Visual Inertial SLAM

Albert Tan
Electrical and Computer Engineering
University of California, San Diego
San Diego, California
aktan@eng.ucsd.edu

I. Introduction

Autonomous vehicles are equipped with sensors that give information about the vehicle's motion, while observations from a stereo camera define the vehicle's observation model. SLAM (Simultaneous Localization and Mapping) is a technique that combines data from these two modes of data (motion and observation) to perform mapping and localization, simultaneously. At each time step, the position is used to create a landmark mapping, which in turn is used to localize the robot. The result is a loop-closure of the entire robot trajectory and its corresponding mapping. The EKF (Extended Kalman Filter), a special case of the more general Bayes filter, is implemented for the task.

II. Problem Formulation

A. SLAM Problem

SLAM is a parameter estimation problem for $\boldsymbol{x}_{0:T}$ and $\mathbf{m}$ given a dataset of the robot inputs $\boldsymbol{u}_{0:T-1}$ and observations $\boldsymbol{z}_{0:T}$. In general, we relate all those parameters as a joint pdf :

$$p(\mathbf{x}_{0:T}, \mathbf{m}, \mathbf{z}_{0:T}, \mathbf{u}_{0:T-1}) = \underbrace{p_0(\mathbf{x}_0, \mathbf{m})}_{\text{prior}} \prod_{t=0}^{T} \underbrace{p_h(\mathbf{z}_t \mid \mathbf{x}_t, \mathbf{m})}_{\text{observation model}} \prod_{t=1}^{T} \underbrace{p_f(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_{t-1})}_{\text{motion model}} \prod_{t=0}^{T-1} \underbrace{p(\mathbf{u}_t \mid \mathbf{x}_t)}_{\text{control policy}}$$

where the joint pdf is decomposed through conditional probability, Bayes rule, and Markov belief network properties. Depending on the observation model and motion model, different types of bayes filters can be used for SLAM.

B. Localization

Given a map $\mathbf{m}$, a sequence control inputs $\boldsymbol{u}_{0:T-1}$, and a sequence of measurements $\boldsymbol{z}_{0:t}$, infer the robot state trajectory $\boldsymbol{x}_{0:t}$

C. Landmark Mapping

Given a robot state trajectory $\boldsymbol{x}_{0:t}$ and a sequence of measurements $\boldsymbol{z}_{0:t}$, build a map $\mathbf{m}$ of the environment where the estimate of the map at each timestep is defined as:

$$\boldsymbol{m} \mid \boldsymbol{z}_{0:t} \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) \text{ where } \boldsymbol{u}_t \in R^{3M} \text{ and } \boldsymbol{\Sigma}_t \in R^{3M \times 3M}$$

D. Robot Motion Model

To determine how the car moves in the environment, we must use a pose kinematics model based on the rotation and twist of the vehicle. The motion model problem reduces to finding a recursive update rule for determining the pose i.e. $T_{k+1} = f(t_k)$ where the motion uses a discrete-time pose kinematics. This motion model must be derived using motion data (linear and angular velocities from an IMU).

E. Robot Observation Model

To determine how the vehicle moves in the environment, an observation model $\boldsymbol{z}_{t,i}$ must be determined as below:

- **Observation Model**: with measurement noise $\mathbf{v}_{t,i} \sim \mathcal{N}(0, V)$

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

where the probability of the observation $\boldsymbol{z}_{t,i}$ at time t uses the stereo camera imaging data with the current pose state $\boldsymbol{T}_t$ and map $\boldsymbol{m}_j$

F. Generalized Bayes Filter

To combine the motion and observation models above, a bayes filter is used. Starting with a prior of any arbitrary particular distribution:

$$\textbf{Prior: } p_{t|t}(\mathbf{x}_t) := p(\mathbf{x}_t \mid \mathbf{z}_{0:t}, \mathbf{u}_{0:t-1})$$

From the initial state distribution, the bayes filter iterates between prediction and update to find the next best estimate of the state through the two equations:

- **Prediction**: $p_{t+1|t}(\mathbf{x}) = \int p_f(\mathbf{x} \mid \mathbf{s}, \mathbf{u}_t) p_{t|t}(\mathbf{s}) d\mathbf{s}$
- **Update**: $p_{t+1|t+1}(\mathbf{x}) = \frac{p_h(\mathbf{z}_{t+1}|\mathbf{x}) p_{t+1|t}(\mathbf{x})}{p(\mathbf{z}_{t+1}|\mathbf{z}_{0:t}, \mathbf{u}_{0:t})} = \frac{p_h(\mathbf{z}_{t+1}|\mathbf{x}) p_{t+1|t}(\mathbf{x})}{\int p_h(\mathbf{z}_{t+1}|\mathbf{s}) p_{t+1|t}(\mathbf{s}) d\mathbf{s}}$

Similar to the prior, the prediction and updates can take on any arbitrary form. Using Taylor series approximations and Gaussian assumptions as described in the technical approach, these equations can be simplified into a tractable implementation where the distributions are all Gaussian.

G. Data Sensors and Dead Reckoning

Since the SLAM problem requires a motion model and observation model, they be evaluated by using the data from sensors above. This introduces the task of processing those sensors to solve the task of dead-reckoning.

The observations come in the form of images from a stereo camera. Each image contains features which are matched with its corresponding landmark via data association. These features of each observation are then used to initialize landmarks if they are first seen, or update landmarks if they have been seen previously.

The kinematics data comes in the form of angular and linear velocities, which are used to define the pose of the car at each timestep.

III. Technical Approach

A. Extended Kalman Filter

The Bayes Filter made no assumptions on neither thee prior nor the nor observation and motion model. In this project, an Extended Kalman Filter is used to produce the best estimate of the position and landmarks at each timestep. It is a special case of the bayes filter with the following assumptions:

- The prior pdf $p_{t|t}$ is Gaussian
- The motion model is linear in the state $\mathbf{x}_t$ with Gaussian noise $\mathbf{w}_t$
- The observation model is linear in the state $\mathbf{x}_t$ with Gaussian noise $\mathbf{v}_t$
- The motion noise $\mathbf{w}_t$ and observation noise $\mathbf{v}_t$ are independent of each other, of the state $\mathbf{x}_t$, and across time

In practice, the motion and observation models are not linear in the state, which breaks the gaussian assumption of the predict and update step of the Kalman Filter. However, the EKF forces this linearity by using Taylor series to get a linear approximation of the motion and observation models.

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

With this linear approximation, the EKF algorithm has motion model:

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t), \quad \mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, W)$$
$$F_t := \frac{df}{d\mathbf{x}}(\boldsymbol{\mu}_{t|t}, \mathbf{u}_t, \mathbf{0}), \quad Q_t := \frac{df}{d\mathbf{w}}(\boldsymbol{\mu}_{t|t}, \mathbf{u}_t, \mathbf{0})$$

where F_t is the Jacobian of the motion model w.r.t the state and Q_T is the Jacobian w.r.t. the control inputs.

The observation model is defined as:

$$\mathbf{z}_t = h(\mathbf{x}_t, \mathbf{v}_t), \quad \mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, V)$$
$$H_t := \frac{dh}{d\mathbf{x}}(\boldsymbol{\mu}_{t|t-1}, \mathbf{0}), \quad R_t := \frac{dh}{d\mathbf{v}}(\boldsymbol{\mu}_{t|t-1}, \mathbf{0})$$

where H_t is the Jacobian of the predicted observations w.r.t to the landmarks and R_T is the Jacobian w.r.t. the noise.

The EKF then alternates between the prediction and update step:

$$\begin{aligned}
&\text{Prediction:} && \boldsymbol{\mu}_{t+1|t} = f(\boldsymbol{\mu}_{t|t}, \mathbf{u}_t, \mathbf{0}) \\
& && \Sigma_{t+1|t} = F_t \Sigma_{t|t} F_t^\top + Q_t W Q_t^\top \\
&\text{Update:} && \boldsymbol{\mu}_{t+1|t+1} = \boldsymbol{\mu}_{t+1|t} + K_{t+1|t}(z_{t+1} - h(\boldsymbol{\mu}_{t+1|t}, 0)) \\
& && \Sigma_{t+1|t+1} = (I - K_{t+1|t} H_{t+1}) \Sigma_{t+1|t} \\
&\text{Kalman Gain:} && K_{t+1|t} := \Sigma_{t+1|t} H_{t+1}^\top \left(H_{t+1} \Sigma_{t+1|t} H_{t+1}^\top + R_{t+1} V R_{t+1}^\top\right)^{-1}
\end{aligned}$$

where due to the linear approximation of the Taylor Series, the distribution at each time-step is Gaussian. As a result, the EKF reduces to finding the mean and variance at each timestep of the prediction and update. The Kalman Gain $K_{t+1|t}$ is used to scale the correction based on how much we trust the measurements, and is based on the variance of the observations. The higher the variance, the less the gain, and vice-versa.

B. IMU-based Localization via EKF Prediction

To first determine the baseline trajectory for SLAM, a prediction-only trajectory must first be obtained. Keeping track of the pose $\mu_{t|t} \in SE(3)$ at each timestep, the prediction can be calculated as follows:

$$\boldsymbol{\mu}_{t+1|t} = \boldsymbol{\mu}_{t|t} \exp\left(\tau \hat{\mathbf{u}}_t\right)$$

where

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

The control input u_t is the concatenation of the linear and angular velocity of the IMU, which is then passed through a hat-map to get the nominal split of the perturbation kinematics. In discrete time, the perturbation scales the current position with $\exp(\tau \widehat{u_t})$, which can be implemented with the scipi.expm function.

C. Landmark Mapping via EKF Update

Once the predicted IMU trajectory was obtained, the next step was to perform landmark mapping to estimate the landmark positions. Specifically, an update-only EKF was used estimate the unknown landmark position $\boldsymbol{m}$ as a state $\boldsymbol{\mu}_{t|t}$. Since the landmarks are assumed static, no prediction is needed for the EKF of the observations.

The extrinsics of the stereo camera were provded via the the matrix M:

$$M := \begin{bmatrix} fs_u & 0 & c_u & 0 \\ 0 & fs_v & c_v & 0 \\ fs_u & 0 & c_u & -fs_u b \\ 0 & fs_v & c_v & 0 \end{bmatrix}$$

If the landmarks were first seen, their positions were calculated based on hw 2 problem 4 by solving the following system of equations:

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

By multiplying through the rows and columns, the landmark coordinates in the camera frame can be found by:

$$\boxed{z = \frac{f s_u b}{u_L - u_R} = \frac{K_{11} b}{u_L - u_R}}$$

$$\boxed{y = z\left(\frac{v_L - c_v}{f s_v}\right) = z\left(\frac{v_L - K_{23}}{K_{22}}\right)}$$

$$\boxed{x = z\left(\frac{u_L - c_u}{f s_u}\right) = z\left(\frac{u_L - K_{13}}{K_{11}}\right)}$$

Because these coordinates are in the camera frame, the world frame coordinates must be found through pose transformations:

$$\underline{S_{World}} = {}_{World}T_{IMU} *_{IMU} T_{camera} * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

The previous result assumed the landmark was seen for the first time. If the landmark was already seen at a previous timestep, the coordinates would be updated based on the corresponding past observations to produce a better estimate of the landmark positions:

$$K_{t+1} = \Sigma_t H_{t+1}^\top \left(H_{t+1}\Sigma_t H_{t+1}^\top + I \otimes V\right)^{-1}$$
$$\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t + K_{t+1}\Big(\mathbf{z}_{t+1} - \underbrace{M\pi\left({}_O T_I\, T_{t+1}^{-1}\underline{\boldsymbol{\mu}}_t\right)}_{\tilde{\mathbf{z}}_{t+1}}\Big)$$
$$\Sigma_{t+1} = (I - K_{t+1}H_{t+1})\Sigma_t$$

We first calculate the innovation, or error between the current pixel observations and the reprojection $\tilde{z}$:

$$\left(\mathbf{z}_{t+1} - \tilde{\mathbf{z}}_{t+1}\right)$$

where the reprojection $\tilde{z}$ is defined as:

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

with the projection equations

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

One big problem is knowing the data association between $\boldsymbol{z_i}$ and the estimates of the map $\boldsymbol{\mu_j}$. With dataset 10.npz, that data association $\Delta_t: \{1, \ldots, M\} \to \{1, \ldots, N_t\}$ was provided and thus the current estimates $\mu_{t,j}$ could be directly matched with the corresponding observation and used to calculate the reprojection and innovation.

The Kalman Gain was then used to scale the innovation based on how much we trust the observation. To determine K_{t+1} we first compute the Jacobian H_{t+1}, or the jacobian of $\tilde{z}$ w.r.t. the observations:

$$H_{t+1,i,j} = \begin{cases} M\frac{d\pi}{d\mathbf{q}}\left({}_O T_I\, T_{t+1}^{-1}\underline{\boldsymbol{\mu}}_{t,j}\right){}_O T_I\, T_{t+1}^{-1} P^\top & \text{if } \Delta_t(j) = i, \\ \mathbf{0}, \in \mathbb{R}^{4\times 3} & \text{otherwise} \end{cases}$$

where $H \in 4N_t \times 3M$ and $K \in 3M \times 4N_t$.

Combining these steps creates an update-only EKF where at each timestep, which creates a better estimate of the map at each time-step by taking in initial observations and correcting them if they were previously seen.

D. Visual Inertial SLAM

Combining the IMU-based prediction with the EKF update landmark mapping, SLAM leverages the map to localize the position at each timestep, which in turn creates a better map.

In the mapping-only step, the landmarks were uncorrelated. In SLAM, the landmarks are now correlated. We perform sensor fusion between IMU and images by maintaining a joint covariance $\Sigma = \begin{bmatrix} \Sigma_T & \mathbf{0} \\ \mathbf{0} & \Sigma_M \end{bmatrix} \in R^{(3M+6)\times(3M+6)}$ where the extra 6 dimensions come from 6 degrees of freedom in the lie algebra of the pose.

The SLAM prediction step (only μ_T and Σ_T, or the mean and variance of the pose T_t , needs to be predicted):

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

Writing in terms of the joint covariance matrices:

$$\mu_T^{predicted} = \mu_T^{updated}\exp\left(\tau\hat{\mu}\right)$$

$$\Sigma_T^{predicted} = \begin{bmatrix}\exp\left(-\tau\hat{\mu}\right) & I\end{bmatrix}\begin{bmatrix}\Sigma_T & \mathbf{0} \\ \mathbf{0} & \Sigma_M\end{bmatrix}\begin{bmatrix}\exp(-\tau\hat{\mu}) & I\end{bmatrix}^T + \begin{bmatrix}W & \mathbf{0} \\ \mathbf{0} & \mathbf{0}\end{bmatrix}$$

The SLAM update can then be written as:

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

We break this update step down into 3 phases:

1. update μ_M
2. update μ_T
3. update the joint covariance $\Sigma = \begin{bmatrix}\Sigma_T & \mathbf{0} \\ \mathbf{0} & \Sigma_M\end{bmatrix}$

Similar to the joint covariance, we use sensor fusion to exploit the correlation between the landmarks by concatenating the Jacobians of the motion and observations as follows:

$$H = [H_T \quad H_M] \in R^{(4N)\times(3M+6)}$$

where $H_T \in R^{4N\times 6}$ and $H_M \in R^{4N\times 3M}$

We know H_M, the Jacobian of the predicted observations $\tilde{z}$ w.r.t. the landmarks $\boldsymbol{m}$:

$$H_{t+1,i,j} = \begin{cases} M\frac{d\pi}{d\mathbf{q}}\left({}_O T_I\, T_{t+1}^{-1}\underline{\boldsymbol{\mu}}_{t,j}\right){}_O T_I\, T_{t+1}^{-1} P^\top & \text{if } \Delta_t(j) = i, \\ \mathbf{0}, \in \mathbb{R}^{4\times 3} & \text{otherwise} \end{cases}$$

We must now find H_T in SLAM, or the Jacobian of the predicted observations $\tilde{z}$ w.r.t. the pose T_{t+1} :

Jacobian of $\tilde{\mathbf{z}}_{t+1,i}$ with respect to T_{t+1} evaluated at $\boldsymbol{\mu}_{t+1|t}$:

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

This joint Jacobian $H = [H_T, H_M]$ can then be obtained. We use it to determine the joint Kalman Gain $K_{t+1} \in R^{(3M+6)\times(4N)}$:

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

where the noise V must be empirically defined due to the very high sensitivity of SLAM

To update μ_M:

$$\mu_M^{update} = \mu_M^{predicted} + \zeta_{7:3M+6}$$

and update μ_T:

$$\mu_T^{update} = \mu_T^{predicted} * \exp\left(\hat{\zeta}_{1:6}\right)$$

where $\zeta = K_{joint}(z - \tilde{z}) = \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \vdots \\ \zeta_{3M+6} \end{bmatrix}$

Finally, we update the joint covariances all at once:

$$\Sigma^{\text{updated}} = \left(I - K_{joint}H_{joint}\right)\Sigma^{\text{predicted}} \in R^{(3M+6)\times(3M+6)}$$

IV. Results

Ideally, the EKF SLAM should create a more accurate map and trajectory compared to just the IMU localization prediction (part a) and landmark mapping via EKF update (part c).

We perform SLAM on the KITTI 360 dataset with reference to the following video of a car navigating through an urban city: https://www.youtube.com/watch?v=uusgX7XPoIg&ab_channel=AlbertTan

For the IMU localization, the trajectory was determined as:

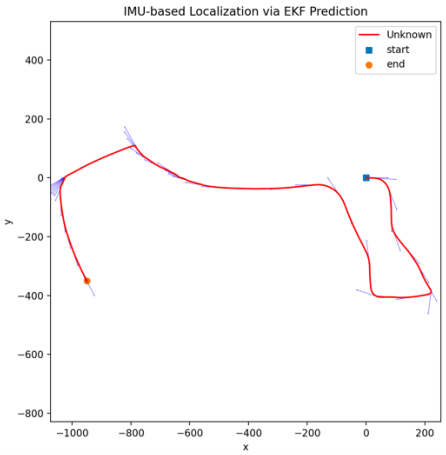

Figure 5. IMU-based localization

Assuming the trajectory from the IMU-based localization is correct, the landmark mapping via EKA update was found over time:

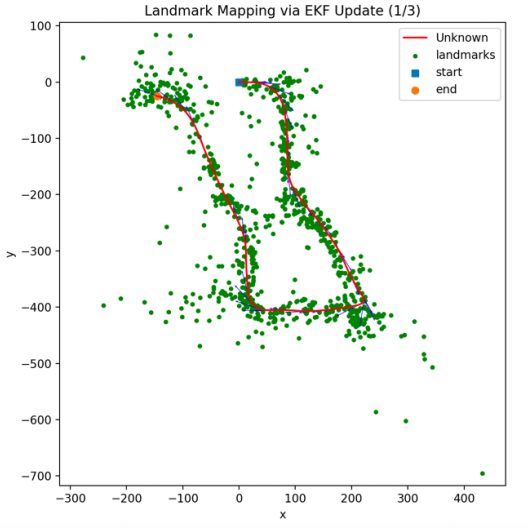

Figure 1. EKF Landmark mapping at 1/3 of completed trajectory

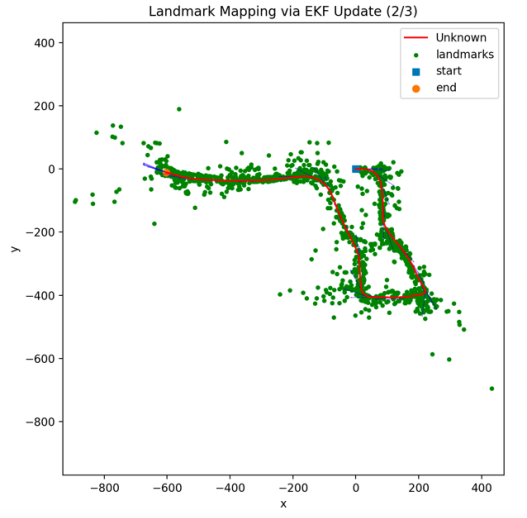

Figure 2. EKF Landmark Mapping at 2/3 of completed trajectory

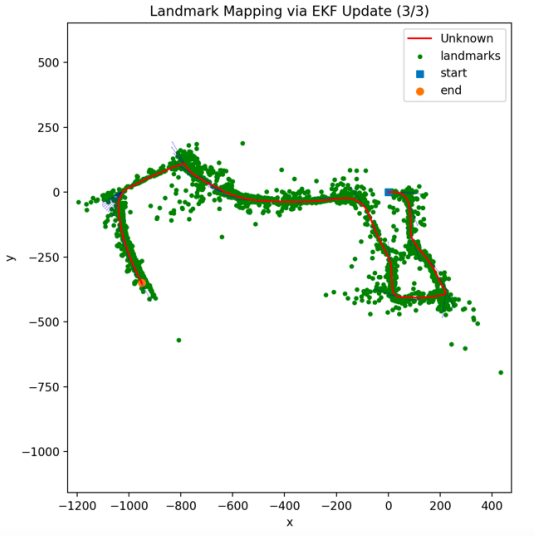

Figure 3. EKF Landmark Mapping over complete trajectory

Based on the results of EKF landmark mapping, the green points are closely aligned with the trajectory of the car in red. I used measurement noise $v_{t,i} \sim N(0, V)$ where $V = 16$, which created the best mapping where the green points were concentrated along the road. Larger values of the noise caused the green landmarks to sporadically appear everywhere on the map.

Overall, Extended Kalman Filters are well suited for the task of leveraging pose data and observations to perform localization and mapping. Even with nonlinearities in the motion and observation model, EKF is capable of maintaining the best Gaussian estimate of the state through Taylor Series linear approximations to create a closed-loop feedback system for SLAM.