Visual-Inertial SLAM

Renu Krishna Gutta, PID: A59018210, ECE 276A, Project-3

1. INTRODUCTION

The goal of this project is to implement the simultaneous localization and mapping (SLAM) using the IMU measurements and the visual images captured by a stereo camera of a moving car.

For an autonomous driving vehicle, it is important to identify the landmarks around and estimate its trajectory with respect to those landmarks, so that appropriate planning can be done for its motion. In this project, we have datasets with IMU measurements and a certain number of static features identified with their pixel locations in the stereo-camera images captured throughout the car's journey.

For the SLAM, I implemented an Extended Kalman Filter, where the car/robot pose and landmark positions were considered as random variables by introducing some noise.

2. PROBLEM FORMULATION

Following data is available:

a. **IMU measurements** – Provides linear velocities $v_t \in \mathbb{R}^3$ (pre-processed) and angular velocities $\omega_t \in \mathbb{R}^3$ along X, Y, Z directions.

b. **Visual feature measurements** – Features were identified from the stereo-camera images and corresponding left and right camera pixel values $z_t \in \mathbb{R}^{4\times M}$ were provided. M is the total number of features observed throughout the motion. Landmark *i* that was not observable at time *t* has a measurement of $z_{t,i} = [-1,-1,-1,-1]^T$

c. **Time stamps** – in UNIX time

d. **Intrinsic calibration** – stereo baseline *b* in meters and left camera calibration matrix;

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

e. **Extrinsic calibration** – ${}_I T_C \in SE(3)$, transformation from the left camera to IMU frame.

$$SE(3) \coloneqq \left\{ T = \begin{bmatrix} R & p \\ 0^T & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4} \,\middle|\, R \in SO(3), p \in \mathbb{R}^3 \right\}$$

2.1. Motion Model: Pose Kinematics

We use the discrete time kinematics for pose $T(t) \in SE(3)$. That is, assume $\xi(t) = \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix}$ is constant for $t \in [t_k, t_{k+1})$

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

where, $T_k = T(t_k)$, $\tau_k = t_{k+1} - t_k$, $\xi_k = \xi(t_k)$

Introduce noise:

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

$$T = \mu \exp(\hat{\epsilon}), \quad \epsilon \sim N(0, \Sigma)$$

$\mu \in SE(3)$ is the deterministic mean and $\epsilon \in \mathbb{R}^6$ is the noise added to the $\xi_k = [v_t^T, \omega_t^T]^T$

Now, we can split pose kinematics into nominal and perturbation kinematics:

Nominal: $\mu_{t+1} = \mu_t \exp(\tau_t \hat{u}_t), \quad u_t = [v_t^T, \omega_t^T]^T$

Perturbation: $\delta\mu_{t+1} = \exp(-\tau_t \hat{u}_t)\, \delta\mu_t + w_t$

$$\hat{u}_t \coloneqq \begin{bmatrix} \hat{\omega} & \hat{v} \\ 0 & \hat{\omega} \end{bmatrix} \in \mathbb{R}^{6\times 6}$$

Here w_t is the effect of noise that we introduced above, which is separated from nominal kinematics.

In summary,

Motion model: $T_{t+1} = f(T_t, u_t, w_t) \sim p_f(.|T_t, u_t)$

2.2. Observation Model: Stereo-camera

Measurement of landmark *i* at time step *t*;

$$z_{t,i} = h(T_t, m_j) + v_{t,i} \coloneqq K_s \pi({}_O T_I T_t^{-1} \underline{m}_j) + v_{t,i}$$

Measurement noise: $v_{t,i} \sim N(0, V)$

Homogeneous landmark world coordinates: $\underline{m}_j \coloneqq \begin{bmatrix} m_j \\ 1 \end{bmatrix}$, with $i = \Delta_t(j)$ at time *t*. (data association)

Projection function: $\pi(q) \coloneqq \frac{1}{q^3} q \in \mathbb{R}^4$

$K_s \in \mathbb{R}^{4\times 4}$: Intrinsic calibration matrix of the stereo-camera constructed using K and b.

In summary,

Observation model: $z_{t+1} = h(x_t, v_t) \sim p_h(.|x_t)$

where, $x_t \coloneqq \{T_t, \boldsymbol{m}\}$, i.e., state is defined as combined robot pose and landmarks locations.

2.3. IMU Localization via EKF Prediction

Assumptions:

- Known world-frame landmark coordinates $m \in \mathbb{R}^{3M}$
- The data association $\Delta_t: \{1,2,\ldots,M\} \to \{1,2\ldots,N_t\}$ stipulating that landmark *j* corresponds to observation $z_{t,i} \in \mathbb{R}^4$ with $i = \Delta_t(j)$ at time *t*.

Objective: Given IMU measurements $u_{0:T}$ with $u_t := [v_t^T, \omega_t^T]^T \in \mathbb{R}^6$ and feature observations $z_{0:T}$, **predict** the IMU poses $T_t := {}_WT_{I,t} \in SE(3)$ using Extended Kalman Filter Prediction.

$$Predict\ pdf: p_{t+1|t}(T_{t+1}) := p(T_{t+1}|z_{0:t}, u_{0:t})$$
$$p_{t+1|t}(T_{t+1}) = \int p_f(T_{t+1}|s, u_t) p_{t|t}(s) ds$$

2.4. Landmark mapping via EKF Update

Assumptions:

- The IMU pose $T_t := {}_WT_{I,t} \in SE(3)$ is known.
- The data association $\Delta_t: \{1,2,\dots,M\} \to \{1,2\dots,N_t\}$ stipulating that landmark *j* corresponds to observation $z_{t,i} \in \mathbb{R}^4$ with $i = \Delta_t(j)$ at time *t*.
- Landmarks $m \in \mathbb{R}^{3M}$ are static. No motion model or EKF prediction needed.

Objective: Given set of observations $z_t := [z_{t,1}^T, \dots, z_{t,N_t}^T]^T \in \mathbb{R}^{4N_t}\ for\ t = 1,2,\dots,T$, and robot pose T_t is known estimate/**update** the coordinates $\boldsymbol{m} := [m_1^T, \dots, m_M^T]^T \in \mathbb{R}^{3M}$ of the landmarks that generated them, using the Extended Kalman Filter Update.

$$Update\ pdf: p_{t|t}(\boldsymbol{m}_t) := p(\boldsymbol{m}_t|z_{0:t}, u_{0:t-1})$$
$$p_{t|t}(\boldsymbol{m}_t) = \frac{p_h(z_t|\boldsymbol{m}_t, T_t) p_{t|t-1}(T_t)}{\int p_h(z_t|s) p_{t|t-1}(s) ds}$$

2.5. Visual-Inertial SLAM

Input:

- Linear velocities and angular velocities.
- Camera: Features

Assumption: The transformation ${}_OT_I \in SE(3)$ from the IMU to the camera optical frame (extrinsic parameters) and the stereo camera calibration matrix K_s (intrinsic parameters) are known.

Output:

- World-frame IMU pose ${}_WT_{I,t} \in SE(3)$ over time
- World-frame coordinates $m_j \in \mathbb{R}^3$ of the $j = 1,2,\dots,M$ point landmarks that generated the visual features $z_{t,i} \in \mathbb{R}^4$

SLAM using Extended Kalman Filter:

Localization: EKF Predict

Predict the world-frame IMU pose.

Landmark locations are assumed to be static. Therefore, no prediction step is required for those, and this step is same as sec.2.3.

$$Predict\ pdf: p_{t+1|t}(T_{t+1}) := p(T_{t+1}|z_{0:t}, u_{0:t})$$
$$p_{t+1|t}(T_{t+1}) = \int p_f(T_{t+1}|s, u_t) p_{t|t}(s) ds$$

Mapping: EKF Update

Assumptions:

- The data association $\Delta_t: \{1,2,\dots,M\} \to \{1,2\dots,N_t\}$ stipulating that landmark *j* corresponds to observation $z_{t,i} \in \mathbb{R}^4$ with $i = \Delta_t(j)$ at time *t*.
- Landmarks $m \in \mathbb{R}^{3M}$ are static.

Objective: Given set of observations $z_t := [z_{t,1}^T, \dots, z_{t,N_t}^T]^T \in \mathbb{R}^{4N_t}\ for\ t = 1,2,\dots,T$, **update** the predicted world-frame IMU pose ${}_WT_{I,t} \in SE(3)$ and the coordinates $m := [m_1^T, \dots, m_M^T]^T \in \mathbb{R}^{3M}$ of the landmarks that generated them, using the Extended Kalman Filter Update.

$$Update\ pdf: p_{t|t}(x_t) := p(x_t|z_{0:t}, u_{0:t-1})$$
$$p_{t|t}(x_t) = \frac{p_h(z_t|x_t) p_{t|t-1}(x_t)}{\int p_h(z_t|s) p_{t|t-1}(s) ds}$$

where, $x_t := \{T_t, \boldsymbol{m}\}$, i.e., state is defined as combined robot pose and landmarks locations.

3. TECHNICAL APPROACH

IMPLEMENTATION OF VISUAL-INERTIAL SLAM

Following is the technical approach to visual-inertial SLAM

3.1.Prior

$$x_t |\ z_{0:t}, u_{0:t-1} \sim \mathcal{N}(\mu_{t|t}, \Sigma_{t|t})$$

We consider the state at time *t* as the collection of both robot's pose and the world-frame locations of all the *M* features at that instant.

Hence, the following symbol representation:

R: Robot, L: Landmarks

State:

$$x_t \equiv \left\{ {}_WT_{I,t} \in SE(3),\quad m := [m_1^T, \dots, m_M^T]^T \in \mathbb{R}^{3M} \right\}$$

State Mean, $\mu_{t|t} \equiv \left\{ \mu_{t|t}^{(R)} \in SE(3),\ \mu_{t|t}^{(L)} \in \mathbb{R}^{3M} \right\}$

State Covariance,

$$\Sigma_{t|t} \equiv \begin{bmatrix} \Sigma_{t|t,L} & \Sigma_{t|t,LR} \\ \Sigma_{t|t,RL} & \Sigma_{t|t,R} \end{bmatrix} \in \mathbb{R}^{(3M+6)\times(3M+6)}$$

where, $\Sigma_{t|t,L} \in \mathbb{R}^{3M\times 3M}$, $\Sigma_{t|t,R} \in \mathbb{R}^{6\times 6}$

Note: {., .} denotes just a collection.

Important: Prediction and Update for state mean are done separately for pose and landmarks, whereas for the state covariance they are done together on a single matrix $\Sigma_{t|t}$ containing both covariances.

Initialization: (t=0)

- **<u>Initialize pose mean:</u>** Assume robot is at origin with axes aligned with that of the world frame,

$$\mu_{0|0}^{(R)} = I_{4\times 4}\ (Identity\ matrix)$$

- **Initialize pose covariance:** The pose covariance is a diagonal matrix with variances for each component of v_t and ω_t, assuming no correlation between noises in the linear and angular velocities.
  σ_v^2 = 1m/s, σ_ω^2 = 0.001 rad/s (These are hyper parameters, can be tuned later)
  $$\Sigma_{0|0,R} = diag(\sigma_v^2, \sigma_v^2, \sigma_v^2, \sigma_\omega^2, \sigma_\omega^2, \sigma_\omega^2) \in \mathbb{R}^{6\times6}$$

- **Initialize landmark mean:** Look at the first feature measurement z_0.
  Discard those with -1's which means they are not observed.
  Initialize the observed landmark locations with the world coordinates obtained by transforming pixel coordinates with the inverse observation model. This is a fair way of initializing as we do not have any other information.
  Reject the outliers: Reject the landmark locations that very far from current robot position, using Euclidean distance.
  Data association is straight forward, $i = \Delta_t(j) = j$.

  N_0, be the number of features observed after discard and outlier removal. Their world-frame locations are initialized using the inverse observation model and corresponding indices are filled in the $\mu_{t|t}^{(L)} \in \mathbb{R}^{3M}$ matrix.
  Observe that, M is the total number of landmarks that we are interested in.
- **Initialize landmark covariance:** Assume an uncertainty of *1m* in the landmark location. This is also a hyper parameter which can be tuned later.
  $$\Sigma_{0|0,L} = I_{3M\times3M}\ (Identity\ matrix)$$
- $\boldsymbol{\Sigma_{t|t,LR}}$ **and** $\boldsymbol{\Sigma_{t|t,RL}}$ are initially zero matrices. Later they get filled with non-zero values due to the correlation between pose and landmarks.

3.2.Motion Model

$$x_{t+1} = f(x_t, u_t, w_t), \qquad w_t \sim \mathcal{N}(0, W)$$
$$F_t \coloneqq \frac{df}{dx}(\mu_{t|t}, u_t, 0), \quad Q_t \coloneqq \frac{df}{dw}(\mu_{t|t}, u_t, 0)$$

Motion model is required in the EKF Predict step. Only the robot pose requires prediction. The landmarks do not require this as their locations are assumed to be static. They just require EKF update based on the observations.
Hence we write the motion model specific to the robot pose with the help of Pose Kinematics and since the robot pose belongs to SE(3) which is an embedded submanifold of the Euclidean space as well as a matrix group, we employ lie algebra to perform perturbation and calculate Jacobians.

Jacobians of motion model:

$$F_t = \exp\left(-\tau_t \hat{u}_t\right) \in \mathbb{R}^{6\times6}$$
$$Q_t = I_{6\times6}$$

3.3.Observation Model

$$z_t = h(x_t, v_t), \quad v_t \sim \mathcal{N}(0, V)$$
$$H_t \coloneqq \frac{dh}{dx}(\mu_{t|t-1}, 0), \quad R_t \coloneqq \frac{dh}{dv}(\mu_{t|t-1}, 0)$$

Here observation at time *t* is the collection of left and right camera pixel locations of the landmarks.
N_t is the number of valid observations at time *t* after discarding unobserved ones and outliers with respect to current nominal robot pose $\mu_{t|t-1}^{(R)}$.
(Data association is straight forward, $i = \Delta_t(j) = j$)

For $i = 1,2,\dots.N_t$;
$$z_{t,i} = h(T_t, m_i) + v_{t,i} \coloneqq K_s\pi\left({}_OT_I T_t^{-1}\underline{m}_j\right) + v_{t,i}$$
where, $T_t = \mu_{t|t-1}^{(R)}$

Note: Measurement noise: $v_{t,i} \sim N(0, \mathrm{V})$. Noise in pixel is taken as 2-pixel standard deviation, i.e., variance = 4. This is a hyper parameter and be tuned. $V = 4I$

All observations are stacked as $4N_t$ vector, at time *t*

$$\boldsymbol{z_t} = K_s\pi\left({}_OT_I T_t^{-1}\underline{\boldsymbol{m}}\right) + \boldsymbol{\vartheta_t}$$

$\boldsymbol{\vartheta_t} \sim \mathcal{N}(0, I \otimes V)$, $\otimes$: Kronecker product
$\boldsymbol{z_t}, \underline{\boldsymbol{m}}, \boldsymbol{\vartheta_t} \in \mathbb{R}^{4\times1}$

For the Jacobians computation, we need the derivative of the projection function which is:

$$\frac{d\pi}{d\boldsymbol{q}}(\boldsymbol{q}) = \frac{1}{q_3}\begin{bmatrix} 1 & 0 & -\frac{q_1}{q_3} & 0 \\ 0 & 1 & -\frac{q_2}{q_3} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{q_4}{q_3} & 1 \end{bmatrix} \in \mathbb{R}^{4\times4}$$

Jacobians of observation model:

$$H_t \coloneqq \frac{dh}{dx}(\mu_{t|t-1}, 0)$$

To compute H_t:
Let $T_t = {}_WT_{I,t} = \mu_{t|t-1}^{(R)}$, current predicted robot pose.
State $x_t \equiv \{ {}_WT_{I,t} \in SE(3),\ m \coloneqq [m_1^T, \dots, m_M^T]^T \in \mathbb{R}^{3M}\}$ is the collection of robot pose and the landmark locations. Hence the Jacobian with respect to state comprises two blocks as below:

$$H_t = \left[H_t^{(L)}, H_t^{(R)}\right] \in \mathbb{R}^{4N_t\times(3M+6)}$$

i. Computing $H_t^{(L)}$

$$H_t^{(L)} = \frac{dh}{d\boldsymbol{m}}(T_t, \underline{\boldsymbol{m}}) \in \mathbb{R}^{4N_t\times3M}$$

$H_t^{(L)}$ can be seen as a block matrix of size $N_t \times M$ and each element is a matrix $H_{t,i,j}^{(L)} \in \mathbb{R}^{4\times3}$

$$H_{t,i,j}^{(L)} = \begin{cases} \frac{\partial}{\partial m_j}h\left(T_t, \underline{m}_j\right) & , if\ \Delta_t(j) = i \\ 0 & , otherwise \end{cases}$$

evaluated at $T_t = \mu_{t|t-1}^{(R)}, m_j = \mu_{t|t-1,j}^{(L)}$

On further simplification:

$$H_{t,i,j}^{(L)} = \begin{cases} K_s \frac{d\pi}{d\boldsymbol{q}}\left({}_OT_I T_t^{-1} \underline{\mu}_{t|t-1,j}^{(L)} \right) {}_OT_I T_t^{-1} P^T & , if \Delta_t(j) = i \\ 0 & , otherwise \end{cases}$$

where $P = [I\ 0] \in \mathbb{R}^{3\times4}$, $j = i = 1,2, \dots N_t$

Hence the diagonal block matrices in $H_t^{(L)}$ will get updated as above.

ii. Computing $H_t^{(R)}$

$$H_t^{(R)} = \frac{dh}{dT_t}(T_t, \underline{\boldsymbol{m}}) \in \mathbb{R}^{4N_t\times6}$$

$H_t^{(R)}$ can be seen as a vector of block matrices of 4x6 size.

That is, $H_t^{(R)} = \begin{bmatrix} H_{t,1}^{(R)} \\ \vdots \\ H_{t,N_t}^{(R)} \end{bmatrix}$

where $H_{t,j}^{(R)} \in \mathbb{R}^{4\times6}\ for\ j = 1,2 \dots, N_t$

$$H_{t,j}^{(R)} = -K_s \frac{d\pi}{d\boldsymbol{q}}\left({}_OT_I \mu_{t|t-1}^{(R)-1} \underline{m}_j \right) {}_OT_I \left(\mu_{t|t-1}^{(R)-1} \underline{m}_j\right)^{\odot}$$

where, $\begin{bmatrix} S \\ 1 \end{bmatrix}^{\odot} \coloneqq \begin{bmatrix} I & -\hat{s} \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{4\times6}$

iii. Jacobian with respect to noise:

$$R_t \coloneqq \frac{dh}{d\vartheta}(\mu_{t|t-1}, 0) = I_{4N_t\times4N_t}\ (identity)$$

3.4.EKF Predict

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

As mentioned before, the prediction is for robot pose only and not for the landmark locations. Derivatives for the pose kinematics model are derived by introducing a perturbation and then approximating the Taylor series expansion. We get the prediction equations as follows:

$$\mu_{t+1|t}^{(R)} = \mu_{t|t}^{(R)} \exp(\tau_t \widehat{u_t})$$

$$\Sigma_{t+1|t} = \mathbb{E}\left[\delta\mu_{t+1|t}^{(R)} \delta\mu_{t+1|t}^{(R)T}\right] = \begin{bmatrix} \Sigma_{t|t,L} & \Sigma_{t|t,LR} F_t^T \\ F_t \Sigma_{t|t,RL} & F_t \Sigma_{t|t,R} F_t^T + W \end{bmatrix}$$

where,

$F_t = \exp(-\tau_t \widehat{u_t})$

$u_t = \begin{bmatrix} v_t \\ \omega_t \end{bmatrix} \in \mathbb{R}^6$,

$\widehat{u_t} = \begin{bmatrix} \widehat{\omega}_t & v_t \\ 0^T & \widehat{\omega} \end{bmatrix} \in \mathbb{R}^{4\times4}$, $\widehat{u_t} \coloneqq \begin{bmatrix} \widehat{\omega}_t & \hat{v}_t \\ 0 & \widehat{\omega}_t \end{bmatrix} \in \mathbb{R}^{6\times6}$

3.5.EKF Update

Kalman Gain:

$$K_{t+1} \coloneqq \Sigma_{t+1|t} H_{t+1}^T (H_{t+1} \Sigma_{t+1|t} H_{t+1}^T + R_{t+1} V R_{t+1}^T)^{-1}$$

Update:

$$\mu_{t+1|t+1} = \mu_{t+1|t} + K_{t+1}(z_{t+1} - h(\mu_{t+1|t}, 0))$$
$$\Sigma_{t+1|t+1} = (I - K_{t+1} H_{t+1}) \Sigma_{t+1|t}$$

Compute Kalman Gain:

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

Compute predicted observations:

$$\tilde{z}_{t+1,i} = K_s \pi\left({}_OT_I \mu_{t+1|t}^{(R)-1} \underline{m}_i \right)\ \ for\ i = 1,2,.., N_t$$

Update for pose mean:

$$\mu_{t+1|t+1}^{(R)} = \mu_{t+1|t}^{(R)} \exp((K_{t+1}(\boldsymbol{z}_{t+1} - \tilde{\boldsymbol{z}}_{t+1}))^\wedge)$$

Update for landmark means:

$$\mu_{t+1|t+1}^{(L)} = \mu_{t+1|t}^{(L)} + K_{t+1}(\boldsymbol{z}_{t+1} - \tilde{\boldsymbol{z}}_{t+1})$$

Combined update for pose and landmark covariance:

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

4. RESULTS

4.1.Plots

Case-1: Variances (velocity = 1, angular velocity = 0.0001, pixel variance = 25)

- **Dataset – 3:**

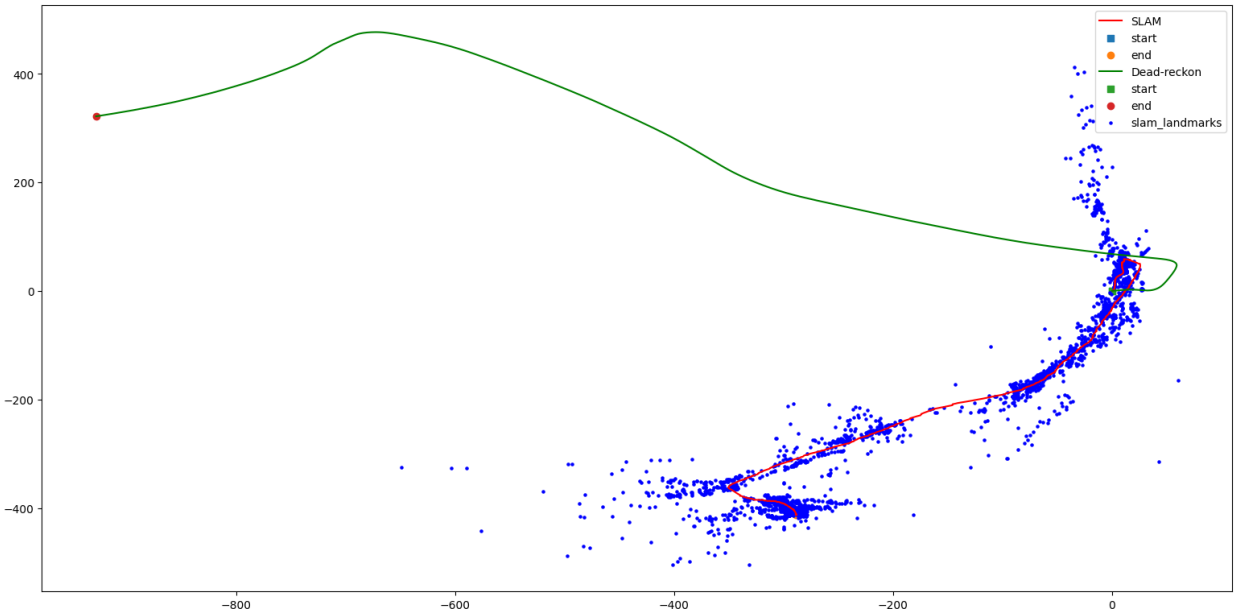

Figure 4.1.1: Plot for dataset-3 for case-1

- **Dataset – 10:**

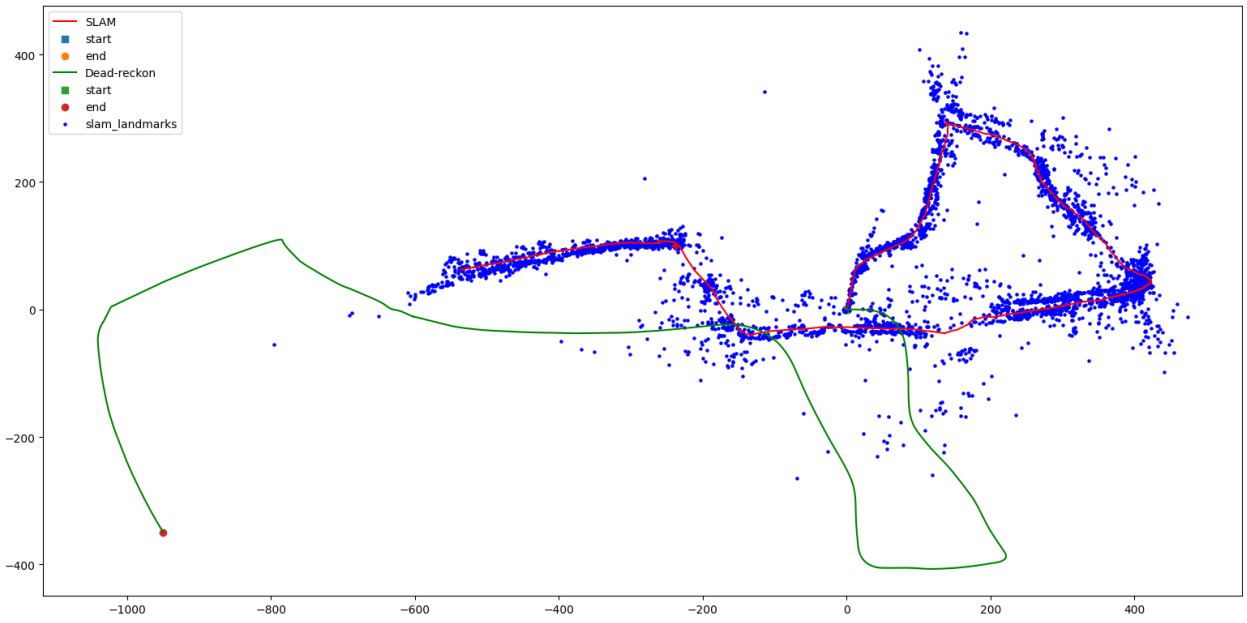

Figure 4.1.2: Plot for dataset-10 for case-1

Case-2: Variances (velocity = 2, angular velocity = 0.001, pixel variance = 25)

- **Dataset – 3:**

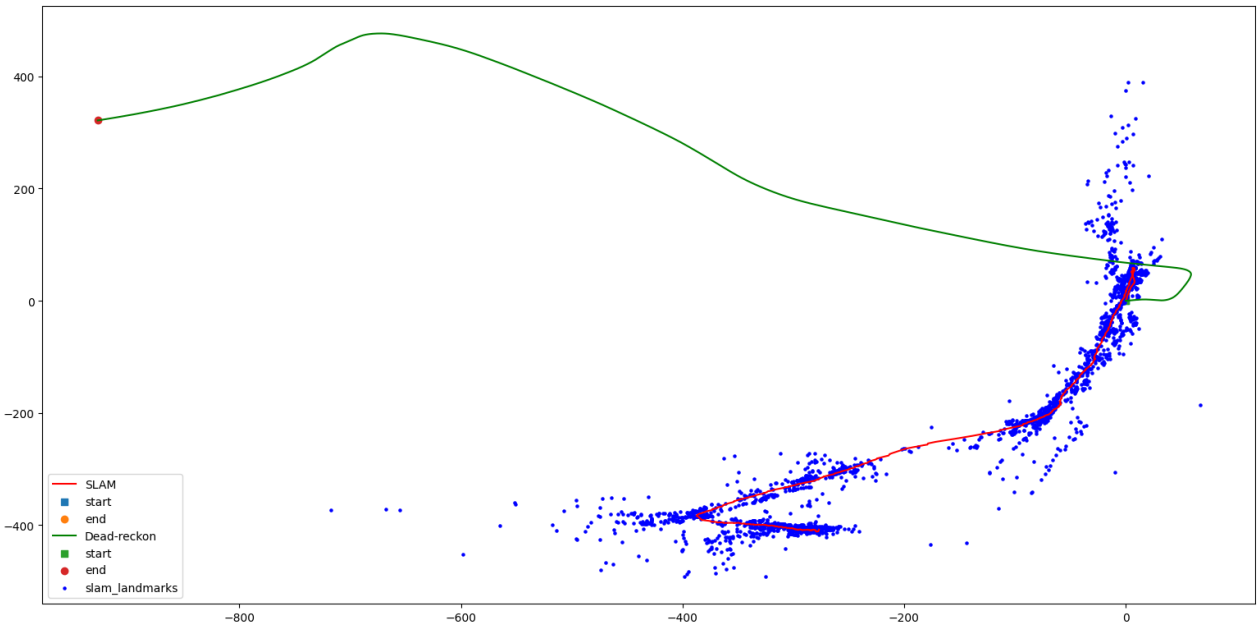

Figure 4.2.1: Plot for dataset-3 for case-2

- **Dataset – 10:**

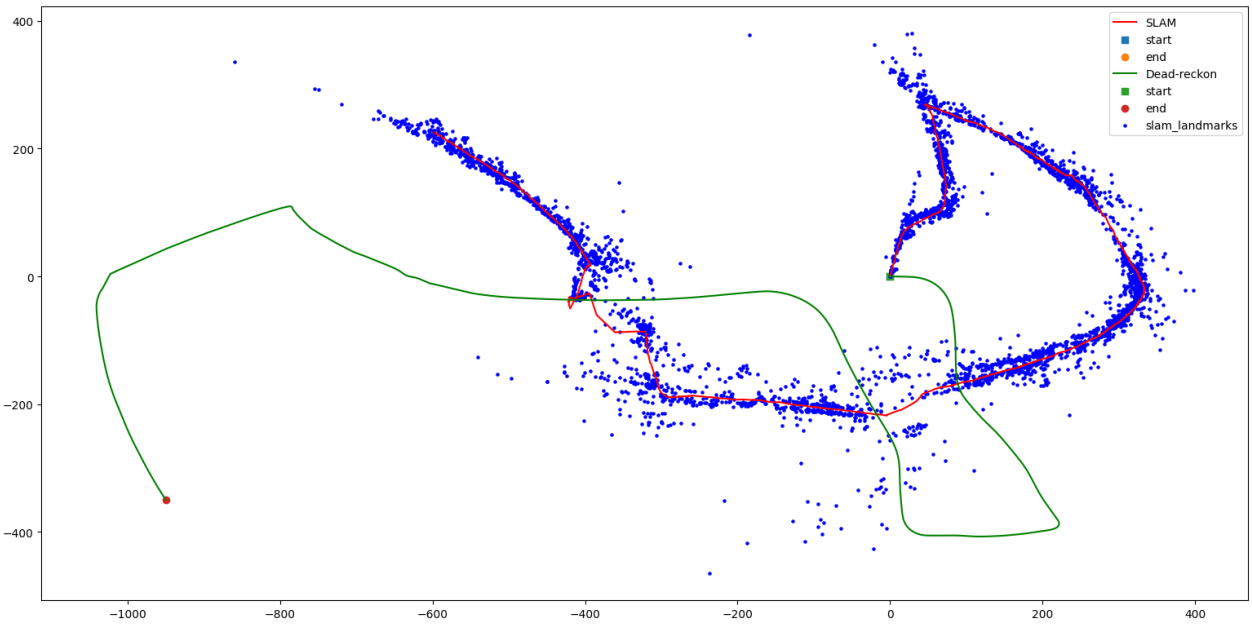

Figure 4.2.2: Plot for dataset-10 for case-2

4.2.Discussion

- For dataset-3, I discarded every one feature alternatively and for the dataset-10, I discarded every 2 features alternatively.

- All the matrix computations were done using scipy sparse library which reduced the computation time by many folds.
- The SLAM estimated trajectories are completely different from the dead-reckoning. In the dead-reckoning, there was no noise involved and trajectory was computed using kinematics equations. Whereas, in SLAM we introduced noise to the kinematics and then corrected the estimates based in the observations.
- In real life scenario, motion is not governed absolutely by kinematics. There are many factors like drag, friction, etc., that cause deviation from the kinematics equations. Hence, it is always safe to estimate the trajectory in a probabilistic sense and using the observations to strengthen the hypothesis.
- Coming to the results, I have tested for different noise levels in the inputs – linear and angular velocities. The estimated paths were very different across different noise parameters.
- Ground truth data would help in checking accuracy. However, we need to study further on accuracy analysis.

ACKNOWLEDGEMENT

I would like to thank Prof. Nikolay Atanasov, TAs – Sambaran Ghosal and Shrey Kansal for guiding me with this project.

REFERENCES

[1] https: //natanaso.github.io/ece276a
[2] https://docs.scipy.org/doc/scipy/reference/sparse.html