Robot Localisation and Navigation Project Report

State Estimation using Unscented Kalman Filter

Project 3

Professor: Dr. Giuseppe Loianno

Pavan Chowdary Cherukuri
N10938396
pc3088
27 April 2022

Introduction:

The main goal of this project is to develop an Unscented Kalman Filter to fuse the inertial data already used in the Project 1 (Extended Kalman Filter for state estimation) and the vision based pose and velocity estimation developed in the project 2. This project is performed primarily in two parts, in the first part of the project, the measurement is considered to be the estimated visual pose and in the second part of the project, the velocity obtained from the optical flow is considered as the measurement. In the upcoming sections the Theory behind the Unscented Kalman Filter is discussed, which is then followed by approach and coding. In the final parts of the report, the Results and Analysis of the result were discussed.

Theory:

There are many ways to linearize the transformation of a Gaussian. The Taylor series expansion applied by the Extended Kalman Filter is only one way to linearize, another linearization method is applied by the Unscented Kalman Filter which performs stochastic linearization through the use of a weighted statistical linear regression process.
Generally the process model consists of noise, this noise can be classified into additive or non-additive noise. In the first case, the calculations can be performed similarly as done for the Extended Kalman Filter, where we already know the effects of additive noise on the mean and covariance calculations. Whereas in the non-additive noise case, there is some procedure to be followed which will be discussed further.
There are two steps involved in the Unscented Kalman Filter algorithm similar to the Extended Kalman Filter algorithm, which are ‘prediction’ and ‘update’ steps.
The prediction step is the same for both parts of the project, so we will be discussing the prediction step combinedly for both parts.

Prediction Step: In the prediction step, the process model is given by following equation:

$$x_t = f(x_{t-1}, u_t, q_t) \qquad x \sim N(\mu, \boldsymbol{\Sigma}) \quad q \sim N(0, Q)$$

In this case, if the dimension of ‘x’ is considered to be as ‘n’ and the dimension of ‘q’ as ‘n_q’ then we will be calculating n’ which is given by: $n' = n + n_q$.

Step 1: Computation of Sigma Points

The UKF algorithm deterministically extracts the “sigma points” from the gaussian and passes through the function ‘f’. Generally, the sigma points are located at the mean and symmetrically along the main axes of the covariance. To compute sigma points, initially, the state x_t has to be augmented to $x_{aug} = (x_t ; q_t)$ and form the sigma points for the augmented random variable. The calculation of sigma points in the prediction step is given by the following equations.

$$\chi_{aug,t-1}^{(i)} = \boldsymbol{\mu}_{aug,t-1} \pm \sqrt{n' + \lambda'} \left[\sqrt{\boldsymbol{P}_{aug}}\right]_i \qquad \boldsymbol{\mu}_{aug,t-1} = \begin{pmatrix} \boldsymbol{\mu}_{t-1} \\ \boldsymbol{0} \end{pmatrix} \qquad \boldsymbol{P}_{aug} = \begin{pmatrix} \boldsymbol{\Sigma}_{t-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}_t \end{pmatrix}$$

Where, $\mu_{aug,t-1}$ is the augmented mean at time step 't-1' and 'P_{aug}' is the augmented Covariance which are given by the above formulas.
Sigma points which are represented by '$X^{(i)}_{aug,t-1}$ are also calculated by the above formula. Here the square root of the 'P_{aug}' matrix is calculated by cholesky decomposition. The 'i' th column of the augmented covariance matrix is used to calculate the sigma points. It is to be noted that for each value of 'i' there are two sigma points one with '+' and the other with '-' in the formula for calculating the sigma points. This thereby results in n'+n' = 2n' sigma points from this formula. These sigma points are spreaded around the first sigma point which is given by: $X^{(0)} = \mu_{aug}$.
So this results in a total of 2n'+1 sigma points. In this formula **λ**' is given by the formula:

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

Where, 'α' and 'k' determine the sigma points spread.

Step 2: Propagation of Sigma points through the non-linear function 'f'
The calculated Sigma points are now propagated through the function 'f' which can be understood by the below mentioned formula.

$$\chi_t^{(i)} = f(\chi_{aug,t-1}^{(i),x}, u_t, \chi_{aug,t-1}^{(i),n})$$

Where i = 0,...,2n'. The calculated sigma points will be having the state (or size of the matrix) as n' so the first n elements are passed in the first argument of 'f' (that represents the state) and the last 'n_q' are passed in the third argument of 'f' (that represents the noise, since the noise is non-additive). This is the basic formula for any kind of process model, this can be adopted to the process model we are concentrating, which is given by:

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{x}_3 \\ G(\mathbf{x}_2)^{-1}(\boldsymbol{\omega}_m - \mathbf{x}_4 - \mathbf{n}_g) \\ \mathbf{g} + R(\mathbf{x}_2)(\mathbf{a}_m - \mathbf{x}_5 - \mathbf{n}_a) \\ \mathbf{n}_{bg} \\ \mathbf{n}_{ba} \end{bmatrix} = f(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{n})$$

This is the same process model that was used in Project 1, so all the definitions of terms remain the same. So the propagation of the sigma points in the function 'f' results in 'x_dot'. We already know that 'x_dot' can be given as $(x_t - x_{t-1})/dt$. Therefore, the current state 'x_t' can be defined as $x_t = x_{t-1}$ + (x_dot*dt), which explains how the discretization of the function is performed.

Step 3: Compute the Mean and Covariance

In the last and final step of the prediction step the mean and covariance are calculated by using the equations given below:

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

Where,

$$W_0^{(m)'} = \frac{\lambda'}{n' + \lambda'} \qquad W_i^{(m)'} = \frac{1}{2(n' + \lambda')}.$$

$$W_0^{(c)'} = \frac{\lambda'}{n' + \lambda'} + (1 - \alpha^2 + \beta) \quad W_i^{(c)'} = \frac{1}{2(n' + \lambda')}$$

$i = 0,....,2n'$ and $\boldsymbol{\beta} = 2$.

From these calculations, the predicted mean and covariance can be found out which can be returned to be used in the update step.

<u>Update Step:</u>

The update step now takes the previously calculated mean and covariances in the prediction step and it outputs the updated mean and covariance. As discussed above, the measurement model is different for Part 1 and Part 2, hence they are explained separately.

Part 1:

In the Part 1 of the project, the measurement model used the values calculated from the visual pose estimation in the Project 2. Since the 'pose' is in the world frame, the linear measurement model that is used in Project 1 applies here. So, we can directly use the normal Kalman Filter equations with the additive noise in this update step. Hence the equations for the update step can be given by the following equations:

$$\boldsymbol{z}_t = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{\eta} \qquad \boldsymbol{\eta} \sim N(\boldsymbol{0}, \boldsymbol{R})$$

$$\boldsymbol{\mu}_t = \bar{\boldsymbol{\mu}}_t + \boldsymbol{K}_t(\boldsymbol{z}_t - C\,\bar{\boldsymbol{\mu}}_t)$$

$$\boldsymbol{\Sigma}_t = \bar{\boldsymbol{\Sigma}}_t - \boldsymbol{K}_t\,\boldsymbol{C}\,\bar{\boldsymbol{\Sigma}}_t$$

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

All the terms retain the same meanings as defined in Project 1. Finally, the updated mean and updated covariance 'μ_t' and '$\boldsymbol{\Sigma}_t$' are calculated.

Part 2:
In the Part 2 of the project the update step is different from that of Part 1. Here in the update step, the noise is assumed to be additive which is given by:

$$\boldsymbol{z}_t = g(\boldsymbol{x}_t) + \boldsymbol{r}_t \qquad\qquad \boldsymbol{r}_t \sim N(\boldsymbol{0}, \boldsymbol{R}_t)$$

Step 1: Computation of Sigma Points
Sigma Points are calculated by using the below mentioned formulas, where the mean and covariance present in the equations are the predicted mean and covariance and 'n' is the dimensionality of 'x'.

$$\chi_t^{(0)} = \bar{\boldsymbol{\mu}}_t \qquad \chi_t^{(i)} = \bar{\boldsymbol{\mu}}_t \pm \sqrt{n+\lambda}\left[\sqrt{\bar{\boldsymbol{\Sigma}}_t}\right]_i \qquad i = 1, \dots n$$

Step 2: Propagation of Sigma points through the non-linear function 'g'
The calculated Sigma points are now propagated through the function 'g' which can be understood by the below mentioned formula.

$$Z_t^{(i)} = g\left(\chi_t^{(i)}\right) \qquad i = 0, \dots 2n$$

Step 3: Compute the Mean and Covariance
In the last and final step of the update step the mean and covariance are calculated by using the equations given below:

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

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

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

By using the equations the updated mean and covariance are calculated which thereby complete our required calculations for the Unscented Kalman Filter.

Approach and Coding:

In this project, we use the IMU driven model from Project 1 and fuse the inertial data from the Camera Pose and Velocity obtained from the Project 2. Hence, there will be a slight change in this project as compared to the measurement model in Project 1. In Part 1 of Project 3, a variable

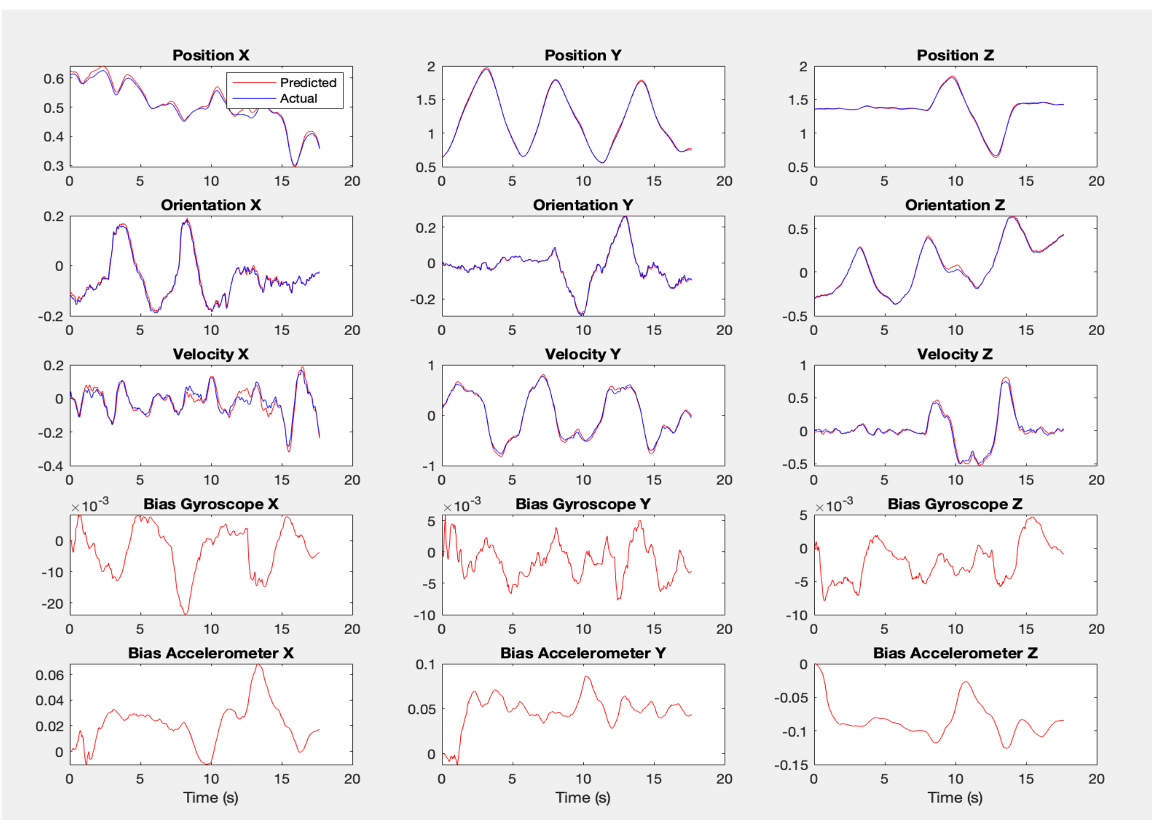
‘Z_vis’ was given as input which is a 6*1 column matrix obtained by concatenating the two column matrices extracted for a particular sample time step from ‘pos’ and ‘pose’. All the other inputs for prediction and update steps were the same as in Project 1.
In Part 2 of the Project, the measurement input in the update step, a variable ‘Z_vis’ was given as input which is a 6*1 column matrix obtained by concatenating the two column matrices extracted for a particular sample time step from ‘vel’ and ‘angVel2’. Also, in this part of the Project, the velocity for the optical flow is expressed in terms of the camera frame, hence appropriate calculations were performed in the code as mentioned in the lecture. The equation of measurement model is mentioned below for better understanding. All the other variables and any other explanations were defined appropriately and also have been commented on in the code for better understanding.

$$z_t = {}^{C}v_C^W = g(x_{2,}, x_3, {}^{B}\omega_B^W) + \eta \qquad \eta \sim N(0, R)$$

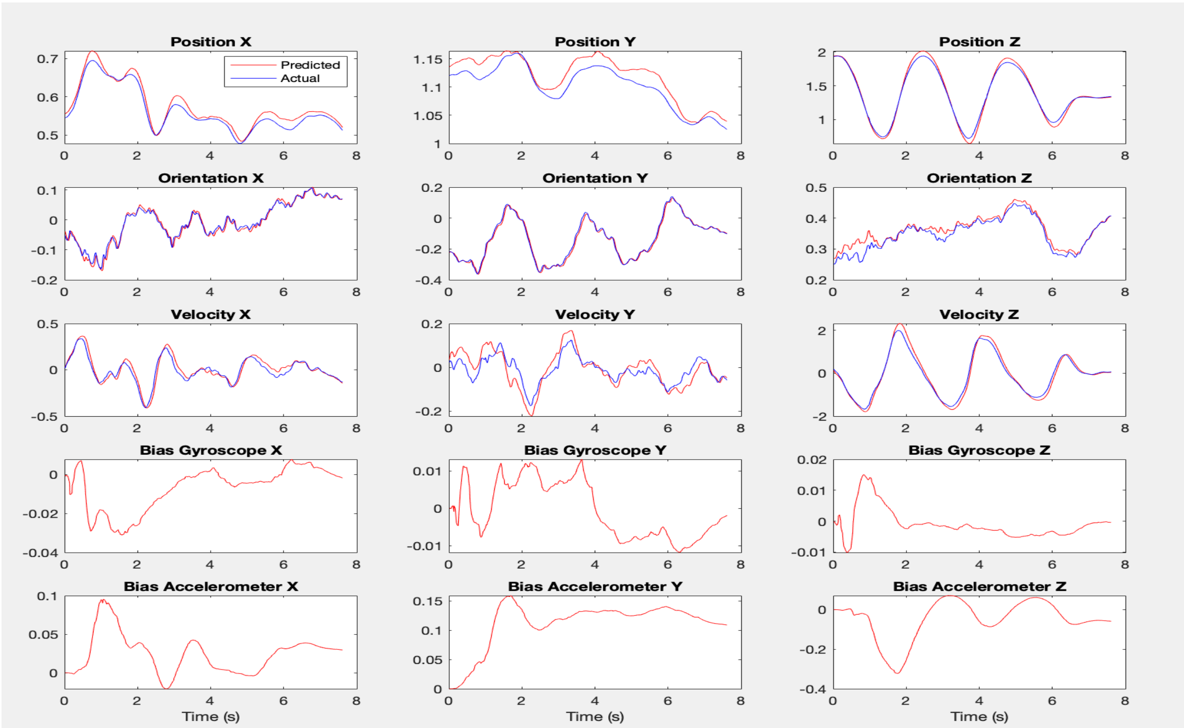
<u>Results:</u>

<u>Part 1:</u>

DataSet 1:

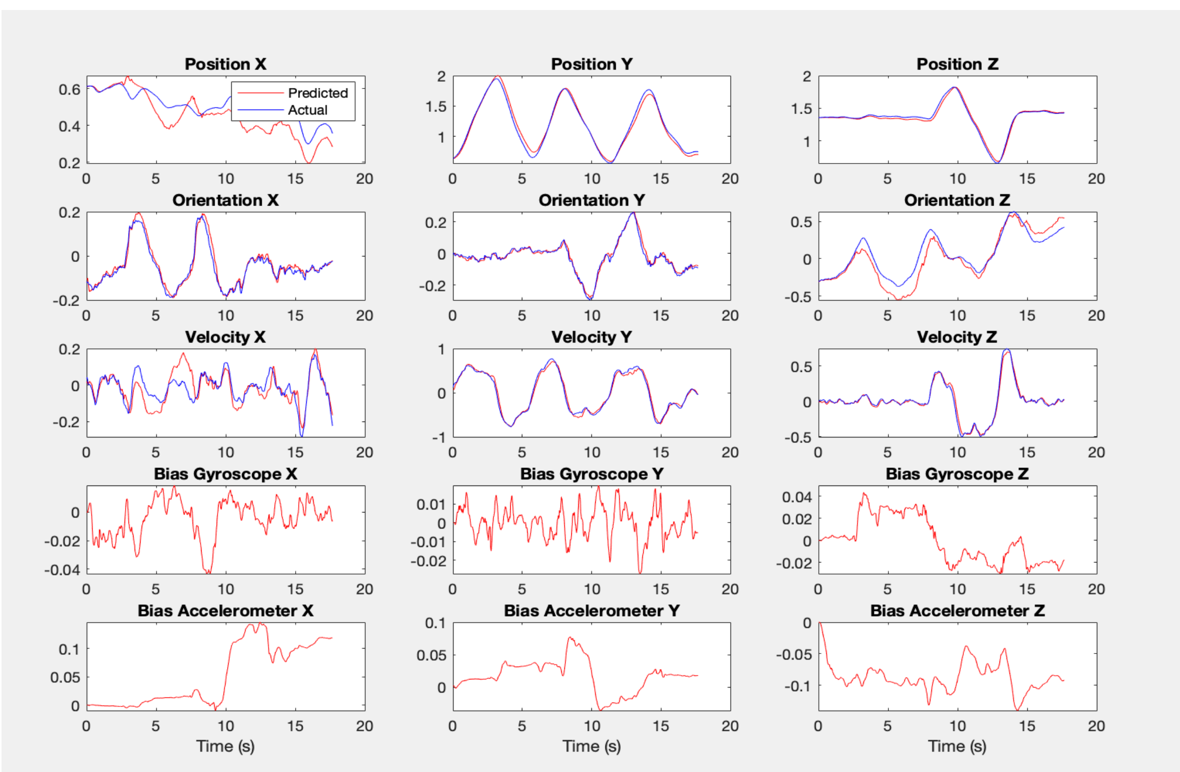


DataSet 4:

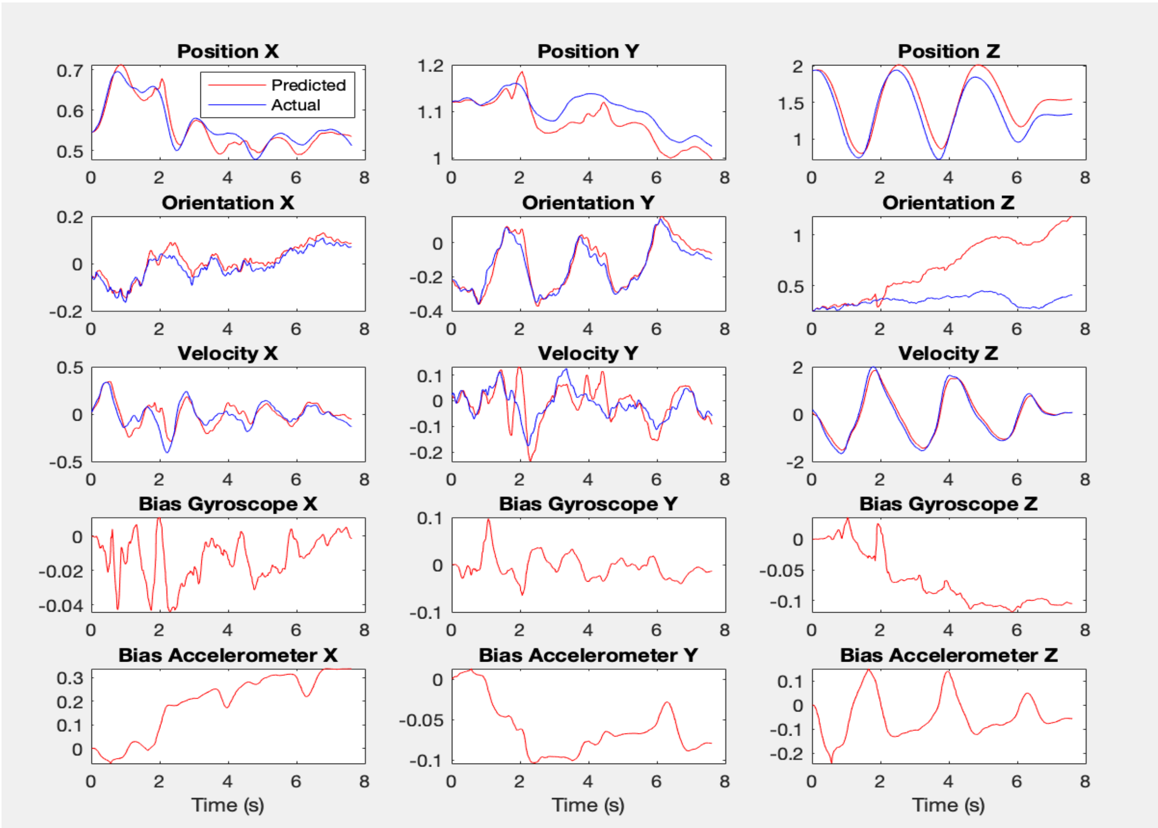


Part 2:

DataSet 1:



DataSet 4:



Analysis:

It is to be noted that the graphs obtained are pretty much accurate in Part 1, and in Part 2 there were slight fluctuations observed and the overlapping is lost at certain places. Since, there might be inconsistencies in the data sets these errors are ignored. Also, point to be taken note is the graphs are subject to change by changing the multipliers for the noise matrices. In the code, the noise multipliers that provide optimal output were chosen.

References:

Lecture Notes by Dr. Giuseppe Loianno.

Probabilistic Robotics Chapter 3.