UKF for Drone Pose Estimation – REPORT

Rajasundaram Mathiazhagan

Part 1

Prediction:

Prediction in UKF involves computing sigma points for the updated / prior state vector spaced distributed according to the prior mean and covariance. These sigma points are propagated through the system dynamics equations to obtain predictions for these sigma points. Finally, predicted mean and covariance is calculated.

Computing sigma points:

Since the dynamics here have non-additive noise, augmented state vector is considered.

$$x_{aug} = \begin{pmatrix} x \\ q \end{pmatrix} = \begin{pmatrix} position \\ euler\ angles \\ v \\ b_g \\ b_a \\ n_g \\ n_a \\ n_{bg} \\ n_{ba} \end{pmatrix}$$

$$\mu_{aug} = \begin{pmatrix} \mu_{t-1} \\ 0 \end{pmatrix} \qquad \Sigma_{aug} = \begin{pmatrix} \Sigma_{t-1} & 0 \\ 0 & Q \end{pmatrix}$$

The sigma points are computed as follows:

$X^{(0)} = \mu_{aug} \quad X^{(i)} = \mu_{aug} \pm \sqrt{n+\lambda}\left[\sqrt{\Sigma_{aug}}\right]_i \quad i = 0,1,2 \dots n$ where n is the size of x_{aug}

$$\lambda = \alpha^2(n+k) - n, \quad \alpha = 0.001, k = 1$$

$\sqrt{\Sigma_{aug}}$ is computed using the `chol` function in MATLAB.

The computed sigma points are propagated through the system dynamics equations.

$$\dot{x} = \begin{bmatrix} x_3 \\ G^{-1}R\left(\omega_m - b_g - n_g\right) \\ g + R(a_m - b_a - n_a) \\ n_{bg} \\ n_{ba} \end{bmatrix} = f(x, u, n)$$

Discretization:

$$x_t = x_{t-1} + \begin{bmatrix} x_3\, dt \\ dtG^{-1}R\left(\omega_m - b_g - n_{gd}\right) \\ dt(g + R(a_m - b_a - n_{ad})) \\ n_{bgd} \\ n_{bad} \end{bmatrix}$$

where,

$n_{gd} \sim N\left(0, dt\ Q_g\right)$ $n_{ad} \sim N(0, dt\ Q_a)$ $n_{bgd} \sim N\left(0, dt\ Q_{bg}\right)$ $n_{bgd} \sim N\left(0, dt\ Q_{bg}\right)$

x_t is computed for all $2n + 1$ sigma points.

Predicted mean and covariance:

From the computed $X_t^{(i)}$ set, mean and covariance are computed as follows:

$$\bar{\mu}_t = \left(\frac{\lambda}{n + \lambda}\right) X_t^{(0)} + \sum_{i=1}^{2n} \frac{X_t^{(i)}}{2(n + \lambda)}$$

$$\bar{\Sigma}_t = \left(\frac{\lambda}{n + \lambda} + (1 - \alpha^2 + \beta)\right)(X_t^0 - \bar{\mu}_t)(X_t^0 - \bar{\mu}_t)^T + \sum_{i=1}^{2n} \left(\frac{1}{2(n + \lambda)}\right)\left(X_t^i - \bar{\mu}_t\right)\left(X_t^i - \bar{\mu}_t\right)^T$$

Update:

Update in UKF involves computing sigma points as prediction step and then estimating measurement values based on the measurement model and the estimated state values. These estimates are then used along with the actual measurement to compute the updated state values.

Sigma points are computed using the same method followed in prediction step. But here the predicted mean and covariance is used. Since the measurement model here has additive noise, the state vector is not augmented with noise variables.

Measurement Model:

In this part, the drone position and orientation (pose) in the world fame is measured. Since these are the first two state variable sets in the state vector, the measurement model is linear.

$$z = \begin{pmatrix} I_{3X3} & 0 & 0 & 0 & 0 \\ 0 & I_{3X3} & 0 & 0 & 0 \end{pmatrix} X + r \quad r \sim N(0, R)$$

Z is estimated for all the $2n + 1$ sigma points computed.

The Kalman gain matrix and the mean and covariance updates are computed as follows:

$$z_{\mu,t} = \frac{\lambda}{n+\lambda} Z_t^0 + \sum_{i=1}^{2n} \left(\frac{1}{2(n+\lambda)}\right) Z_t^i$$

$$C_t = \sum_{i=0}^{2n} W_i \left(X_t^i - \bar{\mu}_t\right)\left(Z_t^i - z_{\mu,t}\right)^T$$

$$S_t = \sum_{i=0}^{2n} W_i \left(Z_t^i - z_{\mu,t}\right)\left(Z_t^i - z_{\mu,t}\right)^T$$

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

$$K_t = C_t S_t^{-1}$$

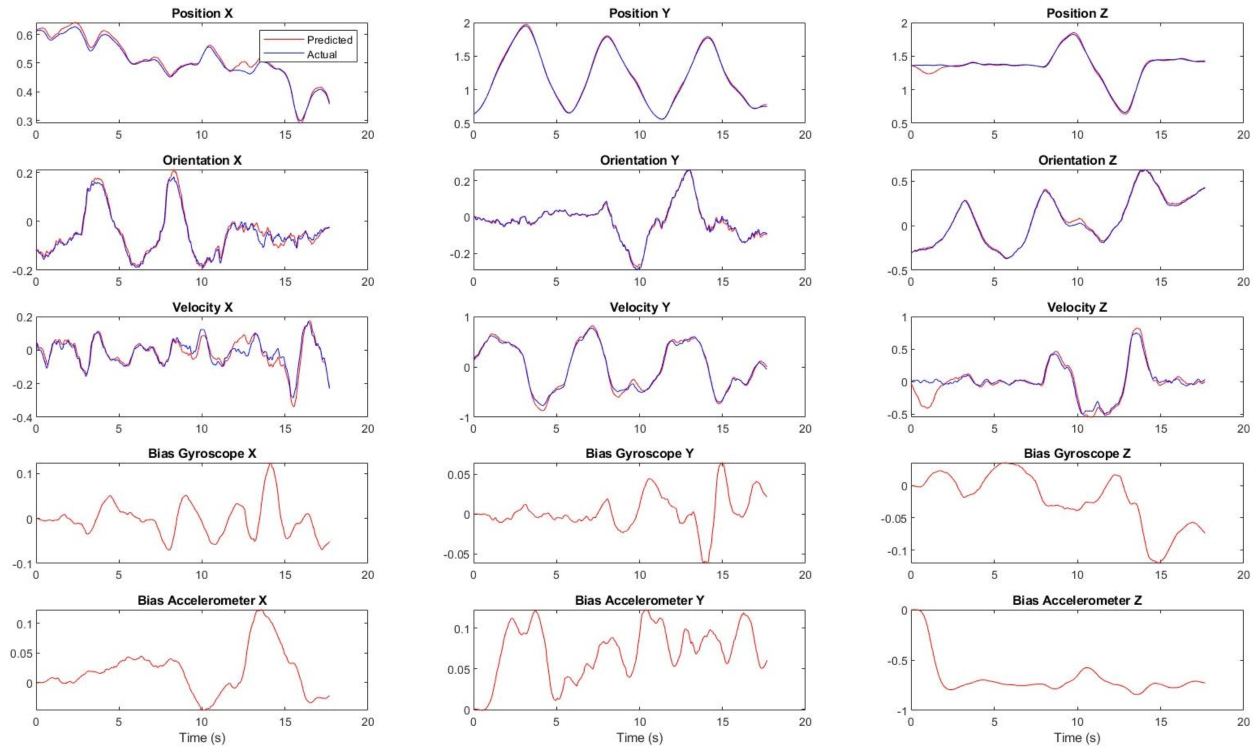
$$\mu_t = \bar{\mu}_t + K_t(z_t - z_{\mu,t})$$

$$\Sigma_t = \Sigma_{t-1} - K_t S_t K_t^T$$

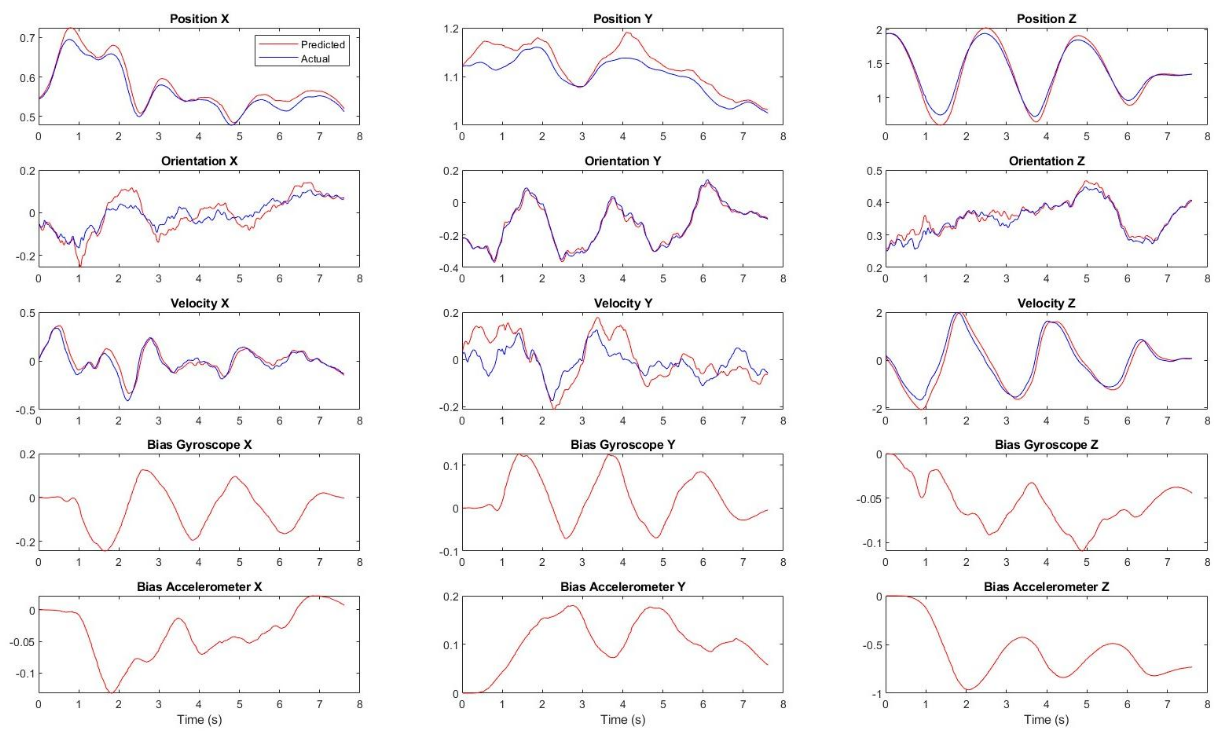
The UKF is finally implemented by repeating this prediction update cycles on the two given datasets and the following results were obtained.

<u>PLOTS</u>

DATASET1:



DATASET 4:



Part 2

In this part, only the velocity measurements from the optical flow are used for update.

Since the system dynamics are the same, the prediction step will remain the same as for part1.

Update:

Since the measured velocity is of the camera frame, as observed in world frame and expressed in the camera frame, the measurement model involves transformations and hence it is non-linear.

$$\begin{bmatrix} {}^{C}\dot{p}_{C}^{W} \\ {}^{C}\omega_{C}^{W} \end{bmatrix} = \begin{bmatrix} R_{B}^{C} & -R_{B}^{C}\, S(r_{BC}^{B}) \\ 0 & R_{B}^{C} \end{bmatrix} \begin{bmatrix} {}^{B}\dot{p}_{B}^{W} \\ {}^{B}\omega_{B}^{W} \end{bmatrix}$$

From the above relation, linear velocity measurement in camera frame is expressed as:

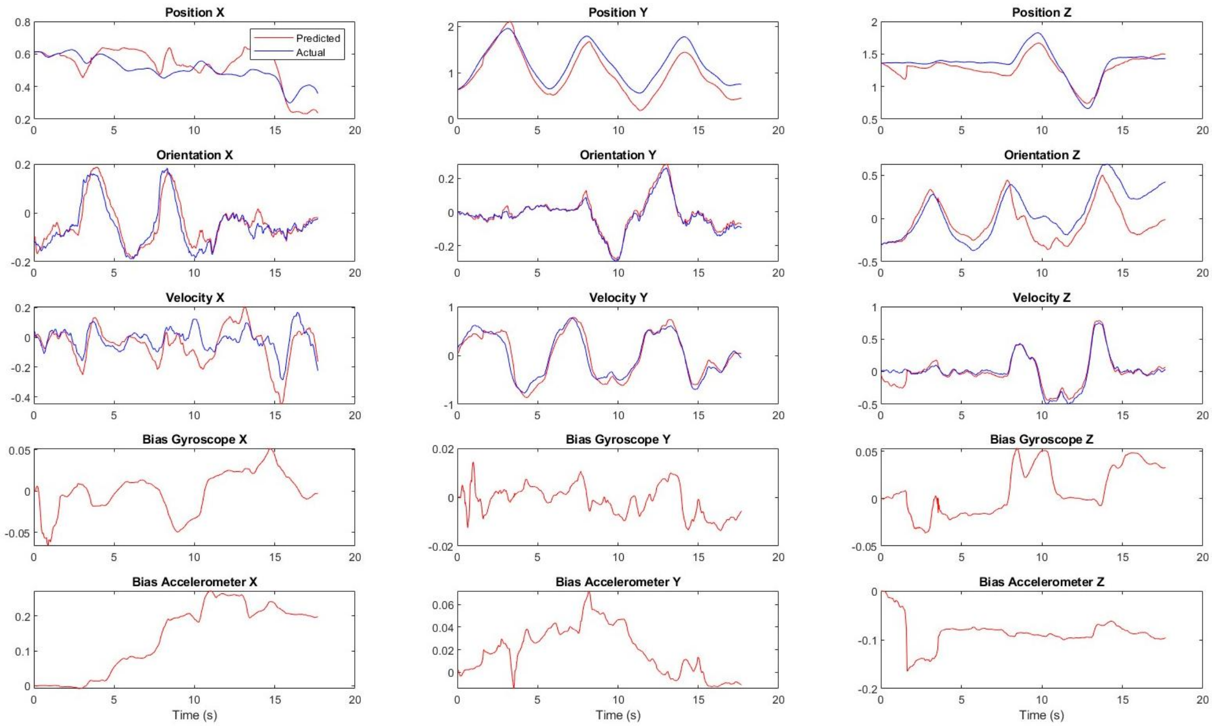
$${}^{C}\dot{p}_{C}^{W} = R_{B}^{C}\, {}^{B}\dot{p}_{B}^{W} - R_{B}^{C}\, S(r_{BC}^{B}) R_{C}^{B} \quad {}^{C}\omega_{C}^{W} + v_t$$

${}^{B}\dot{p}_{B}^{W} = R_{W}^{B} X_3$ where X_3 is the predicted velocity state variable.

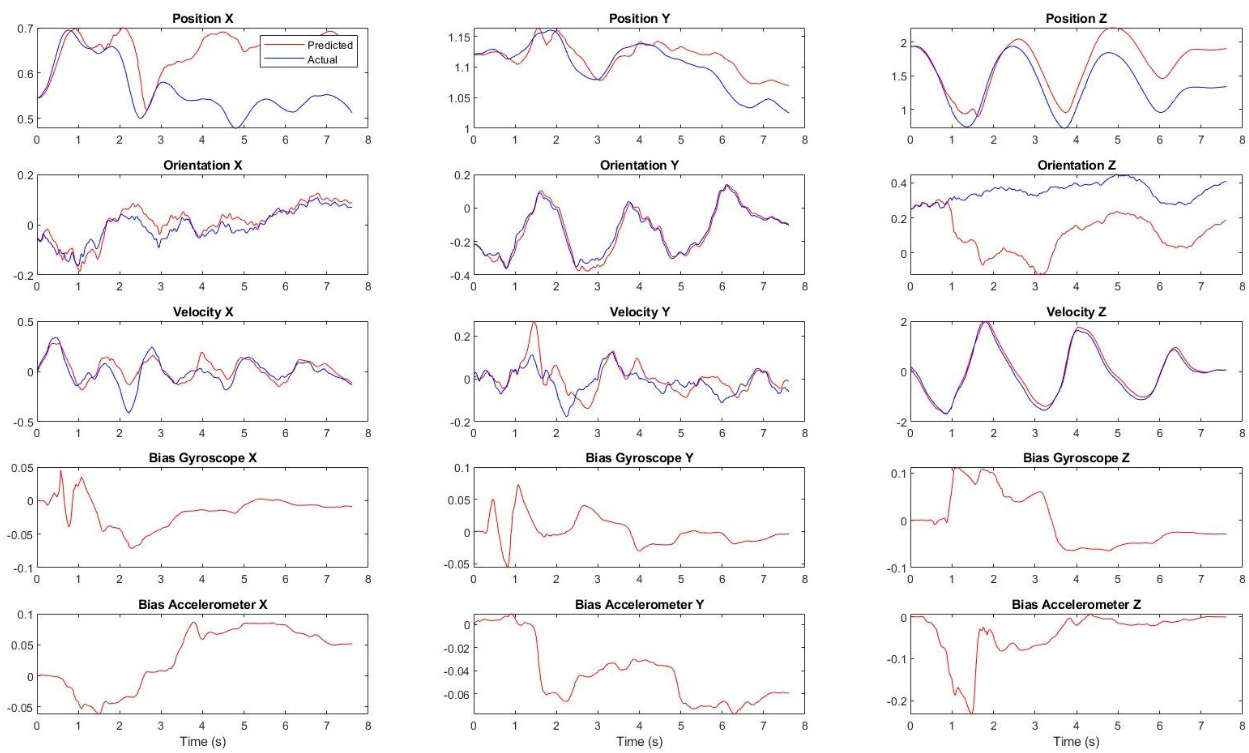
Using this model, measurement estimates are made for all the sigma points at each time step. From these estimates and actual measurement, the Kalman gain, and the mean and covariance updates are calculated following the same procedure as part1.

PLOTS

DATASET 1



DATASET 4



CONCLUSIONS

The unscented Kalman Filter method is known to better capture the non-linearity of the system compared to the Extended Kalman Filter method. From the plots for part1, we can observe that the filter output values very closely follow the actual values.

Since only velocity was measured in part2, we can observe that the velocity values closely follow the actual values. The position estimates also follow the true values in most cases, following the trend with some offset in a few cases.

Overall, the filter is observed to produce good estimates from measurements for the non-linear system.