SLAM using EKF

Harsh Kishorsingh Thakur
Department of Electrical and Computer Engineering
University of California, San Diego
La Jolla, U.S.A.
HThakur@ucsd.edu

Nikolay Atanasov
Department of Electrical and Computer Engineering
University of California, San Diego
La Jolla, U.S.A.
natanasov@eng.ucsd.edu

I. Introduction

Nowadays, robots have become an integral part of society. One of the most common robots that we can see around and everyone is familiar with it is autonomous cars or TESLA. So, we need a map of the world and important features to distinguish things from the environment. Things like traffic lights, traffic signs, trees, other cars, etc. In order to get this, we require something that can map and estimate the trajectory of the car.

This is where **Simultaneous Localization And Mapping (SLAM)** comes into the picture. The basic idea of SLAM is to predict the next location of the robot which is accurate enough to resemble the current world. There are multiple ways in which this can be done those are Rao-Blackwellized Particle Filter, Kalman Filter, Factor Graphs SLAM, Fast SLAM[1], Kinect Fusion. All this method helps us to build a map of the environment. Now, these maps can either be Sparse or Dense depending on our requirement. Point cloud maps, landmark-based, and surfels are some examples of sparse maps. While on the other hand, there are two map types in the dense map, i.e. Implicit surface model, e.g. Occupany map and distance-based map and explicit surface models, i.e. polygon mesh.

Now my goal is to make an autonomous car that has Stereo Camera, and an IMU sensor. Using the stereo camera I can get features of the world and using the IMU I can estimate my location. Thus combining the best of IMU and stereo cameras I can localize and estimate the trajectory of my autonomous car.

II. Problem Formulation

2.1 Stereo and IMU Configuration:

Given IMU is located inside the car. IMU calculated the linear and angular speed of the car. Thus, I also know where my stereo camera is. The features can be obtained from stereo and they are directly given to us. In Fig 1, we can see the edges of the things. Edges mean where there is a sudden change in the color shade. For eg. black to white.

As the trajectory of the car is big all, the landmarks cannot be seen at one timestamp. So, in order to distinguish which landmarks are seen at a particular time, we define $\mathbf{z}_{t,i} = \begin{bmatrix} -1 & -1 & -1 & -1 \end{bmatrix}^\top$, if the features are not visible, it will give [-1, -1, -1, -1]. Thus distinguishing current features from other features.

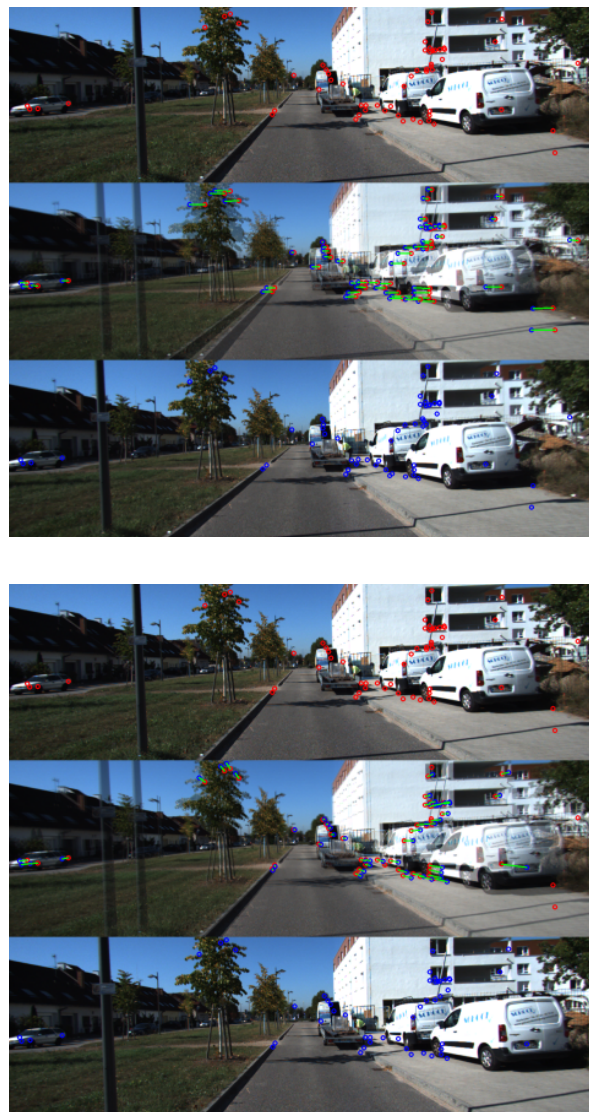

Fig1. Visual features matched across the left-right camera frames (top) and across time (bottom).

Visual features $z_{i,t}$ give us the landmark particle at time t and I represent which feature it represents. As the IMU and Stereo camera is at different location, I need a transformation from IMU to World. Thus the transformation ${}_{o}T_{I} \in SE(3)$ intrinsic parameters i.e. transformation from IMU to Optical frame and the Calibration matrix Ks is given by

$$K_s := \begin{bmatrix} fs_u & 0 & c_u & 0 \\ 0 & fs_v & c_v & 0 \\ fs_u & 0 & c_u & -fs_u b \\ 0 & fs_v & c_v & 0 \end{bmatrix} \qquad \begin{aligned} f &= \text{focal length } [m] \\ s_u, s_v &= \text{pixel scaling } [pixels/m] \\ c_u, c_v &= \text{principal point } [pixels] \\ b &= \text{stereo baseline } [m] \end{aligned}$$

Thus using this transformation, I can go from Stereo Coordinates to World Coordinates and vice versa.

2.2 Simultaneous Localization And Mapping (SLAM)

There is always some kind of noise in the environment. For example, noise in the IMU or in general inaccurate measurements given by sensors. Thus the whole trajectory becomes probabilistic. And this is defined by the probabilistic Makov assumption.

Thus SLAM comes into the picture. Let the robot position, control unit, and noise at time t be x_t, u_t, and w_t respectively. The next predicted step becomes:

$\mathbf{x_{t+1} = f(x_t, u_t, w_t) \sim pf(\cdot|x_t, u_t)}$, where and w_t is the motion noise and the equation gives us the motion of the robot. It can either be written as a linear/non-linear function or can be written as a probabilistic model.

Now once we know the next predicted step of our vehicle, we can observe the environment with the camera and the LiDAR sensor data. And, eventually, try to correlate with the previous map that I have created. Let the observation and noise at time t be z_t and v_t respectively. Thus our observation model is $\mathbf{z_t = h(x_t, m, v_t) \sim ph(\cdot|x_t, m)}$, where m is the map until time t. Where v_t is the observation noise of the map.

2.3 Bayesian Inference and Filtering

Now, we know the motion model and the observation model how do we calculate x_{t+1} and z_t. This is where Bayes Filter comes in place.

There are basically two steps involved in this:

1. Given robot position and control input what will be the next position? **Prediction**
2. Given the position and previous observation, what is the new observation? **Update**

Prediction:

For prediction given the prior pdf or pmf, in our case although it is kind of continuous we take a small difference in time and try to make it discrete. Thus given a pmf at time t i.e. $p_{t|t}$ we try to compute $p_{t+1|t}$. Thus we get the equation

$\mathbf{p_{t+1|t}(x_{t+1}) = \int pf(x_t| s, u_t) p_{t|t}(s)ds.}$

Update:

For update state, we get a new observation, and thus in that case using the updated position of my vehicle in predict step, and the new observation we try to correct our position to correct location.

$$\mathbf{p_{t+1|t+1}(x_{t+1}) = \frac{ph(z_{t+1}|x_{t+1})\, p_{t+1|t}(x_{t+1})}{\int ph(z_{t+1}|s)\, p_{t+1|t}(x_t)\, ds}}$$

2.4 Mapping of Landmark

Here we have M static landmarks which are not moving in the world frame. But, the stereo camera of the autonomous car is picking up this landmark. My aim is to accurately map this landmark based on the motion model and observation model. $M \in R^{3M}$. Each landmark will have $[x_i, y_i, z_i]$ coordinates. As I will be plotting in 2D matrix form, I will only use x and y coordinates. Z coordinates only give the height at which a particular landmark is visible.

Thus, in the mapping problem, I will localize and find the location of the landmarks z_i in the world coordinates. Thus the observation model is $\mathbf{p(z_t \mid x_t, m)}$ and here we will only considere those landmarks visible at a particular time.

III. Technical Approach

3.1 Extended Kalman Filter:

This problem of plotting the landmark and finding the important features is now non-linear because of the noise. There are even multiple assumptions in the Extended Kalman Filter:

1. The prior pdf $p_{t|t}$ is Gaussian.
2. The motion model is non-linear in the state x_t with Gaussian noise w_t.
3. The observation model is non-linear in the state x_t with Gaussian noise v_t.
4. The motion noise w_t and observation noise v_t are independent of each other, of the state x_t, and across time.
5. The predicted and updated pdfs are forced to be Gaussian via approximation.

If I try to solve this situation in a similar way as Linear Kalman Filter, the predicted and updated pdfs will not be Gaussian anymore, thus it cannot be solved in closed form.[2] Hence, Momen Matching is helpful in forcing the predicted and updated pdf to be Gaussian by evaluating their first and second moment.

So, I will use the Extended Kalman Filter, which is nothing but the non-linear version of Kalman Filter. The EKF uses a first-order Taylor Series approximation to the motion and observation models around the state and noise.

$$f(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t) \approx f(\boldsymbol{\mu}_{t|t}, \mathbf{u}_t, \mathbf{0}) + \left[\frac{df}{d\mathbf{x}}(\boldsymbol{\mu}_{t|t}, \mathbf{u}_t, \mathbf{0})\right](\mathbf{x}_t - \boldsymbol{\mu}_{t|t}) + \left[\frac{df}{d\mathbf{w}}(\boldsymbol{\mu}_{t|t}, \mathbf{u}_t, \mathbf{0})\right](\mathbf{w}_t - \mathbf{0})$$

$$h(\mathbf{x}_{t+1}, \mathbf{v}_{t+1}) \approx h(\boldsymbol{\mu}_{t+1|t}, \mathbf{0}) + \left[\frac{dh}{d\mathbf{x}}(\boldsymbol{\mu}_{t+1|t}, \mathbf{0})\right](\mathbf{x}_{t+1} - \boldsymbol{\mu}_{t+1|t}) + \left[\frac{dh}{d\mathbf{v}}(\boldsymbol{\mu}_{t+1|t}, \mathbf{0})\right](\mathbf{v}_{t+1} - \mathbf{0})$$

EKF Prediction:

As I have my prior pdf as Gaussian

$$f(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t) \approx f(\boldsymbol{\mu}_{t|t}, \mathbf{u}_t, \mathbf{0}) + \left[\frac{df}{d\mathbf{x}}(\boldsymbol{\mu}_{t|t}, \mathbf{u}_t, \mathbf{0})\right](\mathbf{x}_t - \boldsymbol{\mu}_{t|t}) + \left[\frac{df}{d\mathbf{w}}(\boldsymbol{\mu}_{t|t}, \mathbf{u}_t, \mathbf{0})\right](\mathbf{w}_t - \mathbf{0})$$

And let

Let $F_t := \frac{df}{d\mathbf{x}}(\boldsymbol{\mu}_{t|t}, \mathbf{u}_t, \mathbf{0})$ and $Q_t := \frac{df}{d\mathbf{w}}(\boldsymbol{\mu}_{t|t}, \mathbf{u}_t, \mathbf{0})$ thus our motion model is:

$$f(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t) \approx f(\boldsymbol{\mu}_{t|t}, \mathbf{u}_t, \mathbf{0}) + F_t(\mathbf{x}_t - \boldsymbol{\mu}_{t|t}) + Q_t\mathbf{w}_t\,.$$

Thus the predicted mean and covariance becomes

$$\begin{aligned}
\boldsymbol{\mu}_{t+1|t} &\approx \iint \left(f(\boldsymbol{\mu}_{t|t}, \mathbf{u}_t, \mathbf{0}) + F_t(\mathbf{x} - \boldsymbol{\mu}_{t|t}) + Q_t\mathbf{w}\right)\phi(\mathbf{x}; \boldsymbol{\mu}_{t|t}, \Sigma_{t|t})\phi(\mathbf{w}; 0, W)d\mathbf{x}d\mathbf{w} \\
&= f(\boldsymbol{\mu}_{t|t}, \mathbf{u}_t, \mathbf{0}) + F_t\left(\int \mathbf{x}\phi(\mathbf{x}; \boldsymbol{\mu}_{t|t}, \Sigma_{t|t})d\mathbf{x} - \boldsymbol{\mu}_{t|t}\right) + Q_t\int \mathbf{w}\phi(\mathbf{w}; 0, W)d\mathbf{w} \\
&= \boxed{f(\boldsymbol{\mu}_{t|t}, \mathbf{u}_t, \mathbf{0})}
\end{aligned}$$

$$\begin{aligned}
\Sigma_{t+1|t} &\approx \iint \left(f(\boldsymbol{\mu}_{t|t}, \mathbf{u}_t, \mathbf{0}) + F_t(\mathbf{x} - \boldsymbol{\mu}_{t|t}) + Q_t\mathbf{w}\right)\left(f(\boldsymbol{\mu}_{t|t}, \mathbf{u}_t, \mathbf{0}) + F_t(\mathbf{x} - \boldsymbol{\mu}_{t|t}) + Q_t\mathbf{w}\right)^\top \phi(\mathbf{x}; \boldsymbol{\mu}_{t|t}, \Sigma_{t|t})\phi(\mathbf{w}; 0, W)d\mathbf{x}d\mathbf{w} \\
&\quad - \boldsymbol{\mu}_{t+1|t}\boldsymbol{\mu}_{t+1|t}^\top \\
&= f(\boldsymbol{\mu}_{t|t}, \mathbf{u}_t, \mathbf{0})\left(\int (\mathbf{x} - \boldsymbol{\mu}_{t|t})^\top \phi(\mathbf{x}; \boldsymbol{\mu}_{t|t}, \Sigma_{t|t})d\mathbf{x}\right)F_t^\top + F_t\left(\int (\mathbf{x} - \boldsymbol{\mu}_{t|t})\phi(\mathbf{x}; \boldsymbol{\mu}_{t|t}, \Sigma_{t|t})d\mathbf{x}\right)f(\boldsymbol{\mu}_{t|t}, \mathbf{u}_t, \mathbf{0})^\top \\
&\quad + F_t\left(\int (\mathbf{x} - \boldsymbol{\mu}_{t|t})(\mathbf{x} - \boldsymbol{\mu}_{t|t})^\top \phi(\mathbf{x}; \boldsymbol{\mu}_{t|t}, \Sigma_{t|t})d\mathbf{x}\right)F_t^\top + Q_t\left(\int \mathbf{w}\mathbf{w}^\top \phi(\mathbf{w}; 0, W)d\mathbf{w}\right)Q_t^\top \\
&= \boxed{F_t\Sigma_{t|t}F_t^\top + Q_tWQ_t^\top}
\end{aligned}$$

As in this project I will be considering that the landmarks don't move at all, so using that I can say that there is no predict step required for the landmark. But there will be a predict-step required for the IMU as our car is moving in the world environment.

EKF Update:

The observation model is given as

Let $H_{t+1} := \frac{dh}{d\mathbf{x}}(\boldsymbol{\mu}_{t+1|t}, \mathbf{0})$ and $R_{t+1} := \frac{dh}{d\mathbf{v}}(\boldsymbol{\mu}_{t+1|t}, \mathbf{0})$ so that:

$$h(\mathbf{x}_{t+1}, \mathbf{v}_{t+1}) \approx h(\boldsymbol{\mu}_{t+1|t}, \mathbf{0}) + H_{t+1}(\mathbf{x}_{t+1} - \boldsymbol{\mu}_{t+1|t}) + R_{t+1}\mathbf{v}_{t+1}$$

Thus using the joint distribution of the state and the observation can be computed using the following equations:

$$\begin{aligned}
\mathbf{m}_{t+1|t} &:= \int\int h(\mathbf{x}, \mathbf{v})\phi(\mathbf{x}; \boldsymbol{\mu}_{t+1|t}, \Sigma_{t+1|t})\phi(\mathbf{v}; 0, V)d\mathbf{x}d\mathbf{v} \approx \boxed{h(\boldsymbol{\mu}_{t+1|t}, \mathbf{0})} \\
S_{t+1|t} &:= \int\int (h(\mathbf{x}, \mathbf{v}) - \mathbf{m}_{t+1|t})(h(\mathbf{x}, \mathbf{v}) - \mathbf{m}_{t+1|t})^\top \phi(\mathbf{x}; \boldsymbol{\mu}_{t+1|t}, \Sigma_{t+1|t})\phi(\mathbf{v}; 0, V)d\mathbf{x}d\mathbf{v} \\
&\approx \boxed{H_{t+1}\Sigma_{t+1|t}H_{t+1}^\top + R_{t+1}VR_{t+1}^\top} \\
C_{t+1|t} &:= \int\int (\mathbf{x} - \boldsymbol{\mu}_{t+1|t})(h(\mathbf{x}, \mathbf{v}) - \mathbf{m}_{t+1|t})^\top \phi(\mathbf{x}; \boldsymbol{\mu}_{t+1|t}, \Sigma_{t+1|t})\phi(\mathbf{v}; 0, V)d\mathbf{x}d\mathbf{v} \\
&\approx \boxed{\Sigma_{t+1|t}H_{t+1}^\top}
\end{aligned}$$

Using this supporting equation I can find the Kalman Gain, and finally updated mean and covariance. Thus the conditional Gaussian distribution of $x_{t+1} | y_{t+1}$ is

$$\begin{aligned}
\boldsymbol{\mu}_{t+1|t+1} &:= \boldsymbol{\mu}_{t+1|t} + K_{t+1|t}(\mathbf{z}_{t+1} - \mathbf{m}_{t+1|t}) \\
\Sigma_{t+1|t+1} &:= \Sigma_{t+1|t} - K_{t+1|t}S_{t+1|t}K_{t+1|t}^\top \\
K_{t+1|t} &:= C_{t+1|t}S_{t+1|t}^{-1}
\end{aligned}$$

3.2 Visual Mapping via the EKF

In order to the visual mapping of the Landmark, we just have to update it using the position of the IMU. As I don't know where my car is in the world space I can take the Pose of the IMU as Identity in SE(3) space.

Observation Model: with measurement noise $v_{t,i} \sim N(0, V)$. $\mathbf{z}_{t,i} = h(T_t, \mathbf{m}_j) + \mathbf{v}_{t,i} := K_s\pi\left({}_OT_I T_t^{-1}\underline{\mathbf{m}}_j\right) + \mathbf{v}_{t,i}$. Here the homogeneous coordinates: is $[m_j\ 1]^T$. The projection function and its derives that I will use to calculate the pixel values in the coordinate frame is:

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

When all the observations are stacked as $4N_t$ vector, at time t with notation abuse, I get:

$$\mathbf{z}_t = K_s\pi\left({}_OT_I T_t^{-1}\underline{\mathbf{m}}\right) + \mathbf{v}_t \qquad \mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, I \otimes V) \qquad I \otimes V := \begin{bmatrix} V & & \\ & \ddots & \\ & & V \end{bmatrix}$$

Mapping Update:

The Mapping does not involve predict step as the landmarks are static with respect to the world environment.

Thus the **prior:** $\mathbf{m} \mid \mathbf{z}_{0:t} \sim \mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t)$ with $\boldsymbol{\mu}_t \in \mathbb{R}^{3M}$ and $\Sigma_t \in \mathbb{R}^{3M\times 3M}$.

The **EKF Update:** when a new observation is observed at time $Z_{t+1} \in \mathbb{R}^{4N_{t+1}}$.

$$\begin{aligned}
K_{t+1} &= \Sigma_t H_{t+1}^\top \left(H_{t+1}\Sigma_t H_{t+1}^\top + I \otimes V\right)^{-1} \\
\boldsymbol{\mu}_{t+1} &= \boldsymbol{\mu}_t + K_{t+1}\Big(\mathbf{z}_{t+1} - \underbrace{K_s\pi\left({}_OT_I T_{t+1}^{-1}\underline{\boldsymbol{\mu}}_t\right)}_{\tilde{\mathbf{z}}_{t+1}}\Big) \\
\Sigma_{t+1} &= (I - K_{t+1}H_{t+1})\Sigma_t
\end{aligned}$$

Here z tilt is the predicted observation of the landmark in the optical coordinate system. Using the pose that is available at time t.

I need the Jacobian H in order to find the updated mean, covariance, and Kalman gain, and this can be derived as follows.

$$H_{t+1,i,j} := \begin{cases} \left.\frac{\partial}{\partial \mathbf{m}_j} h(T_{t+1}, \mathbf{m}_j)\right|_{\mathbf{m}_j = \boldsymbol{\mu}_{t,j}}, & \text{if } \Delta_t(j) = i, \\ \mathbf{0}, & \text{otherwise.} \end{cases}$$

The derivative of the observation model with respect to the particles at time t is derived using the Chain rule.

$$\frac{\partial}{\partial \mathbf{m}_j} h(T_{t+1}, \mathbf{m}_j) = K_s \frac{\partial \pi}{\partial \mathbf{q}}(\mathbf{q}_{t+1,j}) \frac{\partial \mathbf{q}_{t+1,j}}{\partial \mathbf{m}_j}$$
$$= K_s \frac{\partial \pi}{\partial \mathbf{q}} \left({}_O T_I T_{t+1}^{-1} \underline{\mathbf{m}}_j \right) {}_O T_I T_{t+1}^{-1} \frac{\partial \underline{\mathbf{m}}_j}{\partial \mathbf{m}_j}$$
$$= K_s \frac{\partial \pi}{\partial \mathbf{q}} \left({}_O T_I T_{t+1}^{-1} \underline{\mathbf{m}}_j \right) {}_O T_I T_{t+1}^{-1} P^\top$$

These are the parameters i.e. K is the calibration matrix of the Stereo camera, ${}_oT_I$ is the rotation of the IMU to coordinate from and T^{-1}_{t+1} is the rotation from world to IMU frame. And matrix P is the matrix with [I 0]. With mj as the homogenous coordinates of the landmarks in the world frame.

The position of the landmark is stored in $\mu t \in R^{3M}$ and covariance is stored in $\Sigma t \in R^{3Mx3M}$. Suppose a landmark is seen for the very first time. The position is directly updated in the matrix μ and the update step for that particular landmark is not done. This is done because the prior knowledge of the landmark is not known which can cause problems.

3.3 Visual Inertial Odometry:

In visual-inertial odometry, I will assume all the landmarks in the world are known to me and I have to correct just the position of the IMU using the odometry model.

We know the landmarks in the optical frame because of the stereo camera. Thus there is Pose matrix in SE(3) state. Thus direct addition is not possible so instead of simplifying the dynamic equations, we can work according to the kinematics.

Assumption during this kinematics equations are

1. We know the linear velocity instead of linear acceleration,
2. The World Frame coordinated of the Landmark is known
3. The features of the world frame, in my case the features of the edges is provided by an external algorithm.

Motion Model:

As the car is moving, it will have angular velocity and linear velocity. Thus the pose and control are respectively:

$$\dot{T} = T\left(\hat{\mathbf{u}} + \hat{\mathbf{w}}\right) \qquad \mathbf{u}(t) := \begin{bmatrix} \mathbf{v}(t) \\ \omega(t) \end{bmatrix} \in \mathbb{R}^6$$

Thus using the pose at previous time t, and control unit at time t+1, the new pose matrix can be calculated using the kinematics equation. As I am dealing with discrete-time, I will the discrete-time equation for the pose update.

$T_{k+1} = T_k \exp(\tau_k \hat{\zeta}_k)$. Thus using this I will update the pose, where the T is the previous pose and sigh is given below.

$$\zeta(t) := \begin{bmatrix} \mathbf{v}(t) \\ \omega(t) \end{bmatrix} \in \mathbb{R}^6 \text{ is the } \textbf{input}$$

But in our equation, the term inside the exponent is sigh_hat which can be calculated as.

$$\mathfrak{se}(3) := \left\{ \hat{\boldsymbol{\xi}} := \begin{bmatrix} \hat{\boldsymbol{\theta}} & \boldsymbol{\rho} \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{4\times4} \;\middle|\; \boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\theta} \end{bmatrix} \in \mathbb{R}^6 \right\}$$

And the simple hat function is the conversion of cross-product to vector multiplication. Tau is the small difference in the time during which this update needs to take place.

IMU Predict:

Using the motion model I can predict the next step of the car.

Prior: $T_t | \mathbf{z}_{0:t}, \mathbf{u}_{0:t-1} \sim \mathcal{N}(\mu_{t|t}, \Sigma_{t|t})$ with $\mu_{t|t} \in SE(3)$ and $\Sigma_{t|t} \in \mathbb{R}^{6\times6}$

Thus using the motion model kinematics, the new step will be:

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

where

$$\mathbf{u}_t := \begin{bmatrix} \mathbf{v}_t \\ \omega_t \end{bmatrix} \in \mathbb{R}^6 \quad \hat{\mathbf{u}}_t := \begin{bmatrix} \hat{\omega}_t & \mathbf{v}_t \\ \mathbf{0}^\top & 0 \end{bmatrix} \in \mathbb{R}^{4\times4} \quad \overset{\curlywedge}{\mathbf{u}}_t := \begin{bmatrix} \hat{\omega}_t & \hat{\mathbf{v}}_t \\ 0 & \hat{\omega}_t \end{bmatrix} \in \mathbb{R}^{6\times6}$$

Observation model:

In the observation model I consider the noise $v_t \sim N(0, V)$ with mean 0 and variance V.

Prior: $T_{t+1} | z_{0:t}, u_{0:t} \sim \mathcal{N}(\mu_{t+1|t}, \Sigma_{t+1|t})$, with $\mu_{t+1|t} \in SE(3)$ and covariance is stored in $\Sigma_{t+1|t} \in R^{6x6}$.

And the observation with measurement noise v_t is.

$$\mathbf{z}_{t+1,i} = h(T_{t+1}, \mathbf{m}_j) + \mathbf{v}_{t+1,i} := K_s \pi \left({}_O T_I T_{t+1}^{-1} \underline{\mathbf{m}}_j \right) + \mathbf{v}_{t+1,i}$$

The observation model remains the same as visual mapping, but this time we are trying to get an observation of the IMU.

I need the Jacobian H of the observation model of IMU at time t+1 evaluated using $\mu_{t+1|t}$. And $H_{t+1} \in R^{4Nx6}$. N represents the number of visible landmarks at time t.

$$\mathbf{z}_{t+1,i} = K_s \pi \left({}_O T_I \left(\boldsymbol{\mu}_{t+1|t} \exp\left(\widehat{\delta\boldsymbol{\mu}}\right) \right)^{-1} \underline{\mathbf{m}}_j \right) + \mathbf{v}_{t+1,i}$$
$$\approx K_s \pi \left({}_O T_I \left(I - \widehat{\delta\boldsymbol{\mu}} \right) \boldsymbol{\mu}_{t+1|t}^{-1} \underline{\mathbf{m}}_j \right) + \mathbf{v}_{t+1,i}$$
$$= K_s \pi \left({}_O T_I \boldsymbol{\mu}_{t+1|t}^{-1} \underline{\mathbf{m}}_j - {}_O T_I \left(\boldsymbol{\mu}_{t+1|t}^{-1} \underline{\mathbf{m}}_j \right)^{\odot} \delta\boldsymbol{\mu} \right) + \mathbf{v}_{t+1,i}$$
$$\approx \underbrace{K_s \pi \left({}_O T_I \boldsymbol{\mu}_{t+1|t}^{-1} \underline{\mathbf{m}}_j \right)}_{\tilde{\mathbf{z}}_{t+1,i}} \underbrace{- K_s \frac{d\pi}{d\mathbf{q}} \left({}_O T_I \boldsymbol{\mu}_{t+1|t}^{-1} \underline{\mathbf{m}}_j \right) {}_O T_I \left(\boldsymbol{\mu}_{t+1|t}^{-1} \underline{\mathbf{m}}_j \right)^{\odot}}_{H_{t+1,i}} \delta\boldsymbol{\mu} + \mathbf{v}_{t+1,i}$$

where for homogeneous coordinates $\underline{\mathbf{s}} \in \mathbb{R}^4$ and $\hat{\boldsymbol{\xi}} \in \mathfrak{se}(3)$:

$$\hat{\boldsymbol{\xi}} \underline{\mathbf{s}} = \underline{\mathbf{s}}^{\odot} \boldsymbol{\xi} \qquad \begin{bmatrix} \mathbf{s} \\ 1 \end{bmatrix}^{\odot} := \begin{bmatrix} I & -\hat{\mathbf{s}} \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{4\times6}$$

IMU Update:

In the IMU update step we will try to correct the pose of the IMU. Prior: $\mu_{t+1|t} \in SE(3)$ and covariance is stored in $\Sigma_{t+1|t} \in R^{6x6}$. Given the calibration matrix K of the stereo camera, ${}_oT_I$ is the rotation of the IMU to coordinate from, T^{-1}_{t+1} is the rotation from world to IMU frame, new observation $z_{t+1} \in R^{4N}$.

Predicted observation based on predicted IMU pose, $\tilde{\mathbf{z}}_{t+1,i} := K_s\pi\left({}_oT_I\mu^{-1}_{t+1|t}\mathbf{m}_j\right)$ for i = 1, …, N_{t+1}. The Jacobian of z tilt with respect to T_{t+1} is evaluated at $\mu_{t+1|t}$.

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

Thus the update step of the EKF Filter will be.

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

3.4 VSLAM

Now there are two ways of doing VSLAM.

The first way is to do everything separately. Keep separate mean and covariance for IMU and mapping. In this, the predict step of IMU will be not much of the Hassle because there is no predict-step for the Landmark. But during the update step, during the calculation of the IMU pose, we considered that the position of the landmarks was correct and updated the IMU pose. And, while calculating the landmarks we assumed the IMU pose was correct. Thus I am assuming that there is independence in the system, but which is not there.

The other way to do the SLAM is to combine the IMU and Mapping mean and covariance matrix. In this, we calculate everything at a single time. Thus we are not assuming the one as correct at a particular time and evaluating the other.

I am doing the first, keeping them separate and finding the mean and covariance and the loop goes on for the dataset.

IV. Evaluation

I have the data from the IMU and the stereo camera. The stereo camera data is not readily available instead, the features of the world are provided by some algorithm. Now in order to check my algorithm, I have been given 2 data sets namely “03.npz” and “10.npz”.

4.1 Dataset “03.npz”:

I tried my algorithm on the dataset named 03.npz and the picture is provided as in fig 1.

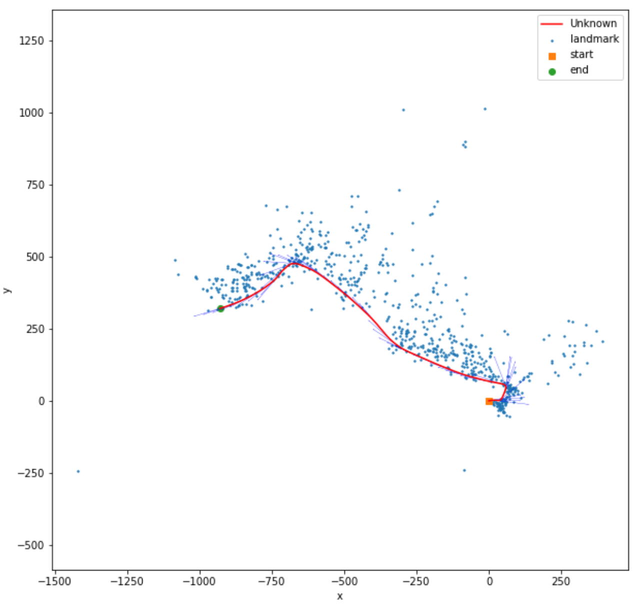

Fig1. Dead Reckoning for “03.npz”

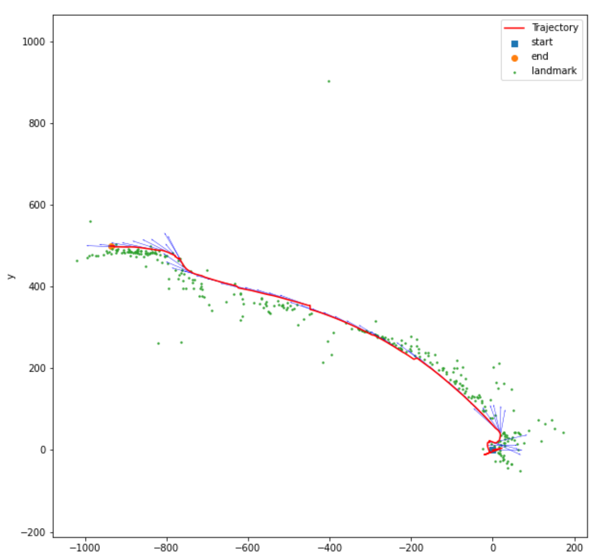

Fig2. SLAM for “03.npz”

4.2 Dataset "10.npz":

I tried my algorithm on the dataset named 03.npz and the picture is provided as in fig 2.

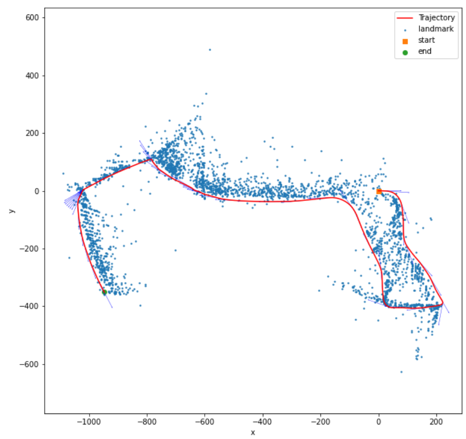

Fig3. Dead Reckoning for "10.npz"

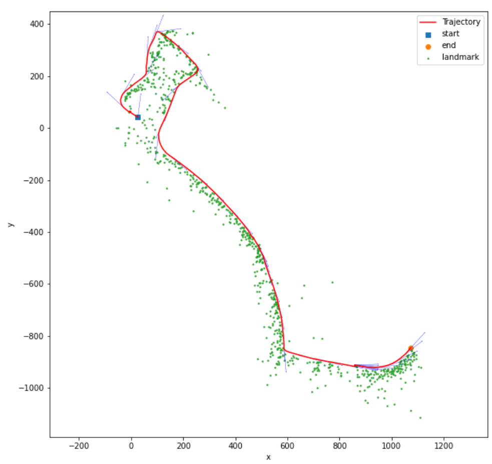

Fig4. SLAM for "10.npz"

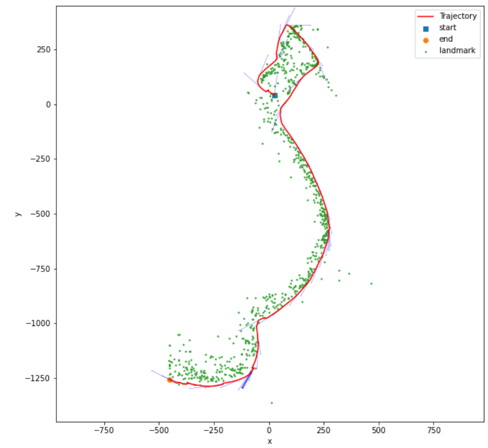

Fig5. SLAM for "10.npz"

The main challenges I faced during the coding step is matrix multiplication and reshaping. In the initial step of my code, I use to just reshape directly but turned out the order got mismatched and thus got strange images. Due to flipped up IMU there is a lot of ambiguity in which one is correct and what to use. But these two maps are pretty close to the actual trajacetory that I saw in the video. From, the video I am not able to guess the correct curve path of vehicle.

Another important thing I noticed is adding noise is okay, but it is not affected me because of the nature of my code. This is happening due to I am separately updating the covariance of IMU and Landmark, which is leading me to inaccurate results. I tried merging them, but it leads to the singular matrix.

References

[1] http://robots.stanford.edu/papers/montemerlo.fastslam-tr.pdf
[2] Slide ECE276A_12_EKF_UKF, all equations are direct screenshots from lecture slides.
[3] Discussed this with Sambaran.