Attitude Estimation Using Unscented Kalman Filter

1st Venkateshkrishna
Masters in Robotics
Worcester Polytechnic Institute
Worcester, MA 01609
vparsuram@wpi.edu

2nd Athithya, Lalith
Masters in Robotics
Worcester Polytechnic Institute
Worcester, MA 01609
lnavaneethakrishnan@wpi.edu

3rd Gampa, Varun
Masters in Robotics
Worcester Polytechnic Institute
Worcester, MA 01609
vgampa@wpi.edu

***Abstract*—This project focuses on the practical implementation of an Unscented Kalman Filter (UKF) to track three-dimensional orientation. The main objective is to utilize IMU sensor data obtained from gyroscopes and accelerometers to accurately estimate the complex 3D orientation. A crucial aspect of this project involves verifying the accuracy of our orientation estimations by comparing them with ground truth data collected through a Vicon motion capture system. This report presents the methodologies, findings, and insights gained from this important endeavor, providing valuable insights into the effectiveness and potential applications of the Unscented Kalman Filter in the field of orientation tracking.**

I. Introduction

The aim of this project is to implement a Unscented Kalman filter to estimate the orientation using gyroscope and accelerometer data obtained from a 6-DOF IMU. The esimated orientation is compared with the ground truth orientation obtained from a vicon motion capture system. Orientations from individual sensors are also compared with the ground truth orientation.

II. Removing Bias From Gyroscope Data And Scaling

The raw values are obtained from the gyroscope are converted to the physical units (rads^{-1}) using

$$\tilde{\omega} = \frac{3300}{1023} \times \frac{\pi}{180} \times 0.3 \times (\omega - b_g) \tag{1}$$

where $\tilde{\omega}$ represents the the value of ω in physical units and b_g is the bias of the gyroscope. b_g is calculated as the average of the first 300 samples.

III. Removing Bias From Accelerometer Data And Scaling

The raw values obtained from the accelerometer are converted to the physical units (ms^{-2}) using

$$\tilde{a}_x = (a_x \times s_x + b_{a,x}) \times g \tag{2}$$

here a_x is the measured quantity from accelerometer, and g

IV. Attitude estimation using Unscented Kalman Filter

A. Filter Explanation

The state is defined as

$$x = \begin{bmatrix} q0 \\ q1 \\ q2 \\ q3 \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{3}$$

Where $[q_0, q_1, q_2, q_3]^T$ represents a unit quaternion with $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$, hence representing only 3 degrees of freedom, not 4.

The filter begins with an initialization process, it involves setting up process noise matrix Q, measurement noise matrix R, and covariance matrix P. The disturbances W used to generate sigma points are computed from covariance and process noise matrices, employing the matrix square root, which is derived using Cholesky Decomposition.

In attitude estimation, the state vector comprises 7 states: [qw qx qy qz wx wy wz]T, which includes attitude quaternion and Euler rotation rates. With 6 degrees of freedom in the states, a dimensionality of n=6 is used. The filter initializes P as a identity matrix of size n x n, as the covariance quickly converges to the correct value. Additionally, sigma points are calculated using $\sqrt{\lambda + n}$, as described to enhance tracking. Also the mean of the distribution is also considered as a sigma point.

The square root matrix is found using,

$$S = \sqrt{Pk_{-1} + Q}$$

Here Q is the process model covariance. The disturbance (noise) of the sigma points is calculated using,

$$W_0 = 0$$

$$W_{i,i+n} = \text{columns}(\pm\sqrt{(}n + \lambda)S)$$

Here, λ is chosen as $n - 3$ it was more numerically stable and converged faster. Here, n is the number of independent variables in the state (6 in this case). The Sigma Points are computed using,

$$Xi = \hat{x}_{k-1} + Wi$$

$$Xi = \begin{bmatrix} q_{k-1}q_W \\ \tilde{\omega}_k + \tilde{\omega}_W \end{bmatrix}$$

where $\tilde{\omega}_W$ is the omega part of W, and q_W is the quaternion part of W. This results in 13 sigma points, each available in both negative and positive cases as well as the mean. These sigma points are calculated by adding disturbances to the current state. These sigma points are weighted as:

$$wt_0 = \frac{\lambda}{\lambda + 2n}$$

$$wt_{i!=0} = \frac{1}{\lambda + 2n}$$

After calculating the sigma points, the process model points are determined. Each sigma point is processed through the system's model to obtain individual state estimates Y_i. In this system, the process model advances the attitude quaternion using the current rotation rates. The quaternion change $q\Delta$ in the process model is computed in a manner similar to how the disturbance for sigma points was determined, but it now takes into account the current angular velocity ω_{t-1} instead of the quaternion disturbance $W_{1:3,i}$.

The mean of the sigma points is calculated iteratively by determining a mean attitude error until a mean attitude is reached. During each iteration, the error ei from each sigma point to the quaternion qt is computed. The average of these error vectors yields an average error e, which is transformed back into a quaternion and applied to qt to adjust it toward the true mean attitude. This iteration process continues until the mean error falls below an acceptable threshold.

The innovation covariance can be calculated by adding the measurement covariance and the measurement noise matrix. To compute the Kalman gain required for updating the state, the cross-correlation matrix Pxz must be computed.

With the Kalman gain determined, the state estimate and state covariance can be updated. The new covariance is calculated by subtracting the product of the Kalman gain, innovation covariance matrix, and Kalman gain transpose from the estimated covariance. The state is updated as the previous state plus the product of the Kalman gain and the difference between the measurement readings zˆ and the estimated measurement readings.

For the dataset used, the process and measurement noise matrix were initialized as follows:

In summary, this filtering process combines elements of sigma point calculations, process and measurement modeling, and iterative adjustments to estimate states and covariances while accounting for non-linearities in the system.

Process Model:

The process model assumes that the angular velocity remains constant during the time interval Δt.

$$\omega_k = \omega_{k-1}$$

$$q^{\Delta} = \left(\cos\left(\frac{|\tilde{\omega}_{k-1}|\Delta t}{2}\right), \frac{\tilde{\omega}_{k-1}}{|\tilde{\omega}_{k-1}|}\sin\left(\frac{|\tilde{\omega}_{k-1}|\Delta t}{2}\right)\right)$$

The updated sigma points are computed using,

$$Y_i = A(X_i, 0) = \begin{pmatrix} q_{k-1}q_W q^{\Delta} \\ \tilde{\omega}_{k-1} + \tilde{\omega}_W \end{pmatrix}$$

Calculate Mean of Sigma Points:

Now, use Intrinsic Gradient Descent to find the mean quaternion. **Data:** Y

Result: $\hat{\bar{x}}_k$

Initialize $\bar{q}$ as X_1 while $t <$ MaxIter or $|e| \leq$ Thld do

$$\text{for all } i, \quad \tilde{e}_i = q_i \bar{q}_t^{-1}$$

First $\tilde{e}_i$ is converted as rotation vector. The compute mean using,

$$\tilde{e} = \sum_{i=1}^{2n+1} wt_i * \tilde{e}_i$$

where $\tilde{e}$ and $\tilde{e}_i$ are rotation vectors. Then $\tilde{e}$ is again converted to quaternion.

$$\bar{q}_{t+1} = e\bar{q}_t$$

$$\bar{\omega} = \frac{1}{2n}\sum_{i=1}^{2n} \omega_i$$

Algorithm 1: Intrinsic Gradient Descent

Update Model Covariance:

$$\bar{P}_k = \sum_{i=1}^{2n+1} wt_i * \bar{W}_i * \bar{W}_i^T$$

Utilizing the mean state $\bar{\mu}$ derived from the sigma points, we can compute the covariance estimate $\bar{P}$ along with the sigma disturbances W_{0i} centered around the mean.

$$\bar{W}_i = \begin{pmatrix} q_i \bar{q}^{-1} \\ \tilde{\omega}_i - \bar{\tilde{\omega}} \end{pmatrix}$$

Now, compute the measurement-updated transformed sigma points,

$$Z_i = \begin{pmatrix} q_i g q_i^{-1} \\ \tilde{\omega}_k \end{pmatrix}$$

where g is the gravity vector. Now, compute $\bar{z}_k = \text{mean} Z_i$.

Compute measurement model covariances:

The covariance corresponding to the measurement model update is computed as,

$$P_{zz} = \sum_{i=1}^{2n+1} wt_i * \phi_i * \phi_i^T$$

here,

$$\phi_i = \begin{pmatrix} q_i \bar{q}^{-1} \\ \tilde{\omega}_i - \bar{\tilde{\omega}} \end{pmatrix}$$

The quaternions and omegas above correspond to the ones in Z.

The innovation term is given by:

$$\nu_k = z_k - \bar{z}_k$$

Here, z_k is the observation, i.e., stacked accelerometer and gyroscope readings.

The innovation covariance is calculated using,

$$P_{\nu\nu} = P_{zz} + R$$

Here, R is the measurement model covariance.

The cross-covariance is calculated as,

$$P_{xz} = \sum_{i=1}^{2n+1} wt_i * \bar{W}_i \phi_i^T$$

Update Kalman Gain, State Covariance, and State:

The Kalman gain is calculated as,

$$K_k = \bar{P}_{xz} P_{\nu\nu}^{-1}$$

Update state as,

$$\hat{x}_k = \bar{\bar{x}}_k + K_k \nu_k$$

Update State Covariance as,

$$P_k = \bar{P}_k - K_k P_{\nu\nu} K_k^T$$

V. Results

The plots for the 6 training data sets are shown in Figs. 1, 2, 3, 4, 5, 6. Each plot has three subplots corresponding to roll, pitch, and yaw respectively. There are 5 legends in each subplot for the orientation computed from gyroscope, accelerometer, complementary filter, vicon data, Madgwick filter and Unscented Kalman Filter.

Furthermore, **Test** data was released 24hrs before the submission deadline of this project. The same code was tested on it as well and the following results are seen:

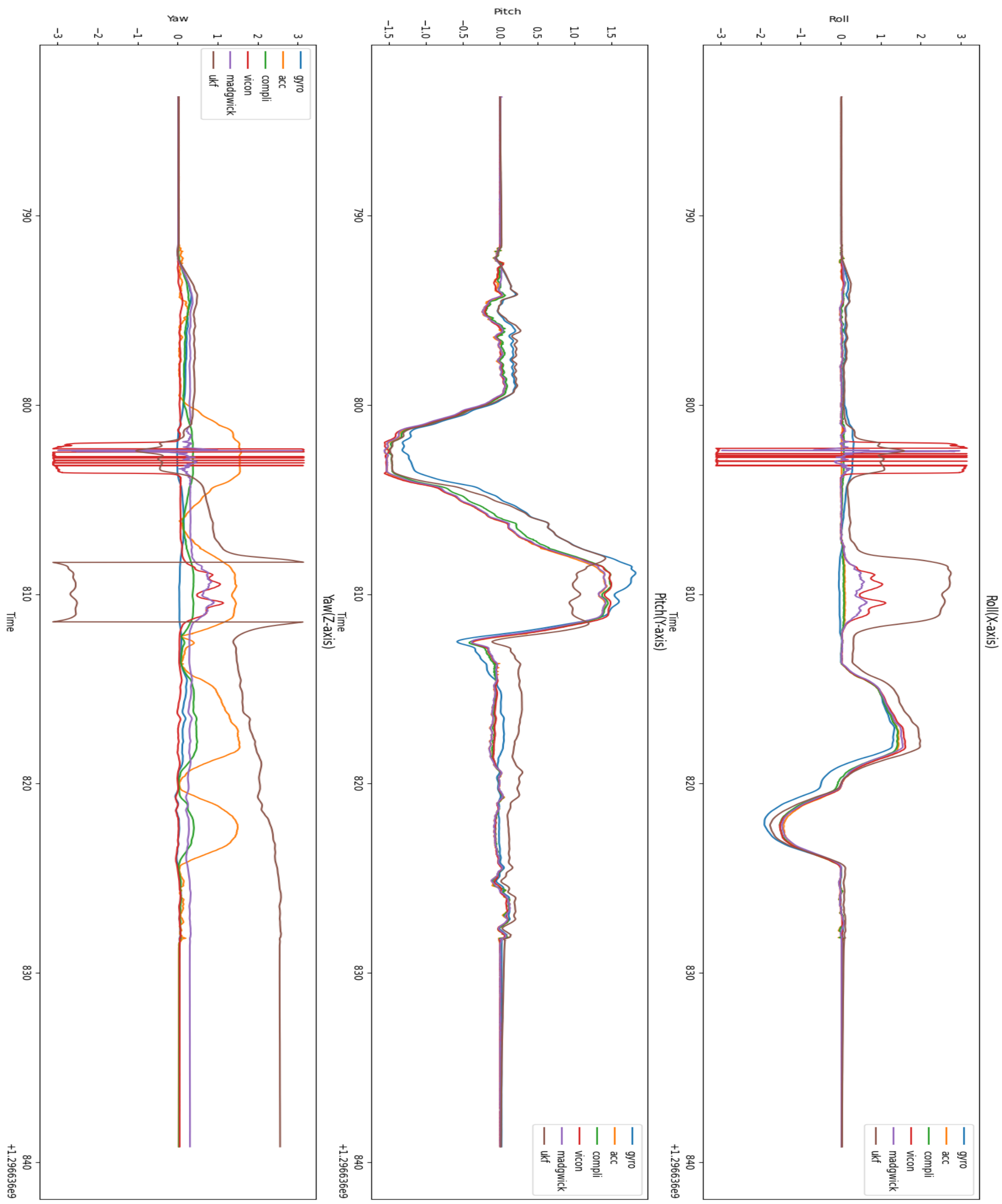

Fig. 1. Roll Pitch Yaw angles for data set 1

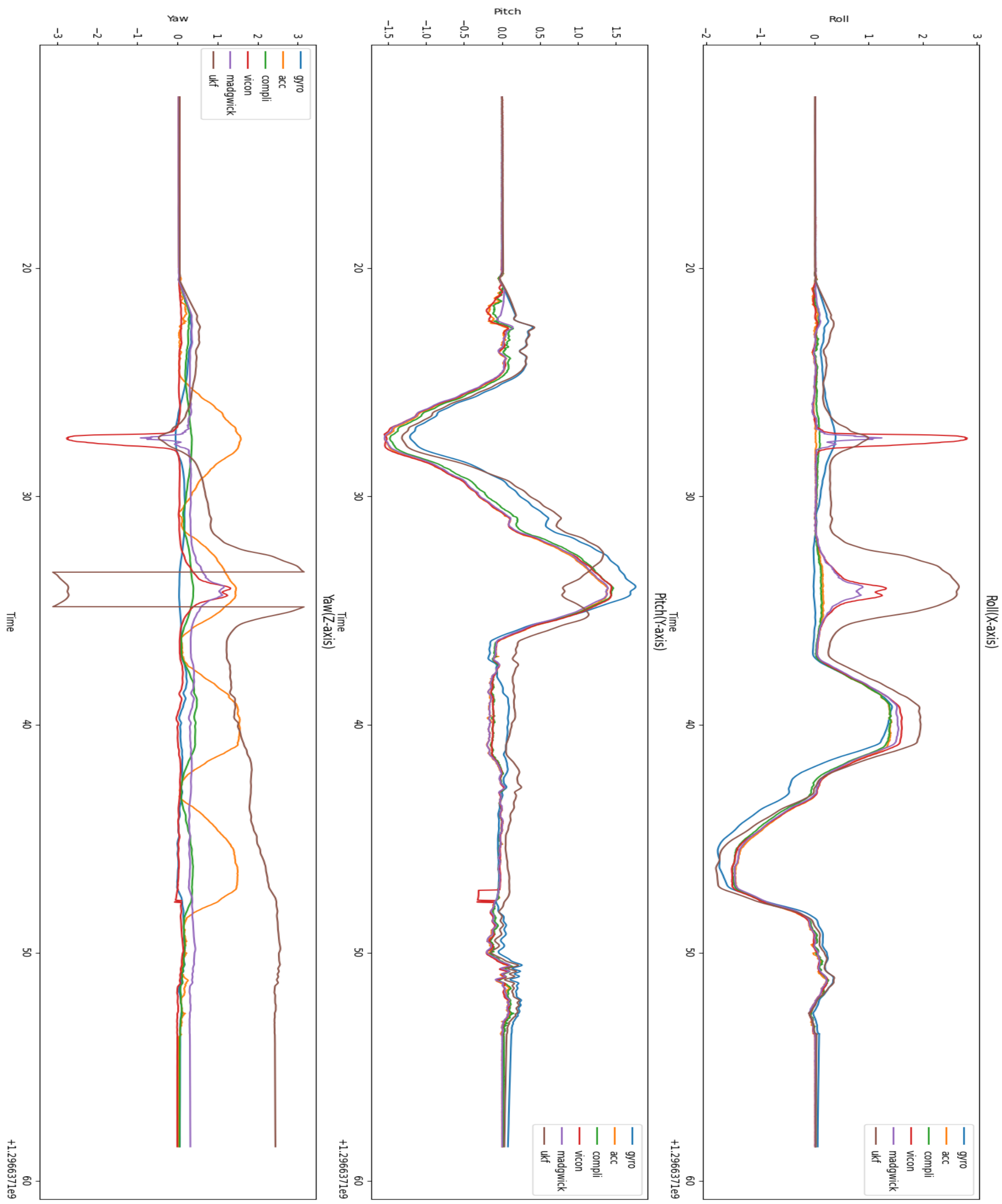

Fig. 2. Roll Pitch Yaw angles for data set 2

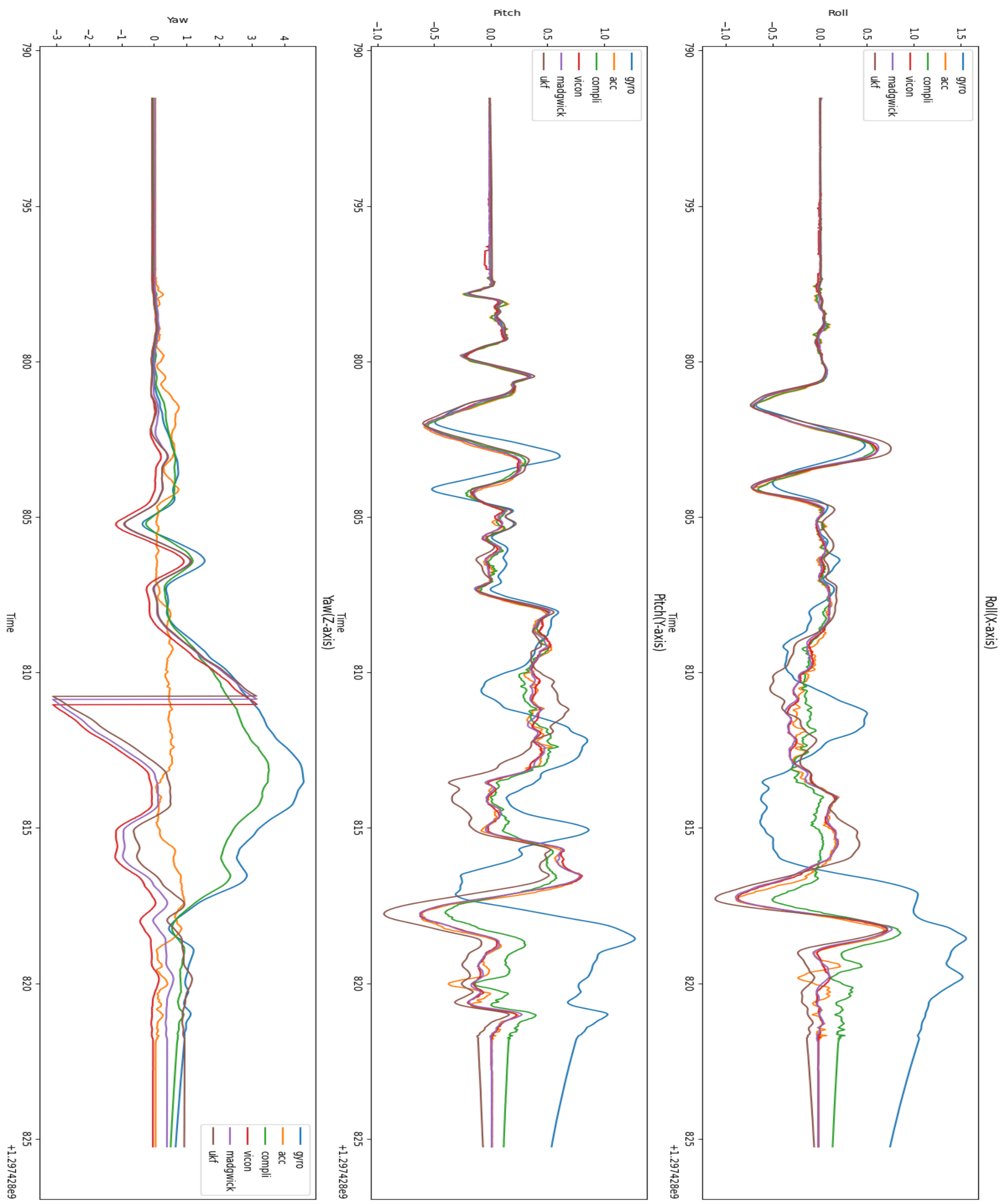

Fig. 3. Roll Pitch Yaw angles for data set 3

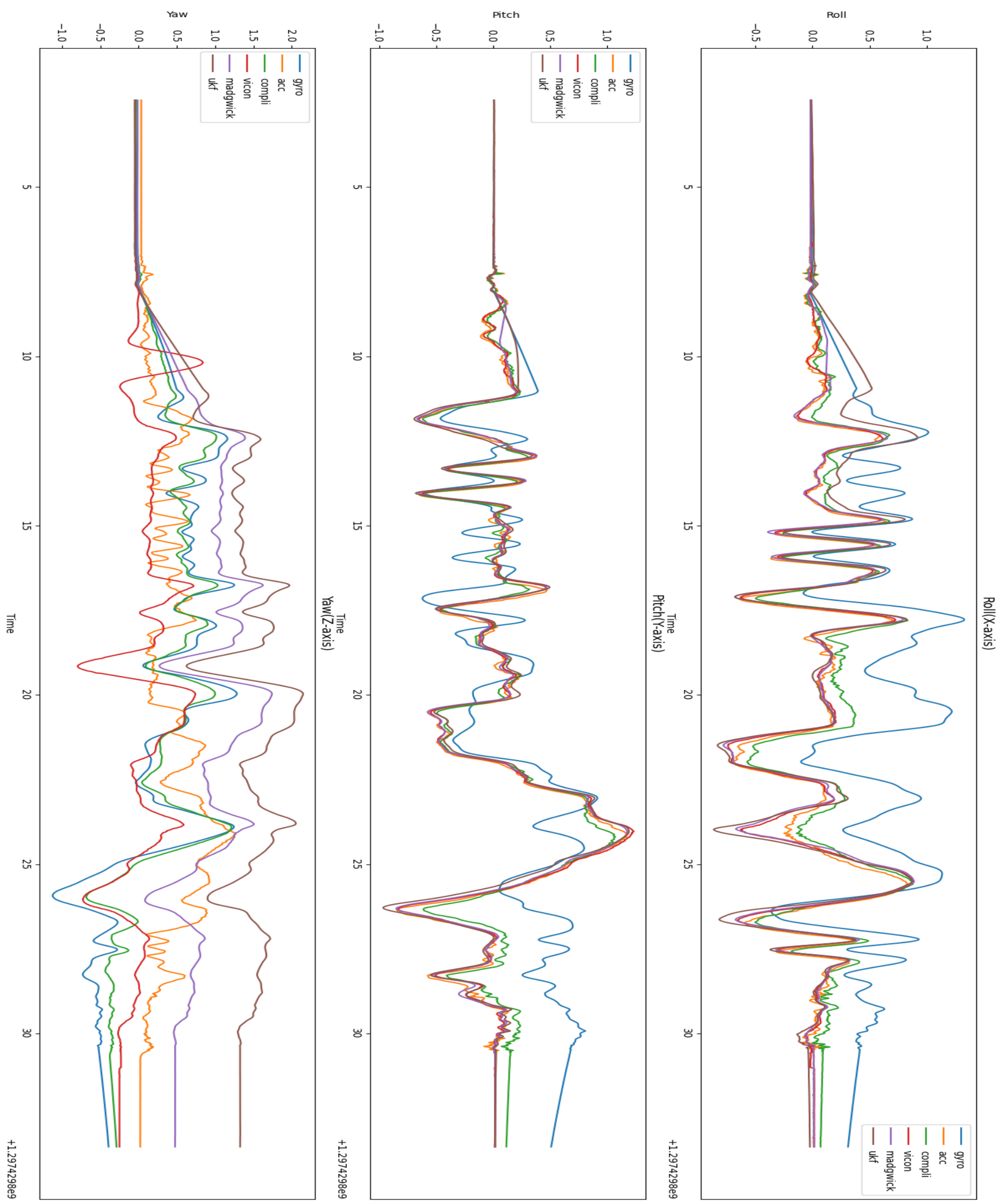

Fig. 4. Roll Pitch Yaw angles for data set 4

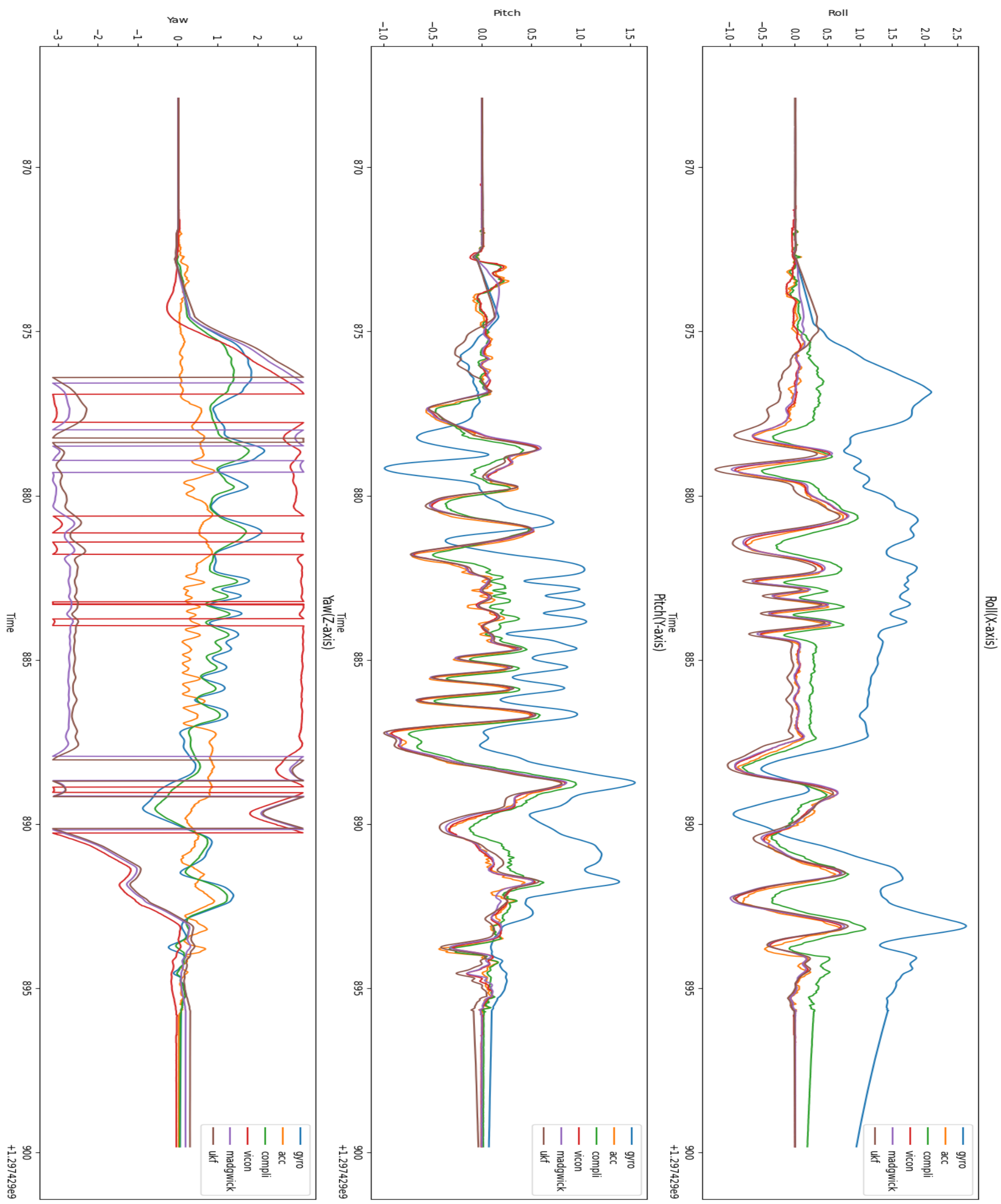

Fig. 5. Roll Pitch Yaw angles for data set 5

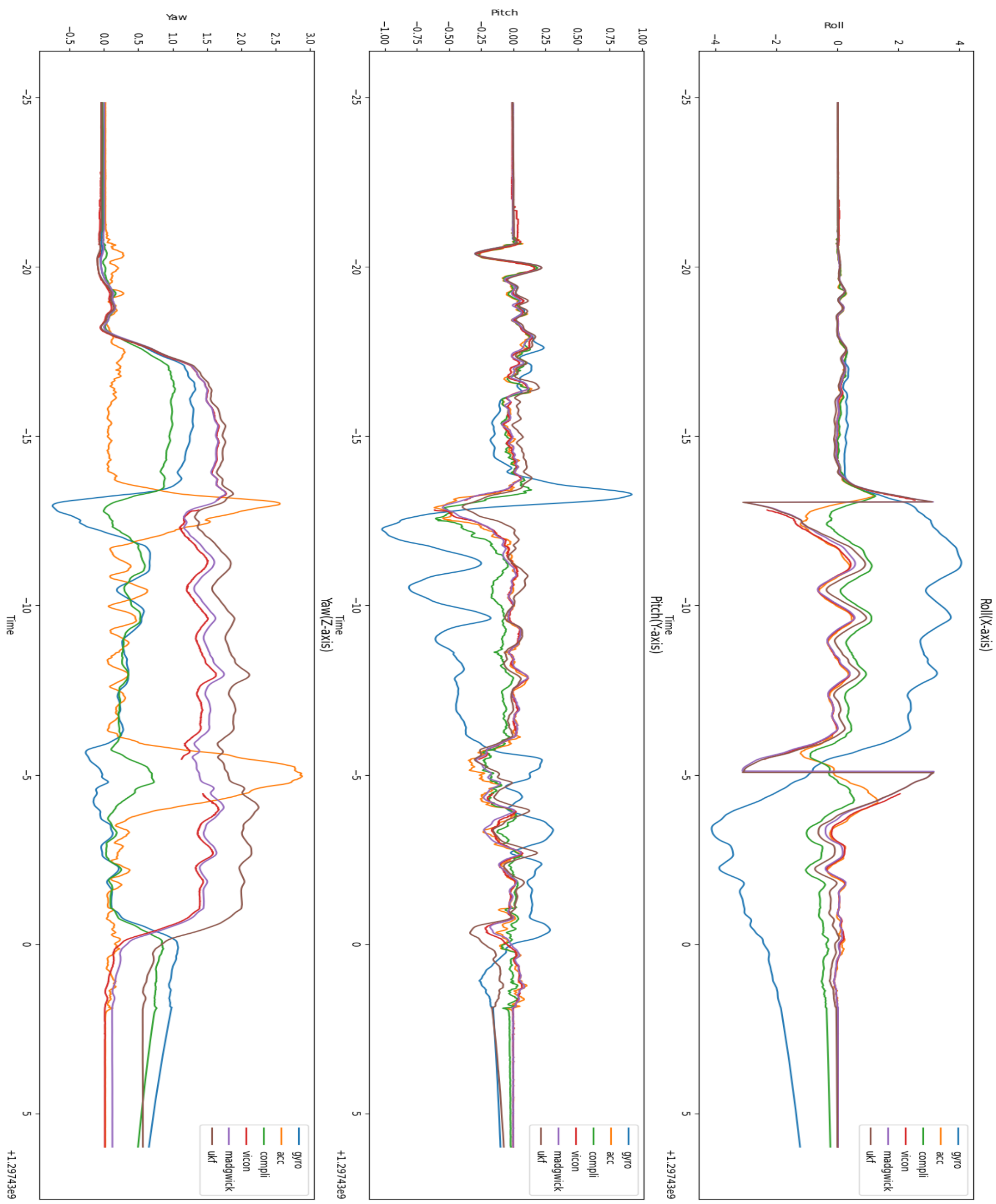

Fig. 6. Roll Pitch Yaw angles for data set 6

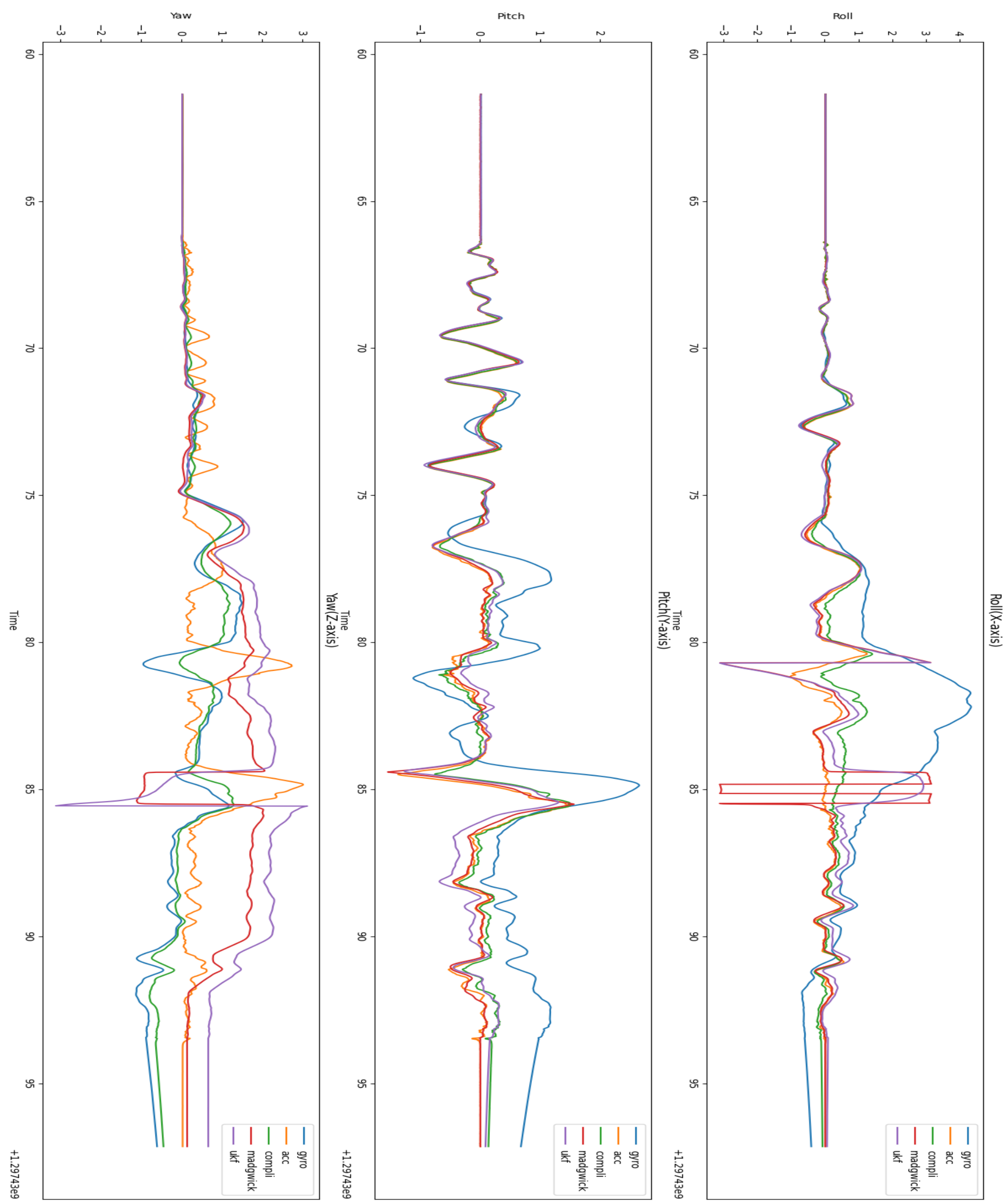

Fig. 7. Roll Pitch Yaw angles for test set 1

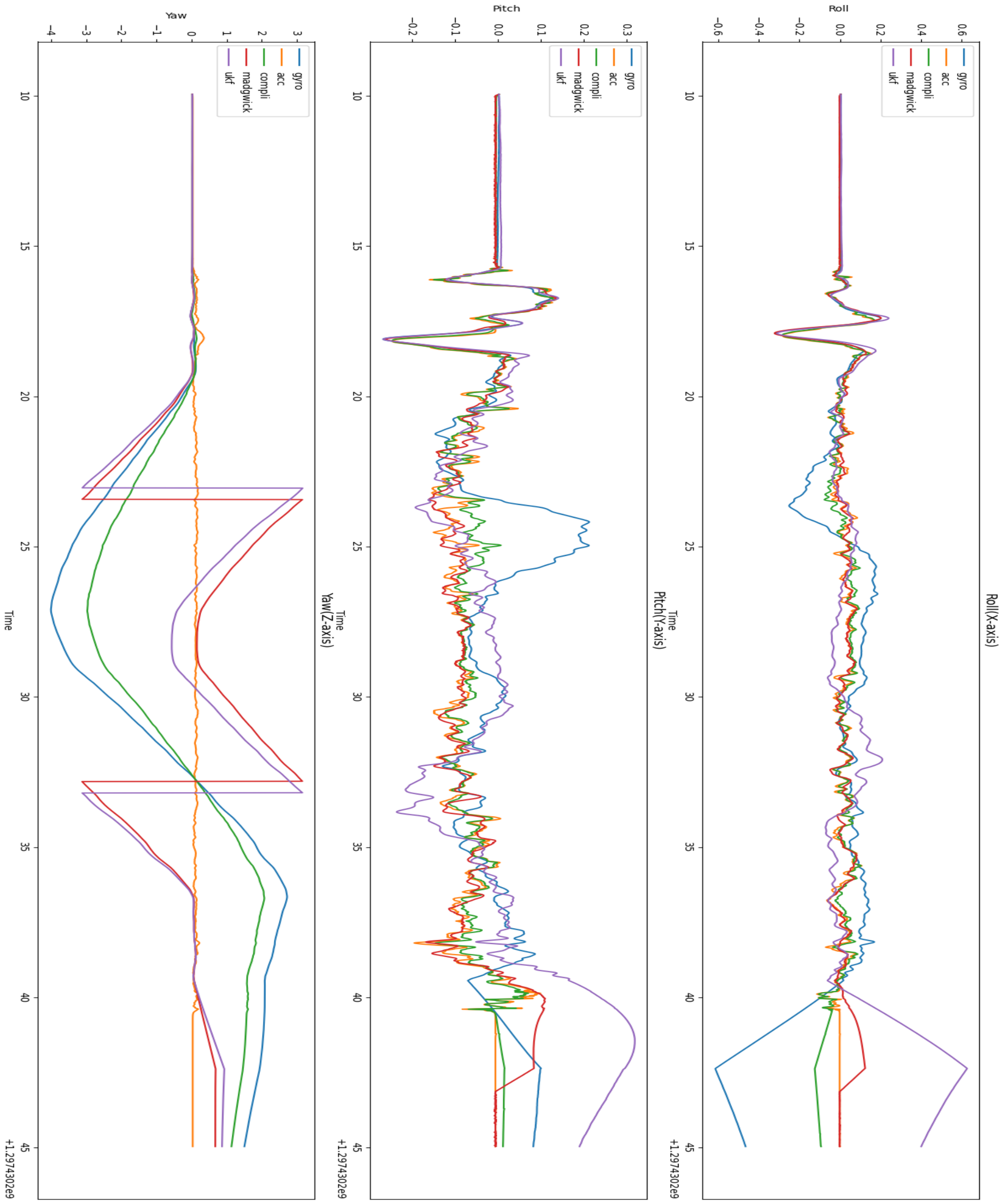

Fig. 8. Roll Pitch Yaw angles for test set 2

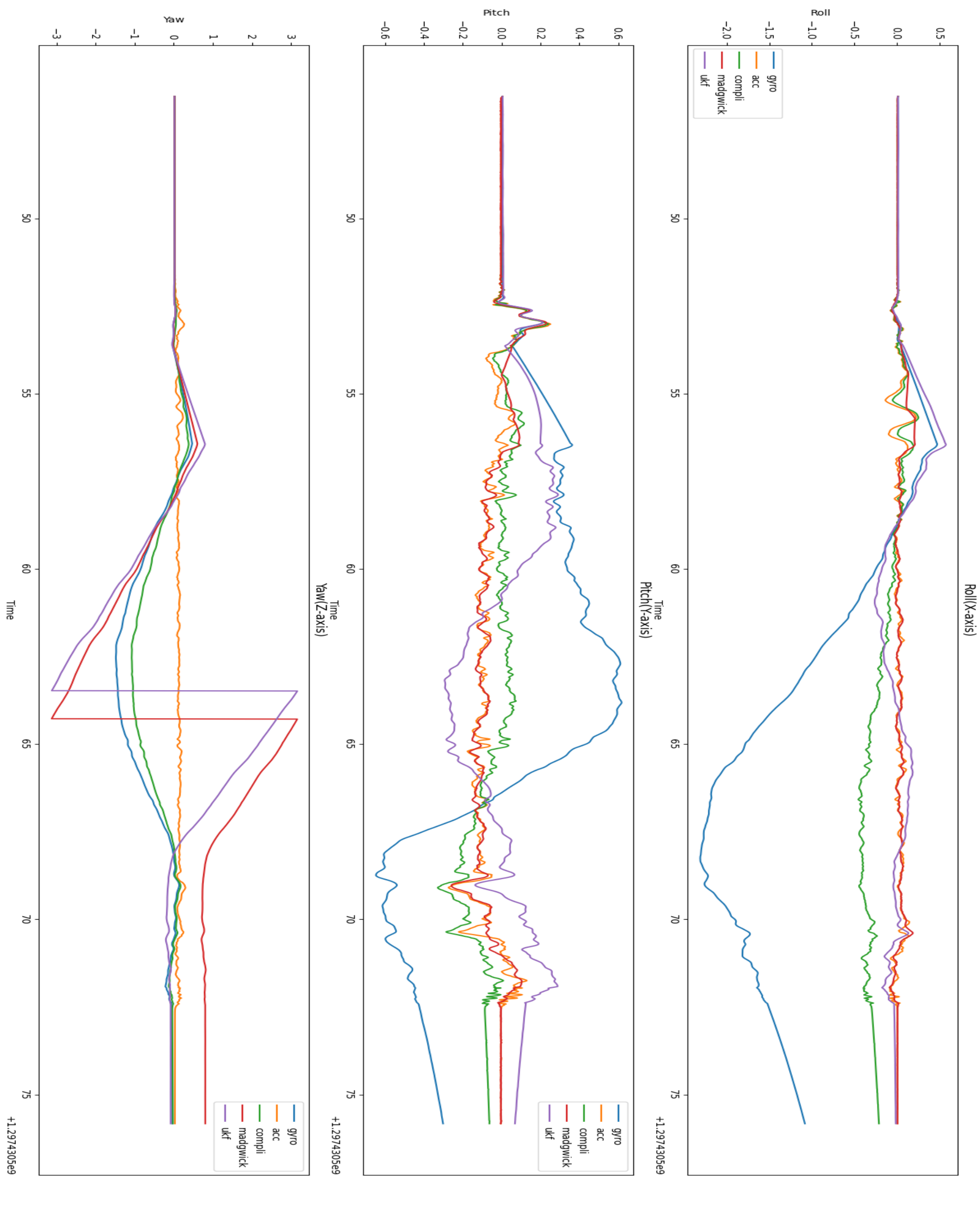

Fig. 9. Roll Pitch Yaw angles for test set 3

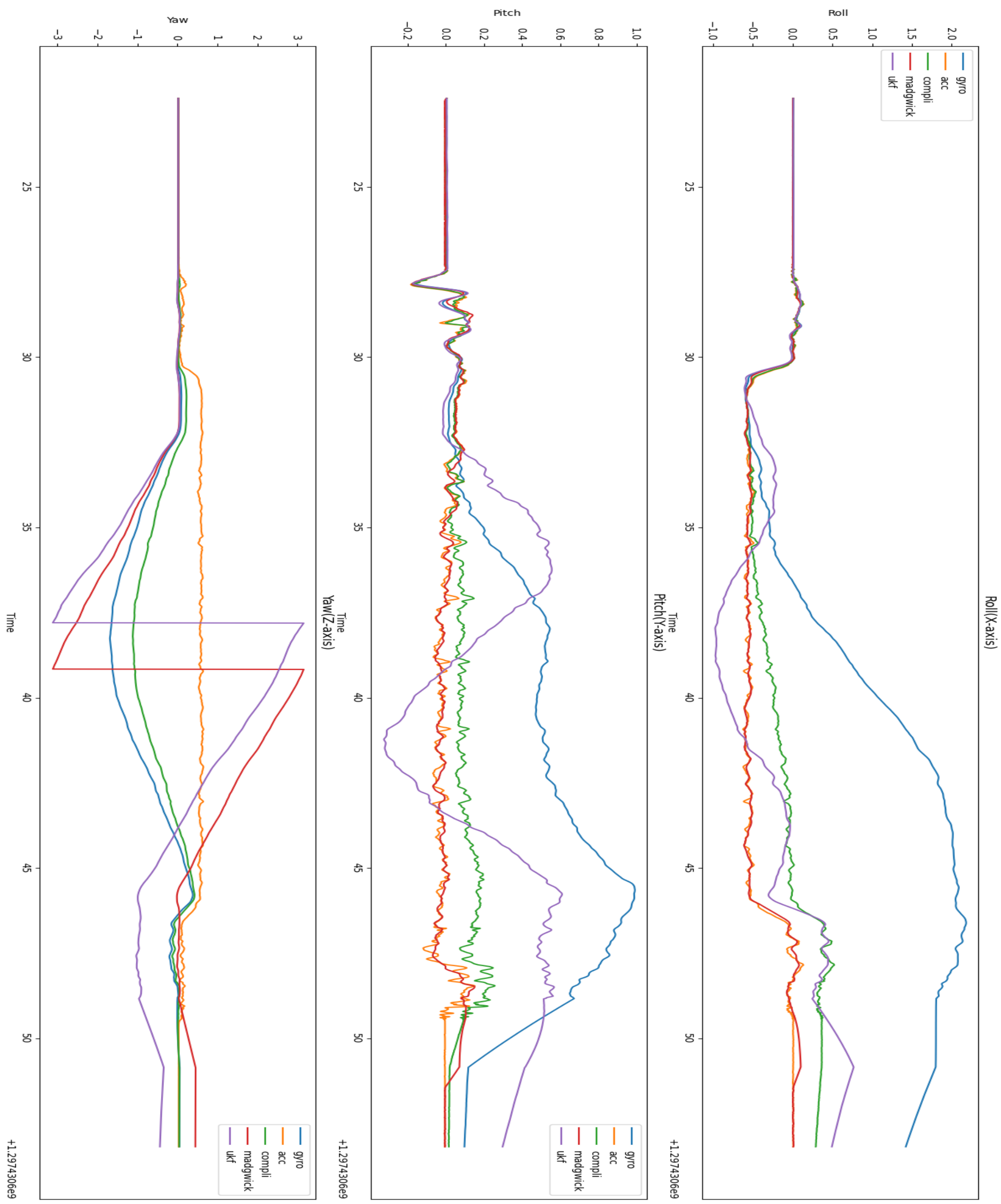

Fig. 10. Roll Pitch Yaw angles for test set 4

VI. Videos

The videos comparing the orientation estimated using the different methods can be found at videos

VII. Conclusion

In this project we implemented a Unscented Kalman filter, Madgwick filter and a complimentary filter to estimate the orientation of a 6-DOF IMU. The orientation estimated using the filters is compared with the ground truth orientation obtained from a vicon motion capture system. The results show that the Uscented Kalman filter is better at estimating the orientation than the independent sensors and the complimentary filter and Madgwick filter especially when there is jerky motions.

References

[1] Complementary filter: link
[2] Madgwick filter: link
[3] Complementary filter: link
[4] Unscented Kalman Filter for Non Linear Estimation: link