Visual-Inertial SLAM using Extended Kalman Filter

Yunhai Han
Department of Mechanical and Aerospace Engineering
University of California, San Diego
y8han@eng.ucsd.edu

***Abstract*—This paper represented approaches for visual-inertial simultaneous localization and mapping problem(also known as SLAM). In this project, we are provided with sensor data, odometry, 2-D laser scans, and RGBD measurements from a humanoid robot. However, all these sensor data are not perfect because they are contaminated by random noise to some extent. For this reason, instead of accumulating raw data without any modification, I introduced extended kalman filter to predict and update the mean and covariance of the state vector. In this case, the state vector contains the pose of the autonomous driving car and the 3D position of all the landmarks. Since, the rotation needs to be updated during each iteration, Lie algebra and Lie group are introduced for this purpose. With this modification, the trajectory recovered from sensor data could be much better.**

***Index Terms*—simultaneous localization and mapping(SLAM), Extended kalman filter(EKF), Autonomous driving car, Lie algebra and Lie group**

I. Introduction

Robotics is the science of sensing and manipulating the physical world through computer-controlled devices. Examples of successful robotic systems include mobile robots for planetary exploration, industrial robotics arms in assembly lines, autonomous driving cars and manipulators that assist surgeons. Robotics systems are situated in the physical world, obtain information from their local environments through onboard sensors, and make reactions. While all of these examples sound amazing, there are still some really challenging problems, which make these applications unavailable right now. First of all, robot environments are inherently unpredictable and dynamically changes. While the degree of uncertainty in well-structured environments is small, environments such as highways and private homes are highly dynamic, thus highly unpredictable. The uncertainty is particularly high for robots with people nearby. Second, sensors are limited in what they can perceive. Limitations arise from several factors. The range and resolution of a sensor is subject to physical limitations. For example, cameras cannot see through walls, and the spatial resolution of a camera image is limited. Sensors are also subject to noise, which makes sensor measurements unpredictable and hence limits the information that can be extracted. Third, robot actuation involves motors that are also unpredictable. Uncertainty arises from effects like control noise or mechanical failure. Some actuators, such as heavy-duty industrial robot arms, are quite accurate and reliable. Others, like low-cost or legged mobile robots, can be extremely flaky(like the humanoid robot in this project). How to handle uncertainty is indeed the most important step towards robust real-world robot systems [1].

During the decades of the development of Bayes filters, there are various efficient algorithms including Extended Kalman Filter, Unscented Kalman Filter and Particle Filter. The main advantage of the extended kalman filter is that it is easy for implementation. However, since most physical models are nonlinear systems, we need to use first-order Taylor expansion for linearization. In most cases, if the system reserves linearity in a local region, the results could be good enough. Besides, it is the most efficient method compared with any other filters. That's one of the reasons why it becomes immensely popular in robotics and some other related areas.

II. Problem Formulation

Given a set of sensor data including IMU and measurement data from stereo camera, the problem is to build a map of each landmark and mark the car's trajectory on the map. However, as I mentioned in the **Introduction** section, there are random noise in these data, which requires me to implement extended kalman filter to estimate the best trajectory and build the feature map.

A. IMU-based Localization via EKF Prediction

The first task is to predict the car's pose only by the IMU data. Also, this is know as localization-only problem. As the professor tells in the class, this time we would only use kinematic rather than dynamic equations. Here is the problem description:

- **Assumption 1**: linear velocity $\mathbf{v}_t \in \mathbb{R}^3$ instead of linear acceleration $\mathbf{a}_t \in \mathbb{R}^3$ measurements are available; angular velosity $\mathbf{w}_t \in \mathbb{R}^3$ instead of angular acceleration $\alpha_t \in \mathbb{R}^3$ measurements are available.
- **Assumption 2**: the world-frame landmark coordinates $\mathbf{m} \in \mathbb{R}^{3\times M}$ are known.
- **Assumption 3**: the data association $\pi_t : \{1, \ldots, M\} \to \{1, \ldots, N_t\}$ stipulating which landmarks were observed at each time t is known or provided by an external algorithm. In most cases, this is the hardest part.
- **Objective**: given the IMU measurements $\mathbf{u}_{0:T}$ with $\mathbf{u}_t := \left[\mathbf{v}_t^\top, \boldsymbol{\omega}_t^\top\right]^\top$ and the visual feature observations $\mathbf{z}_{0:T}$, estimate the inverse IMU pose $U_t := {}_wT_{l,t}^{-1} \in SE(3)$ over time.

B. Landmark Mapping via EKF Update

The second task is to estimate the landmark positions, assuming that the predicted IMU trajectory from the previous problem is correct. In detail, we should implement an EKF with the unknown landmark positions $\mathbf{m} \in \mathbb{R}^{3\times M}$ as a state and perform EKF update step after every visual observation $\mathbf{z}_t$ in order to keep track of the mean and covariance of $\mathbf{m}$. Note that in this case, we are assuming that the landmarks are static so it is not necessary to implement a prediction step. Here is the problem description:

- **Assumption 1**: the inverse IMU pose $U_t := wT_{I,t}^{-1} \in SE(3)$ is known.
- **Assumption 2**: the landmarks are static, i.e., it is not necessary to consider a prediction step.
- **Assumption 3**: the data association $\pi_t : \{1,\dots,M\} \to \{1,\dots,N_t\}$ stipulating which landmarks were observed at each time t is known or provided by an external algorithm.
- **Objective**: given the visual feature observations $z_{0:T}$ and inverse IMU pose U_t, estimate the homogeneous coordinates $\underline{m} \in \mathbb{R}^{4\times M}$ in the world frame of the landmarks that generated the visual observations

The homogeneous coordinate of the position of each landmark m_i could be expressed as:

$$\text{Homogeneous coordinates: } \underline{m}_i := \begin{bmatrix} m_i \\ 1 \end{bmatrix}$$

Finally, we could simply extract x, y from each homogenous coordinate(the first two elements) and then plot the feature map.

C. Visual-Inertial SLAM

The third task is to combine the IMU prediction step from part (A) with the land-mark update step from part (B) and an IMU update step based on the stereo camera observation model to obtain a complete visual-inertial SLAM algorithm. The new function here is to update IMU pose based on the stereo camera observation model. It is very similar to the part (B) except the fact that the first-order Taylor series approximation would be expanded at an inverse IMU pose using a pose perturbation $\delta\boldsymbol{\mu}_{t+1|t+1}$ instead of the perturbation $\delta\boldsymbol{\mu}_{t,j}$ for the position of landmark j in part (B). Here is the problem description:

- **Assumption**: The assumptions are the same as above.
- **Objective**: given the visual feature observations $z_{0:T}$ and inverse IMU pose U_t, update the IMU pose U_t based on the observation model and the sensor value.

To sum up, the output should include both the car's trajectory and the world-frame coordinates of all the landmarks. They are shown in the following figure: The green trajectory and the black landmarks are what we want.

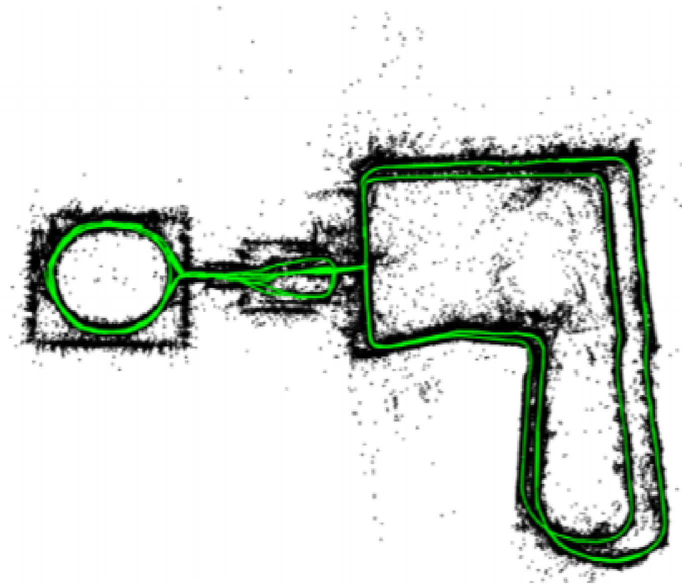

Fig. 1. The final outputs(from page 3)

III. Technical Approach

In the **Problem Formulation** section, I briefly describe the framework of how I deal with this project. In this section, I will tell more details about each tasks.

A. IMU-based Localization via EKF Prediction

First, we could assume we already obtain the prior probability distribution of time t by its mean and covariance:$U_t|\mathbf{z}_{0:t}, \mathbf{u}_{0:t-1} \sim \mathcal{N}\left(\boldsymbol{\mu}_{t|t}, \Sigma_{t|t}\right)$ with $\boldsymbol{\mu}_{t|t} \in SE(3)$ and $\Sigma_{t|t} \in \mathbb{R}^{6\times 6}$. Then our goal is to predict the mean and covariance at time $t+1$:$U_t|\mathbf{z}_{0:t}, \mathbf{u}_{0:t} \sim \mathcal{N}\left(\boldsymbol{\mu}_{t+1|t}, \Sigma_{t+1|t}\right)$.

Here, The covariance is 6×6 because only the six degrees of freedom of $U_t \in SE(3)$ are changing.

The motion model could be described as:

Motion Model: with time discretization τ and noise $\mathbf{w}_t \sim \mathcal{N}(0, W)$

$$U_{t+1} = \exp\left(-\tau\left((\mathbf{u}_t + \mathbf{w}_t)\right)^\wedge\right) U_t \quad \mathbf{u}_t := \begin{bmatrix} \mathbf{v}_t \\ \omega_t \end{bmatrix} \in \mathbb{R}^6$$

Note that $\mathbf{u}_t + \mathbf{w}_t$ is negative above since U_t is the inverse IMU pose.

The reason is very simple: Let the IMU pose in continuous time be $wT_I(t) = T(t) = U^{-1}(t)$

$$\dot{T} = T\hat{\mathbf{u}} \quad TU = 1 \quad \dot{T}U + T\dot{U} = 0$$
$$\dot{U} = -U\dot{T}U = -U(T\hat{\mathbf{u}})U = -\hat{\mathbf{u}}U$$
$$U_{t+1} = \exp\left(-\tau\hat{\mathbf{u}}_t\right) U_t$$

In the above equations, you could see why the $u_t + w_t$ should be negative when U_t is the inverse IMU pose.

Then, we need to define the kinematics model with perturbation. There are four steps:

- Consider what happens with the pose kinematics

$$\dot{T} = -(\hat{\mathbf{u}} + \hat{\mathbf{w}})T$$

if the pose is expressed as a nominal pose $\mu \in SE(3)$ and small perturbation $\delta\hat{\boldsymbol{\mu}} \in \mathrm{se}(3)$:

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

- Substituting the nominal + perturbed pose in the kinematic equations:

$$(\hat{\delta}\dot{\boldsymbol{\mu}})\boldsymbol{\mu} + (1 + \hat{\delta\boldsymbol{\mu}})\dot{\boldsymbol{\mu}} = -(\hat{\mathbf{u}} + \hat{\mathbf{w}})(1 + \hat{\delta\boldsymbol{\mu}})\boldsymbol{\mu}$$
$$(\hat{\delta\boldsymbol{\mu}})\boldsymbol{\mu} + \hat{\delta\mu}\dot{\mu} + \dot{\boldsymbol{\mu}} = -\hat{\mathbf{u}}\boldsymbol{\mu} - \hat{\boldsymbol{w}}\boldsymbol{\mu} - \hat{\mathbf{u}}\delta\hat{\mu}\boldsymbol{\mu} - \hat{\boldsymbol{w}}\delta\hat{\mu}\boldsymbol{\mu}$$
$$\dot{\boldsymbol{\mu}} = -\hat{\mathbf{u}}\boldsymbol{\mu} \quad (\hat{\delta\boldsymbol{\mu}})\boldsymbol{\mu} - \delta\mu\hat{\ }\boldsymbol{\mu} = -\hat{\mathbf{w}}\boldsymbol{\mu} - \hat{\mathbf{0}}\delta\hat{\mu}\boldsymbol{\mu}$$
$$\dot{\boldsymbol{\mu}} = -\hat{\mathbf{u}}\boldsymbol{\mu} \quad \hat{\delta\mu} = \hat{\delta\mu}\hat{\mathbf{u}} - \hat{\mathbf{u}}\hat{\boldsymbol{\delta}}\boldsymbol{\mu} - \hat{\mathbf{w}} = (-\hat{\mathbf{u}}\delta\boldsymbol{\mu})^{\wedge} - \hat{\mathbf{w}}$$

- Using $T \approx (I + \delta\hat{\boldsymbol{\mu}})\boldsymbol{\mu}$, the pose kinematics $\dot{T} = -(\hat{\mathbf{u}} + \hat{\mathbf{w}})T$ can be split into nominal and perturbation kinematics:

$$\text{nominal} : \boldsymbol{\mu} = -\hat{\mathbf{u}}\boldsymbol{\mu} \qquad \widetilde{\mathbf{u}} := \begin{bmatrix} \hat{\omega} & \hat{\mathbf{v}} \\ 0 & \hat{\omega} \end{bmatrix} \in \mathbb{R}^{6\times 6}$$
$$\text{perturbation} : \delta\boldsymbol{\mu} = -\widetilde{\mathbf{u}}\delta\boldsymbol{\mu} + \quad \mathbf{w}$$

- In discrete-time with discretization τ, the above becomes:

$$\text{nominal} \ :\boldsymbol{\mu}_{t+1} = \exp\left(-\tau\hat{\mathbf{u}}_t\right)\boldsymbol{\mu}_t$$
$$\text{perturbation} : \delta\boldsymbol{\mu}_{t+1} = \exp\left(-\tau\widetilde{\mathbf{u}}_t\right)\delta\boldsymbol{\mu}_t + \mathbf{w}_t$$

In the last step, we successfully separate the effect of the noise w_t from the motion of the deterministic part of T_t.

Finally, the EFK prediction step with $\mathbf{w}_t \sim \mathcal{N}(0, W)$ could be defined as:

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

where

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

During each iteration, we extract the IMU data and implement these steps. As a result, we could plot the car's trajectory from raw data and the figures are in **Results**.

B. Landmark Mapping via EKF Update

In this section, we consider the mapping-only problem, assuming the inverse IMU pose

$$U_t := wT_{l,t}^{-1} \in SE(3)$$

is known.

At the time $t+1$, we want to update the mean and covariance of map at time t, which are the prior probability distribution:$\mathbf{m}|\mathbf{z}_{0:t} \sim \mathcal{N}\left(\boldsymbol{\mu}_t, \Sigma_t\right)$ with $\boldsymbol{\mu}_t \in \mathbb{R}^{3M}$ and $\Sigma_t \in \mathbb{R}^{3M\times 3M}$. Here, M represents the number of the landmarks and it is known.

The observation model could be described as: Observation Model: with measurement noise $\mathbf{v}_{t,i} \sim \mathcal{N}(0, V)$

$$\mathbf{z}_{t,i} = h\left(U_t, \mathbf{m}_j\right) + \mathbf{v}_{t,i} := M\pi\left(oT_l U_t \mathbf{m}_j\right) + \mathbf{v}_{t,i}$$

Where, π represents the projection function; $\mathbf{m}_j$ represents the homogeneous coordinate of the landmark j; M represents the intrinsic matrix for stereo camera; oT_l represents the extrinsic matrix from IMU to the left camera; U_t represents the inverse IMU pose; $\mathbf{v}_{t,i}$ represents the measurement noise.

The expression and the derivative for function π is:

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

All observations(stacked as a $4N_t$ vector) at time t could be described in one equation:

$$\mathbf{z}_t = M\pi\left(oT_l U_t \mathbf{m}\right) + \mathbf{v}_t \quad \mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, I \otimes V) \quad I \otimes V := \begin{bmatrix} V & & \\ & \ddots & \\ & & V \end{bmatrix}$$

The EKF update steps are:

$$\begin{aligned} K_t &= \Sigma_t H_t^{\top}\left(H_t\Sigma_t H_t^{\top} + I \otimes V\right)^{-1} \\ \mu_{t+1} &= \mu_t + K_t\left(z_t - \underbrace{M\pi\left(oT_l U_t \mu_t\right)}_{\tilde{z}_t}\right) \\ \Sigma_{t+1} &= \left(I - K_t H_t\right)\Sigma_t \end{aligned}$$

In the above equations, $\tilde{\mathbf{z}}_t$ is the predicted observation based on the landmark position estimates $\boldsymbol{\mu}_t$ at time t and z_t is the measurement data read from sensors.

Hence, We need the observation model Jacobian $H_t \in \mathbb{R}^{4N_t\times 3M}$ evaluated at μ_t. Let the elements of $H_t \in \mathbb{R}^{4N_t\times 3M}$ corresponding to different observations i and different landmarks j be $H_{t,i,j} \in \mathbb{R}^{4\times 3}$.

Consider a perturbation $\delta\boldsymbol{\mu}_{t,j}$ for the position of landmark j :

$$\mathbf{m}_j = \boldsymbol{\mu}_{t,j} + \delta\boldsymbol{\mu}_{t,j}$$

Using the projection matrix P: $P = \begin{bmatrix} I & 0 \end{bmatrix}$ and the first-order Taylor series approximation, we could rewrite the observation model with perturbation $\delta\boldsymbol{\mu}_{t,j}$:

$$\begin{aligned} \boldsymbol{z}_{t,i} &= M\pi\left(oT_1 U_t\left(\boldsymbol{\mu}_{t,j} + \delta\boldsymbol{\mu}_{t,j}\right)\right) + \mathbf{v}_{t,i} \\ &= M\pi\left(oT_1 U_t\left(\underline{\boldsymbol{\mu}}_{t,j} + P^{\top}\delta\boldsymbol{\mu}_{t,j}\right)\right) + \mathbf{v}_{t,i} \\ &\approx \underbrace{M\pi\left(oT_1 U_t \underline{\mu}_{t,j}\right)}_{z_{t,i}} + \underbrace{M\frac{d\pi}{d\mathbf{q}}\left(oT_l U_t \underline{\boldsymbol{\mu}}_{t,j}\right) oT_l U_t P^{\top}}_{H_{t,i,j}}\delta\boldsymbol{\mu}_{t,j} + \mathbf{v}_{t,i} \end{aligned}$$

Then, the jacobian of $\tilde{\mathbf{z}}_{t,i}$ with respect to $\mathbf{m}_j$ evaluated at $\mu_{t,j}$ is:

$$H_{t,i,j} = \begin{cases} M\frac{\partial\pi}{d\mathbf{q}}\left(oT_l U_t \underline{\boldsymbol{\mu}_{t,j}}\right) oT_l U_t P^{\top} & \text{if observation } i \text{ corresponds} \\ & \text{to landmark} j \text{ at time } t \\ \mathbf{0} \in \mathbb{R}^{4\times 3} & \text{otherwise} \end{cases}$$

However, at each time t, the dimension of H_t is too large($\mathbb{R}^{4N_t\times 3M}$), which would cost really a lot time. Hence, the more practical way is to select H_t with the dimension $\mathbb{R}^{4N_t\times 3N_t}$. The modification could greatly save much time.

Besides, if the landmark j has never been detected before, we simply compute its position based on the car's current pose and the measurement data without any update.

Finally, perform the EKF update:

$$K_t = \Sigma_t H_t^\top \left(H_t \Sigma_t H_t^\top + I \otimes V\right)^{-1}$$

$$\mu_{t+1} = \mu_t + K_t\left(z_t - \tilde{z}_t\right) \qquad I \otimes V := \begin{bmatrix} V & & \\ & \ddots & \\ & & V \end{bmatrix}$$

$$\Sigma_{t+1} = \left(I - K_t H_t\right)$$

C. Visual-Inertial SLAM

In the section **IMU-based Localization via EKF Prediction**, we already obtain the predicted mean and covariance based on the motion model. The next step is to update the mean and covariance based on the measurement model.

The prior probability distribution here is the predicted mean and covariance:Prior: $U_{t+1}|z_{0:t}, u_{0:t} \sim \mathcal{N}\left(\boldsymbol{\mu}_{t+1|t}, \Sigma_{t+1|t}\right)$ with $\boldsymbol{\mu}_{t+1|t} \in SE(3)$ and $\Sigma_{t+1|t} \in \mathbb{R}^{6\times 6}$.

The observation model is the same as in the visual mapping problem but this time the variable of interest is the inverse IMU pose $U_{t+1} \in SE(3)$ instead of the landmark positions $\mathbf{m} \in \mathbb{R}^{3\times M}$.

We need the observation model Jacobian $H_{t+1|t} \in \mathbb{R}^{4N_t\times 6}$ with respect to the inverse IMU pose U_t, evaluated at $\mu_{t+1|t}$. Let the elements of $H_{t+1lt} \in \mathbb{R}^{4N_t\times 6}$ corresponding to different observations i be $H_{i,t+1|t} \in \mathbb{R}^{4\times 6}$.

The first-order Taylor series approximation of observation i at time $t+1$ using an inverse IMU pose perturbation $\delta\boldsymbol{\mu}_{t+1|t+1}$ is:

$$\begin{aligned}
&\mathbf{z}_{t+1,i} \\
&= M\pi\left(oT_1 \exp\left(\delta\boldsymbol{\mu}_{t+1|t+1}\right)\boldsymbol{\mu}_{t+1|t}\mathbf{m}_t\right) + \mathbf{v}_{t+1,i} \\
&\approx M\pi\left(oT_t\left(1 + \delta\boldsymbol{\mu}_{t+1|t+1}\right)\boldsymbol{\mu}_{t+1|t}\mathbf{m}_{t,t,i}\right) + \mathbf{v}_{t+1,i} \\
&= M\pi\left(oT_1\boldsymbol{\mu}_{t+1|t}\mathbf{m}_j + oT_t\left(\boldsymbol{\mu}_{t+1|t}\mathbf{m}_j\right)^{\odot}\delta\boldsymbol{\mu}_{t+1|t+1}\right) + \mathbf{v}_{t+1,i} \\
&\approx \underbrace{M\pi\left(oT_t\boldsymbol{\mu}_{t+1|t}\mathbf{m}_j\right)}_{\mathbf{z}_{t+1,i}} \\
&+ \underbrace{M\frac{d\pi}{d\mathbf{q}}\left(o^T I_{t,k}\boldsymbol{\mu}_{t+1|t}\mathbf{m}_j\right) oT_l\left(\boldsymbol{\mu}_{t+1|t}\mathbf{m}_j\right)^{\odot}}_{\mathbf{H}_{t,k+1|t}}\delta\boldsymbol{\mu}_{t+1|t+1} + \mathbf{v}_{t+1,i}
\end{aligned}$$

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

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

Predicted observation based on $\mu_{t+1|t}$ and known correspondences π_t :

$$\tilde{\mathbf{z}}_{t+1,i} := M\pi\left(oT_l\boldsymbol{\mu}_{t+1|t}\mathbf{m}_j\right) \qquad \text{for } i = 1, \ldots, N_t$$

Jacobian of $\tilde{\mathbf{z}}_{t+1,i}$ with respect to U_{t+1} evaluated at $\mu_{t+1|t}$

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

Finally, perform the EKF update:

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

$$\text{where, } H_{t+1|t} = \begin{bmatrix} H_{1,t+1|t} \\ \vdots \\ H_{N_{t+1},t+1|t} \end{bmatrix}$$

D. Joint update for the IMU and landmarks

In the previous subsection, we have successfully implemented the IMU pose update via EKF. Right now, the last step is to put the update for the IMU pose and landmarks together into a large $(3M+6)\times(3M+6)$ covariance matrix and associated mean.

In other words, we have to maintain the mean value vector with the dimension $(3M+6)\times 1$ and the covariance matrix with the dimension $(3M+6)\times(3M+6)$. Since from the beginning, we have no ideas of the the position of each landmarks, the mean value for them could be safely initialized as 0(the same in the part b). Also, we could set the starting point as the origin, so the mean value of car's position at $t = 0$ is also 0. For the covariance matrix, the left upper submatrix(6×6) is the the same as in the previous subsection and the right lower submatrix(3M $\times$3M)) is the same as in the part (b). Hence, the initial value for the mean vector and the covariance matrix could be set as:

$$\mu_0 = \begin{pmatrix} 0 & 0 & 0 & \ldots & 0 \end{pmatrix}^T$$

$$\Sigma_0 = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \infty & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & \infty
\end{pmatrix}$$

In practice, we could set a very large number instead of infinity.

Then, the update step is the same as in the part(b) and in the previous subsection. The elements in H matrix could be computed by the method mentioned before. For each observation, we could compute $H_{t,i,j}$ and $H_{i,t+1|t}$, respectively. Hence, the dimension of the joint H matrix should be $4N_t \times (3\times M + 6)$.

The main reason why we need this joint updata is that a separate update is less accurate because it ignores the nonzero correlation between the map state and the IMU inverse pose.

IV. Results

A. IMU-based Localization via EKF Prediction

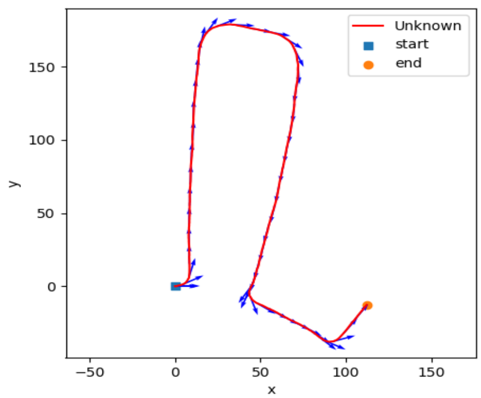

Fig. 2. results for data 0022

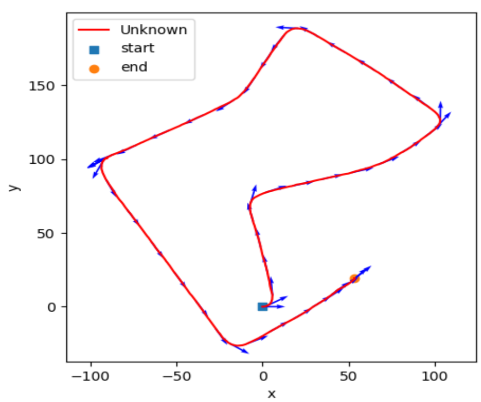

Fig. 3. results for data 0027

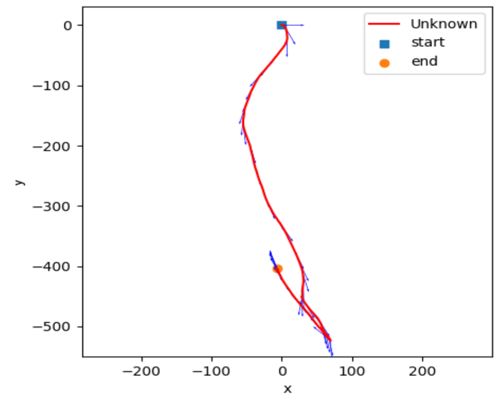

Fig. 4. results for data 0034

In the above figures, you could see the random noise in the IMU data could bring in some problems when we compute the car's trajectory. For example, in Fig 3, the trajectory is not a closed loop, but from the video, it should be. Besides, in Fig 4, the trajectory is a little strange, because it seems the car spin 180° around its own axis, which is impossible for any cars.

In the code, I set the initial mean as 0-vector and the initial covariance as 0-matrix. Besides, I set the motion noise as an identity matrix.

B. Landmark Mapping via EKF Update

I assume the trajectories are perfect and compute the positions of each landmark.

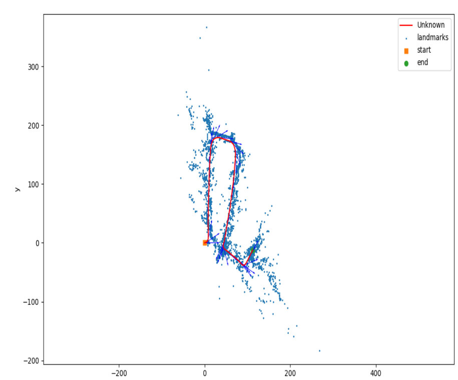

Fig. 5. results for data 0022

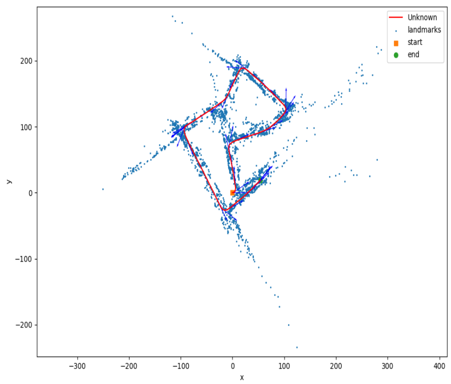

Fig. 6. results for data 0027

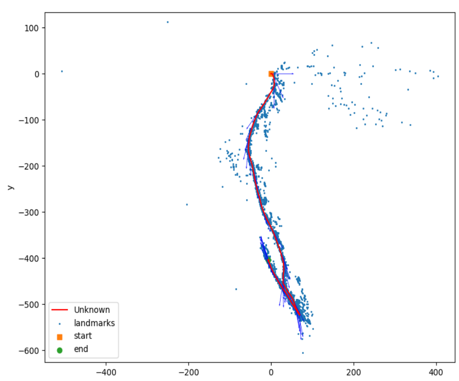

Fig. 7. results for data 0034

In the above figures, I mark the landmarks on the map. Here, I set the measurement noise as 0.5 pixel.

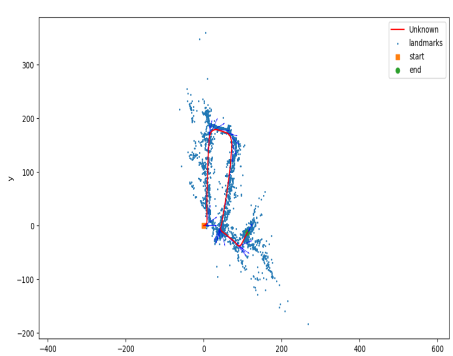

Fig. 8. results for data 0022

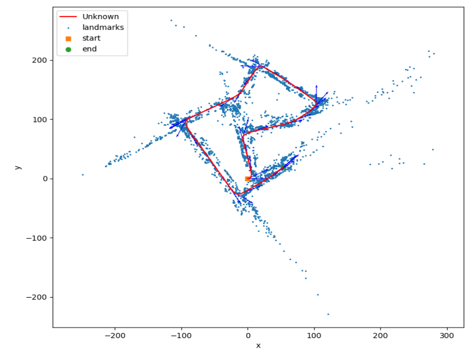

Fig. 9. results for data 0027

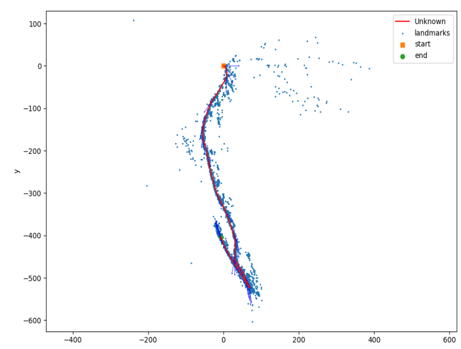

Fig. 10. results for data 0034

Here, I set the measurement noise as 4 pixels.

For the initial value of the mean, since we have no idea of the position of each landmark, I simply set the mean as 0-vector and when the landmark j has been detected for the first time, I compute its position based on the car's current pose and measurement. For the initial value of the covariance, we have no confidence of where they should be, so the uncertainty should be infinite. In practice, they should be a very large number and I simply set them as 1000.

Indeed, there is a little difference shown in the above figures. And it means the mean value of each landmark keeps unchanged. This fact makes me confused at first time, but when I compare the covariance matrix, I find they are not the same. Besides, when the measurement noise is set as 0.5, the elements in the covariance matrix are smaller. This is true, because if we think the sensor data are more reliable, the uncertainty of the final results should be smaller. In this case, if the measurement noise is only 0.5 pixel, the covariance should be lesser compared it with the measurement noise as 4 pixels, which means we are more prone to believe the results are correct.

C. *Visual-Inertial SLAM*

In this part, I need to combine the previous two parts and update the pose based on the observation model.

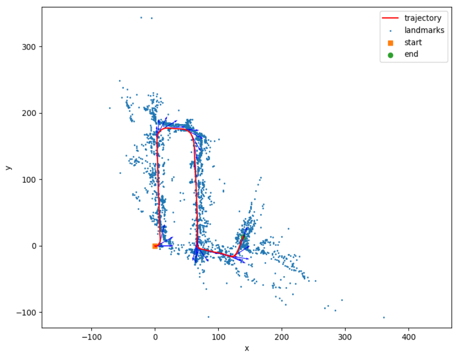

Fig. 11. results for data 0022

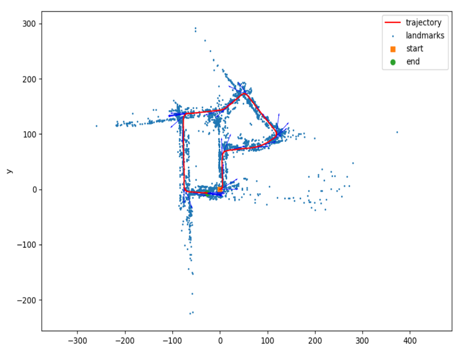

Fig. 12. results for data 0027

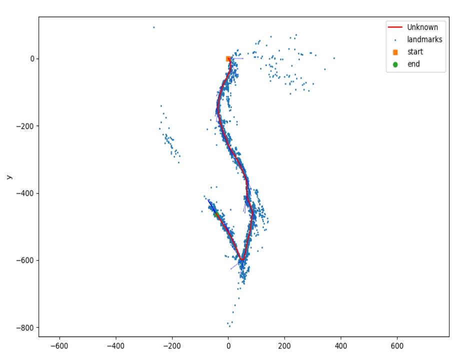

Fig. 13. results for data 0034

In the above figures, I set the measurement noise as 4 pixels and motion noise as an identity matrix.

You could see the estimated trajectories are much better than then in the first part. For example, in Fig 12, the estimated trajectory is right now a closed loop and in Fig 13, the estimated trajectory is not strange any more. You could see there is a U-turn.

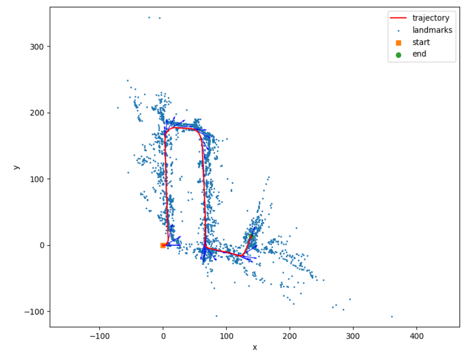

Fig. 14. results for data 0022

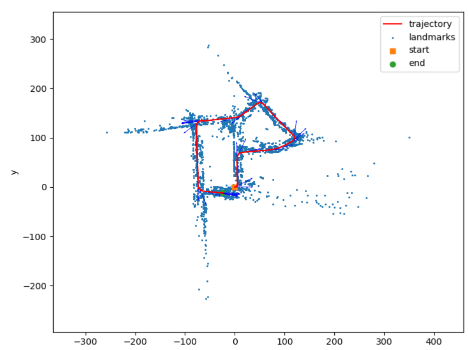

Fig. 15. results for data 0027

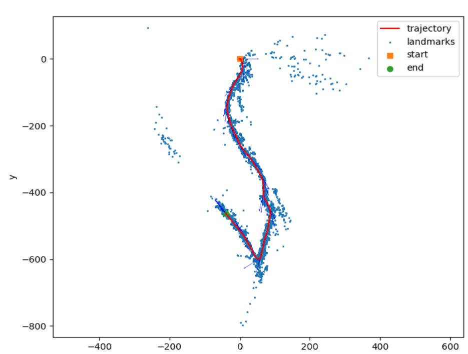

Fig. 16. results for data 0034

In the above figures, I set the measurement noise as 10 pixels and motion noise as an identity matrix. The estimated trajectories are the same, but the covariance matrix of the car's pose are not.

```
4:
[[ 2.10878761e-05  1.91525331e-06 -3.66870809e-06 -
8.80370433e-08 -1.18345135e-07 -1.17757536e-07]
 [ 1.91543679e-06  2.01302448e-05 -1.23379479e-06 -
4.89240536e-07 -6.03962598e-08 -8.85976287e-07]
 [-3.66867548e-06 -1.23386023e-06  1.47382623e-05
6.37254381e-08 6.61001405e-07  6.39452921e-08]
 [-8.72626835e-08 -4.89335791e-07  6.38304394e-08
1.08217423e-07 5.11522985e-09  1.86591318e-08]
 [-1.18799295e-07 -6.01300764e-08  6.61184052e-07
3.40275619e-09 3.73425716e-08  1.77712395e-09]
 [-1.15742407e-07 -8.85145129e-07  6.43645773e-08
2.07795319e-08 2.79741506e-09  4.99649123e-08]]
```

Fig. 17. Covariance matrix for 0022(measurement noise:4)

```
10:
[[ 1.74934013e-04  1.19356463e-05 -2.98637469e-05 -
6.44683821e-07 -9.56159479e-07 -7.97411576e-07]
 [ 1.19356363e-05  1.62803752e-04 -9.24573269e-06 -
3.92342371e-06 -4.53623332e-07 -7.10598624e-06]
 [-2.98637487e-05 -9.24572804e-06  1.20373719e-04
5.74085642e-07 5.36395334e-06  4.86027064e-07]
 [-6.44598860e-07 -3.92341325e-06  5.74077045e-07
8.77311772e-07 3.23092942e-08  1.61816703e-07]
 [-9.56226004e-07 -4.53682680e-07  5.36399817e-06
3.20063148e-08 3.05886019e-07  2.29288127e-08]
 [-7.97656127e-07 -7.10604210e-06  4.85990787e-07
1.61948758e-07 2.28215513e-08  3.69776473e-07]]
```

Fig. 18. Covariance matrix for 0022(measurement noise:10)

From the above two figures, I could tell the elements in the first covariance matrix are smaller(you could compare the elements on the diagonal). As I explain before, it means the first results are more reliable, which corresponds to the fact that the measurement noise is smaller.

D. Joint update for the IMU and landmarks

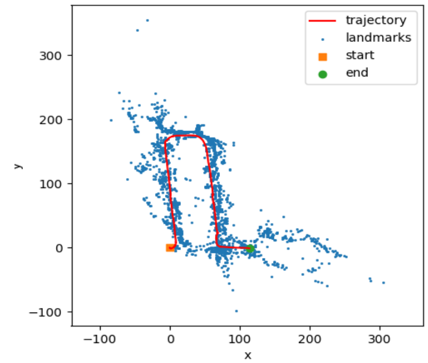

Fig. 19. results for data 0022

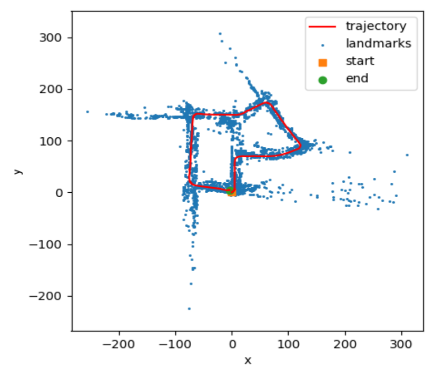

Fig. 20. results for data 0027

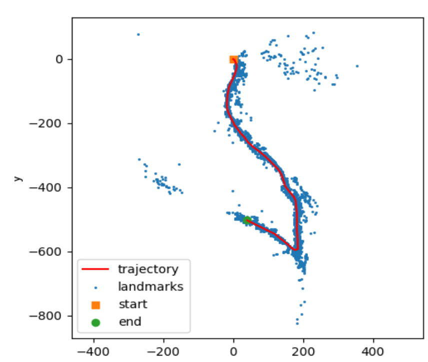

Fig. 21. results for data 0034

From the above three figures, you could see that there is only a few difference compared with the results in the previous subsection, when we don't jointly update the mean and covariance. I think maybe the correlation between robot's pose and the landmarks are not very strong, and even though we choose to ignore them, we still could obtain some good results.

V. Acknowledgment

I am grateful to the whole Python communities for providing us with so many powerful tools.

References

[1] Sebastian Thrun, Wolfram Burgard and Dieter Fox, Probabilistic robotics, MIT Press, 647 pp