Unscented Kalman Filter for State Estimation of a Micro Aerial Vehicle

Methods and Results

Jagennath Hari – jh7454

Robot Localization and Navigation - Project 3

ROBGY-6213 Spring 2022
Dr. Loianno
April 27, 2022

1) Introduction

In this project, I implemented an Unscented Kalman filter to fuse the inertial data already used in project 1 and the vision-based pose and velocity estimation developed in project 2. The project is divided in 2 parts. In the first one, you will use as measurement the visual pose estimation, whereas in the second one you will use only the velocity from the optical flow.

Since the difference in both parts only lies in the measurement models, I will first discuss my approach to deriving the process model for the prediction step. The measurement models and results sections will each be divided in two for both the parts.

2) Process model and Prediction model

The state x, is a 15x1 vector and is defined by the following –

$$X = \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \\ \dot{\mathbf{p}} \\ \mathbf{b}_g \\ \mathbf{b}_a \end{bmatrix}$$

Where p is the position, q is the orientation, ṗ is the velocity of the VICON, b_g and b_a are gyro and accelerometer bias respectively.

Let us define an augmented random variable $\boldsymbol{x}_{aug} = \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{q} \end{pmatrix}$

Having Mean $\boldsymbol{\mu}_{aug} = \begin{pmatrix} \boldsymbol{\mu} \\ \mathbf{0} \end{pmatrix}$

And Covariance $\boldsymbol{\Sigma}_{aug} = \begin{pmatrix} \boldsymbol{\Sigma} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{Q} \end{pmatrix}$

2.1) Computing the sigma points

We must first choose the number of sigma points which in our case is 15 + 12, as 15 for the state and 12 for the noise to make it augmented.

And by applying the following formulas we can compute the sigma points

$$\mathcal{X}^{(0)} = \boldsymbol{\mu}_{aug}$$

$$\mathcal{X}^{(i)} = \boldsymbol{\mu}_{aug} \pm \sqrt{n' + \lambda'}\left[\sqrt{\boldsymbol{\Sigma}_{aug}}\right]_i \qquad i = 1, \ldots n'$$

$$\lambda' = \alpha^2(n' + k) - n'$$

Where λ' is the spread of sigma points and n' is 27.

2.2) Propagating sigma points through non-linear function

$$\chi_t^{(i)} = f(\chi_{aug,t-1}^{(i),x}, u_t, \chi_{aug,t-1}^{(i),n}) \qquad i = 0, \ldots 2n'$$

$$\chi_t^{(i)} = \begin{bmatrix} \mathbf{x}_3 \\ G(\mathbf{x}_2)^{-1}(\omega_m - \mathbf{x}_4 - \mathbf{n}_g) \\ \mathbf{g} + R(\mathbf{x}_2)(\mathbf{a}_m - \mathbf{x}_5 - \mathbf{n}_a) \\ \mathbf{n}_{bg} \\ \mathbf{n}_{ba} \end{bmatrix}$$

2.3) Computing Mean and Covariance

$$\bar{\boldsymbol{\mu}}_t = \sum_{i=0}^{2n'} W_i^{(m)} \chi_t^{(i)} \qquad \bar{\boldsymbol{\Sigma}}_t = \sum_{i=0}^{2n'} W_i^{(c)\prime} \left(\chi_t^{(i)} - \bar{\boldsymbol{\mu}}_t\right)\left(\chi_t^{(i)} - \bar{\boldsymbol{\mu}}_t\right)^T$$

3) Measurement model and Update Step

In the first part we are provided with measurements of the visual pose estimation for the update. In the second part, only velocity from optical flow is provided.

3.1) Update step for part 1

The update model is as follows-

$$z = \begin{bmatrix} p \\ q \end{bmatrix} + v = \begin{bmatrix} I & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ q \\ \dot{p} \\ b_g \\ b_a \end{bmatrix} + v = C\boldsymbol{X} + \boldsymbol{v}$$

In this case, C is a 6x15 matrix.

$$\boldsymbol{\mu}_t = \overline{\boldsymbol{\mu}}_t + \boldsymbol{K}_t(\boldsymbol{z}_t - C\,\overline{\boldsymbol{\mu}}_t)$$
$$\boldsymbol{\Sigma}_t = \overline{\boldsymbol{\Sigma}}_t - \boldsymbol{K}_t\,\boldsymbol{C}\,\overline{\boldsymbol{\Sigma}}_t$$
$$\boldsymbol{K}_t = \overline{\boldsymbol{\Sigma}}_t\,\boldsymbol{C}^T\,(\boldsymbol{C}\,\overline{\boldsymbol{\Sigma}}_t\,\boldsymbol{C}^T + \boldsymbol{R})^{-1}$$

Where K_t, $\boldsymbol{\mu}_t$ and $\boldsymbol{\Sigma}_t$ is the Kalman Gain, current mean and current covariance respectively.

As we can see the update step for part 1 is the same as the update step for the extended Kalman filter.

3.2) Update step for part 2

3.2.1) Computing the sigma points

We must first choose the number of sigma points which in our case is 15, as 15 for the state.

And by applying the following formulas we can compute the sigma points

$$\chi_t^{(0)} = \overline{\boldsymbol{\mu}}_t$$

$$\chi_t^{(i)} = \overline{\boldsymbol{\mu}}_t \pm \sqrt{n+\lambda}\left[\sqrt{\overline{\boldsymbol{\Sigma}}_t}\right]_i \qquad i = 1, \dots n$$

3.2.2) Propagating sigma points through non-linear function

$$\boldsymbol{z}_t = {}^{C}\boldsymbol{v}_C^W = \boldsymbol{g}\left(\boldsymbol{x}_{2,}, \boldsymbol{x}_3, {}^{B}\boldsymbol{\omega}_B^W\right) + \boldsymbol{\eta} \qquad \boldsymbol{\eta} \sim N(\boldsymbol{0}, \boldsymbol{R})$$

$${}^{C}\dot{\boldsymbol{p}}_C^W = \boldsymbol{R}_B^C\boldsymbol{l}(\boldsymbol{x}_{2,}, \boldsymbol{x}_3) - \boldsymbol{R}_B^C\boldsymbol{S}\left(\boldsymbol{r}_{BC}^B\right)\ \boldsymbol{R}_C^B\ {}^{C}\boldsymbol{\omega}_C^W$$

3.2.3) Computing Mean and Covariance

$$\boldsymbol{z}_{\mu,t} = \sum_{i=0}^{2n} W_i^{(m)} Z_t^{(i)}$$

$$\boldsymbol{C}_t = \sum_{i=0}^{2n} W_i^{(c)}\left(\chi_t^{(i),} - \overline{\boldsymbol{\mu}}_t\right)\left(Z_t^{(i)} - \boldsymbol{z}_{\mu k}\right)^T$$

$$S_t = \sum_{i=0}^{2n} W_i^{(c)} \left(Z_t^{(i)} - \boldsymbol{z}_{\mu k}\right) \left(Z_t^{(i)} - \boldsymbol{z}_{\mu,t}\right)^T + \boldsymbol{R}_t$$

3.2.4) Computing Filter Gain and Filtered Mean and Covariance

$$\boldsymbol{\mu}_t = \bar{\boldsymbol{\mu}}_t + \boldsymbol{K}_t \left(\boldsymbol{z}_t - \boldsymbol{z}_{\mu,t}\right)$$
$$\boldsymbol{\Sigma}_t = \bar{\boldsymbol{\Sigma}}_t - \boldsymbol{K}_t \boldsymbol{S}_t {\boldsymbol{K}_t}^T$$
$$\boldsymbol{K}_t = \boldsymbol{C}_t {\boldsymbol{S}_t}^{-1}$$

Where K_t, $\boldsymbol{\mu}_t$ and $\boldsymbol{\Sigma}_t$ is the Kalman Gain, filtered mean and filtered covariance respectively.

4) Results Part 1

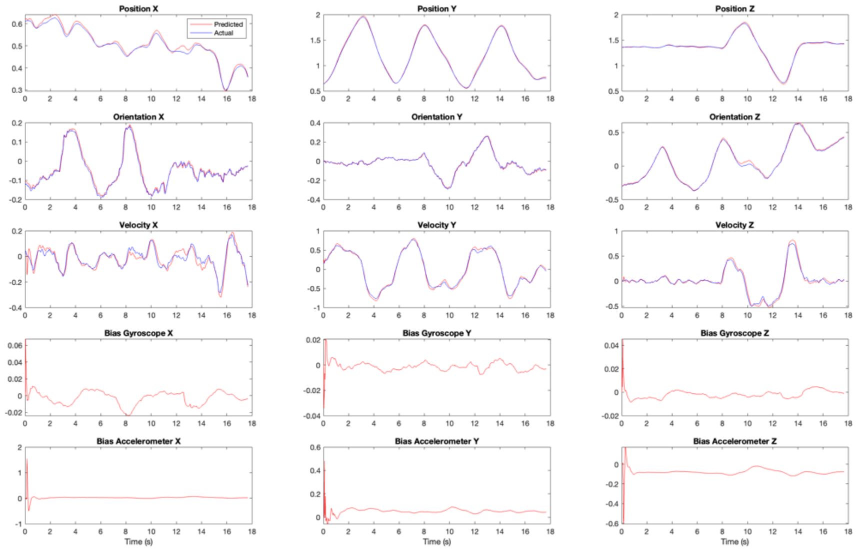

Figure 1: Dataset 1

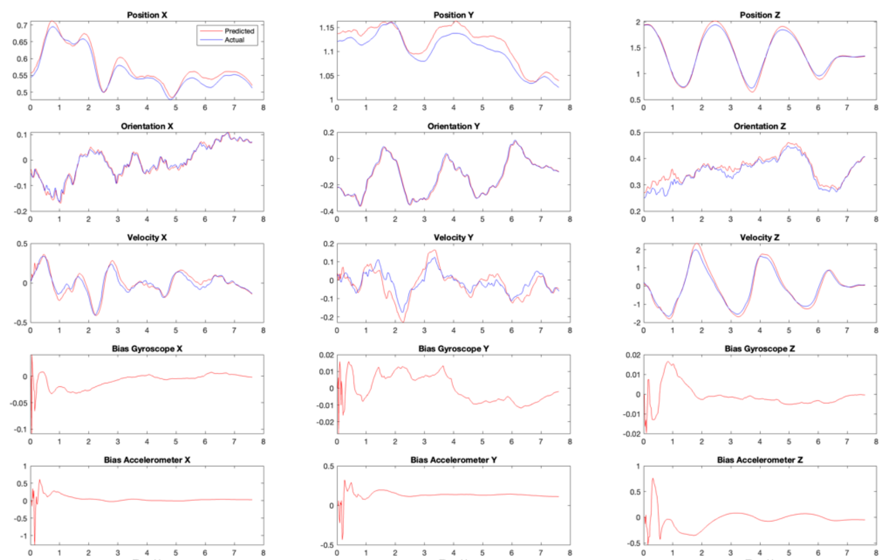

Figure 2: Dataset 4

5) Results Part 2

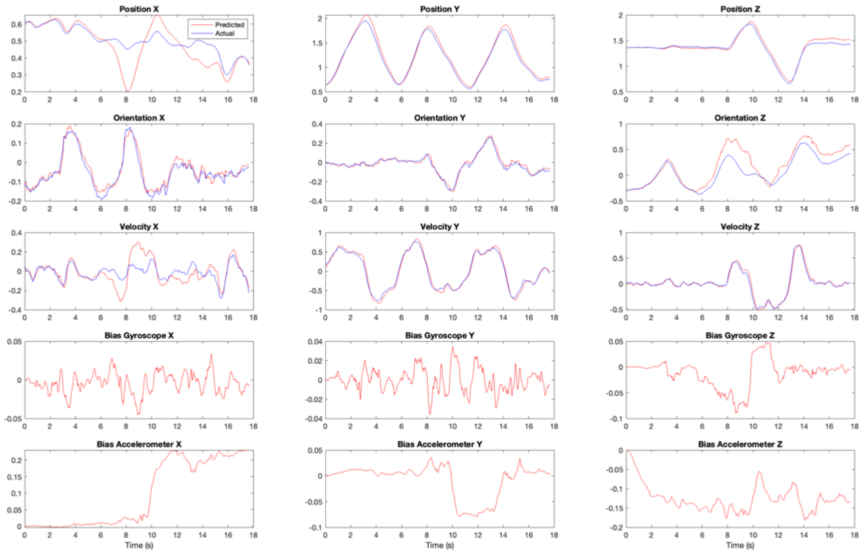

Figure 3: Dataset 1

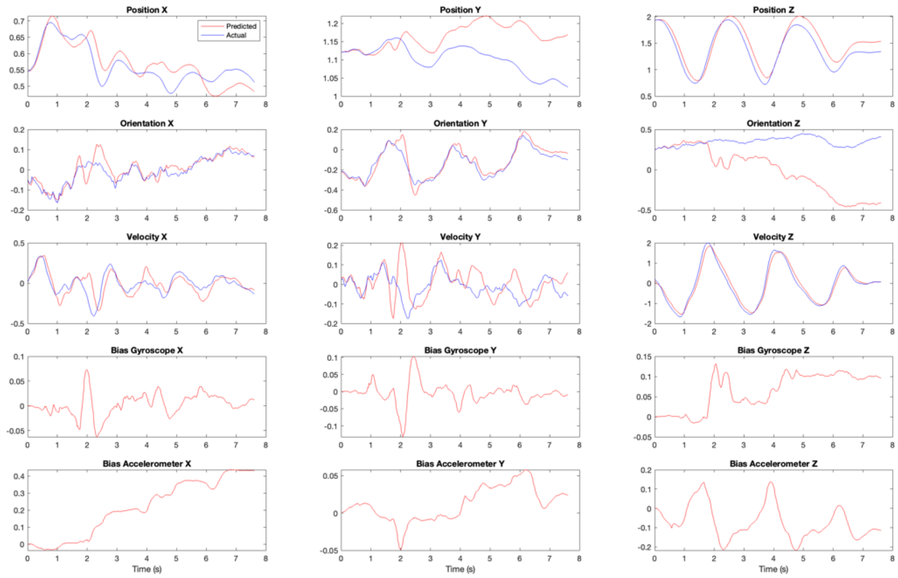

Figure 4: Dataset 4

6) Conclusion

As we can see for part 1 since the measurement is coming from velocity pose estimation the graph matches almost perfected for dataset 1 when plotted. For Dataset 4 the graph is very good with only minor deviations, this is a result of accurate data from visual pose estimation.

For part 2 even after comprehensive turning of noise values the graphs are not perfect. For dataset 1 the graphs match reasonably well with some certain spikes in position x, orientation z, and velocity x. For dataset 4 the graph is very bad with only position x, position z, orientation z, orientation y, velocity y, and velocity z matching to a certain degree. The rest of the graphs are completely off by a scale. This may be because of the inaccurate measurement from the optical flow. Another thing is that noise values for both datasets must be the same, so it is hard to tune the noise values while also maintaining a good balance for both datasets.

Overall, the result from the unscented Kalman filter is extremely good as it is able to fuse the data from both projects, to give a good accurate result of the pose, orientation, and velocity. The sigma points computed help linearize the non-linear function better than the extended Kalman Filter did so in general it is far superior for fusing data than the extended Kalman Filter.