Pose and feature estimation using the Extended Kalman Filter

Per Kvinnesland Omvik
PID: U07974631
UC San Diego

***Abstract*— This paper discusses two implementations of Extended Kalman Filters (EKF) for estimating the pose of a moving IMU and positions of landmarks captured by two cameras in a stereo camera setup moving along the same path. These procedures are treated separately, meaning that they do not solve the SLAM problem that arises, but can serve as an acceptable simplification or starting point.**

I. INTRODUCTION

The Kalman filter have long been a valuable tool for engineers in multiple diciplines. Most notably, the Kalman filter has been an integral part of aerospace mechanics and control, but is a versatile algorithm that can be used for almost any problem involving state estimation. The versatility and robustness of the filter is attributed to it's ability to account for noisy measurements, control inputs and model parameters. There are mainly three factors that impact the performance of the Kalman filter: a) The quality and accuracy of the system model b) The nature of the entering noise and c) The accuracy of the noise estimation [5]. How the noise parameters affected the results will be discussed in section IV.

The two Kalman filters examined in this paper tackle two different problems. The first focuses on estimating the state (or pose) of the IMU using only sensory information from the IMU itself. It does not utilize the feature observations from the camera in this estimation, and therefore the update step is omitted. The second filter is designed to estimating the positions of camera captured landmarks in 3D space using visual inertial odometry (IMU data paired with visual observations). However, because of the assumption that landmarks are stationary, we omit the prediction step. So while neither of the Kalman filters implemented in this report are full-fledged Kalman filters, together they inhibit all the features of a complete EKF. A more in-depth problem formulation and it's proposed solution are found in sections II and III.

II. PROBLEM FORMULATION

A. Pose prediction

The EKF is an extension of the regular Kalman filter that can be applied to nonlinear systems. The algorithms are conceptually almost identical, but involves linearization of the nonlinearities in the EKF. With state $\mathbf{x}_t$, input $\mathbf{u}_t$ and model noise $\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{W})$, the nonlinear system model is expressed as

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

The problem we are solving is estimating the pose of the IMU using its measurements of linear and rotational velocities $v_t, \omega_t \in \mathbb{R}^3$. The linear velocity vector v_t is composed of translational velocity vectors in the x, y, z directions in the IMU frame. ω_t contains the angular velocities about all three rotational axes: roll, pitch, yaw = $[\theta, \phi, \psi]^T$. We define these measurements as control inputs to the model:

$$\mathbf{u}_t = \begin{bmatrix} v_t \\ \omega_t \end{bmatrix} \tag{2}$$

The Kalman prediction step aims to predict the (Gaussian) distribution of the state which includes the mean $\boldsymbol{\mu}_{t+1|t}$ of the next state $\mathbf{x}_{t+1}$ based on the current mean estimate $\boldsymbol{\mu}_{t|t}$ and input $\mathbf{u}_t$. Noise is assumed to have zero mean.

$$\boldsymbol{\mu}_{t+1} = \mathbf{f}(\boldsymbol{\mu}_t, \mathbf{u}_t, 0) \tag{3}$$

It also generates a new estimation of the state covariance matrix using the Jacobian of the system model:

$$\mathbf{F}_t = \frac{d\mathbf{f}}{d\mathbf{x}}(\boldsymbol{\mu}_t, \mathbf{u}_t, 0) \tag{4}$$

yields

$$\boldsymbol{\Sigma}_{t+1} = \mathbf{F}_t \boldsymbol{\Sigma}_t \mathbf{F}_t^T + \mathbf{W}_t' \tag{5}$$

B. Landmark updating

The nonlinear observation model is expressed as follows:

$$\mathbf{z}_t = \mathbf{h}(\mathbf{x}_t) + \mathbf{v}_t \tag{6}$$

The observations $\mathbf{z}_t \in \mathbb{R}^{4 \times N_t}$ are the pixel coordinates of the features as captured by the stereo cameras at time t. An illustrative example is given in figure 1 where blue dots are pixel coordinates of the feature from right camera, and red dots are as observed by the left camera. The objective is to estimate their positions

$$\mathbf{m}_i \in \mathbb{R}^3 \text{ for } i = 1, ..., M$$

For M landmarks in total. Because of the noise $\mathbf{v}_t \sim \mathcal{N}(0, \mathbf{V})$ in the observation model, we implement the EKF to estimate the true landmark positions. We define

$$\hat{\mathbf{z}}_t = \mathbf{h}(\boldsymbol{\mu}_t, 0) \tag{7}$$

$$\mathbf{H}_t = \frac{d\mathbf{h}}{d\mathbf{x}}(\boldsymbol{\mu}_t, 0) \tag{8}$$

We arrive at the equations for the Kalman update that we need to fit to our observation model:

$$\mathbf{K}_t = \boldsymbol{\Sigma}_t \mathbf{H}_t^T (\mathbf{H}_t \boldsymbol{\Sigma}_t \mathbf{H}_t^T + \mathbf{V}_t')^{-1} \tag{9a}$$

$$\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t + \mathbf{K}_t(\mathbf{z}_t - \hat{\mathbf{z}}_t) \tag{9b}$$

$$\boldsymbol{\Sigma}_{t+1} = (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t)\boldsymbol{\Sigma}_t \tag{9c}$$

III. TECHNICAL APPROACH

A. Pose prediction

The general procedure for this implementation can be found in [4]. We can express the motion model with a perturbation $\xi \in \mathbb{R}^6$ of the current pose:

$$\boldsymbol{\mu}_{t+1} = exp(\hat{\xi})\boldsymbol{\mu}_t \approx (\mathbf{I} + \hat{\xi})\boldsymbol{\mu}_t \tag{10}$$

Using time discretization τ and dividing the motion model kinematics into nominal and and perturbation kinematics, we can express the motion model as follows:

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

$$\xi_{t+1} = exp(-\tau\mathbf{u}_t^{\curlywedge})\xi_t + \tau\mathbf{w}_t \tag{11b}$$

with

$$\hat{\mathbf{u}}_t = \begin{bmatrix} \exp(\omega_t) & v_t \\ \mathbf{0}^T & 0 \end{bmatrix} \tag{12a}$$

$$\mathbf{u}_t^{\curlywedge} = \begin{bmatrix} \exp(\omega_t) & \hat{v}_t \\ \mathbf{0}^T & \exp(\omega_t) \end{bmatrix} \tag{12b}$$

The complete EKF prediction step as described in (3) and (5) for the pose of the IMU becomes (11a) and

$$\begin{aligned} \boldsymbol{\Sigma}_{t+1} &= \mathbb{E}[\xi_{t+1}\xi_{t+1}^T] \\ &= exp(-\tau\mathbf{u}_t^{\curlywedge})\boldsymbol{\Sigma}_t exp(-\tau\mathbf{u}_t^{\curlywedge})^T + \tau^2\mathbf{W} \end{aligned} \tag{13}$$

After iterating through all time steps, the complete trajectory of the pose of the IMU is extracted from the estimated means. From the equation one can observe that the means for all t are independent of the covariances when the update step is not implemented. This is a simplification of the pose estimation, and in reality cannot be regarded as an implementation of the EKF, because the Kalman gain is not computed. The performance of this method will be discussed in section IV.

B. Landmark updating

The specified observation model (7) for this problem is:

$$\mathbf{z}_{t,i} = \mathbf{M}\pi({}_O\mathbf{T}_I\mathbf{T}_t(\boldsymbol{\mu}_{t,j} + \mathbf{D}\delta_{t,j})) \tag{14}$$

For a perturbation δ_t of the position of the landmark j. Where ${}_O\mathbf{T}_I$ is the transform from the IMU to the optical frame, $\mathbf{M}$ is the intrinsic matrix of the stereo camera setup, $\pi(\mathbf{q}) = \frac{1}{q_3}\mathbf{q} \in \mathbb{R}^4$.

After generating a complete trajectory

$$\mathbf{T}_t =_W \mathbf{T}_{I,t}^{-1} \text{ for } t = 1, .., T$$

for the IMU in the previous subsection, we can generate a transformation from the pixel coordinates $\mathbf{z}_t \in \mathbb{R}^{4\times N_t}$ to the landmarks $\mathbf{m}_i \in \mathbb{R}^{3\times M}$ in the world frame. For the first time a landmark is observed, we initialize the mean and covariance as

$$\boldsymbol{\mu}_{t,j} = ({}_O\mathbf{T}_I\mathbf{T}_t)^{-1}\pi^{-1}(\mathbf{M}\hat{\mathbf{z}}_{t,j}) \tag{15a}$$

$$\boldsymbol{\Sigma}_{t+1} = \boldsymbol{\Sigma}_t \oplus \mathbf{I} \tag{15b}$$

The identity matrix is appended to the diagonal of $\boldsymbol{\Sigma}_t$, increasing the dimensions by 3×3.

After a landmark is initialized, the next observation of said landmark initializes the Kalman update step. The Jacobian in (8) becomes

$$\mathbf{H}_{i,j,t} = \begin{cases} \mathbf{H}\frac{d\pi}{d\mathbf{q}}({}_O\mathbf{T}_I\mathbf{T}_t\boldsymbol{\mu}_{t,j}){}_O\mathbf{T}_I\mathbf{T}_t\mathbf{D}, & \text{if observation } i \\ & \text{corresponds to landmark } j \text{ at time } t. \\ \mathbf{0} \in \mathbb{R}^{4\times 3}, & \text{otherwise.} \end{cases} \tag{16}$$

We use the Jacobian to complete the EKF update step:

$$\mathbf{K}_t = \boldsymbol{\Sigma}_t\mathbf{H}_t^T(\mathbf{H}_t\boldsymbol{\Sigma}_t\mathbf{H}_t^T + \mathbf{I} \otimes \mathbf{V})^{-1} \tag{17a}$$

$$\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t + \mathbf{K}_t(\mathbf{z}_t - \hat{\mathbf{z}}_t) \tag{17b}$$

$$\boldsymbol{\Sigma}_{t+1} = (\mathbf{I} - \mathbf{K}_t\mathbf{H}_t)\boldsymbol{\Sigma}_t \tag{17c}$$

Because the observation noise is set to be uncorrelated to the observations themselves, the last part in (17a) is simply

$$\begin{bmatrix} \mathbf{V} & & \\ & \ddots & \\ & & \mathbf{V} \end{bmatrix}$$

Where the observation noise $\mathbf{V}$ is assumed to be known. $\mathbf{V}$ can either be estimated using the data, or tuned as a parameter.

IV. RESULTS

A. Pose prediction

The results from the pose prediction were overall reasonable, as they seemed to match the videos closely in their path and distance. The results are displayed in figures 2 - 4. Because of how the prediction step is implemented in (3), one can observe that the prediction for $\boldsymbol{\mu}_{t+1}$ is independent of $\boldsymbol{\Sigma}_t$ and only relies on the previous $\boldsymbol{\mu}_t$. Therefore, the magnitude of the model noise $\mathbf{W}$ doesn't affect the prediction step unless there is also an update step. This means that the prediction step is merely an implementation of dead-reckoning, as only internal velocity measurements from the IMU is effectively integrated to a position and orientation. This in turn makes our estimation very vulnerable to sensor noise which can lead to a bad trajectory. Luckily, as seen from the results, the trajectories are reasonable, and without other means of comparing the results to the test data other than the video itself are as good as we can discern.

B. Landmark Updating

When estimating the landmark positions, the EKF performed well as the results generally coincided with the video of the training data. The same goes for the testset. The results are displayed in figures 5-7. That being said, there is a land in the bottom of fig 6 that is very far away from the others. A potential reason for this is that at this point in the trajectory, there are features placed high up in trees. If the difference between the pictures generated by the cameras at this point is small, this could potentially "trick" the camera model into estimating that the feature is far away. To attempt filtering out such astray landmarks, simple outlier rejection was applied to generate the plot in fig 9. Another interesing

result was that the magnitude of the observation nose had a fairly insignificant effect on the landmarks. However, there were some small differences for extreme cases. A moderate noise parameter generally yielded the best results, and $\mathbf{V} = \mathbf{I}$ was used in the project. The effect of the magnitude of $\mathbf{V}$ can be seen in (17a) where large $\mathbf{V}$ yield small $\mathbf{K}_t$. This will cause the filter to prefer the value of the initialization which also yields good results. When the noise is set to be small, the model trusts the model to a much larger degree. In this case, the model would more often mistakenly set points way outside the trajectory. The difference between $\mathbf{V} = 0.0001 \cdot \mathbf{I}$ and $\mathbf{V} = 1000 \cdot \mathbf{I}$ for dataset 20 is displayed in figures 10 and 11.

Fig. 1: Screencap of video where feature data is captured

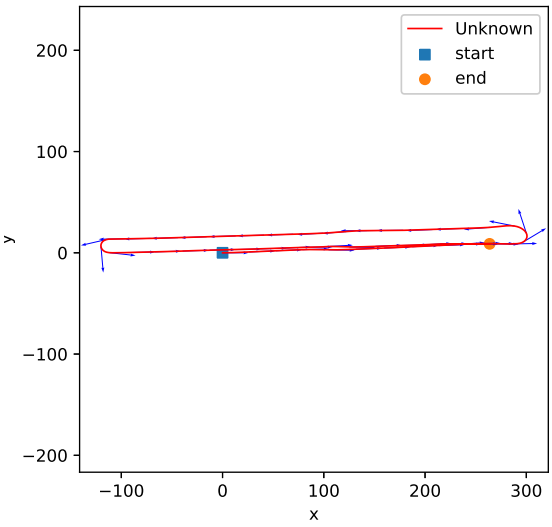

Fig. 2: IMU trajectory in dataset 20

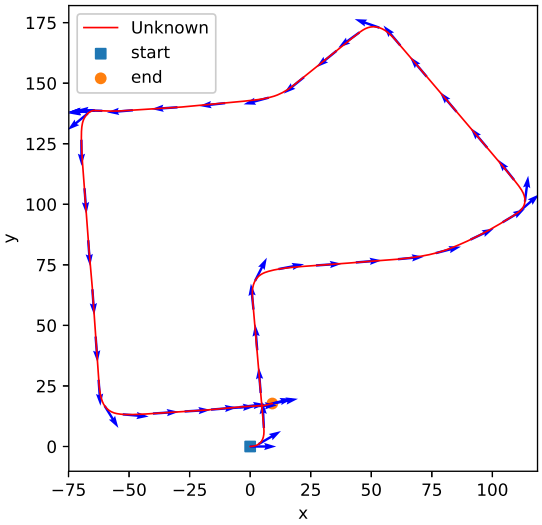

Fig. 3: IMU trajectory in dataset 27

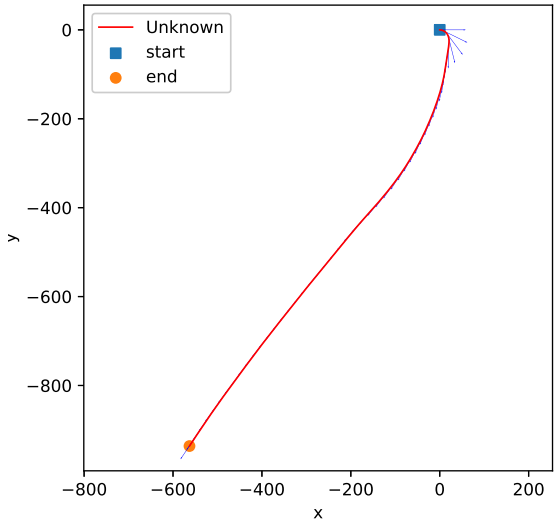

Fig. 4: IMU trajectory in dataset 42

REFERENCES

[1] Antanasov, Nikolay, Lecture 10: Projective Geometry, Camera Model, UC San Diego, 2019

[2] Antanasov, Nikolay, Lecture 12: SE(3) Geometry and Kinematics, UC San Diego, 2019

[3] Antanasov, Nikolay, Lecture 14: Extended and Unscented Kalman Filtering, UC San Diego, 2019

[4] Antanasov, Nikolay, Lecture 15: Visual Intertial SLAM, UC San Diego, 2019

[5] Kalman filter optimality

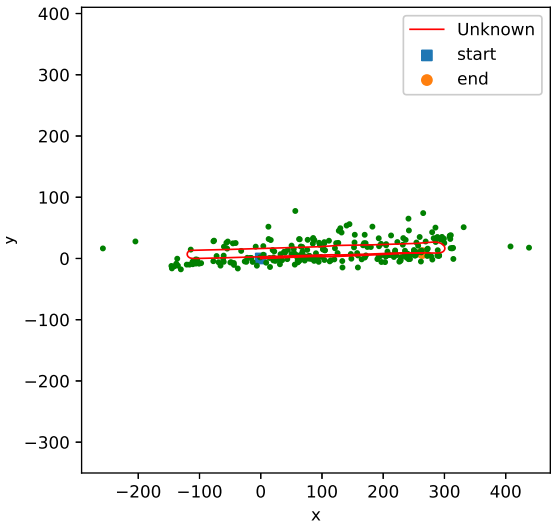

Fig. 5: Plot of landmarks and trajectory in dataset 20

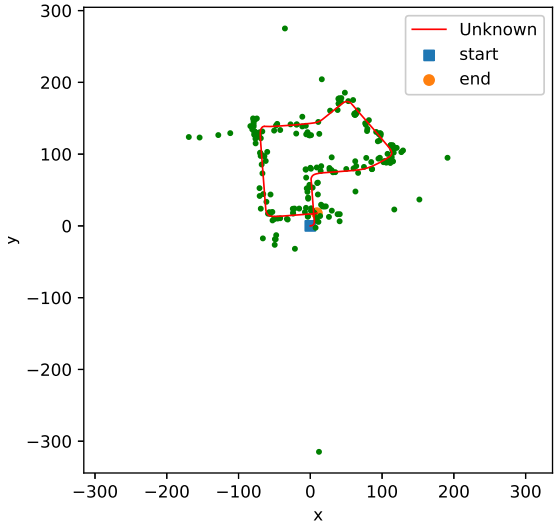

Fig. 6: Plot of landmarks and trajectory in dataset 27

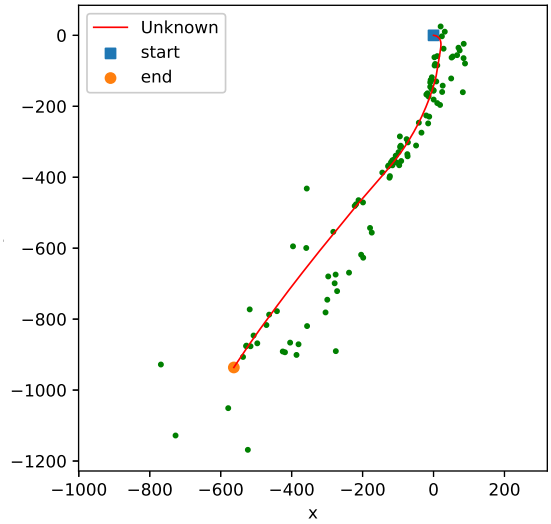

Fig. 7: Plot of landmarks and trajectory in dataset 42

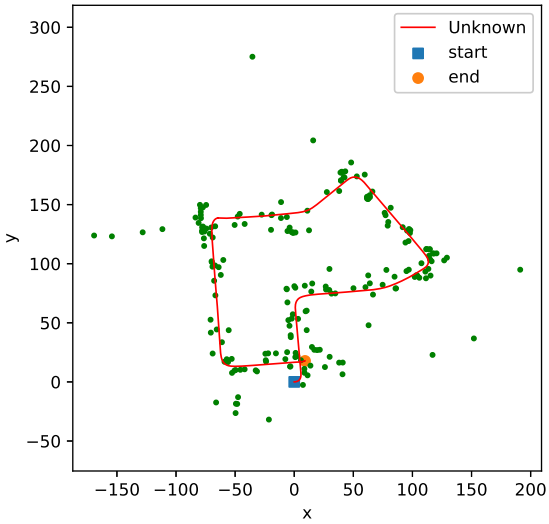

Fig. 8: Plot of filtered landmarks and trajectory in dataset 27

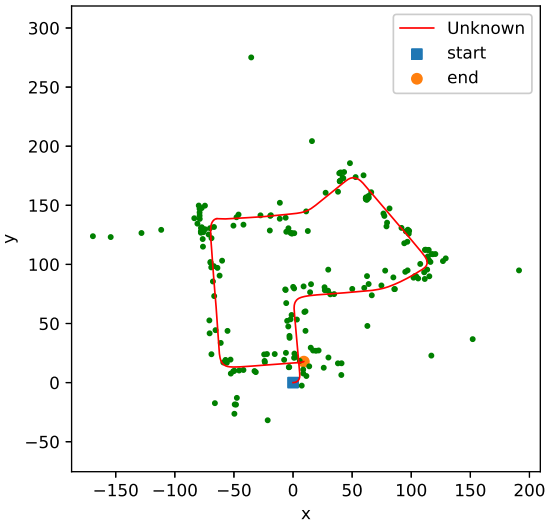

Fig. 9: Plot of filtered landmarks and trajectory in dataset 27

Fig. 10: Plot of landmarks in dataset 20 with $V = 0.0001 \cdot \mathbf{I}$

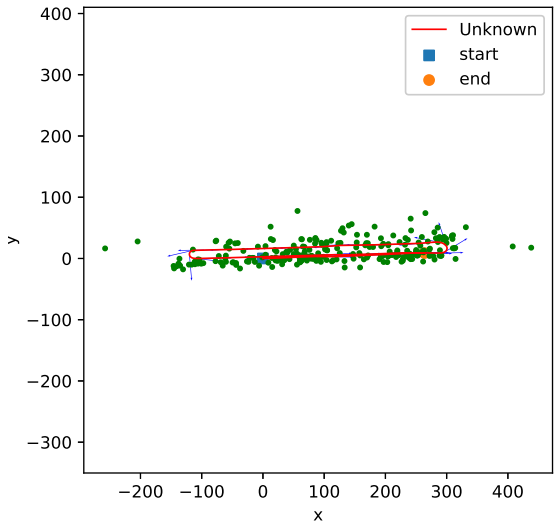

Fig. 11: Plot of landmarks in dataset 20 with $V = 1000 \cdot \mathbf{I}$