Robot Localization and Navigation:Project 3

Abhimanyu Suthar(abs9477) - N10804526

April 25, 2024

1 INTRODUCTION

This report intends to serve as a self-contained description of the implementation of Unscented Kalman Filter(UKF) to fuse the inertial data and the vision-based pose and velocity estimates.
The UKF is implemented in two phases:

- **Part-1**: Visual pose data is utilized for the measurement model.
- **Part-2**: Velocity from the optical flow forms the measurement model

The structure of this report is as follows: Section 2 provides the background to understand the motivation to use UKF and also describes the filter in sufficient detail so that the reader can appreciate the robustness of the implementation in Section 3. Conclusions are drawn in Section 4. Valuable references are mentioned in last Section.

2 BACKGROUND

2.1 Motivation

The Kalman Filter is one of the most widely used estimators for estimating the state of a dynamic system from a series of noisy measurements. The primary goal of the Kalman Filter is to produce an estimate of the true state of the system based on a combination of predictions from the system model and measurement from sensors while taking into account the uncertainties in the both the models. When the system dynamics are linear, the MSME error estimate may be computed using the Kalman Filter. However, as it turns out, most systems of interest are non-linear to varying degrees and therefore require extensions to the base filter.

In the project-1, EKF was utilized where a linearization was applied by using Jacobian at a single point. However it has two major drawbacks which are: The linearization around a single point is only accurate when the non-linearity of the system is low, and derivation of Jacobian matrices are non-trivial in most applications. Therefore, to work around the limitations, unscented Kalman filter is applied to handle non-linear dynamics and measurements.

2.2 UKF

In order to linearize the system, the Extended Kalman Filter applies the Taylor series expansion. However two other approaches are found to yield superior results, one of them being the **unscented Kalman filter** which performs a stochastic linearization of the linear dynamics with a linear model using a weighted regression approach which accounts for probabilistic uncertainty in the data and model fitting.
The UKF uses the unscented transform method where the idea is to deterministically choose a fixed number of sigma points that capture the mean and covariance of the original distribution.[2]

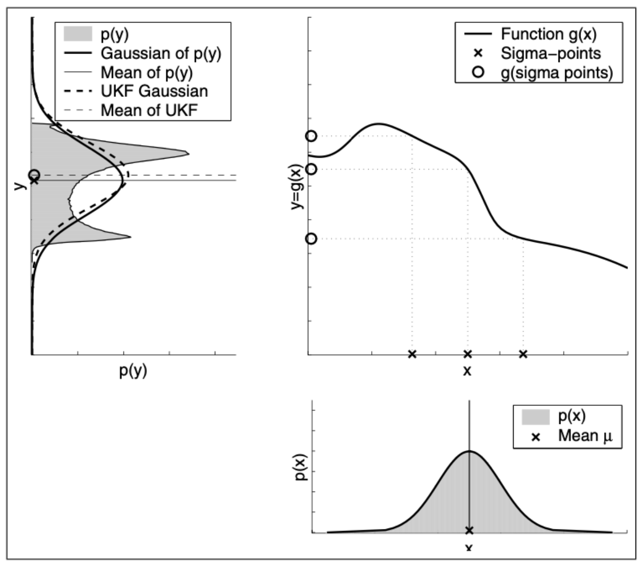

Figure 1: Illustration of linearization applied by the UKF

The visualization can be seen in Fig: 6 where 3 sigma points are propagated through the non-linear function g. The linearized Gaussian is then extracted from the mapped sigma points

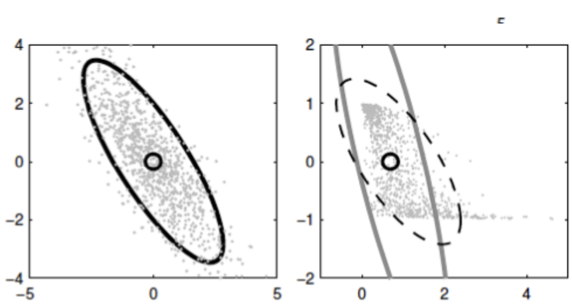

Figure 2: EKF-linearization

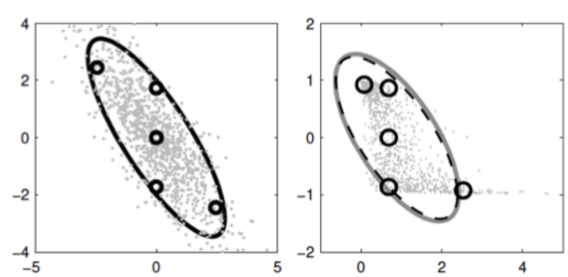

Figure 3: Unscented Transform approximation

As can be seen in Fig:7 that propagating sigma points through the non-linear function approximates the mean and covariance of the distribution much better than just linearizing through a single point (fig6)

The Unscented Transformation is founded on the intuition [1] that it is easier to approximate a probability distribution than it is to approximate an arbitrary non-linear function or transformation.

The concept is most effectively comprehended by delving into the mathematics: For an n-dimensional Gaussian with mean μ and covariance Σ, the resulting 2n + 1 "sigma points" are computed according to the following rule:

$$\begin{aligned}
\mathcal{X}^{[0]} &= \mu \\
\mathcal{X}^{[i]} &= \mu + \left(\sqrt{(n+\lambda)\,\Sigma}\right)_i \quad \text{for } i = 1,\dots,n \\
\mathcal{X}^{[i]} &= \mu - \left(\sqrt{(n+\lambda)\,\Sigma}\right)_{i-n} \quad \text{for } i = n+1,\dots,2n
\end{aligned}$$

where $\lambda = \alpha^2(n+k) - n$ in which α and k are scaling parameters that determine how far the sigma points are spread from the mean. There are also two weights associated with the mean w_m and w_c for calculating the mean and covariance respectively.

$$\begin{aligned}
w_m^{[0]} &= \frac{\lambda}{n+\lambda} \\
w_c^{[0]} &= \frac{\lambda}{n+\lambda} + (1-\alpha^2+\beta)
\end{aligned}$$

$$w_m^{[i]} = w_c^{[i]} = \frac{1}{2(n+\lambda)} \quad \text{for } i = 1,\dots,2n.$$

These sigma points are then passed through the function g

$$y^{[i]} = g(X^{[i]})$$

The mean and covariance of the resulting Gaussian distribution are extracted from the mapped sigma points as follows :

$$\begin{aligned}
\mu' &= \sum_{i=0}^{2n} w_m^{[i]}\,\mathcal{Y}^{[i]} \\
\Sigma' &= \sum_{i=0}^{2n} w_c^{[i]}\,(\mathcal{Y}^{[i]} - \mu')(\mathcal{Y}^{[i]} - \mu')^T.
\end{aligned}$$

There two major types of unscented Transform that we are interested in:

1. Additive Noise
2. Non-Additive Noise

3 Implementation

3.1 Part-1

For the prediction step of the filter, we need to implement the non-additive noise version of the UKF, the process model which is IMU driven is as follows :

$$x_t = f(x_{t-1}, u_t, q_t)$$

$$x \sim N(\mu, \Sigma)$$

$$q_t \sim N(0, Q_t)$$

Let the dimensions of x and q be n and n_q respectively, and $n^{'}$ = n + n_q, the size of state for the quadrotor is 15 and size of noise is 12, hence, $n^{'}$ = 27. This matrix is also called augmented matrix

3.1.1 MATLAB functions

pred-step.m

- Initializtion: alpha, k, n-state, n-noise are defined to set up parameters for the computation. The vectors $U_{augmented}$ and pAug are initialized to create the augmented state vector and its corresponding augmented covariance matrix. $U_{augmented}$ contains the previous state estimate concatenated with zeros to accomomdate additional noise components. pAug is constructed by padding the previous covariance matrix with zeros and adding a scaled indentity matrix for the process noise covariance

$$\boldsymbol{x}_{aug} = \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{q} \end{pmatrix} \text{ with mean } \boldsymbol{\mu}_{aug} = \begin{pmatrix} \boldsymbol{\mu} \\ \boldsymbol{0} \end{pmatrix}$$

- The next step is to use Cholesky decomposition
- After decomposition, sigma points are computed, and then they are propagated through process model based on ZYX Euler Angles notation.
- The process model multiplied by dt is then added to the first 15 rows of the augmented matrix.
- Next, based on the weights, the predicted mean and covariance is computed

upd-step.m

- The model in this case is linear so same as Kalman Filter. The current mean and covariance based on predicted values is updated as follows:

$$\begin{aligned} \mu_t &= \bar{\mu}_t + K_t(z_t - C\,\bar{\mu}_t) \\ \Sigma_t &= \bar{\Sigma}_t - K_t\,C\,\bar{\Sigma}_t \\ K_t &= \bar{\Sigma}_t\,C^T\,(C\,\bar{\Sigma}_t\,C^T + R)^{-1} \end{aligned}$$

plotData.m plots the estimated data with ground truth data.
UKF-KalmanFilt-Part1.m calls all the above functions into one file and updates the mean and covariance estimates at each time step. **init.m** calls the dataset so that the estimates can be compared with ground truth

3.1.2 Plots

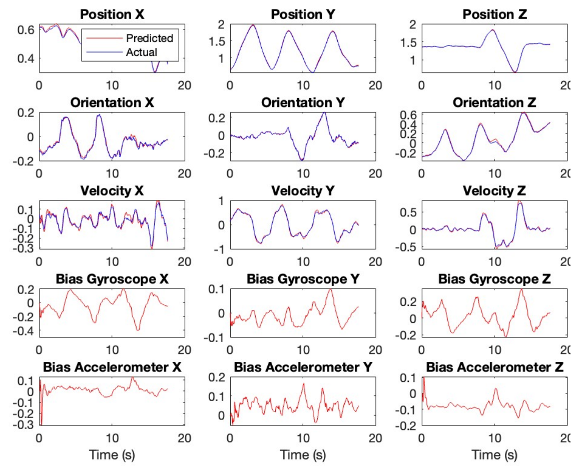

Figure 4: Model-1-Dataset-1

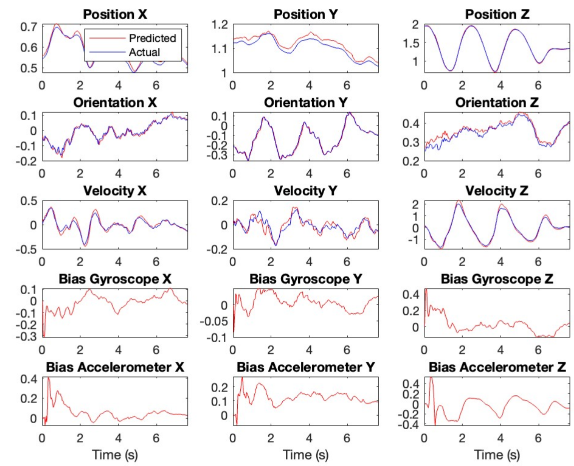

Figure 5: Model-1-Dataset-4

3.1.3 Observations - Part:1

Here, the estimates are very accurate

3.2 Part-2

plotData.m UKF-KalmanFilt-Part2.m calls all the above functions into one file, here the measurement is provided by linear velocity and angular velocity it can and updates the mean and covariance estimates at each time step.
init.m calls the dataset so that the estimates can be compared with ground truth
pred-step.m This is the same model as part-1
upd-step.m

The transformation from camera to robot is defined to be used further

$$z_t = g(x_t) + v_t = {}^{C}\dot{\boldsymbol{p}}_{C}^{W} = \boldsymbol{g}\left(\boldsymbol{x}_{2,}, \boldsymbol{x}_3, {}^{B}\boldsymbol{\omega}_{B}^{W}\right) + \boldsymbol{v}_t \quad \boldsymbol{v}_t \sim N(\boldsymbol{0}, \boldsymbol{V})$$

$${}^{C}\dot{p}_C^W - g_\mu(x_{2,}, x_3, {}^{B}\omega_B^W)$$

$${}^{B}\dot{p}_B^W = l(x_{2,}, x_3)$$

$$\boxed{g_\mu = R_B^C l(x_{2,}, x_3) - R_B^C S(r_{BC}^B) \; R_C^B \; {}^{C}\omega_C^W}$$

3.2.1 Plots

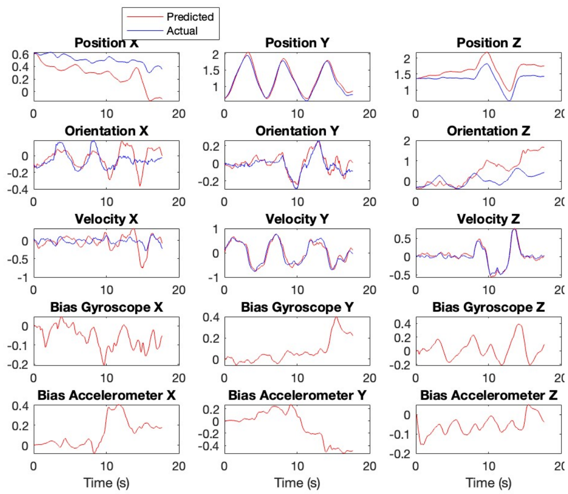

Figure 6: Model-2-Dataset-1

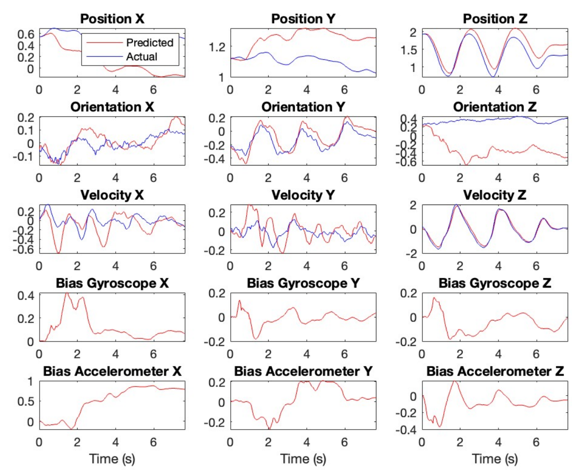

Figure 7: Model-2-Dataset-4

3.2.2 Observations - Part:2

The second part, the estimates are really off because the of the noisy update of velocity from optical flow

Advantages of UKF: It does not require the computation of Jacobians which are difficult to determine in some cases. Therefore, it is also referred to as a derivative-free filter

References

[1] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, pp. 401–422, 2004. [Online]. Available: `https://api.semanticscholar.org/CorpusID:9614092`.

[2] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2005, ISBN: 978-0262201629.