# ECE 276A: Visual-Inertial SLAM via the Extended Kalman Filter

or Why I Lack Social Skills

Roumen Guha
*Department of Electrical and Computer Engineering*
*University of California, San Diego*
La Jolla, United States
roumen.guha@gmail.com

***Abstract*—This paper presents a particular solution to tackle the problem of simultaneous localization and maximization (SLAM) via the Extended Kalman Filter algorithm (EKF) with a visual-inertial sensor configuration. Very briefly, we implemented an extended Kalman filter to localize a vehicle based on IMU odometry and visual key-points from camera data. We then used this filter to compose a map of the vehicle's surroundings. The results section contains pictures and discussion concerning the effectiveness of this algorithm.**

***Index Terms*—visual, inertial, odometry, extended Kalman filter, localization, mapping**

## I. Introduction

The goal of this project was to simultaneously localize a vehicle in its surroundings using the given IMU odometry readings and visual key-point features obtained from a stereo-camera configuration. Further, using these readings, we are able to generate a map this previously unknown environment. Thus, this a visual-inertial SLAM (VI-SLAM) problem. An extended filter approach was used for this task.

Kalman filters (a form of a Bayesian filter) can effectively model a robot's location in the world as a Gaussian distribution based on past control inputs and current observations. The variance of the distribution reduces as we iterate, signalling that we become more confident with our predictions as we continue. The Extended Kalman filter is used when there are non-linearities present in our motion or observation models, which is the situation we have here in our observation model.

In the Problem Formulation section, we will succinctly model the filter in mathematical terms. In the Technical Approach section, we will detail the implementation of the Extended Kalman filter, discussing how the prediction, update, and resampling steps were chose, as well as some implementation details that might be interesting for further study. Finally, in the Results section, we will discuss the performance of the algorithm with the chosen hyperparamaters on the given odometry and visual key-point datasets.

This project was worked on with Maria Harris, Duke Lin, Shubha Bhaskaran, Stephen West, and Will Argus.

## II. Problem Formulation

In this section, we precisely define the quantities we are interested in. We also cover some necessary knowledge necessary to understand how this algorithm was implemented.

SLAM ultimately is a fairly simple idea, though it is a chicken-and-egg problem. Namely:

- **Mapping:** given the robot state trajectory $\mathbf{x}_{0:T}$, build a map $\mathbf{m}$ of the environment
- **Localization:** given a map $\mathbf{m}$ of the environment, localize the robot and estimate its trajectory $\mathbf{x}_{0:T}$

This ends up becoming a parameter estimation problem for the parameters $\mathbf{x}_{0:T}$ and $\mathbf{m}$. Since we are given a dataset of visual key-point observations $\mathbf{z}_{0:T}$ and control inputs $\mathbf{u}_{0:T-1}$, we can use Bayesian filtering to find the appropriate posterior likelihood of the parameters.

Bayesian filtering, like other solutions to the SLAM problem, exploit the decomposition of the joint probability density function via the Markov assumptions:

$$p(\mathbf{x}_{0:T}, \mathbf{m}, \mathbf{z}_{0:T}, \mathbf{u}_{0:T-1}) = p_{0|0}(\mathbf{x}_0, \mathbf{m}) \prod_{t=0}^{T} p_h(\mathbf{z}_t|\mathbf{x}_t, m) \prod_{t=1}^{T} p_f(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_{t-1}) \quad (1)$$

where $p_{0|0}(\mathbf{x}_0, \mathbf{m})$ is the prior, $p_h$ is the distribution of the observation model and $p_f$ is the distribution of the motion model.

### A. Localization Problem and the Motion Model

The motion model in the typical SLAM problem is given by:

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t) \quad (2)$$

where robot state $\mathbf{x}_t \sim \mathcal{N}(\boldsymbol{\mu}_{t|t}, \Sigma_{t|t})$, movement noise $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, W)$ is inherent, and control input $\mathbf{u}_t$ is given and known. We make some assumptions. Namely: the state of a robot $\mathbf{x}_{t+1}$ depends only on the the previous input $\mathbf{u}_t$ and state $\mathbf{x}_t$. Hence the motion model of a robot with Markov assumptions can be described in Eq 2.

### B. Mapping Problem and the Observation Model

The observation model in the typical SLAM problem is similarly defined:

$$\mathbf{z}_{t+1} = h(\mathbf{x}_{t+1}, \mathbf{v}_{t+1}) \tag{3}$$

where observation noise $\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, V)$.

In the case of VI-SLAM, this becomes:

$$\mathbf{z}_{t+1} = M\pi({}_oT_i U_{t+1} \underline{\mathbf{m}}) + \mathbf{v}_{t+1} \tag{4}$$

Where $\underline{\mathbf{m}}$ is a vector of all points being observed (in homogeneous coordinates), $M$ is the stereo camera matrix (which we go into depth about in the next subsection), $\mathbf{z}_{t+1}$ is defined as a vector of observations at time $t$, $\pi$ is the projection function given in Eq. 24, $U_{t+1}$ is the inverse IMU pose, and ${}_oT_i$ is the transformation from IMU frame to optical frame. We will discuss these more in depth in Section III.

We make further assumptions here. Namely: our measurement noise $\mathbf{v}_{t+1}$ is independent of both our robot state $\mathbf{x}_t$ and our movement noise $\mathbf{w}_t$ across timesteps.

*1) Stereo Camera Model:* In the stereo camera configuration of our vehicle, we have two cameras that are rigidly connected to one another with a known transformation (a displacement along the optical-frame $x$-axis, i.e. baseline $b$). This allows us to determine the depth of a point from a single stereo observation.

Given an observation $\mathbf{z} = [u_L, u_R, v_L, v_R]^T$ in pixel-coordinates, we can convert these to coordinates in the camera-frame, representing a point $m$, through the following model:

$$\begin{bmatrix} u_L \\ v_L \\ u_R \\ v_R \end{bmatrix} = \underbrace{\begin{bmatrix} fs_u & 0 & c_u & 0 \\ 0 & fs_v & c_v & 0 \\ fs_u & 0 & c_u & -fs_u b \\ 0 & fs_v & c_v & 0 \end{bmatrix}}_{M} \frac{1}{z} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{5}$$

Because of the stereo setup, two rows of the stereo camera matrix $M$ are identical. The vertical coordinates of the two pixel observations are always the same because the epipolar lines in the stereo configuration are horizontal. Thus, the equations for $v_R$ can be dropped, and we can replace the equation for $u_R$ with a **disparity** measurement $d = u_L - u_R = \frac{1}{z} f s_u b$, leading to:

$$\begin{bmatrix} u_L \\ v_L \\ d \end{bmatrix} = \begin{bmatrix} fs_u & 0 & c_u & 0 \\ 0 & fs_v & c_v & 0 \\ 0 & 0 & 0 & fs_u b \end{bmatrix} \frac{1}{z} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{6}$$

We can then find the necessary depth $z$:

$$z = \frac{fs_u b}{u_L - u_R} \tag{7}$$

and use this to obtain values for $x$ and $y$.

## III. Technical Approach

In this section, we discuss the algorithms implemented in this project.

### A. Localization

In the typical EKF formulation, the motion model given by the function $f$ in Eq. 2 need not be linear, but for the EKF algorithm to work, we must approximate non-linear $f$ using a first-order Taylor series approximation to evaluate $f(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t)$:

$$f(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t) \approx f(\mathbf{x}_t, \mathbf{u}_t, \mathbf{0}) + F(\mathbf{x}_t - \boldsymbol{\mu}_{t|t}) + Q\mathbf{w}_t \tag{8}$$

where Jacobians $F_t := \frac{df}{d\mathbf{x}}(\boldsymbol{\mu}_{t|t}, \mathbf{u_t}, \mathbf{0})$, and $Q_t := \frac{df}{d\mathbf{w}}(\mathbf{x}_t, \mathbf{u}_t, \mathbf{0})$. Since $\mathbf{w_t}$ and $\mathbf{x_t}$ are independent of each other, and because the approximation given in Eq. 8 is linear, we know the distribution of $\mathbf{x}_{t+1}$ is Gaussian, i.e. $\mathbf{x}_{t+1} \sim \mathcal{N}(\boldsymbol{\mu}_{t+1|t}, \Sigma_{t+1|t})$, where:

$$\boldsymbol{\mu}_{t+1|t} = f(\boldsymbol{\mu}_{t|t}, \mathbf{u}_t, \mathbf{0}) \tag{9}$$

$$\Sigma_{t+1|t} = F_t \Sigma_{t|t} F_t^T + Q_t W Q_t^T \tag{10}$$

We begin our localization problem in any given timestep $t$ using our IMU readings to get some idea of our new position in space, after a control input $\mathbf{u}_t$ has been applied. Our motion model, in this system, is a discretized version of nominal and perturbed kinematics. This allows us, with some time-discretization $\tau$, to employ the exponential map, allowing us to perform the perturbations from $\mathfrak{se}(3)$ in SE(3):

$$\boldsymbol{\mu}_{t+1|t} = exp(-\tau \hat{\mathbf{u}}_t) \boldsymbol{\mu}_{t|t} \tag{11}$$

where $\hat{\mathbf{u}}_t \in \mathbb{R}^{4\times4}$ represents the hat map (skew symmetric matrix) of the velocity vector $\mathbf{u}_t = [\boldsymbol{\lambda}_t, \boldsymbol{\omega}_t]^T \in \mathbb{R}^6$ composed of the linear velocity vector $\boldsymbol{\lambda}_t \in \mathbb{R}^3$ and rotational velocity vector $\boldsymbol{\omega}_t \in \mathbb{R}^3$:

$$\hat{\mathbf{u}}_t := \begin{bmatrix} \hat{\boldsymbol{\omega}}_t & \hat{\boldsymbol{\lambda}}_t \\ \mathbf{0} & 0 \end{bmatrix} \in \mathbb{R}^{4\times4} \tag{12}$$

Here, for any vector $\mathbf{s} \in \mathbb{R}^3$,

$$\hat{\mathbf{s}} = \begin{bmatrix} 0 & -s_3 & s_2 \\ s_3 & 0 & -s_1 \\ -s_2 & s_1 & 0 \end{bmatrix} \in \mathbb{R}^{3\times3} \tag{13}$$

Similarly, we can form the prediction step for the covariance:

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

where $\overset{\curlywedge}{\mathbf{u}}_t \in \mathbb{R}^{6\times6}$ represents the twist on the velocity vector:

$$\overset{\curlywedge}{\mathbf{u}}_t := \begin{bmatrix} \hat{\boldsymbol{\omega}}_t & \hat{\boldsymbol{\lambda}}_t \\ 0 & \hat{\boldsymbol{\omega}}_t \end{bmatrix} \in \mathbb{R}^{6\times6} \tag{15}$$

In our implementation, $W$ was set to:

$$W = \begin{bmatrix} \sigma^2_{\lambda} I_3 & 0 \\ 0 & \sigma^2_{\omega} I_3 \end{bmatrix} \in \mathbb{R}^{6\times6} \tag{16}$$

where ${\sigma^2}_{\lambda} = 10^{-5}$, and ${\sigma^2}_{\omega} = 10^{-4}$.

### B. Mapping

Similarly, in the typical EKF formulation, the observation model function $h$ need not be linear because we can use the linear approximation instead:

$$h(\mathbf{x}_{t+1}, \mathbf{v}_{t+1}) \approx h(\boldsymbol{\mu}_{t+1|t}, \mathbf{0}) + H_{t+1}(\mathbf{x}_{t+1} - \boldsymbol{\mu}_{t+1|t}) + R_{t+1}\mathbf{v}_{t+1} \quad (17)$$

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

The conditional Gaussian distribution $x_{t+1}|z_{t+1}$ can then be parameterized by:

$$\boldsymbol{\mu}_{t+1|t+1} = \boldsymbol{\mu}_{t+1|t} + K_{t+1|t}(\mathbf{z}_{t+1} - h(\boldsymbol{\mu}_{t+1|t}, \mathbf{0})) \quad (18)$$

$$\Sigma_{t+1|t+1} = (I - K_{t+1|t}H_{t+1})\Sigma_{t+1|t} \quad (19)$$

where

$$K_{t+1|t} := \Sigma_{t+1|t}H_{t+1}^T(H_{t+1}\Sigma_{t+1|t}H_{t+1}^T + R_{t+1}VR_{t+1}^T)^{-1} \quad (20)$$

is the Kalman gain of the system.

We begin our mapping problem in any given timestep $t$ by transforming the previously unseen features present in the current timestep into world-frame coordinates, initializing the landmarks using these coordinates. We then model the locations of landmarks as Gaussians, and we subsequently update the means and covariances for each landmark at every timestep where we observe their associated feature *after* seeing it for the first time.

As mentioned when we discussed the Stereo Camera Model, we are using the left camera's frame as the basis. This allows us to obtain the world-frame (homogeneous) coordinates by performing the following calculation:

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = {}_wT_{i\,o}T_i^{-1}Z_0K^{-1}\begin{bmatrix} u_L \\ v_L \\ 1 \end{bmatrix} \quad (21)$$

where ${}_wT_i$ represents the transformation matrix from body-frame (IMU) to world-frame coordinates, and ${}_oT_i$ (provided) represents the transformation matrix from body-frame to camera-frame (optical) coordinates, and $K$ is the provided camera calibration matrix, and $Z_0$ is obtained as per Eq. 7. Note that ${}_wT_i = {}_iT_w^{-1} = U_t \in SE(3)$ is simply the inverse pose of the vehicle; i.e. the distribution whose mean and covariance we seek to track and narrow down respectively. Thus, we have ${}_wT_{i,t} = \mathbb{E}[U_t]^{-1}$.

When we encounter a feature corresponding to a previously seen landmark, we update our corresponding mean and covariance using the standard EKF approach. This means calculating a prediction of the landmark in pixel-coordinates:

$$\tilde{\mathbf{z}}_{t,i} = M\pi({}_oT_i\underline{\boldsymbol{\mu}}_{t,j}) \quad (22)$$

where $M$ refers to the stereo camera model given in Eq. 5, and $\pi$ refers to the projection function:

$$\pi(\mathbf{q}) := \frac{1}{q_3}\mathbf{q} \in \mathbb{R}^4 \quad (23)$$

and whose corresponding derivative is given by:

$$\frac{d\pi}{d\mathbf{q}} = \frac{1}{q_3}\begin{bmatrix} 1 & 0 & -\frac{q_1}{q_3} & 0 \\ 0 & 1 & -\frac{q_2}{q_3} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{q_4}{q_3} & 1 \end{bmatrix} \quad (24)$$

Thus the stereo camera Jacobian $H_{t,i,j} \in \mathbb{R}^{4\times3}$ is given by:

$$H_{t,i,j} = M\frac{d\pi}{d\mathbf{q}}({}_oT_{iw}T_i^{-1}\underline{\boldsymbol{\mu}}_{t,j})\,{}_oT_{iw}T_i^{-1}P^T \quad (25)$$

when feature $i$ corresponds to landmark $j$ at time $t$, or else is filled with zeros, and where $P := [I, \mathbf{0}] \in \mathbb{R}^{3\times4}$.

Thus, the landmark update steps are the following:

$$\boldsymbol{\mu}_{t+1,j} = \boldsymbol{\mu}_{t,j} + K_{t,j}(\mathbf{z}_{t,j} - \tilde{\mathbf{z}}_{t,j}) \quad (26)$$

$$\Sigma_{t+1,j} = (I - K_{t,j}H_{t,j})\Sigma_{t,j} \quad (27)$$

where

$$K_{t,j} := \Sigma_{t,j}H_{t,j}^T(H_{t,j}\Sigma_{t,j}H_{t,j})^T + V)^{-1} \quad (28)$$

In our implementation, $V \in \mathbb{R}^{4\times4}$ was set to a value of $3,500I_4$.

Finally, we can update our vehicle location based upon our observations.

At this step, we model the inverse pose $U_{t+1}$ as a conditional distribution: $U_{t+1}|\mathbf{z}_{0:t}, \mathbf{u}_{0:t} \sim \mathcal{N}(\boldsymbol{\mu}_{t+1|t}, \Sigma_{t+1|t})$, with $\boldsymbol{\mu}_{t+1|t} \in SE(3)$ and $\Sigma_{t+1|t} \in \mathbb{R}^{6\times6}$.

The observation model (with inherent measurement noise as defined in Eq. 3) is then the following:

$$\mathbf{z}_{t+1,i} = h(U_{t+1}, \underline{\mathbf{m}}_j) + \mathbf{v}_{t+1,i} := M\pi({}_oT_iU_{t+1}\underline{\mathbf{m}}_j) + \mathbf{v}_{t+1,i} \quad (29)$$

This is to point out that the observation model remains unchanged, but now here the variable of interest is the inverse IMU pose $U_{t+1} \in SE(3)$ instead of the landmark positions $\mathbf{m_j} \in \mathbb{R}^3$.

Thus, the definition of the Jacobian $H_{i,t+1|t} \in \mathbb{R}^{4\times6}$ changes, since it is now with respect to $U_t$, evaluated at $\boldsymbol{\mu}_{t+1|t}$:

$$H_{i,t+1|t} = M\frac{d\pi}{d\mathbf{q}}({}_oT_i\boldsymbol{\mu}_{t+1|t}\underline{\mathbf{m}}_j)\,{}_oT_i(\boldsymbol{\mu}_{t+1|t}\underline{\mathbf{m}}_j)^{\odot} \quad (30)$$

where, for given homogeneous coordinates $\underline{s} \in \mathbb{R}^4$:

$$\underline{\mathbf{s}}^{\odot} = \begin{bmatrix} \mathbf{s} \\ 1 \end{bmatrix}^{\odot} := \begin{bmatrix} I_3 & -\hat{\mathbf{s}} \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{4\times6} \quad (31)$$

And finally we arrive at the EKF update equations for the IMU pose:

$$\boldsymbol{\mu}_{t+1|t+1} = \exp(\widehat{(K_{t+1|t}(\mathbf{z}_{t+1} - \tilde{\mathbf{z}}_{t+1}))})\boldsymbol{\mu}_{t+1|t} \tag{32}$$

$$\Sigma_{t+1|t+1} = (I - K_{t+1|t}H_{t+1|t})\Sigma_{t+1|t} \tag{33}$$

where

$$K_{t+1|t} := \Sigma_{t+1|t}H_{t+1|t}^T(H_{t+1|t}\Sigma_{t+1|t}H_{t+1|t}^T + I \otimes V)^{-1} \tag{34}$$

## IV. Results

### A. Discussion

All the results shown here use the noise parameters as specified in Eqns. 15 and 27, namely, ${\sigma^2}_\lambda = 10^{-5}$, ${\sigma^2}_\omega = 10^{-4}$, and $V = 3,500I_4$.

We noticed that the noise covariances of $W$ are quite small, and indeed would lead to poor results if increased above $10^{-3}$. We also noticed that $V$ was quite robust to change, with maps changing only slightly between $1,000I$ to $4,000I$. Values above $10,000I$ were obviously too strong, though.

We comment on our results in the figure captions below, and include animations in the code submission.

We observe that, in all of the three datasets provided, the EKF performs well at visual-inertial SLAM (provided the guesses of the covariances belonging to the inherent Gaussian movement noise $\mathbf{w}_t$ and measurement noise $\mathbf{v}_t$ are good). By comparing each raw path (in red) to its corresponding filtered path (in blue), we can see that the localization of the vehicle benefits tremendously from localizing based on visual keypoint tracking. If the Kalman filter is black magic, the EKF is advanced black magic.

Finally, we thank the reader for their patience throughout the quarter. We will never forget their kindness and mercy.

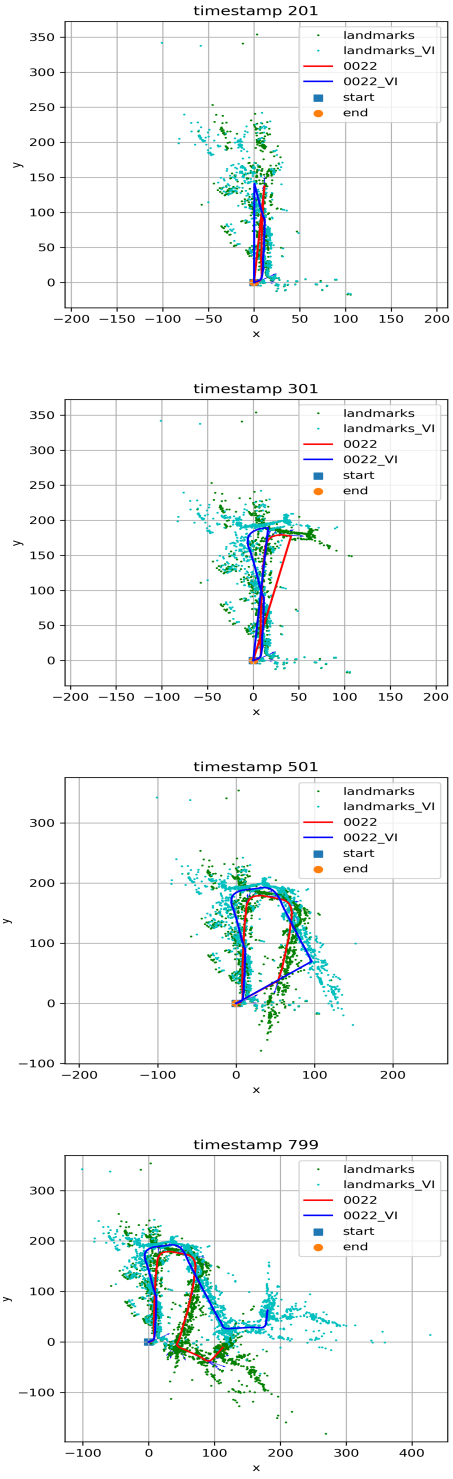
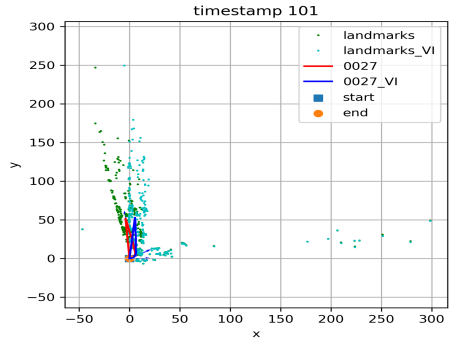
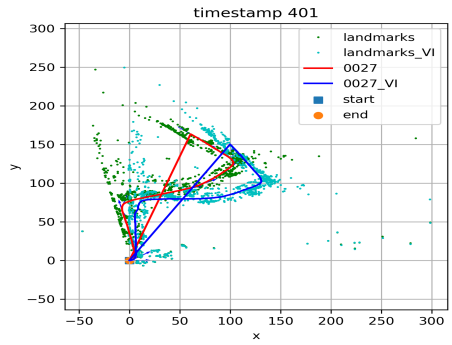
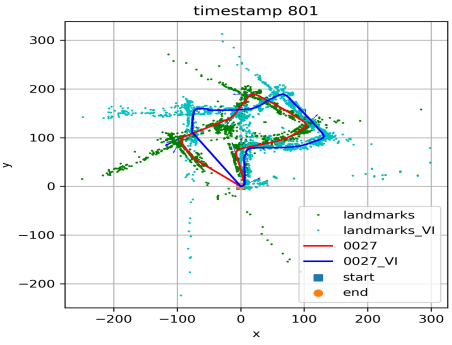



Fig. 1: EKF result for dataset 22. We see at timestamp 201 that the path and landmarks are relatively unchanged, and from timesteps 301 and 501, we see that the biggest changes come from rotations. We know from watching the associated video that the vehicle moves around blocks, so some streets should be parallel, and streets should all be straight, and this is precisely what we observe in the final map above at timestamp 799. Animation submitted with code.



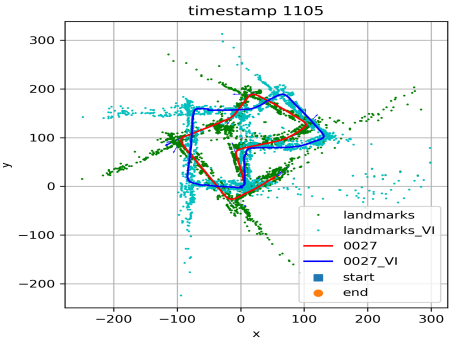
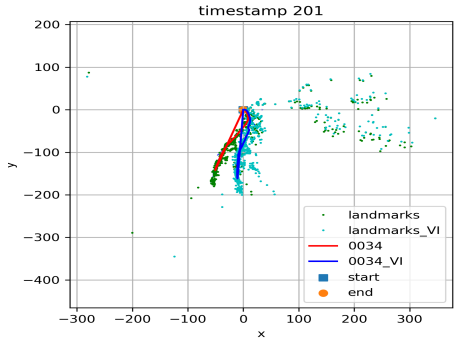
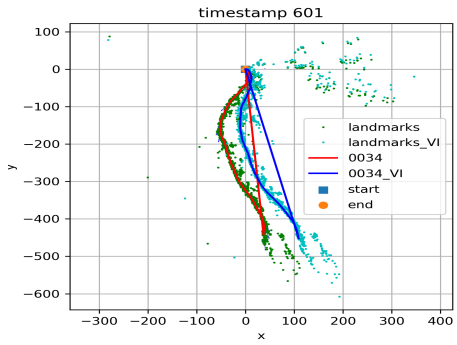
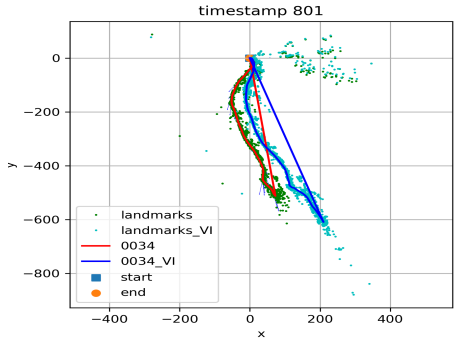




Fig. 2: EKF result for dataset 27. We see at timestamp 101 that the path and landmarks are more deviated at the beginning here than in other datasets, but the filtered path is much cleaner and regular compared to the raw odometry readings. We know from watching the associated video that the vehicle returns to its point of origin, and this is precisely what we observe in the final map above at timestamp 1105. Animation submitted with code.



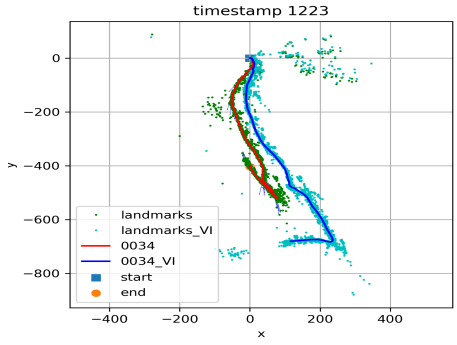




Fig. 3: EKF result for dataset 34. We see at timestamp 201 that the path and landmarks are relatively unchanged, and this remains the case until around timestamp 601. However, we see major deviation in timestamp 801, where the vehicle initiates the turn. We know from watching the associated video that the vehicle turns relatively sharply, but not to the extent of doing a U-turn, and this is precisely what we observe in the final map above at timestamp 1223. Animation submitted with code.