# SLAM - EKF

1st Weixiao Zhan
weixiao-zhan[at]ucsd[dot]edu

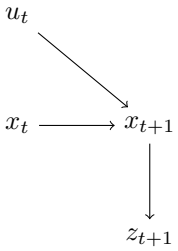
## I. Introduction

Kalman filter is a powerful algorithm for estimating the state of a linear system with Gaussian distribution and Gaussian noise. Extended Kalman Filter (EKF) adapts this approach for nonlinear systems using first-order approximation. These filters are indispensable in robotics, as they provide a probabilistic-based, computation-feasible model for solving simultaneous localization and mapping (SLAM) problem.

In this project, EKF SLAM over a vehicle equipped with IMU and stereo camera is implemented to estimate the vehicle pose trajectory and 3d location of feature points extracted from stereo camera.

## II. Problem Formulation

EKF is defined by three components (state $x$, control input $u$, and observation $z$), two models (motion model $f$ and observation model $h$), and two operations (predict and update). EKF is based on Markov assumption and assumes all components are Gaussian distributed:




### A. State

Without loss of generality, let the vehicle body frame align with IMU sensor frame and the world frame align with the body frame at timestamp 0.

At timestamp $t$, EKF state $x_t$ contains both the vehicle pose and world frame location of feature points extracted from stereo camera. let $T_t$ denote the vehicle pose; $M_t$ denote the set of all the feature points have been extracted, including $t$; $m_{t,j}$ denote the world frame coordinate of $j$ feature point.

$$
\begin{aligned}
x_t &= \{T_t\} \cup \{m_{t,j} | \forall j \in M_t\} \\
x_t[\mu] &= \begin{Bmatrix} T_t[\mu] \in SE(3) & \\ m_{t,j}[\mu] \in \mathbb{R}^3 & \forall j \in M_t \end{Bmatrix} \\
x_t[\Sigma] &= \begin{bmatrix} M_t[\Sigma] & cov(M_t, T_t) \\ cov(T_t, M_t) & T_t[\Sigma] \end{bmatrix} \\
&\in \begin{bmatrix} R^{[3M_t \times 3M_t]} & R^{[3M_t \times 6]} \\ R^{[6 \times 3M_t]} & R^{[6 \times 6]} \end{bmatrix}
\end{aligned}
$$

in which, $(\cdot)[\mu]$ and $(\cdot)[\Sigma]$ are the mean and covariance of the random variables.

### B. Control

In vehicle kinetic, the control input would be steering angle, throttle, and breaking. To simplify motion model and EKF, this project use generalized velocity gathered from IMU as control input.

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

in which, $v_t$ is the linear velocity and $w_t$ is the angular velocity in body frame at $t$.

### C. Observation

Not all feature points are visible at timestamp $t$. Let $N_t \subseteq M_t$ denote the set of visible feature points at $t$. Observation $z_t$ are the stereo camera pixel coordinates of visible feature points.

$$
z_t = \begin{bmatrix} \vdots \\ z_{t,j} \\ \vdots \end{bmatrix} \in \mathbb{R}^{4N_t}, z_{t,j} = \begin{bmatrix} u_l \\ v_l \\ u_r \\ v_r \end{bmatrix}, \forall j \in N_t
$$

in which, $(u_l, v_l)$ is the feature point's pixel coordinate in left camera, $(u_r, v_r)$ is the feature point's pixel coordinate in right camera.

### D. Motion model

Motion model takes in definitive state $x_t$, control $u_t$, noise $w$, and returns estimated next state $x_{t+1}$.

*1) vehicle pose:* Based on rigid body kinematics and assuming IMU reading is noisy, motion model of pose is

$$
\begin{aligned}
T_{t+1} &= f(T_t, u_t, w) \\
&= T_t \exp\left(\tau_t \widehat{(u_t + w)}\right) \\
&\approx T_t \exp(\tau_t \widehat{u_t}) + w'
\end{aligned}
$$

in which, $\tau_t$ is the difference between timestamp $t+1$ and $t$; $\widehat{\cdot}$ converts $R^6$ to $se(3)$ group; $w'$ is approximately linear combination of $w$, thus $w'$ is also Gaussian.

*2) feature points:* We are assuming the feature points are stationary, thus:

$$
m_{t+1,j} = m_{t,j}
$$

### E. Observation model

Observation model takes in definitive state $x_t$, noise $v$, and returns estimated observation $z_t$.

In vehicle configuration, the transformation between the two stereo cameras is only a displacement along the x-axis (optical frame) of size $b$. Thus the stereo camera observation model is:

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

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

$$\begin{aligned} z_j = \begin{bmatrix} u_l \\ v_l \\ u_r \\ v_r \end{bmatrix} &= h(T_t, m_j, v), \forall j \in N_t \\ &= K_s \pi \left( {}_{opt}T_{body} \left(T_t\right)^\dagger \begin{bmatrix} m_j \\ 1 \end{bmatrix} \right) + v \end{aligned}$$

in which, $K$ is the intrinsic parameters of stereo camera; $(\cdot)^\dagger$ represents $SE(3)$ inverse.

### F. EKF Predict

EKF predict step extends definitive motion model to Gaussian random variable and approximate nonlinear motion model using Jacobian.

Assuming motion model noise $w$ is Gaussian, which is equivalent to $w' \sim N(0, W)$. Predicted state $x_{t+1|t}$ is obtained as following:

$$\begin{aligned} F_t &= \frac{\partial f}{\partial x}(T_t[\mu], u_t, 0) \\ &= \exp\left(-\tau_t \hat{\mathbf{u}}_t\right) \\ Q_t &= \frac{\partial f}{\partial w}(T_t[\mu], u_t, 0) \\ &= I \\ T_{t+1|t}[\mu] &= f(T_t[\mu], u_t, 0) \\ m_{t+1|t,j}[\mu] &= m_{t,j}[\mu] \\ x_{t+1|t}[\Sigma] &= \begin{bmatrix} M_{t+1|t}[\Sigma] & cov\left(T_{t+1|t}, M_{t+1|t}\right) \\ cov\left(T_{t+1|t}, M_{t+1|t}\right) & T_{t+1|t}[\Sigma] \end{bmatrix} \\ &= \begin{bmatrix} M_t[\Sigma] & cov\left(T_t, M_t\right) F_t^T \\ F_t cov\left(T_t, M_t\right) & F_t\left(T_t[\Sigma]\right) F_t^T + W \end{bmatrix} \end{aligned}$$

### G. EKF Update

EKF update step uses Bayes rule to correct its states when given observations. Assuming observation model noise $v \sim N(0, V)$, updated states $x_{t+1}$ are obtained as following:

$$\begin{aligned} H_{t,j}^{(T)} =& \frac{\partial h}{\partial T_t}(T_{t+1|t}[\mu], m_{t,j}[\mu], 0) \\ =& -K_s \frac{d\pi}{dq} \left( {}_{opt}T_{body} \left(T_{t+1|t}[\mu]\right)^\dagger \begin{bmatrix} m_{t,j}[\mu] \\ 1 \end{bmatrix} \right) \cdot \\ & {}_{opt}T_{body} \left( \left(T_{t+1|t}[\mu]\right)^\dagger \begin{bmatrix} m_{t,j}[\mu] \\ 1 \end{bmatrix} \right)^\odot \\ & \in \mathbb{R}^{[4\times 6]} \\ H_{t,j}^{(m_j)} =& \frac{\partial h}{\partial m_j}(T_{t+1|t}[\mu], m_{t,j}[\mu], 0) \\ =& K_s \frac{d\pi}{dq} \left( {}_{opt}T_{body} \left(T_{t+1|t}[\mu]\right)^\dagger \begin{bmatrix} m_{t,j}[\mu] \\ 1 \end{bmatrix} \right) \cdot \\ & {}_{opt}T_{body} \left(T_{t+1|t}[\mu]\right)^\dagger P^T \\ & \in \mathbb{R}^{[4\times 3]} \\ H =& \begin{bmatrix} H_{t,0}^{(m_0)} & \cdots & 0 & \cdots & 0 & H_{t,0}^{(T)} \\ 0 & \cdots & H_{t,j}^{(m_j)} & \cdots & 0 & H_{t,j}^{(T)} \\ 0 & \cdots & 0 & \cdots & H_{t,N_t}^{(m_{Nt})} & H_{t,N_t}^{(T)} \end{bmatrix} \\ & \in \mathbb{R}^{[4N_t \times (3M_t+6)]} \\ Q =& \frac{\partial h}{\partial v}(T_{t+1|t}[\mu], m_{t,j}[\mu], 0) \\ =& I \\ K =& x_{t+1|t}[\Sigma] H^T \left(H(x_{t+1|t}[\Sigma]) H^T + V\right)^{-1} \\ T_{t+1}[\mu] =& T_{t+1|t}[\mu] \exp\left(\left(K(z_{t+1} - \widetilde{z_{t+1}})\right)^\wedge\right) \\ m_{t+1,j}[\mu] =& m_{t,j}[\mu] + K(z_{t+1} - \widetilde{z_{t+1}}) \\ x_{t+1}[\Sigma] =& (I - KH)\left(x_{t+1|t}[\Sigma]\right) \end{aligned}$$

in which,

$$\begin{aligned} \begin{bmatrix} s \\ 1 \end{bmatrix}^\odot &= \begin{bmatrix} I_3 & -\hat{s} \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{[4\times 6]} \\ \widetilde{z_{t+1}} &= h\left(T_{t+1|t}[\mu],\ m_{j,t}[\mu],\ 0\right) \\ P &= \begin{bmatrix} I_3 & 0 \end{bmatrix} \in \mathbb{R}^{[3\times 4]} \end{aligned}$$

### H. Localization, mapping, and SLAM

EKF has no requirement for the order in which predicts and updates are invoked. Three special orders of running EKF are picked to simulate different application usage:

1) localization: only run predict step. This method builds a distribution based pose trajectory using IMU reading.
2) mapping: fix $T$, and only run update step on $m_j$. This method assumes no error in poses and creates a 3d mapping of feature points.
3) SLAM: run predict and update alternately. This method builds and corrects pose trajectory and world mapping at the same time, achieving simultaneous localization and mapping.

## III. Technical Approach

### A. Prior distribution

Kalman filter relies on the prior distribution to perform predict and update steps.

Pose state prior are set by world frame definition:

$$T_0[\mu] = I_4$$
$$T_0[\Sigma] = 0^{[6\times 6]}$$

As for the feature points, suppose feature point $j$ is first time observed at time $t$ with observation $z_j = [u_l \quad v_l \quad u_r \quad v_r]^T$. Its world frame coordinate $m_j$ can be estimated by solving stereo projection:

$$d = u_l - u_r$$
$$z = \frac{fs_u b}{d}$$
$$x = z\frac{u_l - c_u}{fs_u}$$
$$y = z\frac{\frac{v_l+v_r}{2} - c_u}{fs_v}$$

In theory, $v_l$ should equals $v_r$. However, due to observation noise, Their average is used to estimated $y$ coordinate in optical frame. Random variable $m_{t,j}$ is initialized to following prior:

$$\begin{bmatrix} m_{t,j}\,[\mu] \\ 1 \end{bmatrix} = T_t\,({}_{opt}T_{body})^{\dagger} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$
$$m_{t,j}[\Sigma] = V^{(0)} \in \mathbb{R}^{[3\times 3]}$$
$$cov(m_{t,j}, s) = 0, \forall s \in x_t$$

### B. Feature selection

In previous calculation of solving stereo projection, when $d$ is small, i.e. the point is far way, $z$ estimation is less accurate due to noise. So I propose following feature selection criteria: only handle features with $d \in [d_{\min}, d_{\max}]$. This selection is enforced when handling new points as well as in EKF update step.

In practice, I chose $d_{\min} = 8$ to remove faraway points. and $d_{\max} = 40$ based on cut off in the histogram of $d$ . This criteria has selected 3081 out of 5105 and 8793 out of 13289 feature points in dataset 03 and 10 respectively. Their comparison are further discussed in Results section.

### C. Variance

Following noise and prior variances are used in practice:

$$W = \text{diag}\left(\left[10^{-8}, 10^{-8}, 10^{-5}, 10^{-5}, 10^{-5}, 10^{-3}\right]\right)$$
$$V = \begin{bmatrix} 10 & 5 & 0 & 0 \\ 5 & 10 & 0 & 0 \\ 0 & 0 & 10 & 5 \\ 0 & 0 & 5 & 10 \end{bmatrix}$$
$$V^{(0)} = 5 * I_3$$

I allow more noise in $z$ and $yaw$ compare to other linear and angular part of the pose. $covar(u_l, v_l)$ and $covar(u_r, v_r)$ are set to a non-zero value to reflect that they are readings from same camera, thus their noise may be correlated.

### D. Time window EKF

The number of feature points increases when vehicle drives. Quickly, it is computationally not stable and exhausting to run EKF over all feature points. The covariance matrix encounters singularity and `nan` due to gradient explosion. One solution is to sample from feature points. However, as the number of points keeps increasing, the sampled feature points will be too sparse and lose information. Thus, I propose time window EKF as an alternative approach. The insight is points observed at long time intervals are presumably uncorrelated.

Time window EKF distinguish states vs active states. All feature points are in states, with full-size mean and covariance matrix maintained. However, only feature points seen in past $\lambda$ stamps are active, since they are likely to be correlated with currently observation. Time window EKF only change the mean and covariance of active states.

Formally, Let $M_t^A$ denote the active points at timestamp $t$, and $x_t^A$ denote the active states:

$$M_t^A = \bigcup_{k=\max(t-\lambda,0)}^{t} N_k$$
$$x_t^A = \{T_t\} \cup \left\{m_{t,j} | j \in M_t^A\right\}$$

Note the pose is always active. The active mean and active variance matrix are slices of the full mean and variance matrix using masks.

In practice, $\lambda$ is set to 200, as a balance point between computation speed and loosing correlation information.

### E. computation tricks

*1) PyTorch and GPU:* I have implemented EKF and time window EKF in PyTorch to harvest GPU power.

*2) NumPy slices and PyTorch view:* The flat mean with shape $(3M_t,)$ and covariance matrix with shape $(3M_t + 6, 3M_t + 6)$ is hard for indexing and slicing. Thus the mean and covariance matrix are reshaped to $(M_t, 3)$ and $(M_t + 2, 3, M_t + 2, 3)$ respectively, index feature points on corresponding axis and then flatten out for computation. This trick happens to work because pose covariance shape $(6, 6)$ is a multiple of feature point covariance shape $(3, 3)$.

*3) compact covariance:* In mapping application, $x_t[\Sigma]$ is $3 \times 3$ block diagonal matrix. Off diagonal elements are initialized to and stay at zero. Through out update steps, the mean and variance of currently unseen points are not updated. Thus I used sparse matrix and time window EKF with $\lambda = 0$ in mapping application to accelerate computation.

*4) symmetric covariance:* To enforce covariance matrix stay symmetric and avoid potential singularity. I used $\Sigma \leftarrow \frac{1}{2}(\Sigma + \Sigma^T)$ every 50 timestamp in conventional EKF. Time window EKF can proceed without such normalization to avoid singularity.

## IV. Results

Dataset 03 and dataset 10 are used to produce following results.

### A. localization

Figure 1 and 2 show the EKF localization of pose trajectory (projected on 2d plane). Red arrows show orientation and orange ellipse curves show one standard deviation (100 times exaggerated) in $x$ and $y$ axis. As the vehicle drives, noise in predict step accumulates, and the standard deviation curves expand. Mean while, as I allow $yaw$ rotation to have higher noise, the standard deviation curves are stretched in the direction perpendicular to the orientation.

### B. mapping

EKF mapping results are shown in figure 3 and 4. The left columns are result of all feature points, the right columns are results of selected feature points.

First row shows initial positions. Second row shows position mean after EKF mapping. Street and building alike structures already present in initial mapping, and some are still preserved after EKF mapping, especially in 4b and 4d.

As EKF update step iterates, the total uncertainty of the feature points, measured as covariance trace, is reduced (shown in 3e, 3g, 4e, and 4g).

### C. EKF SLAM

To obtain results in reasonable time, features are sampled with step 10 and 25; thus, 510 and 531 feature points are used in dataset 03 and 10 respectively. Based on practice experience, EKF is extremely sensitive to hyper-parameters, such as prior and noise variance. And I could not tune a reasonable parameters that works on both dataset.

Figure 5 and 6 shows one example of a set of hyper-parameters, that works on dataset 03 but failed to generalize on dataset 10. Figure 5c and 6c show the EKF was able to reduce the total variance on dataset 03 and the first 50 seconds on dataset 10. However, after that, EKF is no longer stable and not able to produce reasonable pose trajectory on dataset 10.

### D. time window EKF SLAM

Figure 7 and 8 show results of time window EKF SLAM. Left column are results using all feature points, right column are results using selected feature points. Other hyper-parameters are using same ones specified in previous section.

Comparing figure 7c-7e and 8c-8e, the update innovation is smaller with selected points, and the trajectory is also more smooth and closer the pure localization. This suggests there are less divergences between predict and update steps, and confirms the selection criteria that faraway points estimation are indeed noisy.

Because the time window EKF dynamically expands covariance matrix. The covariance trace, shown in 7d, 7f, 8d, and 7f, are increasing as a result of new states. In fact, the covariance of pose is maintained at a very small number, the standard deviation ellipse curves are barely visible on 7i and 8i trajectory graphs.

## V. Summary

Time window EFK with feature selection criteria provides a complete solution for vehicle SLAM problem. Compare to conventional EKF, feature selection and time window approach makes computation more stable and less exhausting. It would be interesting to run factor graph with ICP over feature points and compare the result of factor graph and EKF.

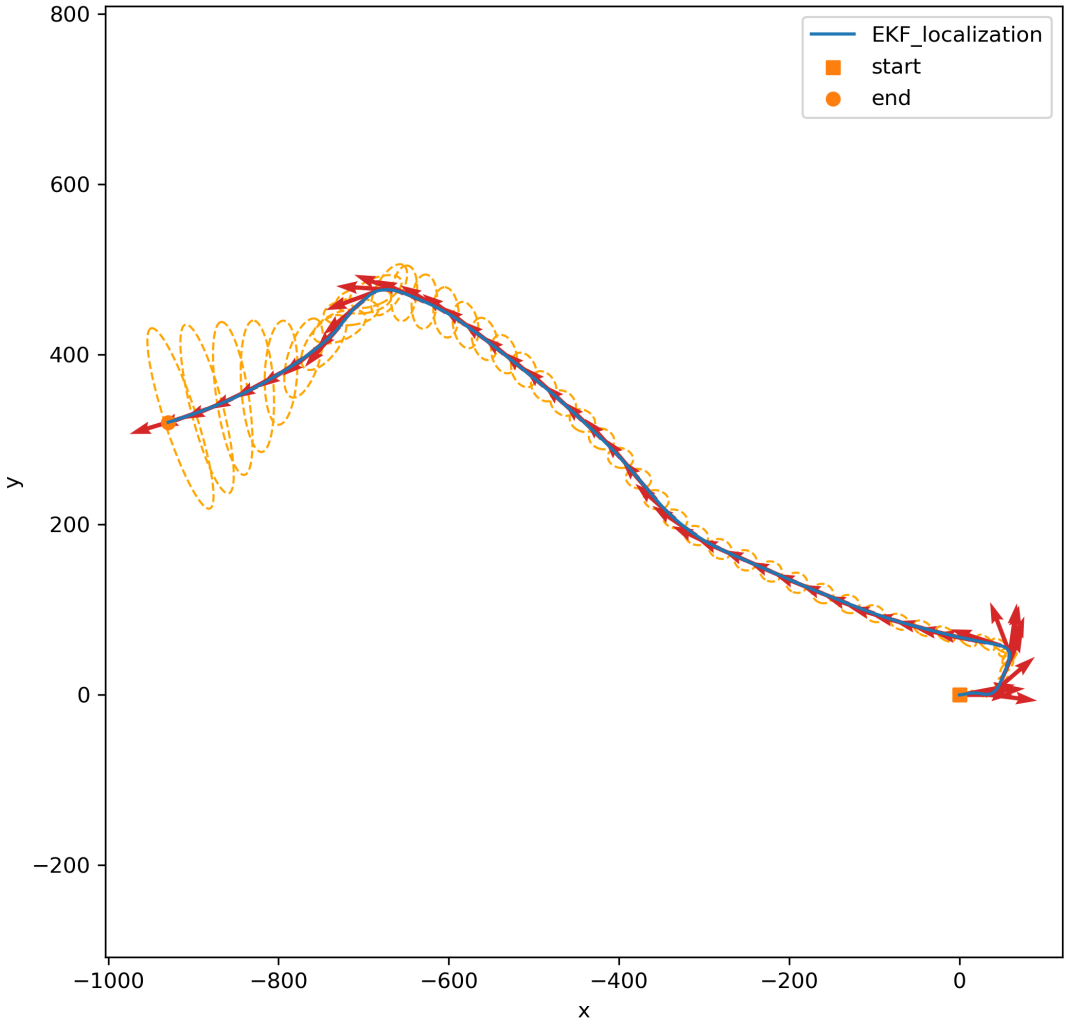



Fig. 1: EKF localization on dataset 03

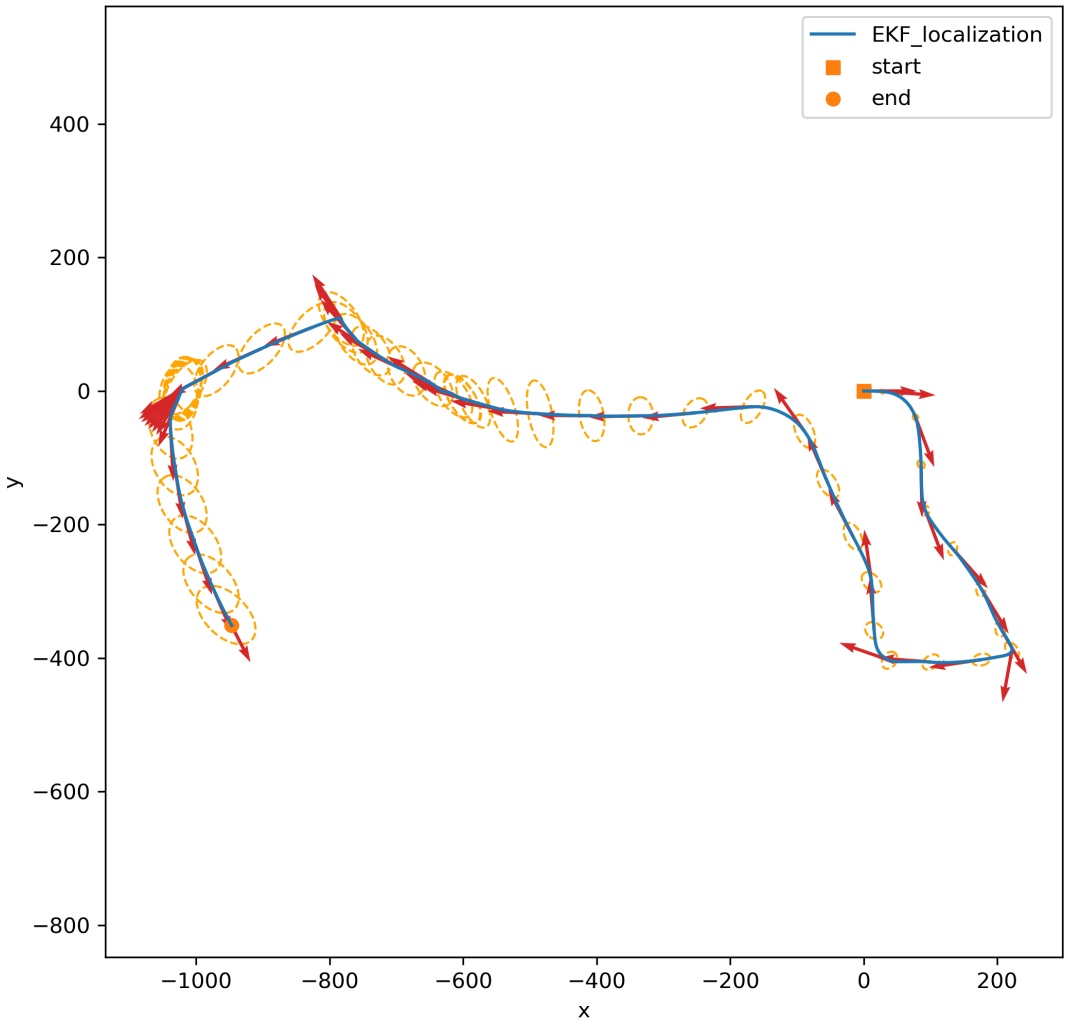
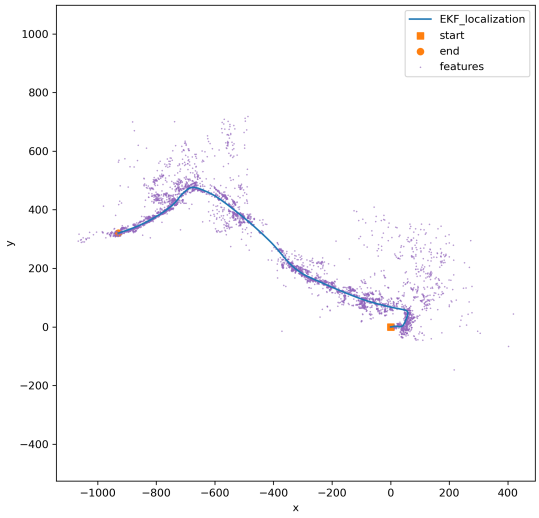
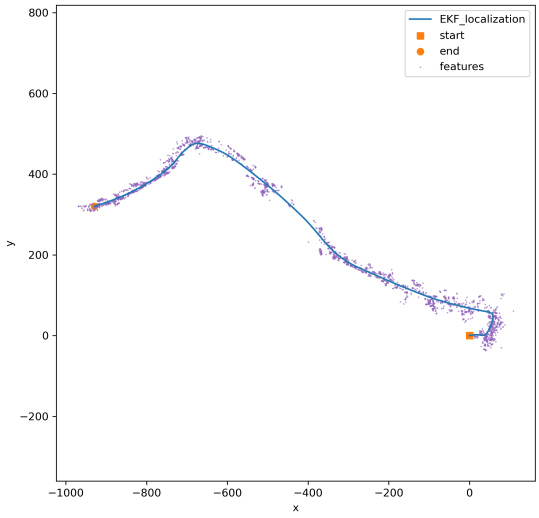


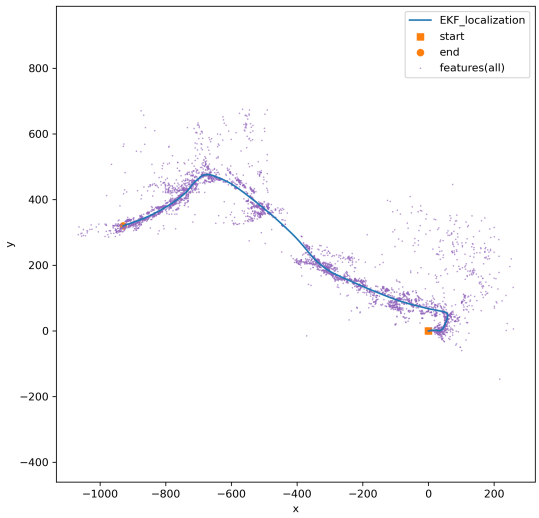

Fig. 2: EKF localization on dataset 10



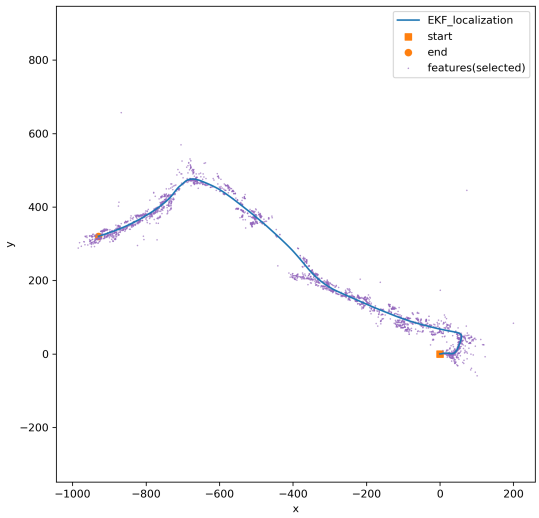

(a) initial positions (all)



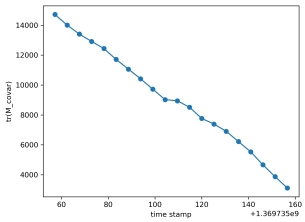

(b) initial positions (selected)



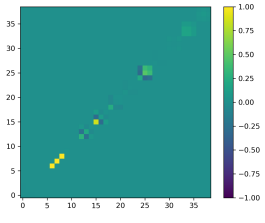

(c) EKF mapping mean positions (all)



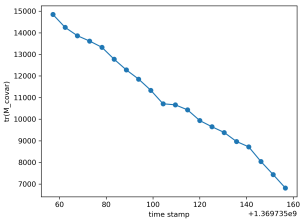

(d) EKF mapping mean positions (selected)



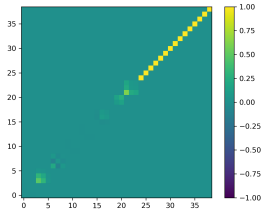

(e) timestamp vs covariance trace (all)



(f) covariance matrix (sampled all to plot)

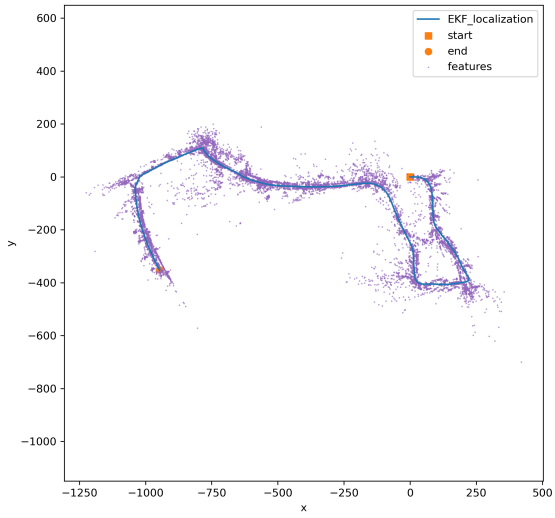



(g) timestamp vs covariance trace (selected)

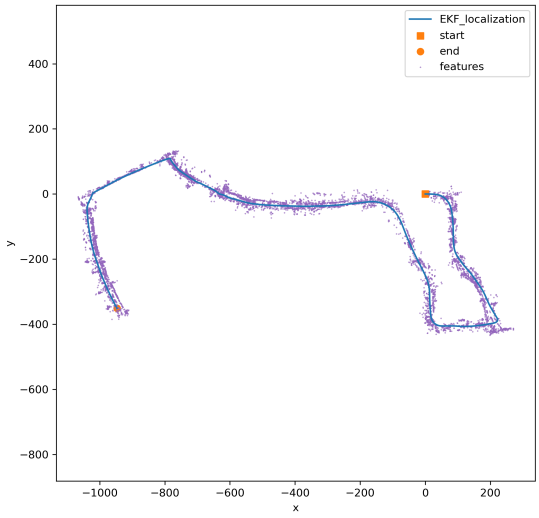


(h) covariance matrix (sampled selected to plot)

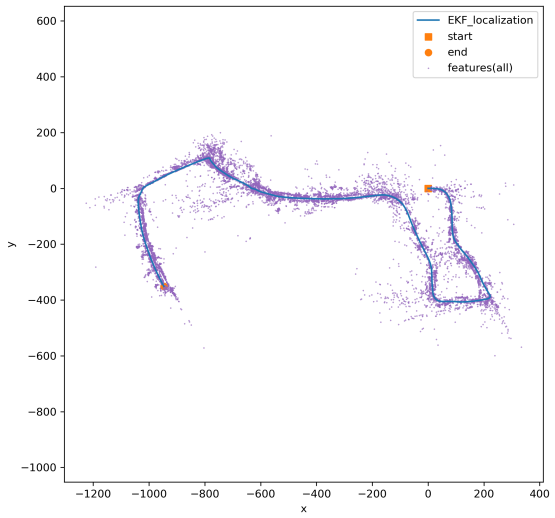
Fig. 3: EKF mapping on dataset 03



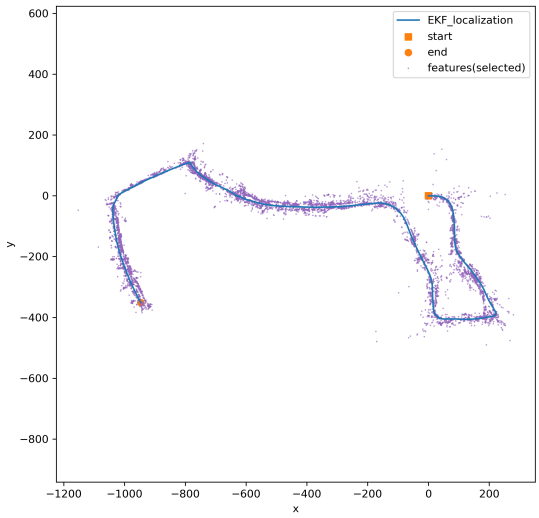

(a) initial positions (all)



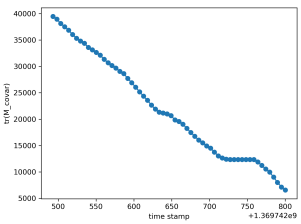

(b) initial positions (selected)



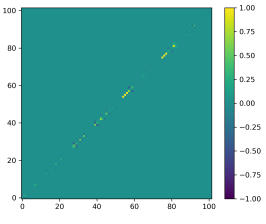

(c) EKF mapping mean positions (all)



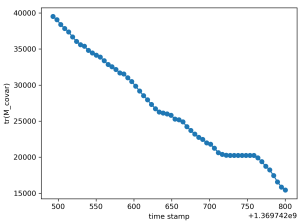

(d) EKF mapping mean positions (selected)



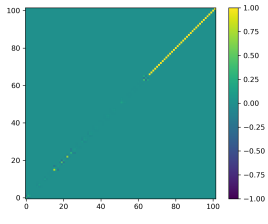

(e) timestamp vs covariance trace (all)



(f) covariance matrix (sampled all to plot)

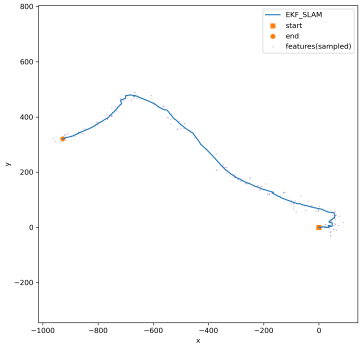



(g) timestamp vs covariance trace (selected)

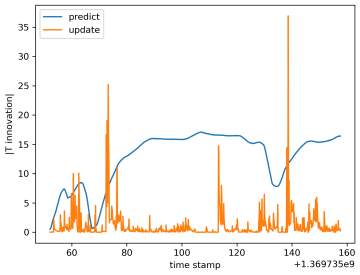


(h) covariance matrix (sampled selected to plot)

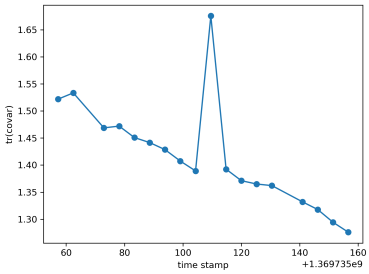
Fig. 4: EKF mapping on dataset 10




(a) EKF SLAM trajectory and mapping

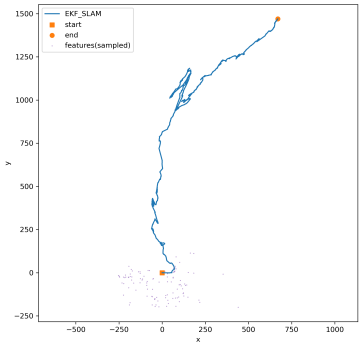



(b) timestamp vs pose innovation

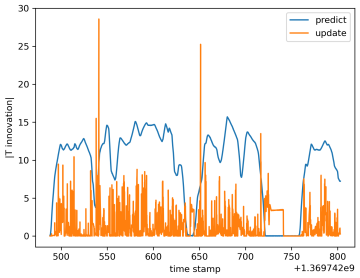



(c) timestamp vs covariance trace

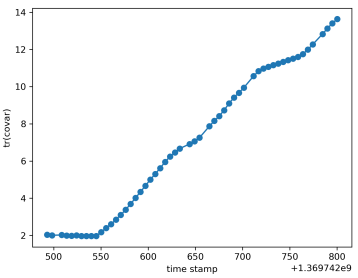
Fig. 5: EKF SLAM on dataset 03




(a) EKF SLAM trajectory and mapping

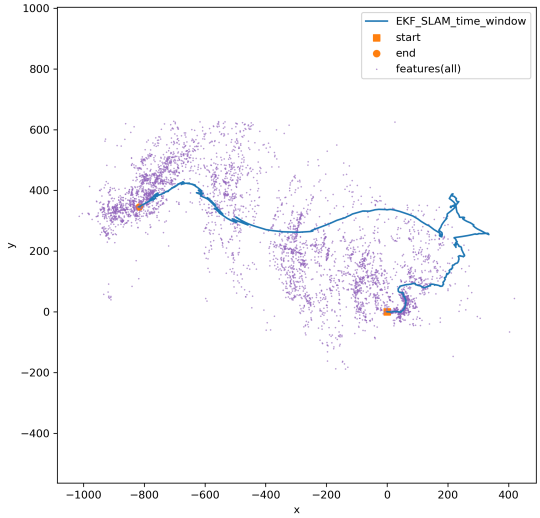



(b) timestamp vs pose innovation

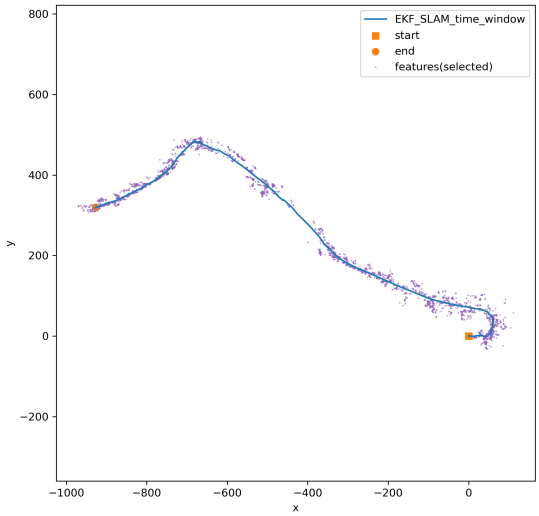



(c) timestamp vs covariance trace

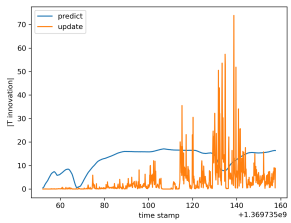
Fig. 6: EKF SLAM on dataset 10



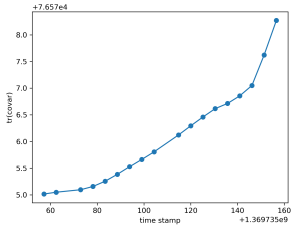

(a) time window EKF SLAM trajectory and mapping (all feature points)



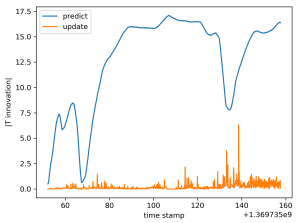

(b) time window EKF SLAM trajectory and mapping (selected feature points)



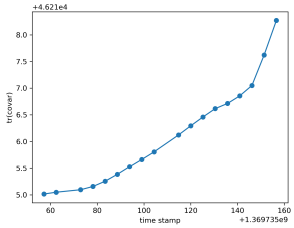

(c) timestamp vs pose innovation (all feature points)



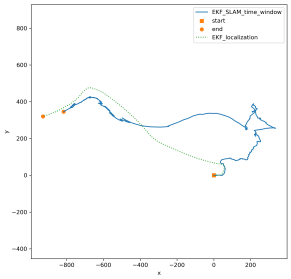

(d) timestamp vs covariance trace (all feature points)



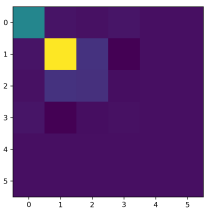

(e) timestamp vs pose innovation (selected feature points)



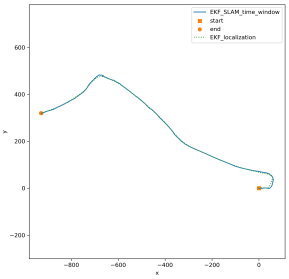

(f) timestamp vs covariance trace (selected feature points)



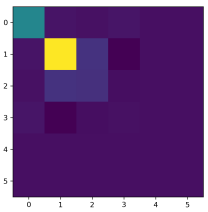

(g) SLAM trajectory vs localization trajectory (all feature points)



(h) pose covariance (all features)

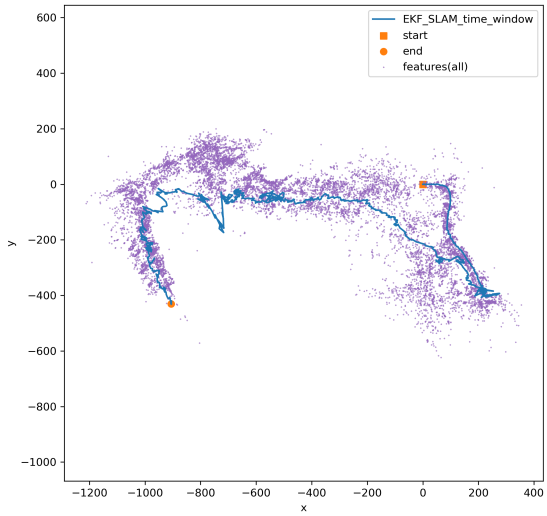



(i) SLAM trajectory vs localization trajectory (selected feature points)

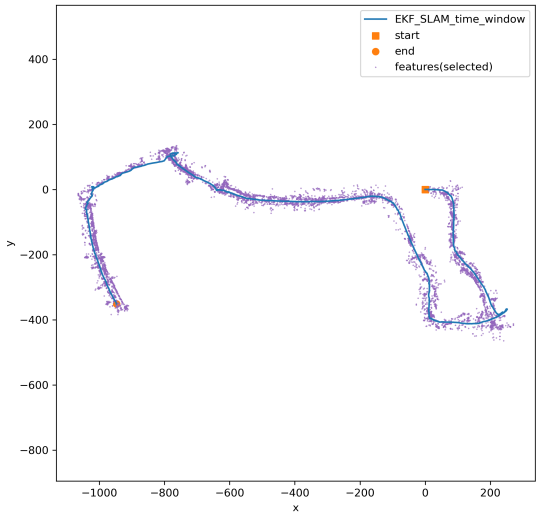


(j) pose covariance (selected features)

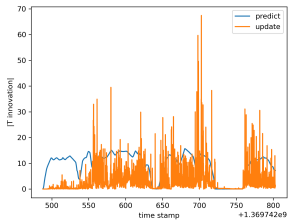
Fig. 7: time window EKF SLAM on dataset 03



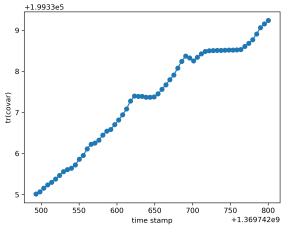

(a) time window EKF SLAM trajectory and mapping (all feature points)



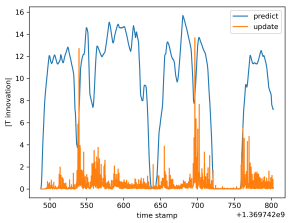

(b) time window EKF SLAM trajectory and mapping (selected feature points)



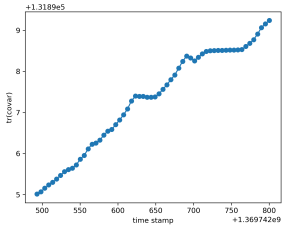

(c) timestamp vs pose innovation (all feature points)



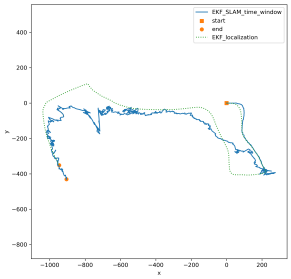

(d) timestamp vs covariance trace (all feature points)



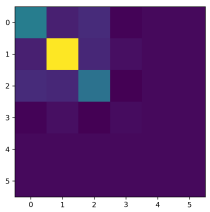

(e) timestamp vs pose innovation (selected feature points)



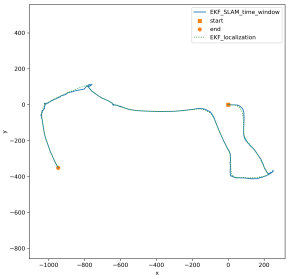

(f) timestamp vs covariance trace (selected feature points)



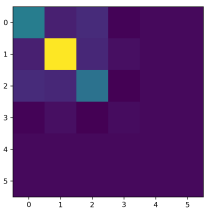

(g) SLAM trajectory vs localization trajectory (all feature points)



(h) pose covariance (all features)




(i) SLAM trajectory vs localization trajectory (selected feature points)



(j) pose covariance (selected features)

Fig. 8: time window EKF SLAM on dataset 10