NYU TANDON SCHOOL OF ENGINEERING

PROJECT REPORT

Robot Localization and Navigation

Submitted by:

Name: Harshavardhan Vibhandik
NET ID: hsv2015
NYU ID: N13023471

Spring 2024
Submitted on: March 24, 2024

ROB-GY6213 Robot Localization and Navigation

Instructor: Prof. Giuseppe Loianno

Introduction

In Project 3, we have to develop the Unscented Kalman Filter using the Data for Projects 1 and 2. Project 1: Inertial data; Project 3: pose and velocity data. The UKF is used to improve the system's non-linearity.

Here, we have to develop an algorithm for the prediction step that utilizes the sigma points to capture the probability distribution and the system's non-linearity.

Explanation & Results

Part I:

In this part, we need to use the pose in the world frame and the update step from Project 1 by considering the inertial IMU-driven data

Here, we consider the points on the probability distribution curve known as sigma points.

$$\mathcal{X}_{aug}^{(0)} = \boldsymbol{\mu}_{aug}, \qquad \mathcal{X}_{aug}^{(i)} = \boldsymbol{\mu}_{aug} \pm \sqrt{n' + \lambda'} \left[\sqrt{\boldsymbol{\Sigma}_{aug}} \right]_i \qquad i = 1, \dots n' \longrightarrow \mathcal{X}_{aug}^{(i)}$$

Where x_aug at 0 was the mean value and the x_aug at i were the values to the right and left of the mean using the above formula. i varies from 0 to n'. Where n' is 27.

n is the size or the number of points augmented through the size of the state (size 15) and noise(12). The state is mentioned below:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \\ \mathbf{x}_5 \end{bmatrix} = \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \\ \dot{\mathbf{p}} \\ \mathbf{b}_g \\ \mathbf{b}_a \end{bmatrix} = \begin{bmatrix} \text{position} \\ \text{orientation} \\ \text{linear velocity} \\ \text{gyroscope bias} \\ \text{accelerometer bias} \end{bmatrix} \in \mathbf{R}^{15}$$

nb = gyroscope noise,

na = the accelerometer noise and

nbg &, nba = the bias noise

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

Where alpha and k determine the sigma points spread

By using the above formula, we determine the sigma point spread where alpha = 0.001, k= 1, and beta =2.

$$\mathcal{X}_{aug}^{(i)} = \begin{bmatrix} \mathcal{X}_{aug}^{(i),x} \\ \mathcal{X}_{aug}^{(i),q} \end{bmatrix}$$

Here, Xaug (x) is the size of the state, and Xaug(q) is the size of the noise.

Then, we have to find the covariance considering the noise of size 12 and formulating a matrix

After finding the sigma points of the state and the covariance, we propagate these values through a non-linear function h.

$$\mathcal{Y}^{(i)} = h\left(\mathcal{X}_{aug}^{(i),x}, \mathcal{X}_{aug}^{(i),q}\right) \qquad i = 0, \dots 2n'$$

We then exploit the values of X-aug, which is a 27x55 matrix size, to formulate the x_dot. We iterate through X_aug and capture the state values of position, orientation, velocity, gyroscope bias and accelerometer bias from the first 15 elements from each column, respectively, and finally, the noises from the last 12 elements from each column, respectively. After formulating x_dot, we have to determine the system's state to finally localise the robot.

Finally, we find and use the corresponding weight values for the mean0 and mean(i) and covariance0 and covariance(i). We compute the final mean and the covariance for the UKF.

$$\boldsymbol{m}_U = \sum_{i=0}^{2n'} W_i^{(m)'} \mathcal{Y}^{(i)}$$

$$\boldsymbol{C}_U = \sum_{i=0}^{2n'} W_i^{(c)'} \left(\mathcal{X}^{(i),x} - \boldsymbol{\mu}\right)\left(\mathcal{Y}^{(i)} - \boldsymbol{m}_U\right)^T$$

Update step:

- $\mu_t = \bar{\mu}_t + K_t\,(z_t - g(\bar{\mu}_t, 0))$
- $\Sigma_t = \bar{\Sigma}_t - K_t\,C_t\,\bar{\Sigma}_t$
- $K_t = \bar{\Sigma}_t\,C_t^T\,(C_t\,\bar{\Sigma}_t\,C_t^T + W_t\,R\,W_t^T)^{-1}$

The update step uses the function z = g(x,v), where x is the Prev state.

$$\mathbf{z} = \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \\ \dot{\mathbf{p}} \end{bmatrix} + \mathbf{v} = \begin{bmatrix} I & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \\ \dot{\mathbf{p}} \\ \mathbf{b}_g \\ \mathbf{b}_a \end{bmatrix} + \mathbf{v} = C\,\mathbf{x} + \mathbf{v}$$

In Part I, we consider only the position and the orientation from the vicon, so the equation will change to

[p ; q] +v = [I3x3 03x3 03x3 03x3 03x3; 03x3 I3x3 03x3 03x3 03x3] * [uPrev] = Cx + v

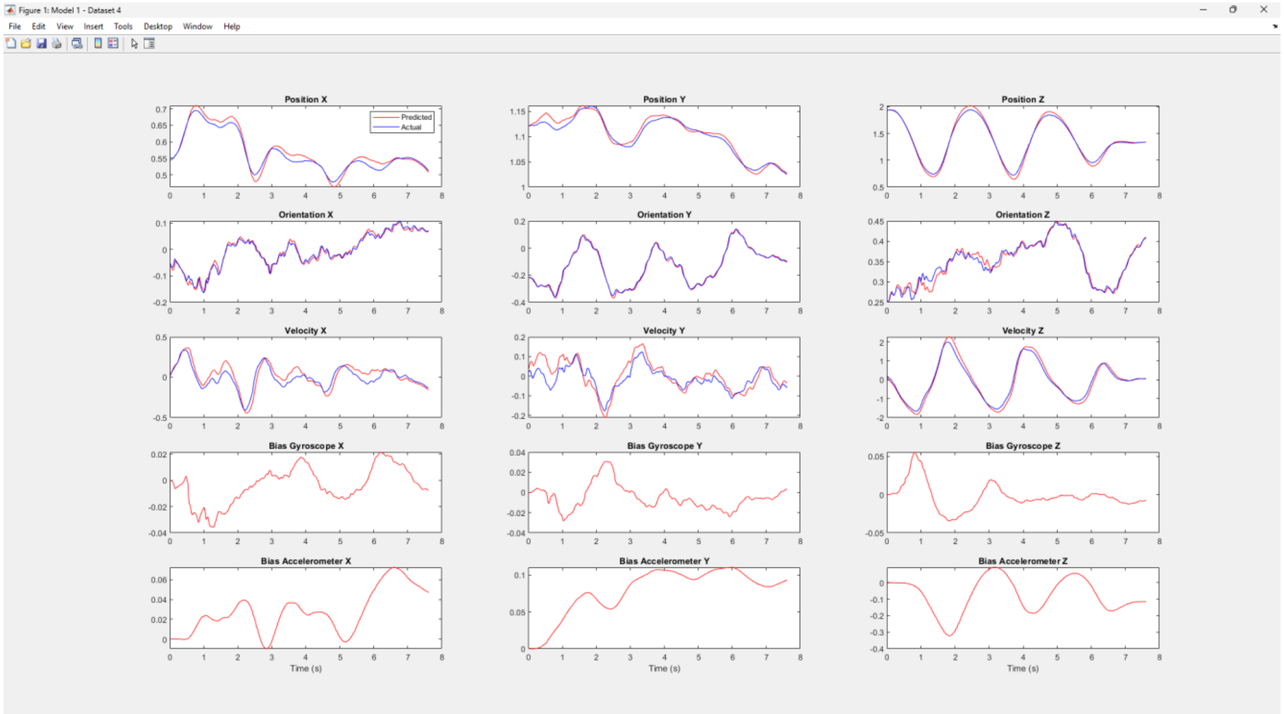
Here, v is considered a matrix of zeroes since we consider the noise from the vicon data to be 0 as we linearise the matrix by eliminating the noise.

The mean in the updated step is calculated using the predicted mean in the pred_function, and then by correcting the data using the actual measurement data from the Vicon (Z), the matrix 'z' is calculated.

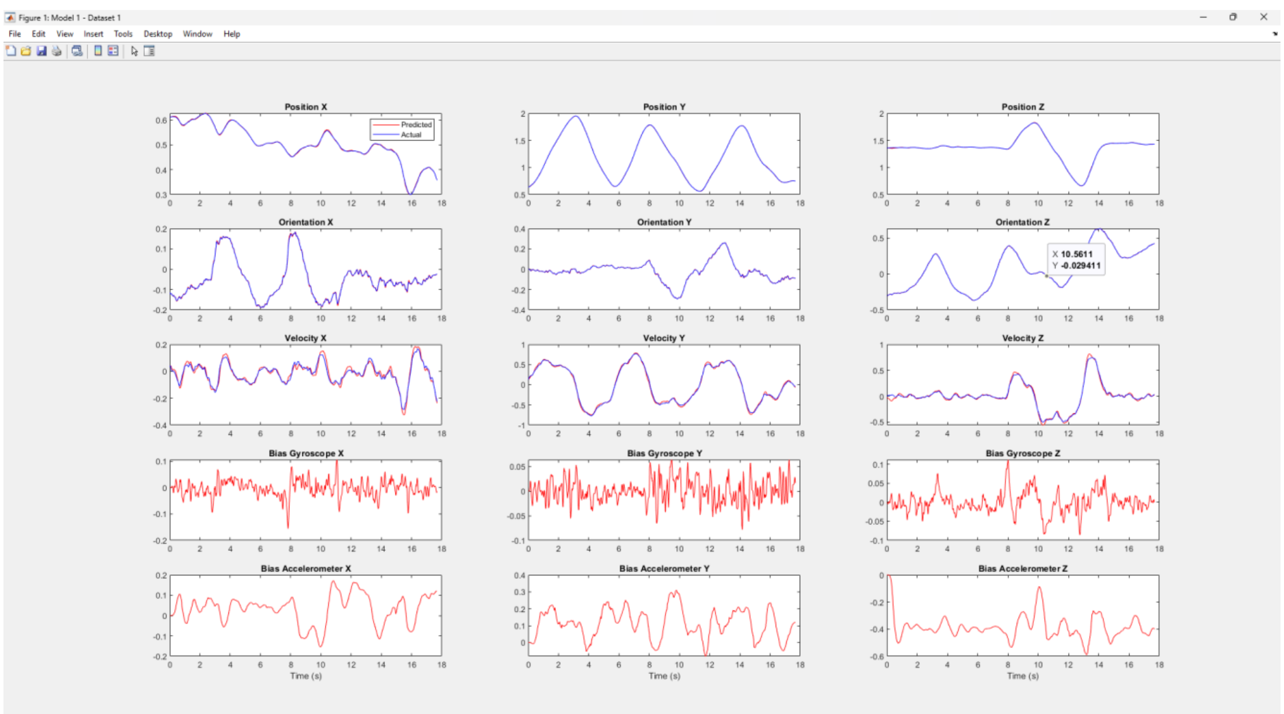
Kt is the Kalman gain, which acts like a weighted matrix to correct the difference between the actual and calculated values of the measurements.

In conclusion, the predicted step gives the estimated mean and estimated covariance of the state, which is given as the input to the updated step for correcting the measurement from the Vicon data.

Results:



Part1: Dataset: 4



Part 1: Dataset 1

Part II:

In this part, we need to convert the pose and velocity from the camera to the world frame.

The prediction step is the same as part I, as it is useful for calculating the uEst and CovarEst.

These outputs are inputs to the update step to calculate the uCurr and covarCurr.

Here, the data is exploited in the camera frame, and thus, the measurement model changes in the project II

$$\chi_{aug,t}^{(i)} = \boldsymbol{\mu}_{aug,t} \pm \sqrt{n'' + \lambda''}\left[\sqrt{\boldsymbol{\Sigma}_{t,aug}}\right]_i$$

The same formula calculates the sigma points using the uEst, where we take 2n+1 points. Where n is 15 points. The importance of n is that these are points on either side of the mean, and the mean is also taken as a sigma point. Hence, we take in 31 points. We then calculate the x_state for each of these points by propagating these sigma points using a function ‘g’

$$Z_t^{(i)} = g(\chi_{aug,t}^{(i),x}, \chi_{aug,t}^{(i),v})$$

After calculating the sigma points, we calculated their predicted mean and the system's covariance.

The predicted covariance and the cross-covariance of the model were also calculated using the weights and the below-mentioned formula

$$\boldsymbol{z}_{\mu,t} = \sum_{i=0}^{2n''} W_i^{(m)''} Z_t^{(i)}$$

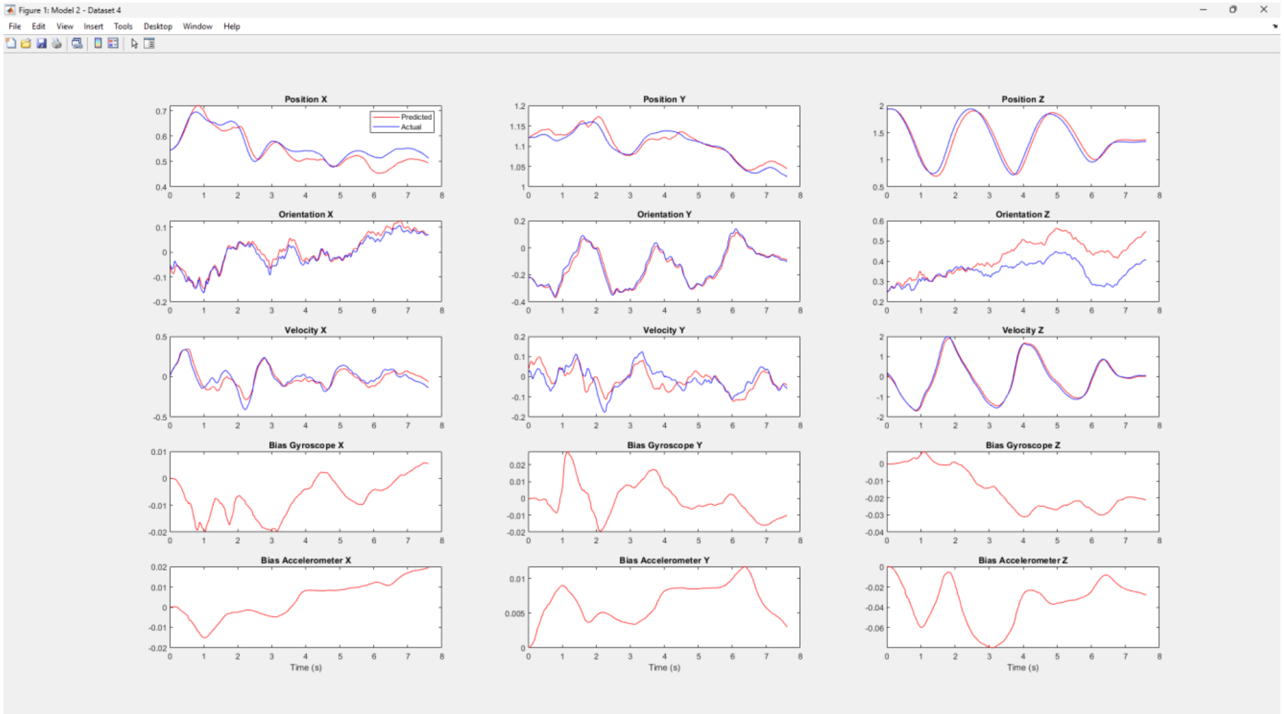
Update

$$\boldsymbol{C}_t = \sum_{i=0}^{2n''} W_i^{(c)''}\left(\chi_{aug,t}^{(i),x} - \bar{\boldsymbol{\mu}}_t\right)\left(Z_t^{(i)} - \boldsymbol{z}_{\mu,t}\right)^T \qquad \boldsymbol{S}_t = \sum_{i=0}^{2n''} W_i^{(c)''}\left(Z_t^{(i)} - \boldsymbol{z}_{\mu,t}\right)\left(Z_t^{(i)} - \boldsymbol{z}_{\mu,t}\right)^T$$

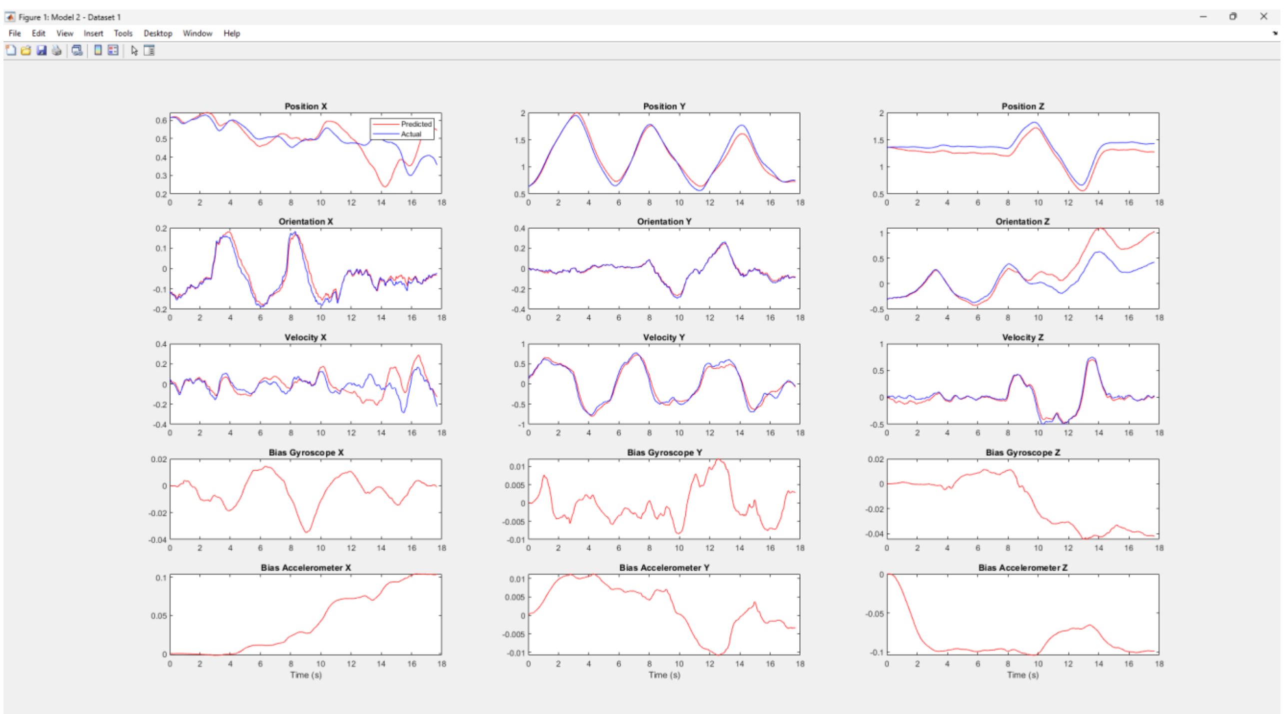
Once the above values are calculated, the updated mean and the updated covariance is calculated:

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

Results:



Part 2: Dataset: 4



Part 2: Dataset: 1