NYU TANDON SCHOOL OF ENGINEERING

ROBOT LOCALIZATION AN NAVIGATION

Project – 3

Name: Rallapalli Dinesh Sai, Naidu
ID: N19602446
NetID: dr3696

Masters in Mechatronics and Robotics Engineering, Department of Mechanical and Aerospace Engineering, Tandon School of Engineering

ROB-GY_6213_1_A **Spring'24 (04/24/2024)**

Instructor: Prof. Giuseppe Loianno

Rallapalli Dinesh Sai, Naidu

N19602446
NetID: Dr3696

1. Overview

Overview

Project-3 aims to implement an Unscented Kalman Filter (UKF) to fuse data from an Inertial Measurement Unit (IMU) with visual pose and velocity estimations obtained from a camera. Divided into two parts, the first part focuses on fusing IMU data with visual pose estimation, utilizing a linear measurement model due to the pose being provided in the world frame. In the second part, the UKF will fuse IMU data with optical flow velocity, requiring a careful transformation of the nonlinear measurement model into the body frame, coincident with the IMU frame. Testing will be conducted using Dataset 1 and Dataset 4, with results from Project 2 provided separately. The project's challenges include ensuring effective capture of system nonlinearity by the UKF and managing runtime considerations due to its potentially higher computational demands compared to other filters. Ultimately, the goal is to demonstrate improved filtering performance over previous techniques.

1. Unscented Kalman Filter

The Unscented Kalman Filter (UKF) is a nonlinear extension of the classic Kalman Filter, designed to handle systems with non-Gaussian and highly nonlinear dynamics. Unlike the Extended Kalman Filter (EKF), which linearizes the system dynamics and measurement equations, the UKF employs a deterministic sampling technique called the Unscented Transform to propagate a set of sigma points through the nonlinear functions. These sigma points capture the mean and covariance of the state distribution more accurately than linearization. By passing these sigma points through the nonlinear functions, the UKF approximates the mean and covariance of the predicted and updated states. This approach often yields more accurate estimates compared to the EKF, especially in highly nonlinear systems where linearization may introduce significant errors. Additionally, the UKF retains the efficiency of the Kalman Filter, making it suitable for real-time applications.

Using the Unscented Transform (UT) with Non-Additive noise in the Unscented Kalman Filter (UKF) involves the following steps:

1. Selection of Sigma Points: Sigma points are chosen deterministically around the mean of the state distribution. These sigma points capture the distribution's characteristics and are propagated through the nonlinear functions.

2. State Prediction: The selected sigma points are propagated through the process model (system dynamics) to predict the state's evolution. This step allows the UKF to capture the nonlinearities in the system's behavior.

3. Covariance Prediction: After predicting the state, the covariance matrix is updated to reflect the uncertainty in the predicted state. This involves computing the spread of the propagated sigma points around the predicted mean.

4. Observation Prediction: Similarly, the sigma points are transformed through the measurement model to predict the expected measurements based on the predicted state.

5.Incorporating Measurement Information: The predicted measurements are compared with the actual measurements obtained from sensors. The difference between predicted and actual measurements is used to correct the predicted state and covariance using the Kalman gain.

6. Update State and Covariance: Finally, the corrected state estimate and covariance matrix are updated based on the Kalman gain, incorporating the information from both the process model and the measurement model.

By using the Unscented Transform with Non-Additive noise, the UKF is capable of handling nonlinearities and non-Gaussian noise distributions more accurately compared to linearization-based methods like the Extended Kalman Filter (EKF). This makes it particularly suitable for applications where the system dynamics are highly nonlinear or where the noise characteristics are complex and non-additive.

2. Process

Prediction Step

In the prediction step, the first task was to calculate the value of 'lambda' using the equation provided.

$$\lambda' = \alpha^2(n' + k) - n' \quad \alpha \text{ and } k \text{ determine sigma points spread}$$

For a Gaussian prior distribution good parameters are $\alpha = 0.001$, $k = 1$ and $\beta = 2$

Then, an augmented state space was defined, along with augmented mean and covariance matrices. The Cholesky decomposition was applied to find the square root of the augmented covariance matrix.

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

$$\boldsymbol{\mu}_{aug,t-1} = \begin{pmatrix} \boldsymbol{\mu}_{t-1} \\ \mathbf{0} \end{pmatrix} \begin{matrix} \leftarrow \text{size 15} \\ \leftarrow \text{size 12} \end{matrix} \qquad \boldsymbol{P}_{aug} = \begin{pmatrix} \boldsymbol{\Sigma}_{t-1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{Q}_t \end{pmatrix}$$

$$\mathcal{X}_{aug,t-1}^{(i)} = \begin{bmatrix} \mathcal{X}_{aug,t-1}^{(i),x} \\ \mathcal{X}_{aug,t-1}^{(i),n} \end{bmatrix} \begin{matrix} \leftarrow \text{size 15} \\ \leftarrow \text{size 12} \end{matrix}$$

Subsequently, a for-loop was used to compute the sigma points, and all the sigma points were stored in a 27 x 55 sized matrix. These sigma points were then passed through the process model, which is similar to the process model used in project one, but with some modifications.

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

$$\longrightarrow \qquad \boldsymbol{x}_t = f(\boldsymbol{x}_{t-1}, \boldsymbol{u}_t, \boldsymbol{n}_t)$$

Afterward, the weights were computed to calculate the estimated mean and covariance. The equations used to calculate the weights were derived from the Unscented Kalman Filter framework.

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

$$W_0^{(m)\prime} = \frac{\lambda'}{n' + \lambda'} \qquad W_i^{(m)\prime} = \frac{1}{2(n' + \lambda')} \qquad i = 1, \ldots 2n'$$

To compute the estimated mean (uEst) and covariance (covarEst), the following equations were iteratively applied within a for loop, aggregating the contributions from each sigma point:

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

Here, Wi represents the weight assigned to each sigma point, Xi represents the sigma point, and n is the number of sigma points. These equations ensure that the estimated mean and covariance capture the weighted contributions from all sigma points, providing an approximation of the system's state and its uncertainty.

Update Step

In the update step, the predicted state from the prediction step is refined using the sensor measurements. The update process begins by defining the observation model g(xt,vt) , where xt is the predicted state and vt represents the measurement noise with zero mean. Subsequently, the Kalman Gain, updated mean, and covariance are calculated using the following equations

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

These equations represent the core of the update step, where the Kalman Gain determines the extent to which the measurement is incorporated into the updated state estimate, ensuring a balance between the predicted state and the observed measurements while considering the respective uncertainties.

Unscented Kalman Filter

To implement the Unscented Kalman Filter, we begin by initializing the necessary data files and parsing the data into the main script. This involves obtaining observed

data for position and orientation from the Vicon system and initializing the mean values using the initial values from the Vicon. In the main loop, which iterates over the timestamps available in the dataset, we calculate the time difference (dt) between consecutive timestamps by defining variables for the current and previous timestamps (t2 and t1). For each iteration, we extract angular velocity and acceleration values from the IMU data corresponding to the current timestamp. These values, along with the previous mean and covariance estimates (uPrev and covarPrev) and dt, are passed to the pred_step function to obtain the estimated covariance and mean (covarEst and uEst).

After updating t1 with the value of t2, we extract the observed data (z_t) from project 2 data for the position and orientation of the robot. This data is then passed along with the output from the pred_step function to the upd_step function to compute the updated mean and covariance (uCurr and covarcurr). Finally, we update the covariance and mean for the next iteration, save the updated mean in the savedStates matrix for plotting, and proceed with the next iteration of the loop.

Prediction Step Part-2

The prediction step will be followed in a similar way as that in Part 1.

Unscented Kalman Filter Part-2

The program is similar to that in the previous part, although, the data z_t, this time is the velocity data from the project 2.

Update Step Part-2

In this part, we utilize data from the optical flow obtained in project 2. Similarly to the previous part, we begin by computing the value of lambda and then proceed to calculate the augmented state, mean, and covariance. It's essential to transform the frames to ensure consistency, as the data from project 2 is in the camera frame whereas we require it in the world frame. Afterward, we compute the sigma points following the same procedure as before. These sigma points are then passed through the non-linear function. Next, we calculate the weights using the current values of n, lambda, alpha, and beta. Subsequently, the predicted mean is computed using the equation Zut, Further, I calculated the predicted cross-covariance Ct and predicted covariance of the measurement St using the equations

$$z_{\mu,t} = \sum_{i=0}^{2n''} W_i^{(m)''} Z_t^{(i)} \quad \longleftarrow \text{Update}$$

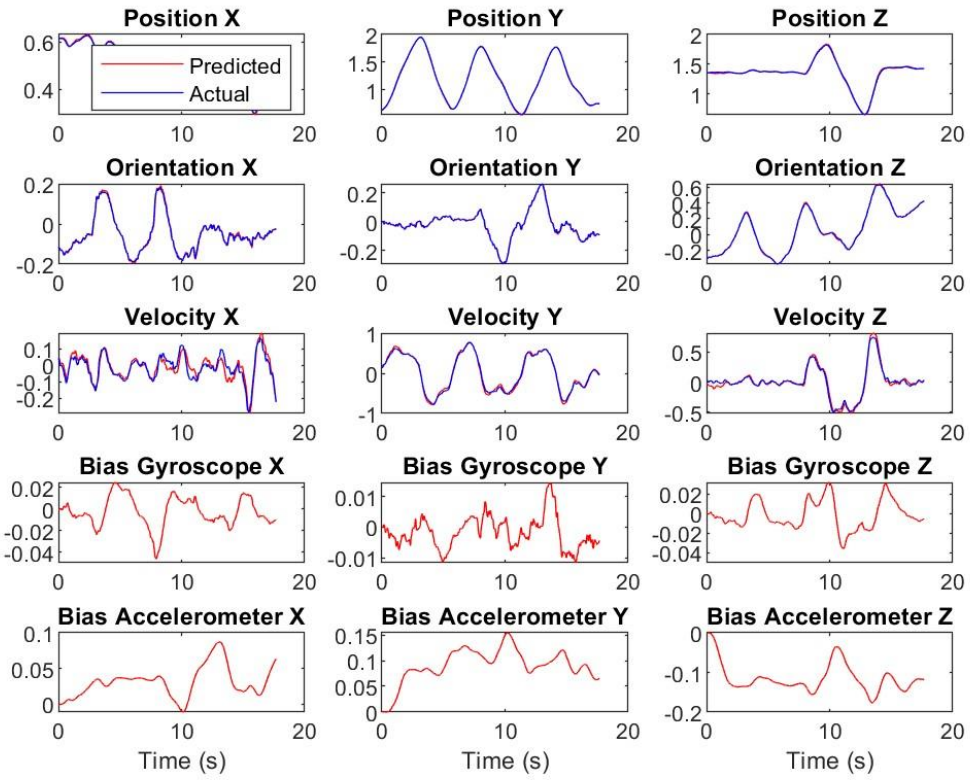
$$C_t = \sum_{i=0}^{2n''} W_i^{(c)''} \left(\chi_{aug,t}^{(i),x} - \bar{\mu}_t\right)\left(Z_t^{(i)} - z_{\mu,t}\right)^T \qquad S_t = \sum_{i=0}^{2n''} W_i^{(c)''} \left(Z_t^{(i)} - z_{\mu,t}\right)\left(Z_t^{(i)} - z_{\mu,t}\right)^T$$

3. Plots

Note:

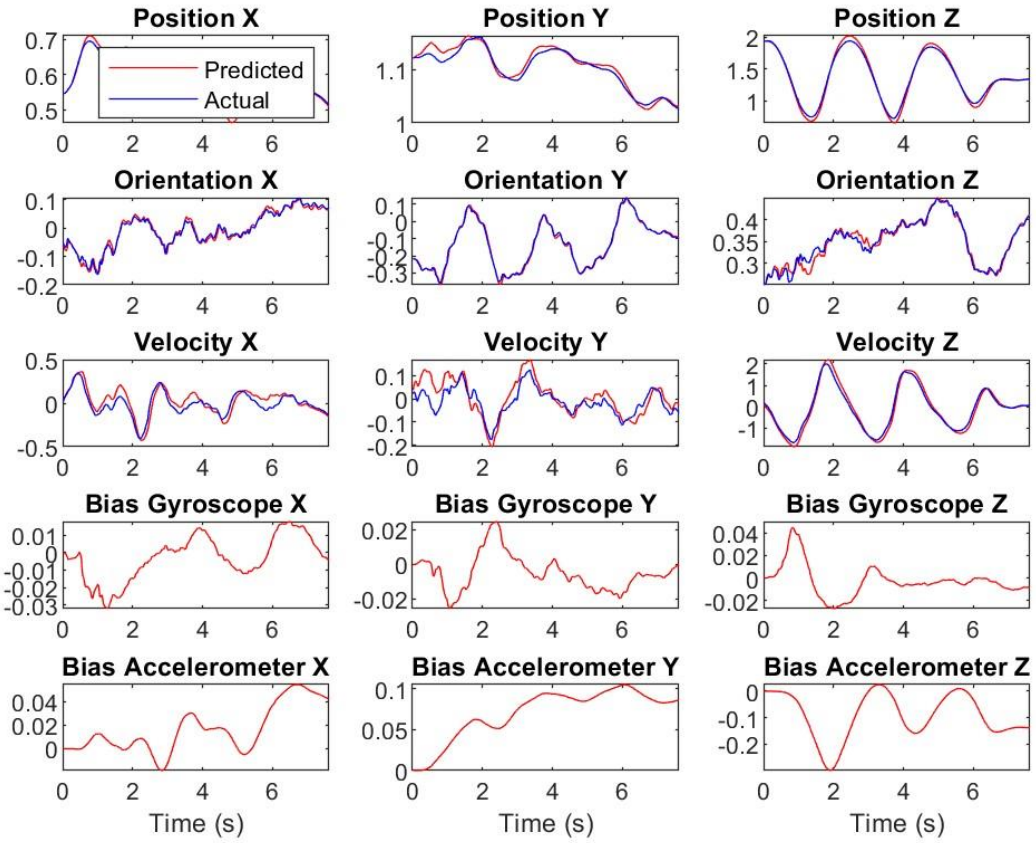
- **Red Line** – Predicted Curve
- **Blue Line** – Actual Curve

Part1-Dataset-1



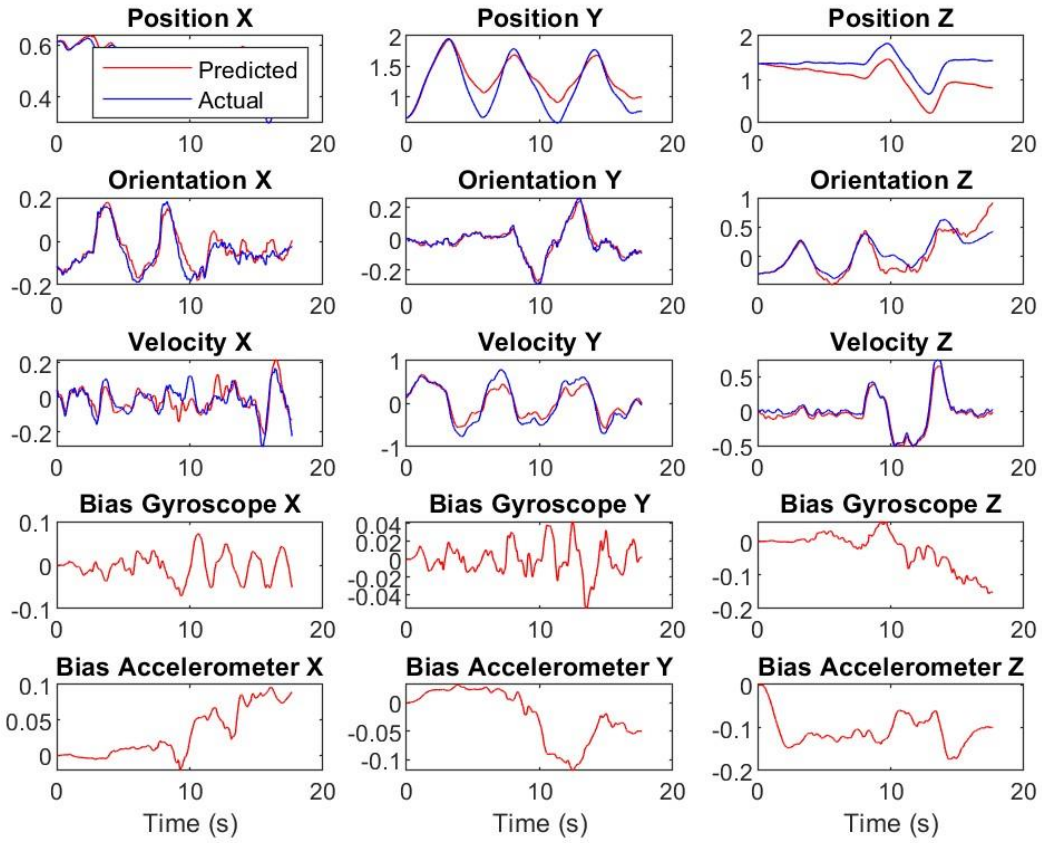
- Predicted Curve and Actual Curve are perfectly coincided almost in Every Parameter.

Part1-Dataset-4



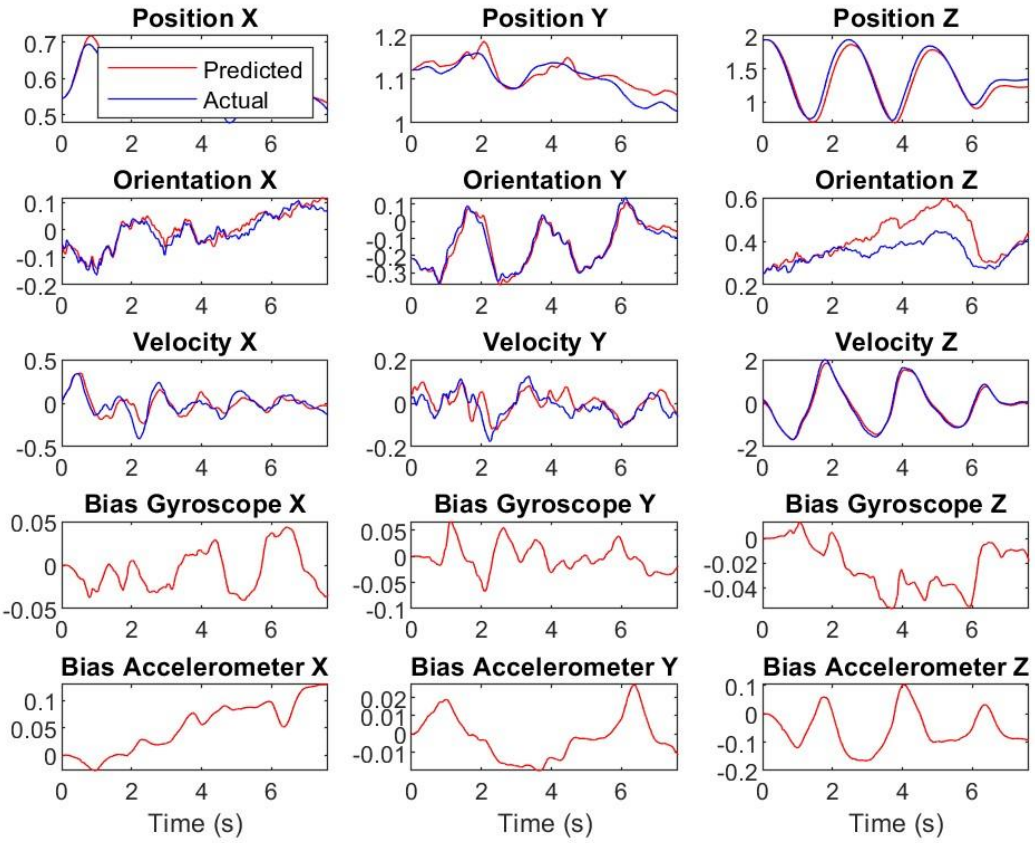
- Predicted Curve and Actual Curve are perfectly coincided almost in Every Parameter.

Part-2-Dataset-1



- Predicted Curve and Actual Curve are coincided almost in Every Parameter but a little deviation could be observed in Position Z and Orientation Z

Part-2-Dataset-4



- Predicted Curve and Actual Curve are coincided almost in Every Parameter but a little deviation could be observed in Orientation Z