P1 b: UKF for Attitude Estimation

Mayank Bansal
Robotics Engineering
Worcester Polytechnic Institute
Email: mbansal1@wpi.edu

Siyuan 'Oliver' Huang
Robotics Engineering
Worcester Polytechnic Institute
Email: shuang4@wpi.edu

Miheer Diwan
Robotics Engineering
Worcester Polytechnic Institute
Email: msdiwan@wpi.edu

***Abstract*—The aim of this project was to estimate the 3D orientation (attitude) of an IMU using four different methods — using an ideal gyroscope, an ideal accelerometer, Complementary Filter, Madgwick Filter and Unscented Kalman Filter.**

I. Phase 1

In Phase 1 of this project, we performed IMU attitude estimation using four different methods — using a gyroscope, an accelerometer, a Complementary filter, and a Madgwick filter. The four methods were tested on six different training datasets and four test sets. The results were plotted for analysis against the ground truth.

A. Data Processing

The data was collected with an ArduIMU+ V2, a six degree of freedom Inertial Measurement Unit and the ground truth data was collected using a Vicon motion capture system. Each data file consists of six values and their corresponding timestamps — three linear acceleration values and three angular velocity values in the x, y and z direction, in the form $\begin{bmatrix} a_x & a_y & a_z & \omega_z & \omega_x & \omega_y \end{bmatrix}^T$.

The accelerometer bias and scale parameters were also provided in a separate file in the form of a 2×3 vector where the first row denotes the scale values — $\begin{bmatrix} s_x & s_y & s_z \end{bmatrix}$ and the second row denotes the bias values — $\begin{bmatrix} b_{a,x} & b_{a,y} & b_{a,z} \end{bmatrix}$.

The Vicon ground truth data is extracted in a similar manner. It contains the rotation matrices estimated from ZYX Euler angles in the form of $3 \times 3 \times N$ matrices and their corresponding timestamps.

1) Data Conversion: The IMU data is not in physical units. Therefore, we need to convert it before attitude estimation. Converting acceleration values to ms^{-2}:

$$\tilde{a_x} = \frac{a_x + b_{a,x}}{s_x} \tag{1}$$

where, $\tilde{a_x}$ represents the value of a_x in physical units, ba_x is the bias and s_x is the scale factor of the accelerometer. Converting angular velocities to rad/s:

$$\tilde{\omega} = \frac{3300}{1023} \times \frac{\pi}{180} \times 0.3 \times (\omega - b_g) \tag{2}$$

Here, $\tilde{\omega}$ represents the value of ω in physical units and b_g is the bias. b_g was calculated as the average of the first hundred samples (assuming that the IMU is at rest in the beginning).

2) Time Stamp Alignment Using Slerp: Since the timestamps of the IMU data and the Vicon ground truth data are not aligned, I aligned them using Slerp.

II. Attitude Estimation Using an Ideal Gyroscope

The gyroscope mathematical model is given by:

$$\omega = \hat{\omega} + \mathbf{b}_g + \mathbf{n}_g \tag{3}$$

Here, ω is the measured angular velocity from the gyroscope, $\hat{\omega}$ is the latent ideal angular velocity we wish to recover, $\mathbf{b}_g$ is the gyroscope bias which changes with time and other factors like temperature, and $\mathbf{n}_g$ is the white Gaussian gyroscope noise.We estimate the orientation using only gyroscope data by integration. Since, we cannot perform integration without knowing the initial values, we assume that the initial orientation from Vicon is known. The initial values can also be estimated from other sensors or by starting from rest. We have the angular velocities at timestamp t and we want to estimate the orientations at timestamp $t+1$. The eqn. can be given by:

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}_{t+1} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}_t + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}_t \times \delta t \tag{4}$$

The computed output for imuRaw1.mat is shown in Fig. 1

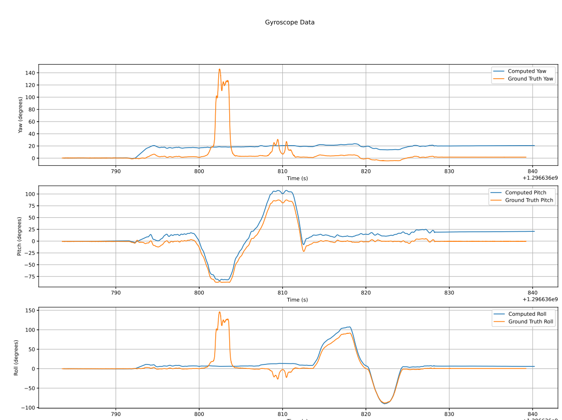

Fig. 1. Gyroscope Attitude Estimation for imuRaw1.mat

III. Attitude Estimation Using an Ideal Accelerometer

The accelerometer mathematical model is given by:

$$\mathbf{a} = {}^{W}R_B{}^{T}(\hat{\mathbf{a}} - \mathbf{g}^{W}) + \mathbf{b}_a + \mathbf{n}_a \tag{5}$$

Here, $\mathbf{a}$ is the measured acceleration from the acc, $\hat{\mathbf{a}}$ is the latent ideal acceleration we wish to recover, $\mathbf{R}$ is the orientation of the sensor in the world frame, $\mathbf{g}$ is the acceleration due to gravity in the world frame, $\mathbf{b}_a$ is the accelerometer bias which changes with time and other factors like temperature and $\mathbf{n}_a$ is the the white gaussian accelerometer noise.

We assume that our IMU is only rotating and we want to estimate the orientations in the next state ($\begin{bmatrix}\phi & \theta & \psi\end{bmatrix}_{t+1}^{T}$) when we have the linear accelerations in x, y and z direction in the previous state ($\begin{bmatrix}a_x & a_y & a_z\end{bmatrix}_{t}^{T}$). We also assume that the world frame is oriented in such a way that the negative z axis coincides with the gravity vector. Now, the only forces acting on the accelerometer are coming from the gravity vector. So,

$$Roll, \phi = \tan^{-1}\left(\frac{a_y}{\sqrt{{a_x}^2 + {a_z}^2}}\right)$$

$$Pitch, \theta = \tan^{-1}\left(\frac{-a_x}{\sqrt{{a_y}^2 + {a_z}^2}}\right)$$

It should be noted that since the accelerometer is symmetric with respect to the z axis, it does not provide any information about Yaw. However, since the IMU is never completely vertical, the Yaw can be computed as:

$$Yaw, \psi = \tan^{-1}\left(\frac{\sqrt{{a_x}^2 + {a_y}^2}}{a_z}\right)$$

The computed output for imuRaw1.mat is shown in Fig. 2

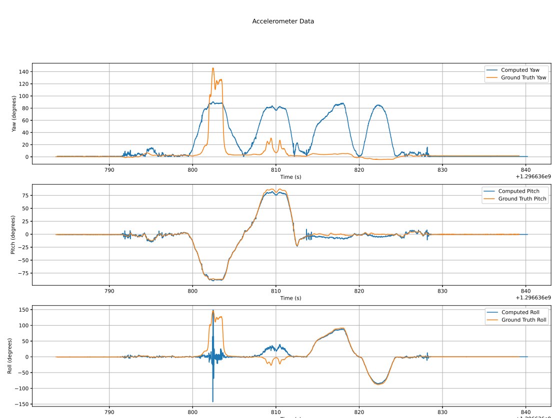

Fig. 2. Accelerometer Attitude Estimation for imuRaw1.mat

IV. Attitude Estimation Using Complementary Filter

In orientation estimation, gyroscope and accelerometer data play crucial roles, each with distinct advantages and limitations. However, gyroscopes are susceptible to bias and drift over time due to varying biases. Accelerometers, on the other hand, offer stability in long-term orientation estimation by utilizing gravity-influenced readings, but they falter in accurately capturing rapid movements. To counter noise in accelerometer data, low-pass filtering is applied, albeit at the cost of introducing a proportional lag represented by the filter's time constant (α). To tackle these challenges, a complementary filter is employed. Gyroscope data is high-pass filtered to alleviate bias-related drift, while accelerometer data is low-pass filtered to minimize noise. These filtered outputs are then weighted, scaled, and combined, harnessing the strengths of both sensors while mitigating their weaknesses. This approach results in accurate orientation estimates that are both responsive to fast changes and resistant to long-term drift.

Low pass filter formula: [α = 0.9]

$$\hat{\mathbf{a}}_{t+1} = (1-\alpha)\mathbf{a}_{t+1} + \alpha\hat{\mathbf{a}}_t \tag{6}$$

High pass filter formula: [α = 0.003]

$$\hat{\omega}_{t+1} = (1-\alpha)\hat{\omega}_t + (1-\alpha)(\omega_{t+1} - \omega_t) \tag{7}$$

Complementary Filter formula: [α = 0.5]

$$Ang_{t+1} = (1-\alpha)(Ang_t + \omega_{t+1}dt) + \alpha\mathbf{a}_{t+1} \tag{8}$$

The complementary filter is shown in Fig. 3.

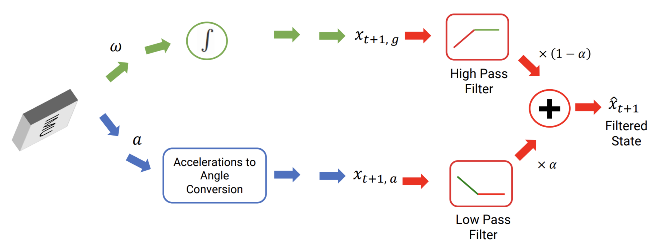

Fig. 3. Complementary Filter

The computed output for imuRaw1.mat is shown in Fig. 4

V. Attitude Estimation Using Madgwick Filter

The Madgwick filter operates by fusing data from the accelerometer and gyroscope to produce accurate and real-time attitude estimates. It combines sensor measurements with a quaternion representation $q = \begin{bmatrix}q1, q2, q3, q4\end{bmatrix}$ in w,x,y,z format which deals with the gimbal lock problem caused because of Euler angles. The major steps involved in implementing a Madgwick Filter are shown below and a detailed mathematical explanation for the same can be found in [1]. Fig. 5 shows the overview of the Madgwick Filter.

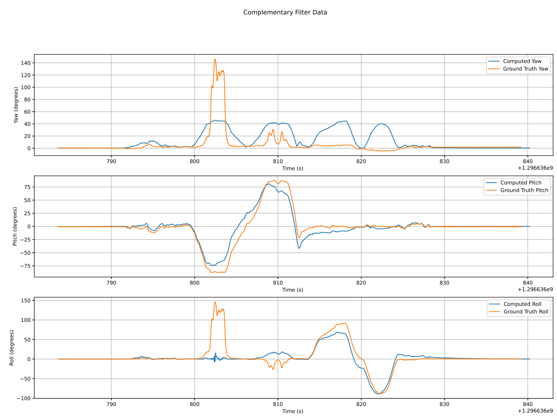

Fig. 4. Complementary Filter Attitude Estimation for imuRaw1

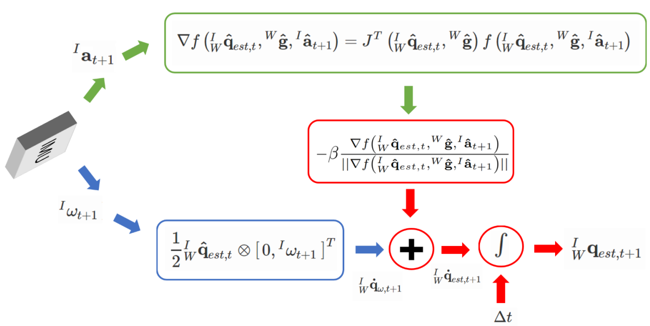

Fig. 5. Madgwick Filter

A. *Calculating Orientation Increment from Accelerometer:*

We model the attitude estimation of the accelerometer by modeling it as an optimization problem. We basically minimize the function (f).

$$\min_{{}^I_W\hat{\mathbf{q}} \in R^{4\times1}} f\left({}^I_W\hat{\mathbf{q}}, {}^W\hat{\mathbf{g}}, {}^I\hat{\mathbf{a}}\right) \tag{9}$$

$$f\left({}^I_W\hat{\mathbf{q}}, {}^W\hat{\mathbf{g}}, {}^I\hat{\mathbf{a}}\right) = {}^I_W\hat{\mathbf{q}}^* \otimes {}^W\hat{\mathbf{g}} \otimes {}^I_W\hat{\mathbf{q}} - {}^I\hat{\mathbf{a}} \tag{10}$$

where, $\hat{\mathbf{q}}$ is the normalized quaternion in w,x,y,z format, ${}^W\hat{\mathbf{g}}$ is the gravity vector in the world frame given by ${}^W\hat{\mathbf{g}} = \begin{bmatrix}0 & 0 & 0 & 1\end{bmatrix}^T$ and ${}^I\hat{\mathbf{a}}$ is the normalized accelerometer measurements in the Inertial frame. To minimize this function, we use a gradient descent algorithm:

$$\nabla f\left({}^I_W\hat{\mathbf{q}}_{est,t}, {}^W\hat{\mathbf{g}}, {}^I\hat{\mathbf{a}}_{t+1}\right) = \tag{11}$$
$$\mathbf{J}^T\left({}^I_W\hat{\mathbf{q}}_{est,t}, {}^W\hat{\mathbf{g}}\right) f\left({}^I_W\hat{\mathbf{q}}_{est,t}, {}^W\hat{\mathbf{g}}, {}^I\hat{\mathbf{a}}_{t+1}\right)$$

$$f\left({}^I_W\hat{\mathbf{q}}_{est,t}, {}^W\hat{\mathbf{g}}, {}^I\hat{\mathbf{a}}_{t+1}\right) = \begin{bmatrix} 2\left(q_2q_4 - q_1q_3\right) - a_x \\ 2\left(q_1q_2 + q_3q_4\right) - a_y \\ 2\left(\frac{1}{2} - q_2^2 - q_3^2\right) - a_z \end{bmatrix} \tag{12}$$

J is the Jacobian of the function given by,

$$J\left({}^I_W\hat{\mathbf{q}}_{est,t}, {}^W\hat{\mathbf{g}}\right) = \begin{bmatrix} -2q_3 & 2q_4 & -2q_1 & 2q_2 \\ 2q_2 & 2q_1 & 2q_4 & 2q_3 \\ 0 & -4q_2 & -4q_3 & 0 \end{bmatrix} \tag{13}$$

The gradient update is given as:

$${}^I_W\mathbf{q}_{\nabla,t+1} = -\beta \frac{\nabla f\left({}^I_W\hat{\mathbf{q}}_{est,t}, {}^W\hat{\mathbf{g}}, {}^I\hat{\mathbf{a}}_{t+1}\right)}{||f\left({}^I_W\hat{\mathbf{q}}_{est,t}, {}^W\hat{\mathbf{g}}, {}^I\hat{\mathbf{a}}_{t+1}\right)||} \tag{14}$$

where β is a tunable parameter that models the magnitude of gyroscope error in the direction of the accelerometer measurements. After making some assumptions we are able to get this simplified equation which will help us converge in one step. These assumptions are given in detail in [1].

B. *Calculating Orientation Increment from Gyroscope:*

When it comes to gyroscopic data, the filter performs an incremental update to refine the orientation estimation. Suppose gyroscope measurements, represented as $\boldsymbol{\omega}$, provide information about the angular velocity of the object. To perform an incremental update, we apply a mathematical operation called quaternion multiplication, denoted as $\otimes$, which combines our current orientation quaternion q with a new quaternion derived from the gyroscope data. This is defined by equation (15):

$$\mathbf{q}_1 \otimes \mathbf{q}_2 = \begin{bmatrix} w_1w_2 - x_1x_2 - y_1y_2 - z_1z_2 \\ w_1x_2 + x_1w_2 + y_1z_2 - z_1y_2 \\ w_1y_2 - x_1z_2 + y_1w_2 + z_1x_2 \\ w_1z_2 + x_1y_2 - y_1x_2 + z_1w_2 \end{bmatrix} \tag{15}$$

$${}^I_W\dot{\mathbf{q}}_{\omega,t+1} = \frac{1}{2} {}^I_W\hat{\mathbf{q}}_{est,t} \otimes \left[0, {}^I\omega_{t+1}\right]^T \tag{16}$$

In the equation (16): - ${}^I_W\hat{\mathbf{q}}_{est,t}$ represents our current orientation estimate in the inertial frame. - ${}^I_W\dot{\mathbf{q}}_{\omega,t+1}$ represents the orientation increment from gyroscope measurements. - $\left[0, {}^I\omega_{t+1}\right]$ represents the angular velocity as a quaternion.

The result of this computation represents the incremental change in orientation caused by the gyroscope measurements over the given time interval. This incremental update is then applied to the current orientation estimate, to refine the estimate of the object's orientation.

C. *Fuse Measurements:*

The orientation estimates from both the gyroscope and accelerometer are fused to obtain the estimated attitude ${}^I_W\mathbf{q}_{est,t+1}$:

$${}^I_W\dot{\mathbf{q}}_{est,t+1} = {}^I_W\dot{\mathbf{q}}_{\omega,t+1} + {}^I_W\mathbf{q}_{\nabla,t+1} \tag{17}$$

$${}^I_W\mathbf{q}_{est,t+1} = {}^I_W\hat{\mathbf{q}}_{est,t} + {}^I_W\dot{\mathbf{q}}_{est,t+1}\Delta t \tag{18}$$

This is repeated for each time step to calculate the orientation over time.

VI. Attitude Estimation using Unscented Kalman Filter

The Kalman filter is effective in estimating system attitude but relies on linear process models, which may not suit all systems. The Unscented Kalman filter (UKF) [2] addresses this limitation by processing the estimated state and covariance matrix through the actual system dynamics, allowing it to handle nonlinear processes. Unlike the Extended Kalman Filter (EKF), which linearizes model equations, the UKF approximates the Gaussian probability distribution using a set of sample points. This approach often yields more accurate results since it retains the original equations while also being computationally efficient due to the absence of Jacobi matrix computations. The filter starts by initializing a process noise matrix Q, measurement noise matrix R and a covariance matrix P. The state vector is defined as

$$\mathbf{x} = [q_w \quad q_x \quad q_y \quad q_z \quad \omega_x \quad \omega_y \quad \omega_z] \tag{19}$$

This is composed of the attitude quaternion and the angular velocities.

A. Prediction Step

1) Computing Sigma Points: The square root points are calculated as:

$$S = \sqrt{P_{k-1} + Q} \tag{20}$$

Here Q is the process model noise covariance matrix and P is the state covariance matrix. S is calculated using the Cholesky Root Decomposition method. The noise of the sigma points is calculated as:

$$\mathcal{W}_{i,i+n} \;=\; \text{columns}(\pm\sqrt{n \cdot (P_{k-1} + Q)}) \tag{21}$$

Here n was chosen instead of 2n because it converges faster and is numerically more stable. Finally, the sigma points are calculated as:

$$\chi_i = \begin{pmatrix} q_{t-1} q_{\mathcal{W}} \\ \omega_{t-1}^{\vec{}} + \vec{\omega}_{\mathcal{W}} \end{pmatrix} \tag{22}$$

Here $\vec{\omega}_{\mathcal{W}}$ is the angular velocity part of $\mathcal{W}$ and $q_{\mathcal{W}}$ is the quaternion part of $\mathcal{W}$.

$$\text{angle}: \qquad \alpha_{\mathcal{W}} = |\tilde{\mathcal{W}}_{\mathrm{q}}| \tag{23}$$

$$\text{axis}: \qquad \vec{e}_{\mathrm{w}} = \frac{\vec{\mathrm{w}}_k}{|\vec{\mathrm{w}}_q|}. \tag{24}$$

$$q_{\mathcal{W}} \;=\; \left[\cos\left(\frac{\alpha_{\mathcal{W}}}{2}\right),\; \vec{e}_{\mathcal{W}} \sin\left(\frac{\alpha_{\mathcal{W}}}{2}\right)\right] \tag{25}$$

2) Transforming Sigma Points: The sigma points are passed through the process model to project each point ahead in time, resulting in a different set of state vectors. It is given as:

$$\mathcal{Y}_i = \begin{pmatrix} q_{t-1} q_{\mathcal{W}} q_{\Delta} \\ \omega_{t-1}^{\vec{}} + \vec{\omega}_{\mathcal{W}} \end{pmatrix} \tag{26}$$

$$\text{angle}: \qquad \alpha_{\Delta} = |\tilde{\omega}_{\mathrm{k}}|\Delta \mathrm{t} \tag{27}$$

$$\text{axis}: \qquad \vec{e}_{\Delta} = \frac{\vec{\omega}_k}{|\vec{\omega}_k|}. \tag{28}$$

$$q_{\Delta} \;=\; \left[\cos\left(\frac{\alpha_{\Delta}}{2}\right),\; \vec{e}_{\Delta} \sin\left(\frac{\alpha_{\Delta}}{2}\right)\right] \tag{29}$$

3) Computing the Mean with Intrinsic Gradient Descent: The mean value is simply the sum over all elements of the set divided by the number of addends (2n). This is called the barycentric mean and is given by,

$$\overline{\mathcal{Y}} = \frac{1}{2n} \sum_{i=1}^{2n} \mathcal{Y}_i \tag{30}$$

The orientation component of $\{\mathcal{Y}_i\}$ is more difficult because orientations are periodic. In other words, they are members of a homogenous Riemannian manifold (the four dimensional unit sphere) but not of a vector space. We employ Intrinsic Gradient Descent to obtain the mean of sigma points. Initially, q_t is set as the first sigma point. In each iteration, we compute the error $\vec{e}_i$ from every sigma point to q_t. The average of these error vectors forms an average error $\vec{e}$, which we then transform back into a quaternion and apply to q_t. This iterative process continues until the mean error falls below a predefined threshold, resulting in the computation of the mean attitude.

Algorithm 1 Intrinsic Gradient Descent

Data $\mathcal{Y}$
Output $\bar{\hat{x}}$
$t \leftarrow 1$
$\bar{q}_t \leftarrow \mathcal{Y}_1$
while $t <$ *MaxIter* **and** $|e| >$*Thld* **do**
  for $\forall i$ **do**
    $q_i \leftarrow \mathcal{Y}_i$
    $\vec{e}_i \leftarrow q_i \bar{q}_t^{-1}$
  end
  $\vec{e} \leftarrow \frac{1}{2n}\sum_{i=1}^{2n} \vec{e}_i$
  $\bar{q}_{t+1} \leftarrow \vec{e}\ \bar{q}_t$
  $t \leftarrow t+1$
end while
$\bar{\omega} \leftarrow \frac{1}{2n}\sum_{i=1}^{2n} \omega_i$
$\bar{\hat{x}} \leftarrow [\bar{q}_t^T \quad \bar{\omega}^T]^T$

4) Computing the Covariances: The term $[\mathcal{X}_i - \bar{x}]$ is the difference between the sigma point and the mean of the distribution. With the mean state $\bar{\mu}$ from the sigma points, the covariance estimates $\bar{P}$, and the mean-centered sigma disturbances $\{\mathcal{W}_i\}$ can be computed.

$$P = \frac{1}{2n} \sum_{i=-1}^{2n} [\mathcal{X}_i - \bar{x}][\mathcal{X}_i - \bar{x}]^T \tag{31}$$

$$P_k^- = \frac{1}{2n} \sum_{i=1}^{2n} \mathcal{W}_i^{'} \mathcal{W}_i^{'T} \tag{32}$$

Similar to $\mathcal{W}_i, \mathcal{W}_i'$ has a rotation vector component $\vec{r}_{\mathcal{W}'}$ and an angular velocity vector component $\vec{\omega}\mathcal{W}'$

$$\mathcal{W}_i' = \begin{pmatrix} \vec{r}_{\mathcal{W}'} \\ \vec{\omega}_{\mathcal{W}'} \end{pmatrix} \tag{33}$$

$\vec{w}_{\mathcal{W}'}$ is the (standard vectorial) difference of the angular velocity components of $\mathcal{Y}_{\text{i}}$ and $\hat{x}_k^-$, denoted by

$$\vec{\omega}_{\mathcal{W}'} = \vec{\omega}_i - \overline{\vec{\omega}} \tag{34}$$

$\vec{r}_{\mathcal{W}'}$ is a representation of the rotation that turns the orientation part of $\hat{x}_k^-$ into $\mathcal{Y}_i$. The corresponding quaternion $r_{\mathcal{W}'}$ of this rotation is given by,

$$r_{\mathcal{W}'} = q_i \bar{q}^{-1} \tag{35}$$

Therefore,

$$\mathcal{W}_i^{'} = \begin{pmatrix} q_i \bar{q}^{-1} \\ \vec{\omega}_i - \bar{\vec{\omega}} \end{pmatrix} \tag{36}$$

The measurement model starts by calculating the estimated system measurements $\bar{z}_i$ from the sigma points. g is the gravity frame represented through quaternions.

$$\bar{z}_i = \begin{bmatrix} \mathcal{Y}_q^{-1} g \mathcal{Y}_q \\ \mathcal{Y}_\omega \end{bmatrix}_i \tag{37}$$

The mean of these is given by the barycentric mean formula as per equation 30.

The covariance of the measurement estimates is given as,

$$P_{zz} = \frac{1}{2n} \sum_{i=1}^{2n} [\mathcal{Z}_i - z_k^-][\mathcal{Z}_i - z_k^-]^T \tag{38}$$

From the measurement covariance, the innovation covariance can be calculated as the sum of the measurement covariance and the measurement noise matrix.

$$P_{\nu\nu} = P_{zz} + R \tag{39}$$

The cross-correlation matrix is computed as,

$$P_{xz} = \frac{1}{2n} \sum_{i=1}^{2n} [\mathcal{W}_i'][\mathcal{Z}_i - z_k^-]^T \tag{40}$$

B. Update Step:

The Kalman Gain is computed as:

$$K_k = P_{xz} P_{\nu\nu}^{-1} \tag{41}$$

The state update is given as:

$$\hat{x_k} = \hat{x}_{\bar{k}} + K_k \nu_k \tag{42}$$

The covariance update is given as:

$$P_k = P_k^- - K_k P_{vv} K_k^T \tag{43}$$

The process and measurement noise covariance matrices we used are as follows:

$$Q = \begin{bmatrix} 105 & 0 & 0 & 0 & 0 & 0 \\ 0 & 105 & 0 & 0 & 0 & 0 \\ 0 & 0 & 105 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1 \end{bmatrix} \tag{44}$$

$$R = \begin{bmatrix} 11.2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 11.2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 11.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.01 \end{bmatrix} \tag{45}$$

C. Summary of the Unscented Kalman Filter

1) The sum of the previous estimate error covariance P_{k-1} and process noise covariance Q is transformed into a set $\{\mathcal{W}_i\}$ of $2n$ six-dimensional vectors. This set is distributed around zero with the covariance $P_{k-1} + Q$.
2) The previous state estimate $\hat{x}_{k-1}$ is applied to $\{\mathcal{W}_i\}$, resulting in the set $\{\mathcal{X}_i\}$ of $2n$ seven-dimensional state vectors (sigma points).
3) The process model $A()$ transforms $\{\mathcal{X}_i\}$ into $\{\mathcal{Y}_i\}$.
4) The a priori estimate $\hat{x}_k^-$ is computed as the mean of the transformed sigma points $\{\mathcal{Y}_i\}$.
5) The set $\{\mathcal{Y}_i\}$ is transformed into the six-dimensional set $\{\mathcal{W}_i'\}$ by first removing the mean vector $\hat{x}_k^-$ from each element and then converting the quaternion part into a rotation vector.
6) The a priori process covariance P_k^- is computed from $\{\mathcal{W}_i'\}$. This concludes the time update step ("prediction").
7) The mean of $\{\mathcal{Z}_i\}$ is computed, giving the measurement estimate z_k^-. This is compared to the actual measured value z_k, their difference being ν_k, the innovation.
8) The innovation covariance $P_{\nu\nu}$ is determined by adding the measurement noise R to the covariance P_{zz} of the set $\{\mathcal{Z}_i\}$.
9) The cross correlation matrix P_{xz} is computed from the sets $\{\mathcal{W}_i'\}$ and $\{\mathcal{Z}_i\}$.
10) The Kalman gain K_k is first computed from P_{xz} and $P_{\nu\nu}$ and then used to calculate the a posteriori estimate $\hat{x}_k$ and its estimate error covariance P_k, which concludes the measurement update step ("correction").

VII. Results

- The plots for the estimated orientations from the Complementary Filter, Madgwick Filter, Unscented Kalman Filter plotted against the Vicon ground truths for the six training data sets are shown in Fig. 6, Fig. 7, Fig. 8, Fig. 9, Fig. 10 and Fig. 11 respectively.
- The plots for the estimated orientations from the Complementary Filter, Madgwick Filter, and Unscented Kalman Filter for the four test data sets are shown in Fig. 12, Fig. 13, Fig. 14, Fig. 15 and Fig. 11 respectively.
- Rotplot Output Link

VIII. Conclusion

- From the figures, we can observe that the gyroscope data is pretty accurate but suffers from drift over time while the accelerometer data has noise.

- The complementary filter is able to reduce the drift and noise after fusing the outputs of the accelerometer and the gyroscope data. However, the complementary filter still drifts over time, although not as bad as the gyroscope. Also, if we have violent motions, the complementary filter does not work well as it depends on the individual values of the gyroscope and the accelerometer.
- It was also observed that removing the bias, in the beginning, will give a better output in the complementary filter.
- From the plots, we can observe that the Madgwick Filter performs better than the Complementary filter for attitude estimation.
- Avoids Gimbal Lock: The use of quaternions eliminates the problem of gimbal lock, a limitation of Euler angles, ensuring robust and stable attitude estimation.
- Computationally Efficient: The Madgwick filter is computationally efficient as we use the gradient descent algorithm to optimize the function in a single step.
- The performance of UKF is comparable to Madgwick Filter and it even surpasses its performance at many points throughout the different datasets. Though its heavily influenced by the choice of initial process model noise and sensor noise covariance matrices. Nevertheless, it does a great job of recovering after periods of not aligning with the Vicon data.

REFERENCES

[1] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of imu and marg orientation using a gradient descent algorithm," in *2011 IEEE International Conference on Rehabilitation Robotics*, 2011, pp. 1–7.

[2] E. Kraft, "A quaternion-based unscented kalman filter for orientation tracking," in *Sixth International Conference of Information Fusion, 2003. Proceedings of the*, vol. 1, 2003, pp. 47–54.

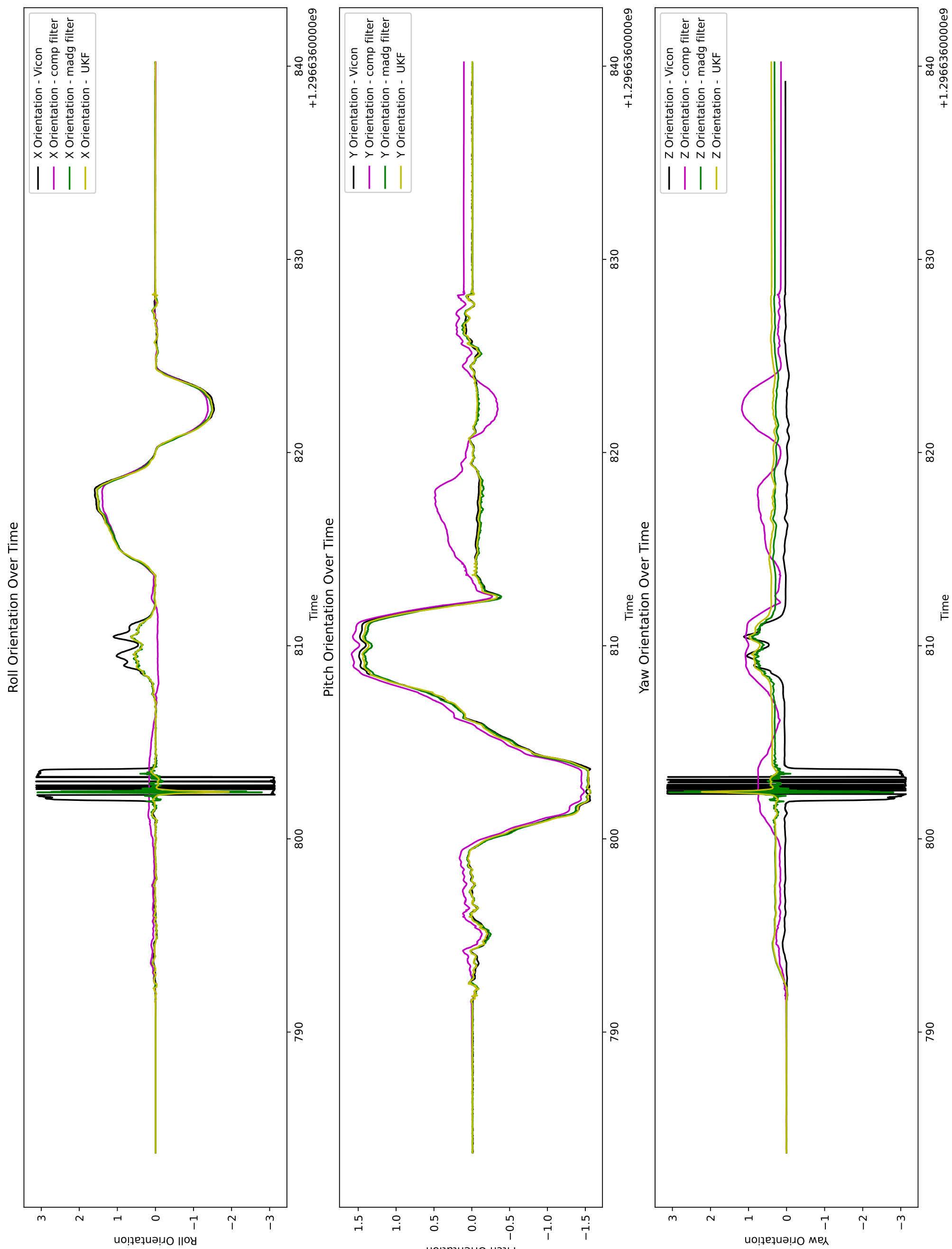

Fig. 6. Orientations for Dataset 1

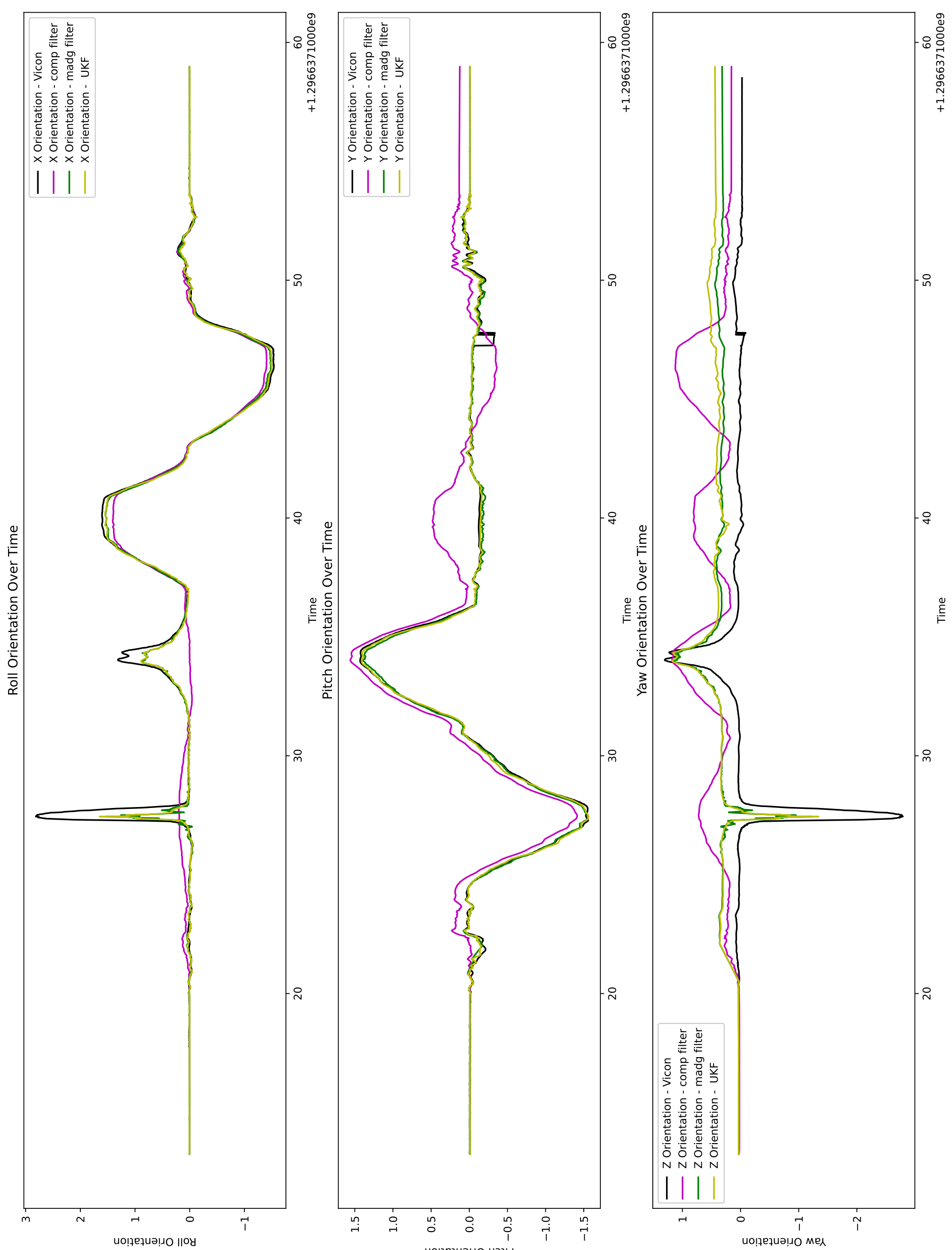

Fig. 7. Orientations for Dataset 2

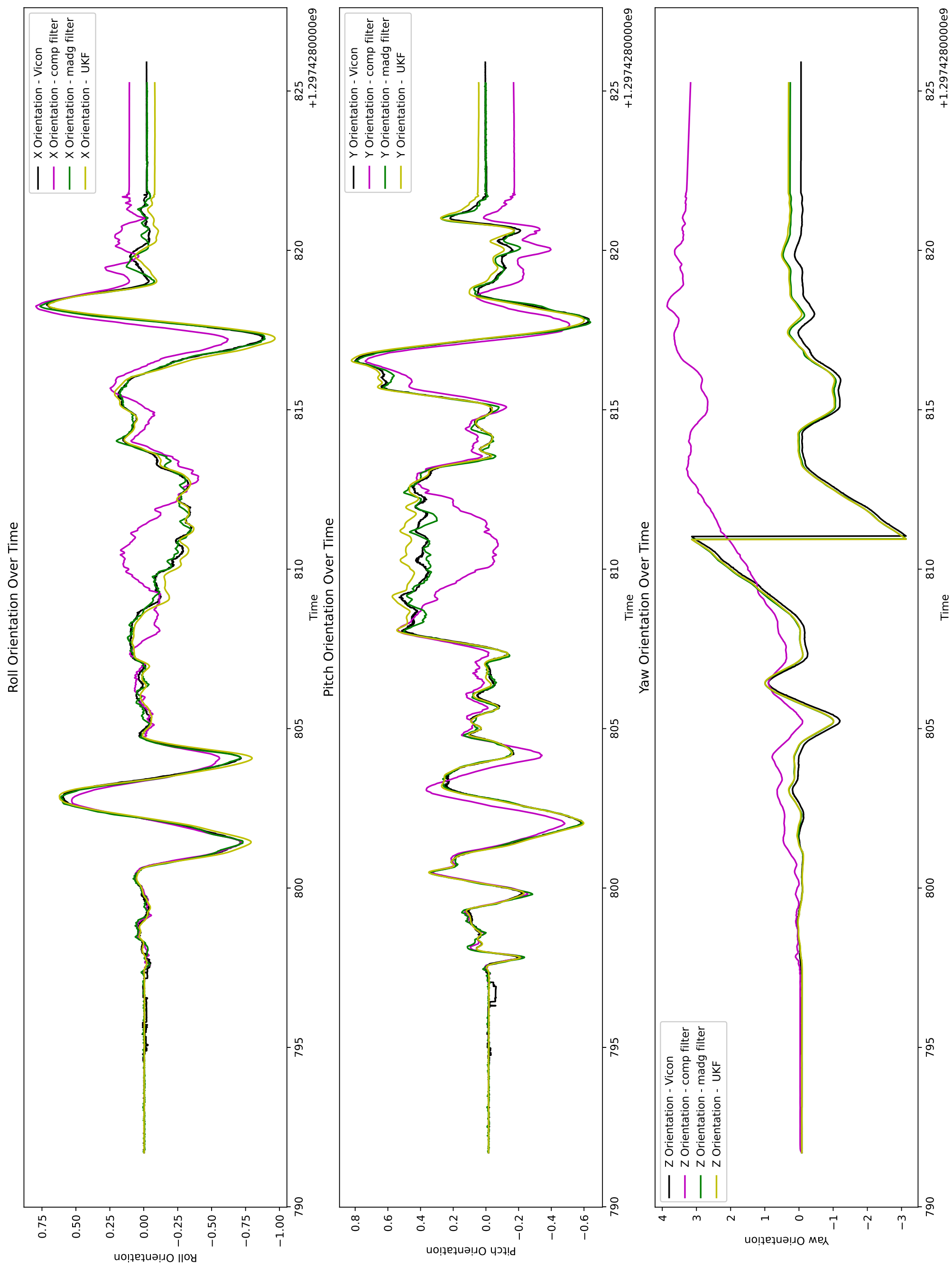

Fig. 8. Orientations for Dataset 3

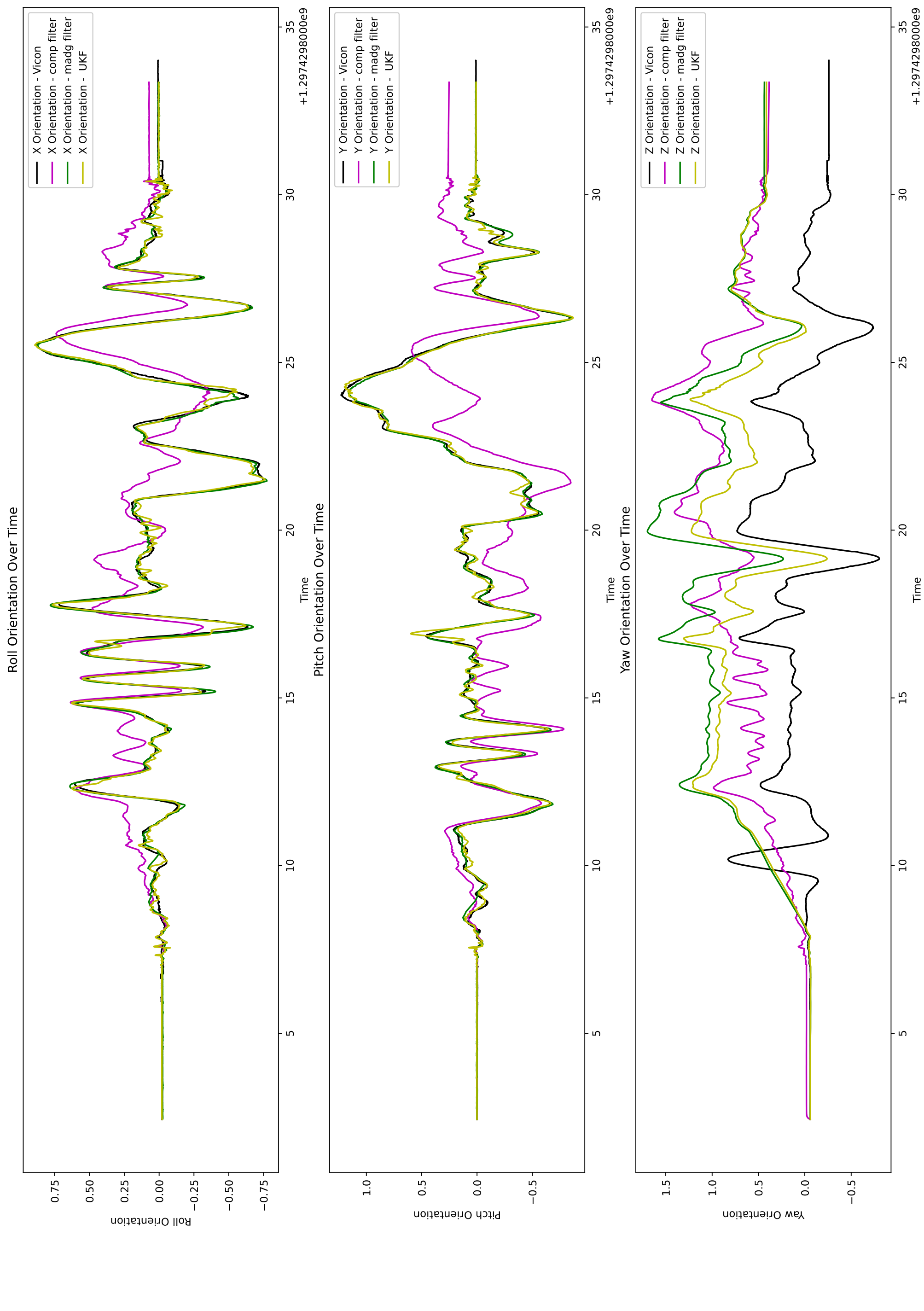

Fig. 9. Orientations for Dataset 4

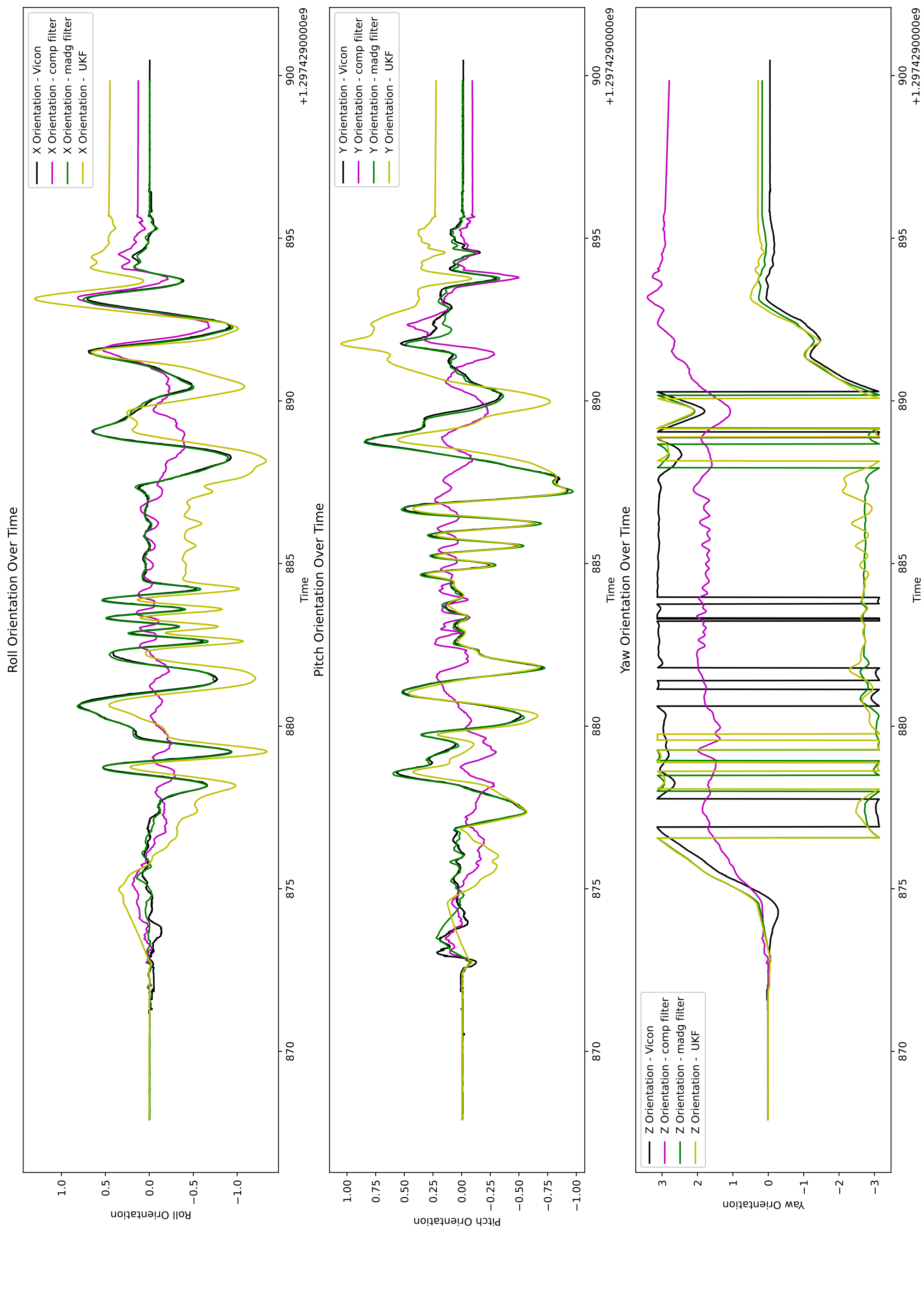

Fig. 10. Orientations for Dataset 5

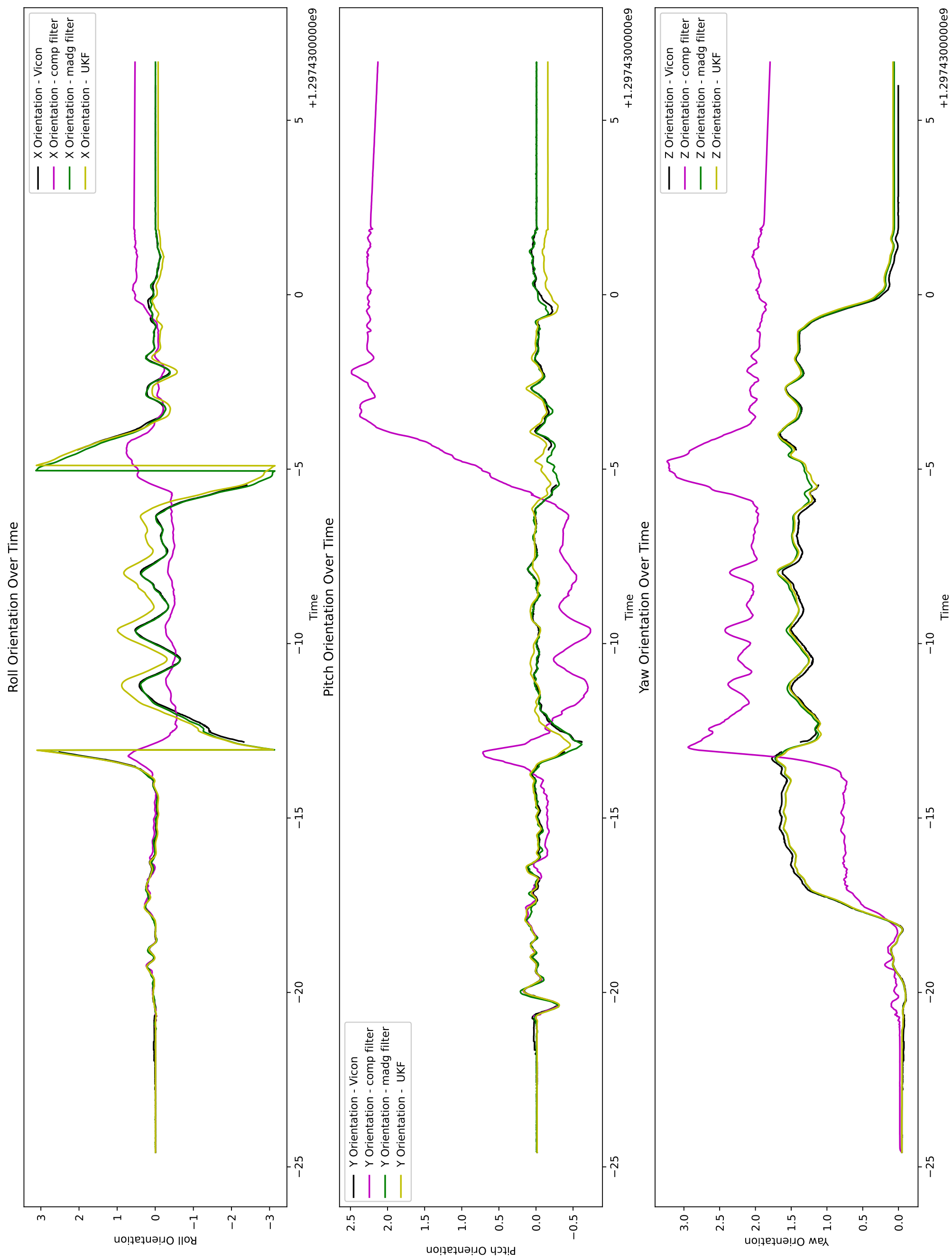

Fig. 11. Orientations for Dataset 6

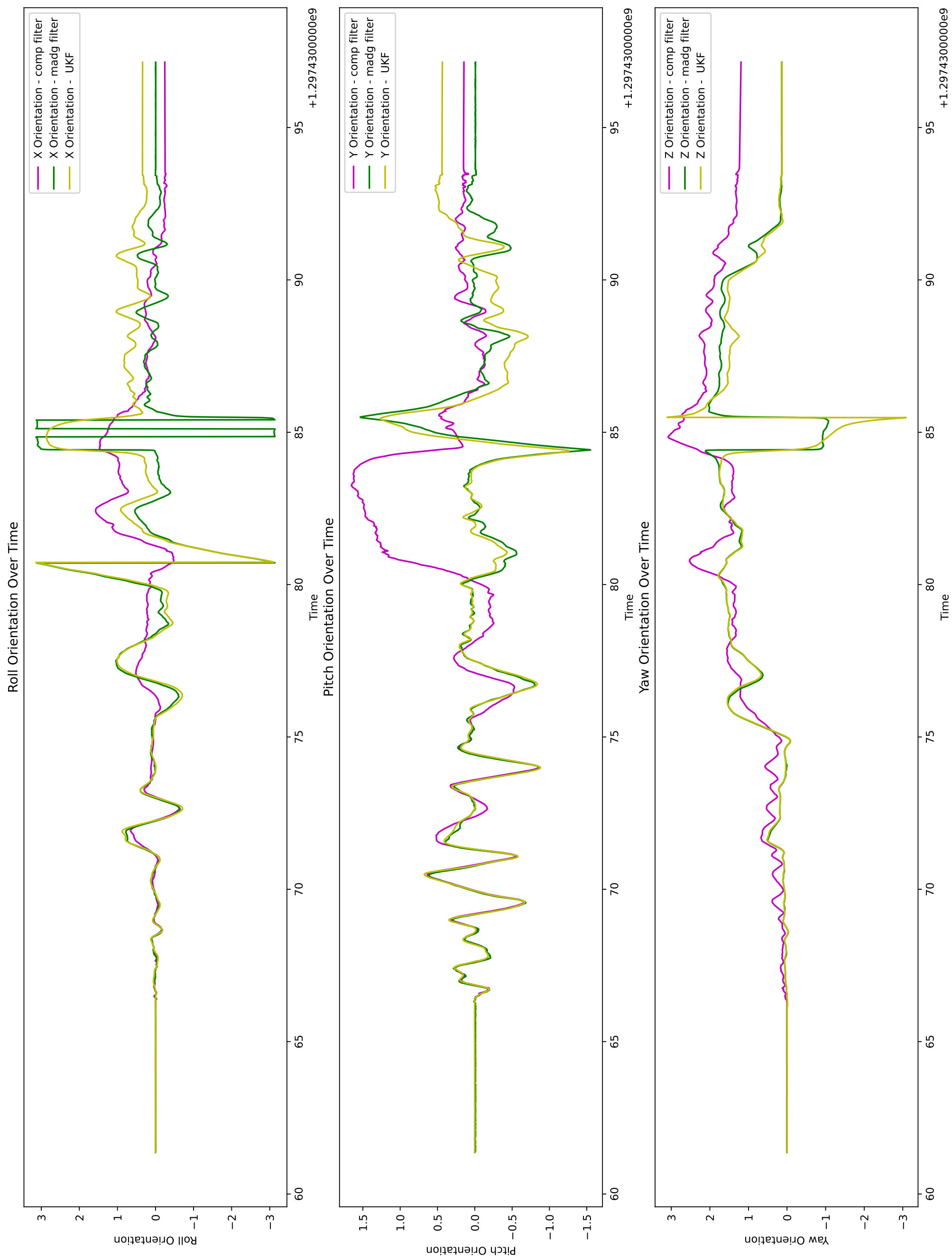

Fig. 12. Orientations for Dataset 7

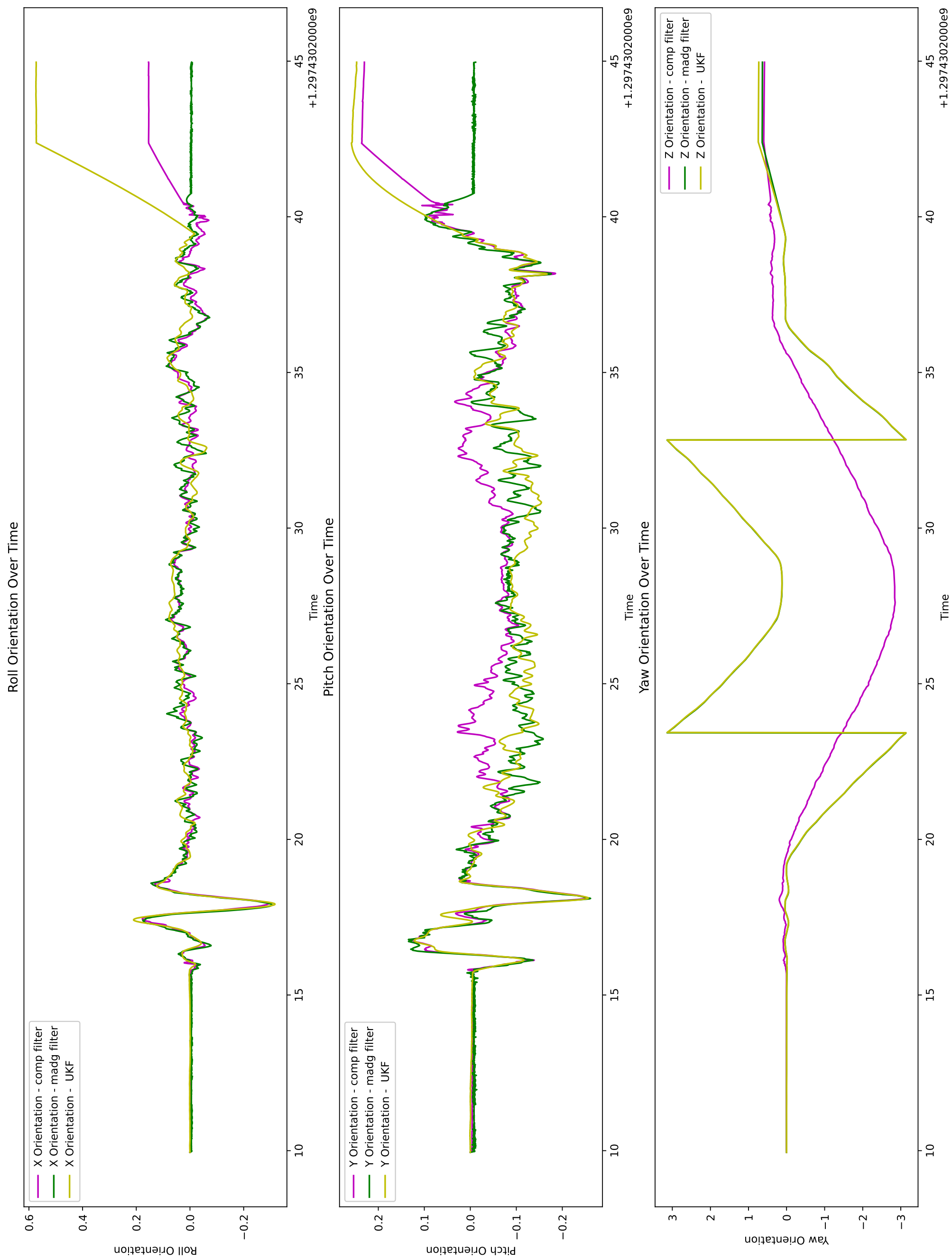

Fig. 13. Orientations for Dataset 8

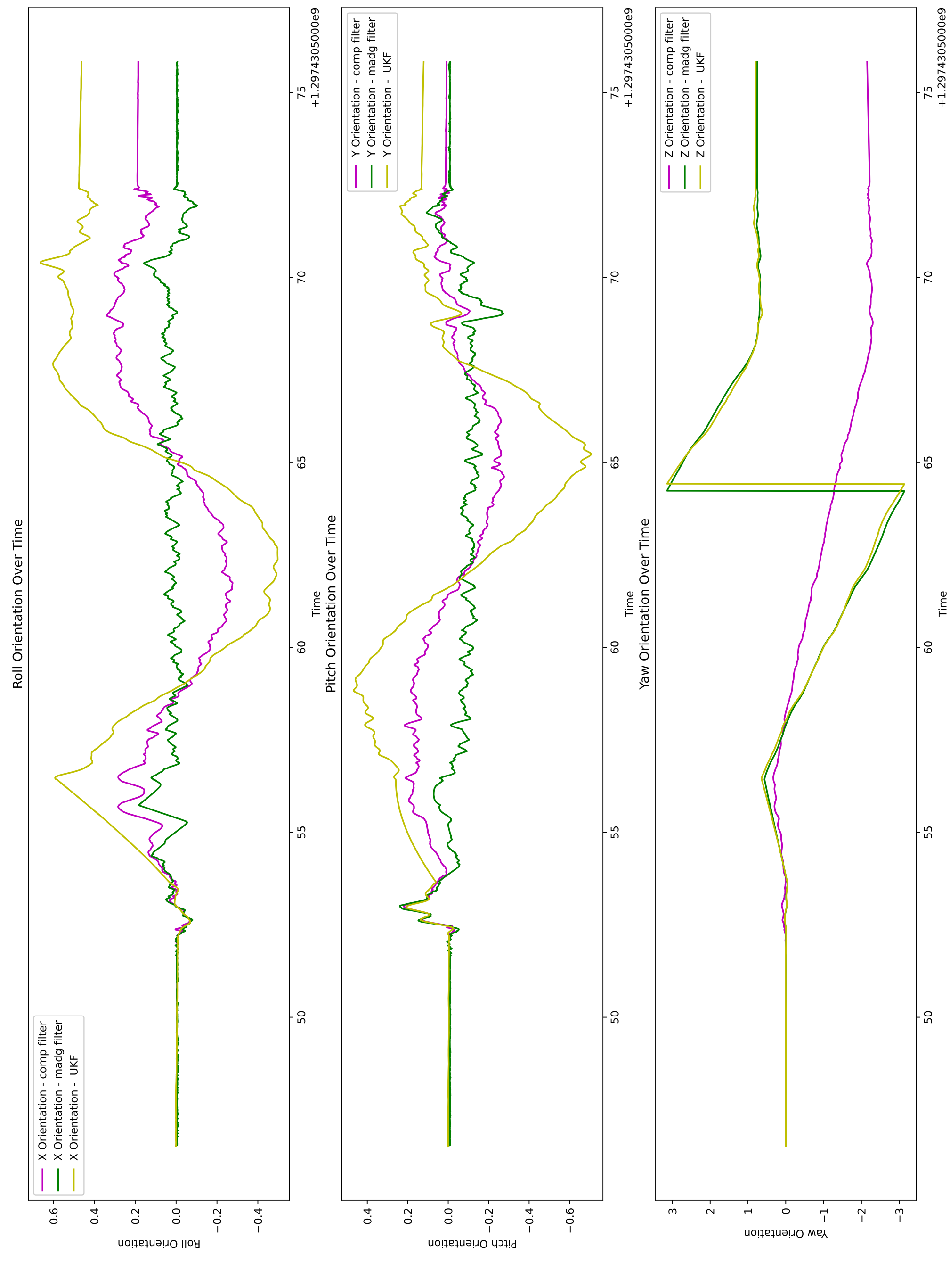

Fig. 14. Orientations for Dataset 9

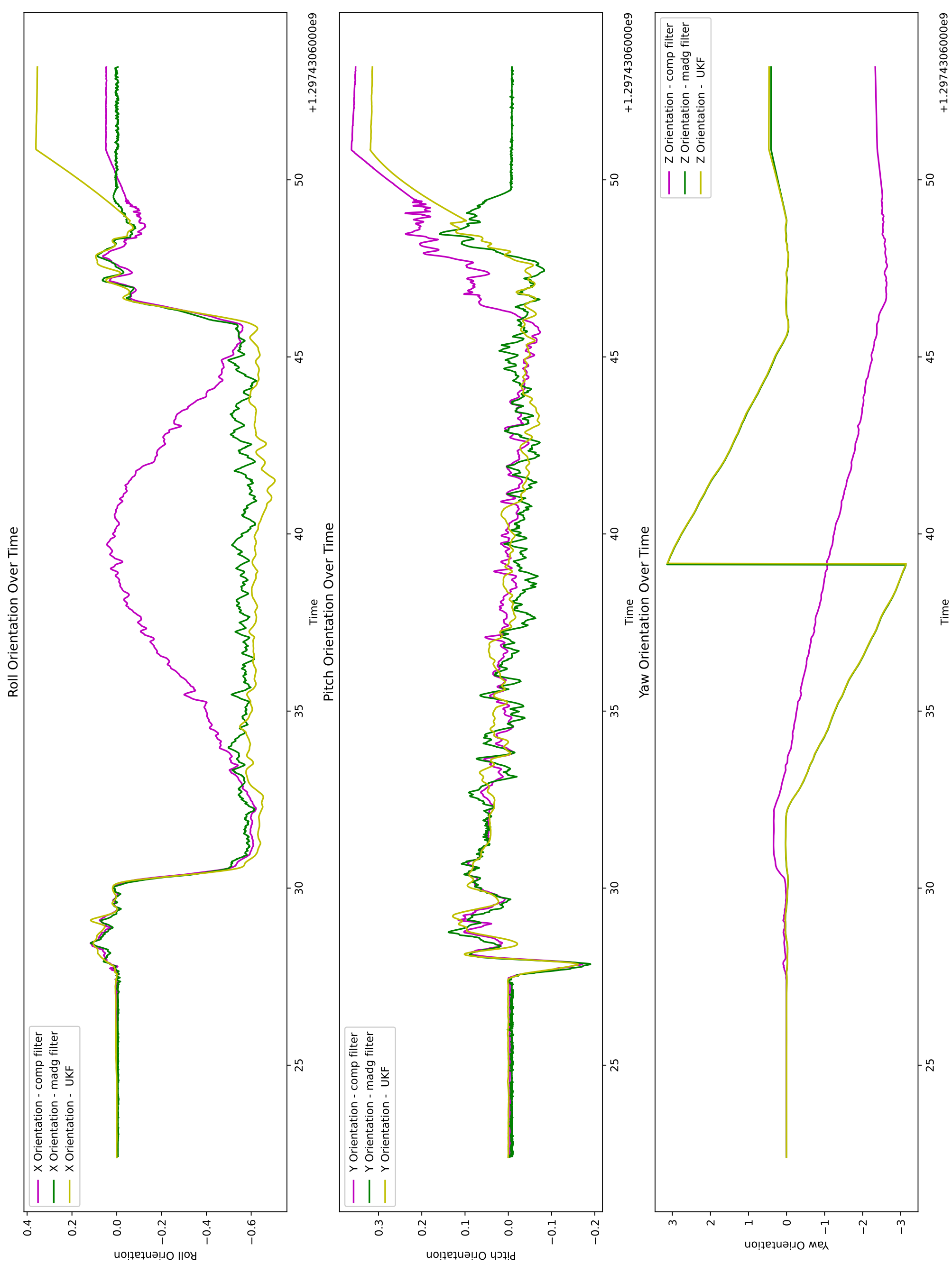

Fig. 15. Orientations for Dataset 10