol{u}}_t$ represents the *hat map* of the vector $\boldsymbol{u}_t = [\boldsymbol{v}_t,\ \boldsymbol{\omega}_t]^T$ and it is given by:

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

$\overset{\curlywedge}{\boldsymbol{u}}_t$ is the adjoint of the *hap-map*, and it is given by:

$$\overset{\curlywedge}{\boldsymbol{u}}_t = \begin{pmatrix} \hat{\boldsymbol{\omega}}_t & \hat{\boldsymbol{v}}_t \\ 0 & \hat{\boldsymbol{\omega}}_t \end{pmatrix} \in \mathbb{R}^{6\times 6}$$

Since our formulation is inherently probabilistic, our poses over time T_t can be modeled as *Gaussian* random variables, and by using equations (13) and (14) over T time stamps we are essentially predicting the mean $\boldsymbol{\mu}_{t+1|t}$ of a *Gaussian* distribution in $SE(3)$ thus giving us a pose over T time stamps. It is also important to note that covariance matrix W has been initialized as $W = \epsilon I$, where $\epsilon \approx 10^{-3}$. A similar initialization scheme has also been employed when defining $\boldsymbol{\mu}_{0|0}$ and $\Sigma_{0|0}$.

C. Landmark mapping via EKF update

In this section, we discuss how we have used the *Update* step of the EKF to perform the *Mapping* part of SLAM. We assume that the poses estimated in the previous section are correct. Before diving into how the *Update* step has been performed, we will start by discussing how we have initialized the landmarks.

1) Landmark map Initialization: As we have discussed in the *Data Pre-Processing* section, we have down-sampled the number of landmarks and excluded un-observed landmarks at each time step t. The first time a landmark is seen, it's coordinates in the *Pixel frame* are projected into the *World frame* $\{W\}$.
Firstly, lets consider the stereo camera *Observation model* with measurement noise $\boldsymbol{v}_t \sim \mathcal{N}(\boldsymbol{0}, V)$(for observation i):

$$\boldsymbol{z}_{t,i} = K_s \pi({}_O T_{IMU} T_t^{-1} \underline{\boldsymbol{m}_j}) + \boldsymbol{v}_{t,i} \tag{13}$$

Where the function π is a projection of the form:

$$\pi(\boldsymbol{x}) = \frac{1}{\boldsymbol{e}_3^T \boldsymbol{x}} \boldsymbol{x},\ \boldsymbol{x} \in \mathbb{R}^3$$

The point $\underline{\boldsymbol{m}_j}$ represents the point $\boldsymbol{m}_j$ in *homogeneous* coordinates. The matrix ${}_O T_{IMU}$ specifies a transformation from the *IMU frame* to the *Optical frame*. All observations are stacked in a $4N_t$ vector, such that the model in (13) with measurement noise $\boldsymbol{v}_t \sim \mathcal{N}(\boldsymbol{0}, I \otimes V)$ ($\otimes$ is the *Kroneker* product) becomes:

$$\boldsymbol{z}_t = K_s \pi({}_O T_{IMU} T_t^{-1} \underline{\boldsymbol{m}}) + \boldsymbol{v}_t \tag{14}$$

It is also important to note that the landmark $\boldsymbol{m}_j$ ans its observation $\boldsymbol{z}_i$ are associated. In other words landmark $\boldsymbol{m}_j$ produces observation $\boldsymbol{z}_i$. Since the intrinsic matrix K_s in (14) is *rank-deficient*, it is not invertible. Therefore we solve the following system of equations such that the observation $\boldsymbol{m}_j$ has optical coordinates:

$$
\begin{aligned}
d &= u_L - u_R \\
z &= -f_{su}\frac{b}{d} \\
x &= (u_L - c_u)\frac{z}{f_{su}} \\
y &= (v_L - c_v)\frac{z}{f_{sv}} \qquad (15)
\end{aligned}
$$

Once the above equations (15) are solved, we project the coordinates $(x, y, z, 1)$ back to world coordinates for the initial estimate of the landmark positions. These are then updated only when the same landmark is observed for a second time.

2) EKF Update Step: In this section we discuss the EKF *Update Step* for *Mapping* purposes. We assume that the landmarks are static, therefore no *Prediction* step is needed. We consider the *Observation* model defined in (14) and a prior $\boldsymbol{m}|\boldsymbol{z}_{0:t} \sim \mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t)$ with $\boldsymbol{\mu}_t \in \mathbb{R}^{3M}$ and $\Sigma_t \in \mathbb{R}^{3M\times 3M}$. As it was discussed in the *Data Pre-Processing* section, the number of landmarks are down-sampled such that $\Sigma_t \in \mathbb{R}^{3M\times 3M}$ can be computed in a reasonable amount of time. Given an incoming observation $\boldsymbol{z}_{t+1} \in \mathbb{R}^{4N_t}$ and the equations (10), (11), (12), the EKF *Update* step is given by:

$$
\begin{aligned}
\boldsymbol{\mu}_{t+1} &= \boldsymbol{\mu}_t + K_{t+1}(\boldsymbol{z}_{t+1} - K_s\pi({}_OT_{IMU}T_t^{-1}\boldsymbol{\mu}_t)) \qquad (16)\\
\Sigma_{t+1} &= (I - K_{t+1}H_{t+1})\Sigma_t \qquad (17)\\
K_{t+1} &= \Sigma_t H_{t+1}^T(H_{t+1}\Sigma_t H_{t+1}^T + I\otimes V)^{-1} \qquad (18)
\end{aligned}
$$

For a landmark-observation pair (i, j) the *Observation* model Jacobian $H_{t+1} \in \mathbb{R}^{4N_t\times M}$ at time $t+1$ with block elements $H_{t+1,(i,j)} \in \mathbb{R}^{4\times 3}$ is given by (for $\Delta(j) = i$):

$$
H_{t+1,(i,j)} = \frac{\partial}{\partial \boldsymbol{m}_j} h(T_{t+1}, \boldsymbol{m}_j = \boldsymbol{\mu}_{t,j})
$$

The partial derivative gives us the following expression for the Jacobian, where $P = [I\ 0] \in \mathbb{R}^{3\times 4}$:

$$
H_{t+1,(i,j)} = K_s\frac{\partial \pi}{\partial \boldsymbol{q}}({}_OT_{IMU}T_t^{-1}\underline{\boldsymbol{m}_j})\,{}_OT_{IMU}T_t^{-1}P^T
$$

We note that the covariance matrix V for the measurement noise $\boldsymbol{v}_t$ is initialized as $V = \epsilon I$ for some small choice of ϵ.

3) Visual Inertial SLAM: In this section, we highlight that the key difference to the previous approaches, mentioned in *IMU localization via EKF prediction* and *EKF Update Step*, is that the vehicle's pose and landmarks are updated simultaneously. In other words, instead of assuming that the poses estimated in *IMU localization via EKF prediction* are correct, we use the latest, incoming observation from the stereo camera to both update our belief of the landmark positions, but also to re-calibrate our belief of the vehicle's true pose. We use an expanded covariance matrix that now contains cross-covariances between the robot's pose and landmark positions, whereas earlier the covariance matrix was separate for both. The joint covariance matrix is now of the form:

$$
\begin{pmatrix} \Sigma_{LL} & \Sigma_{LR} \\ \Sigma_{RL} & \Sigma_{RR} \end{pmatrix} \in \mathbb{R}^{(3M+6)\times(3M+6)}
$$

During the *Prediction* step, we re-formulate the predictions in a different manner. We wish to capture cross-correlations between the robot pose and landmarks. In order to do that we apply the linearized motion model Jacobian to the off-diagonal entries as follows (the bottom right block has the same form as in the *Localization* step):

$$
\Sigma^J = \begin{pmatrix} \Sigma_{LL} & \Sigma_{LR}F^T \\ F\Sigma_{RL} & F\Sigma_{RR}F^T + W \end{pmatrix}
$$

In the *Update* step, the process is similar to the *Mapping* section, but instead we have an expanded version for the Jacobian H and the Kalman gain K. The equations for the *Update* step are as follows (denote "landmark" as l and "robot" as r):

$$
\begin{aligned}
\boldsymbol{\mu}_{t+1|t+1,l} &= \boldsymbol{\mu}_t + K_{t+1}(\boldsymbol{z}_{t+1} - \tilde{\boldsymbol{z}_{t+1}}) \qquad (19)\\
\boldsymbol{\mu}_{t+1|t+1,r} &= \boldsymbol{\mu}_{t+1|t,r}exp(K_{t+1}(\boldsymbol{z}_{t+1} - \tilde{\boldsymbol{z}_{t+1}})) \qquad (20)\\
\Sigma^J_{t+1|t+1} &= (I - K_{t+1}H_{t+1})\Sigma^J_{t+1|t+1} \qquad (21)
\end{aligned}
$$

Where:

$$
\tilde{\boldsymbol{z}_{t+1,i}} = K_s\pi({}_OT_{IMU}\boldsymbol{\mu}^{-1}_{t+1|t,r}\boldsymbol{\mu}_{t,j,l}) \qquad (22)
$$

$$
H_{t+1,(i,j)} = K_s\frac{\partial \pi}{\partial \boldsymbol{q}}({}_OT_{IMU}\boldsymbol{\mu}^{-1}_{t+1|t,r}\boldsymbol{\mu}_{t,j,l})\,{}_OT_{IMU}(\boldsymbol{\mu}^{-1}_{t+1|t,r}\boldsymbol{\mu}_{t,j,l})^{\odot} \qquad (23)
$$

$$
K_{t+1} = \Sigma^J_{t+1|t+1}H^T_{t+1}(H_{t+1}\Sigma^J_{t+1|t+1}H^T_{t+1} + I\otimes V)^{-1} \qquad (24)
$$

In the above equations we define $H_{t+1} = [H_{t+1,l}\ H_{t+1,r}]$. Additionally, we have $H_{t+1,(i,j)} \in \mathbb{R}^{4N_t\times(3M+6)}$ and $K_{t+1} \in \mathbb{R}^{(3M+6)\times 4N_t}$. The operator "$\odot$" denotes the following:

$$
\begin{pmatrix} \boldsymbol{s} \\ 1 \end{pmatrix}^{\odot} = \begin{pmatrix} I & -\hat{\boldsymbol{s}} \\ 0 & 0 \end{pmatrix}
$$

IV. Results

We present the results in the following order for both data sets 10 and 3:

- *IMU localization via EKF prediction*
- *Landmark mapping via EKF update*
- *Visual-Inertial SLAM*
- *Kalman gain and Outliers*

1) IMU localization via EKF prediction: In this section we show the results obtained by performing the EKF *Prediction* step with the IMU data. Figure 1 illustrates the results for DataSet 10, and Figure 2 shows the results for DataSet 3.

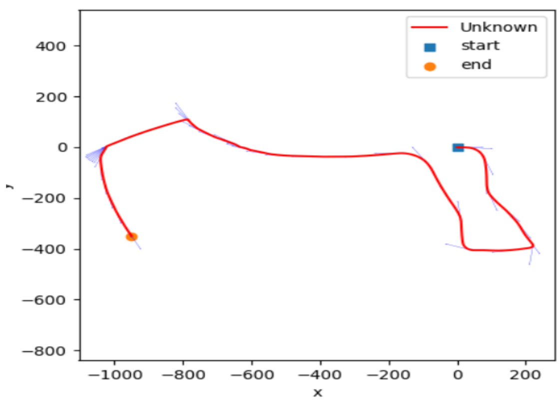

Fig. 1. DataSet 10 - IMU localization via EKF prediction

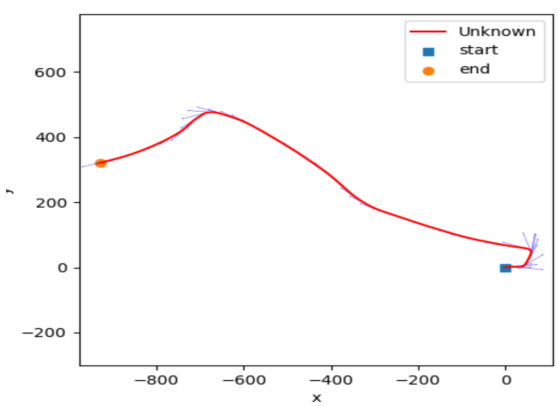

Fig. 2. DataSet 3 - IMU localization via EKF prediction

When comparing these plots with the ground truth data. We notice that the plots reflect well the trajectory that the vehicle undergoes. However, the final right turn before the car parks is missed in these plots. We will later see that when *Outlier* removal is implemented in SLAM, some of these details, such as some quick and sharp turns, will be somewhat visible.

2) Landmark mapping via EKF update: In this section we address the results obtained from the EKF *Update* step. As we have mentioned in the *Technical Approach* section, we initialize the landmarks according to the scheme described in the *Landmark map Initialization* section. Figure 3 and Figure 4 illustrate the results for DataSet 10 and DataSet 3 respectively.

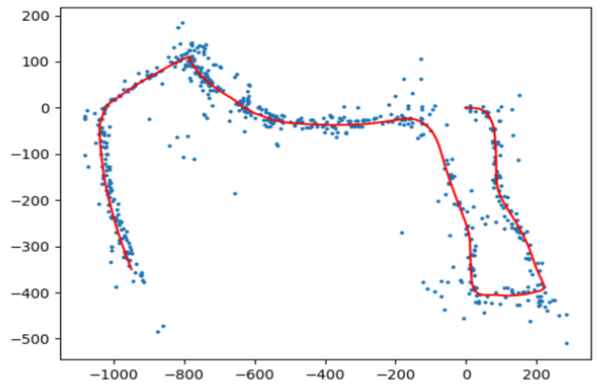

Fig. 3. DataSet 10 - Landmark mapping via EKF update

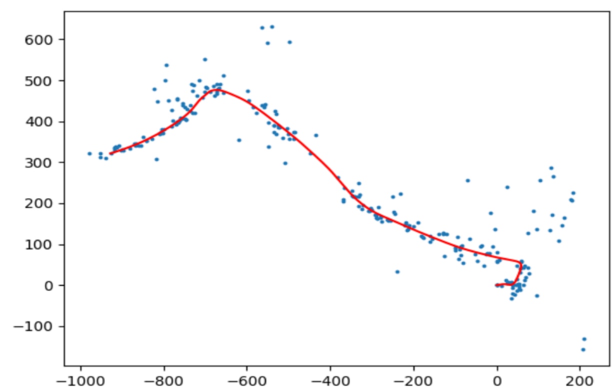

Fig. 4. DataSet 3 - Landmark mapping via EKF update

3) Visual-Inertial SLAM: In this section, we discuss the results obtained from performing the EKF *Prediction* and *Update* step simultaneously. We will also talk about *noise tuning* and how this aspect affected the results that we obtained. Firstly, we highlight the fact that the noise terms $\boldsymbol{w}_t$ and $\boldsymbol{v}_t$ added to both the *Motion* and *Observation* model, reflects the uncertainty in how we have modelled the Kinematics of the robot and how it taked measurements. Therefore, with this scheme by increasing the noise variance, it means that our uncertainty in the model also increases. Conversely, by decreasing the variance, it means that we trust our model. Recall, that the covariance matrices for both $\boldsymbol{w}_t$ and $\boldsymbol{v}_t$ are W and V. For DataSet 10, we define $W = \epsilon I$, for $\epsilon = 10^{-3}$, and $V = 5I$ (meaning that we trust our *Motion* model more than our *Observation* model). Figure 5 and Figure 6 illustrates this result for both DataSet 10 and 3 respectively.

As we can see from the above plots, if we trust the *Motion* model much more than the *Observation* model by a large

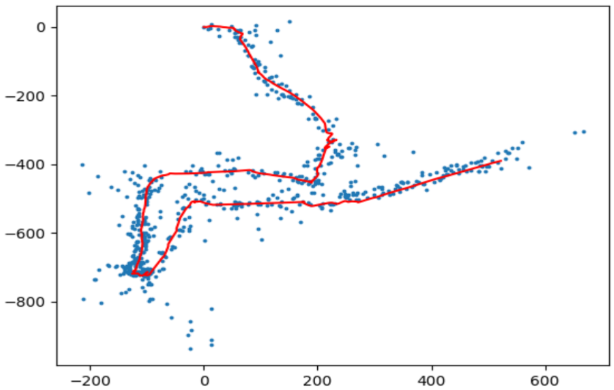

Fig. 5. Dataset 10 - Visual-Inertial SLAM, $W = 10^{-3}I,\ \ V = 5I$

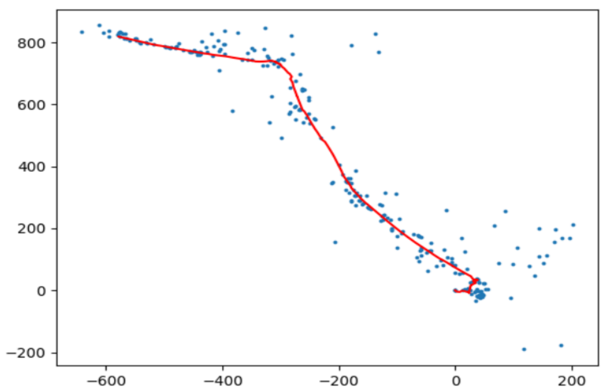

Fig. 6. Dataset 3 - Visual-Inertial SLAM, $W = 10^{-3}I,\ \ V = 5I$

margin, we can start to see that the trajectory of the vehicle will closely resemble to the trajectory of the vehicle in the *IMU localization via EKF prediction* section. This is especially clear in Figure 6. Therefore we can tune the noise a bit further such that there is no big mismatch between the two models. Suppose $W = 10^{-3}I$ and $V = I$ for both DataSet 10 and DataSet 3. Figure 7 and Figure 8 shows the results for both DataSet 10 and DataSet 3.

As we can see from the plot in Figure 7 and Figure 8, not much has changed for DataSet 3, but we notice that the trajectory for DataSet 10 starts to resemble Figure 1.
Now, if we trust out *Observation* model and our *Motion* model equally, we get the following results, as illustrated in Figure 9 and Figure 10 (for $W = 10^{-3}I,\ \ V = 10^{-4}I$). For Figure 10 we see that our trajectory completely changes. But, for DataSet 10 in Figure 9, we see that there is a change in trajectory, in the sense that the coordinates seem flipped, and there are some sudden turns taken during the path. However, at the end of the trajectory (always for Figure 9), we can see that the parking motion which is shown at the end of the video is somewhat illustrated.

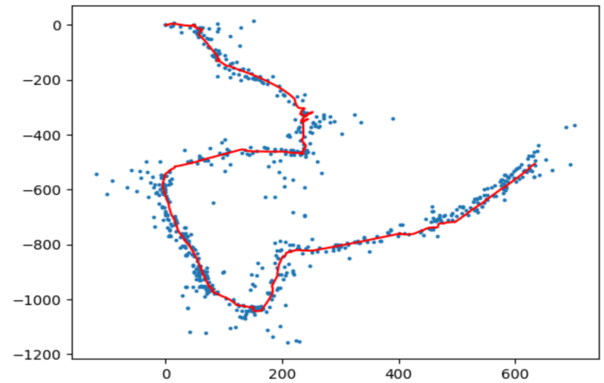

Fig. 7. Dataset 10 - Visual-Inertial SLAM, $W = 10^{-3}I,\ \ V = I$

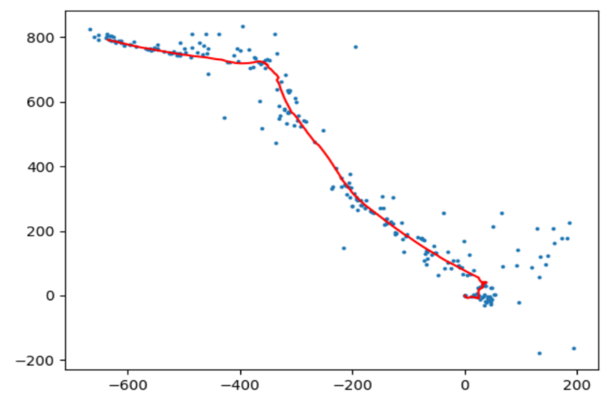

Fig. 8. Dataset 3 - Visual-Inertial SLAM, $W = 10^{-3}I,\ \ V = I$

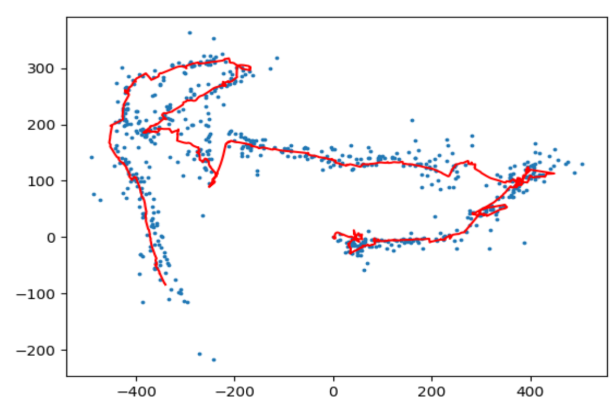

Fig. 9. Dataset 10 - Visual-Inertial SLAM, $W = 10^{-3}I,\ \ V = 10^{-4}I$

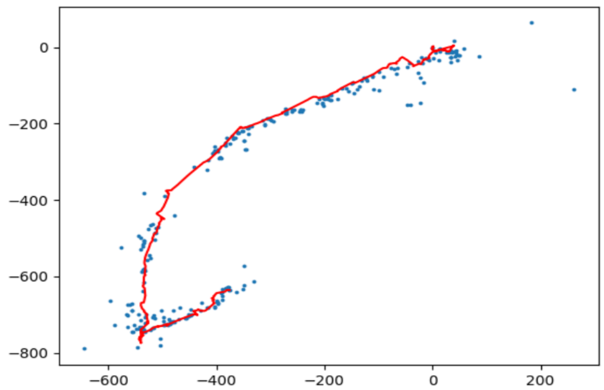

Fig. 10. Dataset 3 - Visual-Inertial SLAM, $W = 10^{-3}I,\ \ V = 10^{-4}I$

4) Kalman gain and Outliers: During this project, it is worth mentioning issues encountered when computing the *Kalman gain* matrix. During the matrix inversion process, errors involving *Singular* matrices were encountered. In order to get passed this issue we considered forcing the matrix $\Sigma_{t+1|t+1}$ to be *Positive-Semi-Definite*. We proceeded in doing the following at each iteration [3]:

$$\Sigma_{t+1|t+1} = \frac{(\Sigma_{t+1|t+1} + \Sigma_{t+1|t+1}^{T})}{2}$$

The above expression ensured that the *Kalman gain* matrix was computed with success.
A quick note on *Outliers*. We have considered the distance between a landmark in coordinates $\{W\}$ and the current robot's position, this distance is then compared to a threshold t defined as (where ”r” corresponds robot, and ”l” to landmark):

$$t = ||\boldsymbol{x}_{t,r} - \boldsymbol{x}_{t,l}||_2^2$$

If the distances are greater then t, then the landmark is discarded, unless it seen again and it is in between the threshold t.

References

[1] ”https://natanaso.github.io/ece276a/ref/ECE276A_7_BayesFilter.pdf”
[2] ”https://natanaso.github.io/ece276a/ref/ECE276A_5_LocalizationOdometry.pdf”
[3] ”https://stackoverflow.com/questions/29459158/numerical-instability-kalman-filter-in-matlab”
[4] ”https://natanaso.github.io/ece276a/ref/ECE276A/12/VI/SLAM.pdf”