Magic filtering for Attitude Estimation

Team Nimbus Navigators

Chaitanya Sriram Gaddipati
Department of Robotics Engineering
Worcester Polytechnic Institute
Worcester, Massachusetss 01609
Email: cgaddipati@wpi.edu

Ankit Talele
Department of Robotics Engineering
Worcester Polytechnic Institute
Worcester, Massachusetss 01609
Email: amtalele@wpi.edu

Shiva Surya Lolla
Department of Robotics Engineering
Worcester Polytechnic Institute
Worcester, Massachusetss 01609
Email: slolla@wpi.edu

***Abstract*—In this project attitudes are estimated for a given IMU dataset. The orientations are found using five methods: from only accelerometer measurements, from only gyroscope measurements, through a complementary filter, from a Madgwick filter, and from an Unscented Kalman filter. The resulting estimates are plotted and compared against the ground truth data from Vicon motion capture system for training datasets. The approaches are plotted and compared again for test datasets.**

I. Introduction

The aim of the project is to implement five methods to estimate the three dimensional orientation/attitude from the data collected from a six degree of freedom Inertial Measurement Unit (6-DoF IMU) sensor i.e., readings from a 3-axis gyroscope and a 3-axis accelerometer. The raw acceleration measurements are corrected and converted to SI units using the bias and scale parameters given. The gyroscope bias is estimated from the average of the initial gyroscope readings and is then used to convert the raw sensor data to SI units. Then attitudes need to be estimated from only accelerometer data, only gyroscope data, a complementary filter, a Madgwick filter, and a UKF filter. The estimates from only gyroscope suffer from drift due to error accumulation overtime as a result of numerical integration. Accelerometer estimates are affected by vibration and other external forces that causes translation. To get better estimates a complementary filter is implemented that combines both of the above estimates. To get even better estimates a Madgwick filter is implemented which performs the calculations using quaternions so that issues of singularity associated with Euler angles are avoided. A non-linear probabilistic UKF filter is also implemented. All the results are then compared with the ground truth data from Vicon motion capture system.

II. IMU and Vicon data pre-processing

In this section the raw sensor data from accelerometer and gyroscope are converted into physical values with corresponding SI units.

A. Accelerometer

From the IMU data the first three values in each measurement represent the accelerometer readings given as $\mathbf{a} = [a_x, a_y, a_z]^T$. To convert them to SI units (m/s^2) the following equation is used:

$$\tilde{a_i} = (a_i \times s_i + b_{i,a}) \times 9.81 \tag{1}$$

Here i is the sample number of the data, s_i is the scale for each axis, and $b_{i,a}$ is the accelerometer bias for each axis.

B. Gyroscope

From the IMU data the last three values in each measurement represent the gyroscope readings and are given as $\boldsymbol{\omega} = [\omega_z, \omega_x, \omega_y]^T$. To convert them to SI units (rad/s) the following equation is used:

$$\tilde{\omega_i} = \frac{3300}{1023} \times \frac{\pi}{180} \times 0.3 \times (\omega - b_{i,g}) \tag{2}$$

Here $b_{i,g}$ is the gyroscope bias for each axis and is obtained by taking the average of the first 200 readings of the gyroscope.

$$b_{i,g} = \frac{1}{200} \sum_{k=1}^{200} \omega_k \tag{3}$$

C. Vicon data (Ground truth)

The Vicon capture system measured the rotation matrix at every instance. The ZYX Rotation matrix is converted to roll, pitch, and yaw angles for comparing them with the estimates. Consider a single instance of ZYX rotation matrix R as shown below:

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \tag{4}$$

The Euler angles are then obtained as:

$$roll(\phi) = \arctan \frac{r_{32}}{r_{33}} \tag{5}$$

$$pitch(\theta) = \arctan \frac{-r_{31}}{\sqrt{r_{32}^2 + r_{33}^2}} \tag{6}$$

$$yaw(\psi) = \arctan \frac{r_{21}}{r_{11}} \tag{7}$$

III. Attitude from accelerometer

Attitude estimates from the accelerometer are obtained from the projection of the acceleration vector with onto the IMU body axis which are known from the gravity vector that is pointing downwards (-Z direction). The angles are given by the following formulas:

$$roll(\phi) = \arctan \frac{a_y}{\sqrt{a_x^2 + a_z^2}} \tag{8}$$

$$pitch(\theta) = \arctan \frac{-a_x}{\sqrt{a_y^2 + a_z^2}} \tag{9}$$

The estimates of yaw from this method are not perfect due to symmetrical nature of Z-axis with gravity vector. But since IMU is not perfectly vertical we can estimate yaw from the following formula:

$$yaw(\psi) = \arctan \frac{\sqrt{a_x^2 + a_y^2}}{a_z} \tag{10}$$

IV. Attitude from gyroscope

From the gyroscope we get the body angular velocities and the attitude estimates are obtained through the numerical integration. Before doing the integration the body angular velocities are converted to Euler angle rates. At every time step during the numerical integration the conversion is done using the following formula [1]:

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix}_{t_i} = \begin{pmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{pmatrix}_{t_i} \times \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix}_{t_i} \tag{11}$$

The Euler rates obtained from the above equation at every time step are then used find the attitude estimates for the next time step through the discrete numerical integration step shown below:

$$\begin{pmatrix} \phi \\ \theta \\ \psi \end{pmatrix}_{t_{i+1}} = \begin{pmatrix} \phi \\ \theta \\ \psi \end{pmatrix}_{t_i} + \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix}_{t_i} \times (t_{i+1} - t_i) \tag{12}$$

During the integration for the initial step the initial value of the estimate is obtained by taking the average of the first 200 estimates from the Vicon ground truth attitude as shown below:

$$\begin{pmatrix} \phi \\ \theta \\ \psi \end{pmatrix}_{t_0} = \frac{1}{200} \sum_{k=1}^{200} x_{vicon} \tag{13}$$

V. Attitude using complementary filter

The attitude estimates from accelerometer and the gyroscope each have complementary issues. The accelerometer estimates are prone to noise and the gyroscope estimates suffer from drift in the long term. These two estimates are combined using low and high pass filters respectively to reduce the error from individual estimates. First the raw sensor data obtained from the accelerometer is sent through a low pass filter by using the following equation to get new sensor values by combining the present sensor value with previous value:

$$\hat{a_{t+1}} = (1 - n) \times a_{t+1} + n \times \hat{a_t} \tag{14}$$

And similarly the sensor data of gyroscope is sent through high pass filter using the following equation:

$$\hat{\omega_{t+1}} = (1 - n)\hat{\omega_t} + (1 - n)(\omega_{t+1} - \omega_t) \tag{15}$$

Here n is taken as 0.8 and the variable $\hat{x}$ represents filtered value. Then the filtered values are used to obtain the orientation estimates which are fused using the following equation to get the new estimates:

$$\begin{pmatrix} \phi \\ \theta \\ \psi \end{pmatrix}_{comp} = \begin{pmatrix} 1-\alpha & 0 & 0 \\ 0 & 1-\beta & 0 \\ 0 & 0 & 1-\gamma \end{pmatrix} \times \begin{pmatrix} \phi \\ \theta \\ \psi \end{pmatrix}_{gyro} + \begin{pmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & \gamma \end{pmatrix} \times \begin{pmatrix} \phi \\ \theta \\ \psi \end{pmatrix}_{acc} \tag{16}$$

Here the parameters α, β, and γ are chosen to be $0.8, 0.8$, and 0.9 respectively.

VI. Attitude from Madgwick filter

The complimentary filter tries to fix the issues associated with accelerometer readings and gyroscope by combining both estimates in a linear manner. But since the estimates from only accelerometer are not accurate due to the issues with vector decomposition, complementary filter doesn't capture the ground truth properly. Additionally the usage of Euler angles in the previous filters leads to the condition of Gimbal lock or a singularity at certain angles. To avoid these issues and get better estimates a Madgwick filter is implemented. This filter utilises quaternions to represent the orientation and for the below equations it should be noted that the scalar element of the quaternion is the first element. For this project the initial estimate of quaternion is found from the average of first 200 values of vicon data for train data and is assumed to be zero for test data. Following are the steps followed to implement the filter:

A. Estimates from accelerometer:

To get the attitude estimates from accelerometer, the problem is formulated as a optmization problem. A gradient descent algorithm is used to find the estimate that minimizes the loss function. The loss function here tries reduce the difference between the measured acceleration and the attitude estimate that rotates the gravity vector to point in the direction of this measured acceleration vector. The following equations describe the optimization problem:

$$\arg\min_{\hat{q}} f(\hat{q}, \hat{g}, \hat{a}) \tag{17}$$

Where

$$f(\hat{q}, \hat{g}, \hat{a}) = \hat{q}^* \otimes \hat{g} \otimes \hat{q} - \hat{a} \tag{18}$$

Here $\hat{q}$ is the unit quaternion estimate , $\hat{q}^*$ is the conjugate of $\hat{q}$, $\hat{g} = [0, 0, 0, 1]^T$ is the unit gravity vector pointing down, and $\hat{a}$ is the IMU measured accelerometer values.
The gradient of the function is shown below:

$$\nabla f(\hat{q_{est,t}}, \hat{g}, \hat{a_{t+1}}) = J^T(\hat{q_{est,t}}, \hat{g}) f(\hat{q_{est,t}}, \hat{g}, \hat{a_{t+1}}) \quad (19)$$

The function can be calculated as:

$$f(\hat{q_{est,t}}, \hat{g}, \hat{a_{t+1}}) = \begin{pmatrix} 2(q_2 q_4 - q_1 q_3) - a_x \\ 2(q_1 q_2 + q_3 q_4) - a_y \\ 2(0.5 - q_2^2 - q_3^2) - a_z \end{pmatrix} \quad (20)$$

And the Jacobian is calculated as:

$$J(\hat{q_{est,t}}, \hat{g}) = \begin{pmatrix} -2q_3 & 2q_4 & -2q_1 & 2q_2 \\ 2q_2 & 2q_1 & 2q_4 & 2q_3 \\ 0 & -4q_2 & -4q_3 & 0 \end{pmatrix} \quad (21)$$

Assuming that the covergence rate of the quaternion estimate is equal to or greater than physical rate at which the orientation is changing the update term can be approximated as [2]:

$$q_{\nabla,t+1} = -\beta \frac{\nabla f(\hat{q_{est,t}}, \hat{g}, \hat{a_{t+1}})}{||\nabla f(\hat{q_{est,t}}, \hat{g}, \hat{a_{t+1}})||} \quad (22)$$

Where β is the tunable parameter that makes the gardient descent converge to the solution. For our project the best value is $\beta = 0.1$.

B. Estimates from gyroscope:

In this step the gyroscope measurements are used to find the quaternion derivative which will be used for numerical integration.

$$\dot{q}_{\omega,t+1} = \frac{1}{2} \hat{q_{est,t}} \otimes [0, \omega_{t+1}]^T \quad (23)$$

C. Fusion of estimates:

The values obtained from the previous two steps are fused to get a final estimate.

$$\dot{q}_{est,t+1} = \dot{q}_{\omega,t+1} + q_{\nabla,t+1} \quad (24)$$

$$q_{est,t+1} = \hat{q}_{est,t} + \dot{q}_{est,t+1} \Delta t \quad (25)$$

Here Δt is obtained by the difference in timestamps. It should be noted that the resulting quaternion estimate for $t+1$ is not unit quaternion so it must be normalized again.

VII. Attitude from Unscented Kalman filter

Unscented Kalman Filter has the advantage of using a non-linear model that accurately represents real-world scenarios. Our implementation uses 2 main steps, process update and measurement update [3]. We have a state vector of 7 states first four values give the attitude quaternion and last three are angular velocities,

$$x = [q_w, q_x, q_y, q_z, w_x, w_y, w_z]^T \quad (26)$$

In the process update the following equations are used:

A. Process update:

The covariance matrix P_{k-1} is initialized to a 6x6 matrix of zeros. The matrices Q and R that represent process noise and measurement noise are tunable matrices that needed fine-tuning to get the current results, but they were initialized with some random values to start with.
First, the Cholesky decomposition is done from P and Q matrices to compute the matrix square root S and subsequently the disturbances $\mathcal{W}$.

$$S = \sqrt{P_{k-1} + Q} \quad (27)$$

$$\mathcal{W}_{i,i+2n} = columns(\pm\sqrt{n}S) \quad (28)$$

Each disturbance computed corresponds to a sigma point with an overall number of sigma points of 12. The first 3 components of each $\mathcal{W}_i$ are used to compute the quaternion part of each sigma points and the next 3 components are used to compute the angular velocity part, given as follows:

$$(q_{\mathcal{W}})_i = \left[\cos\left(0.5 \times |\mathcal{W}_{1:3,i}|\right), \frac{W_{1:3}}{|\mathcal{W}_{1:3,i}|} \sin\left(0.5 \times |\mathcal{W}_{1:3,i}|\right) \right]^T \quad (29)$$

$$(\omega_{\mathcal{W}})_I = [\mathcal{W}_{4,i}, \mathcal{W}_{5,i}, \mathcal{W}_{6,i}] \quad (30)$$

From this the sigma points are calculated as:

$$\mathcal{X}_i = \begin{pmatrix} q_{k-1} q_{\mathcal{W}_i} \\ \omega_{k-1} + \omega_{\mathcal{W}_i} \end{pmatrix} \quad (31)$$

where q_{k-1} and ω_{k-1} are the quaternion and the angular velocity components of the previous state.
Each of the sigma points obtained is run through the process model to get individual state estimates $\mathcal{Y}_i$:

$$\mathcal{Y}_i = A(\mathcal{X}_i, 0) \quad (32)$$

Calculating the process model:

$$q_\Delta = \left[\cos\left(0.5 \left|\omega_{k-1}\right| dt\right), \frac{\omega_{k-1}}{|\omega_{k-1}|} \sin\left(0.5 \left|\omega_{k-1}\right| dt\right) \right] \quad (33)$$

The individual state estimate is then:

$$\mathcal{Y}_i = \begin{pmatrix} q_{t-1} q_{\mathcal{W}_i} q_\Delta \\ \omega_{t-1} + \omega_{\mathcal{W}_i} \end{pmatrix} \quad (34)$$

Algorithm 1 Algorithm for intrinsic gradient descent
Input: X_1, Y
Output: $\bar{x_k}$, vector of $\bar{e_i}$ (vector)
Initialisation :
Initialize $\bar{q}$ as X_1, mean error $\bar{e}$, and vector of $\bar{e_i}$
for $i = 1$ to $2n$ **do**
  $e_i = q_i \bar{q_t}^{-1}$
  $\bar{e_i} = quattorot(e_i)$
  vector.append($\bar{e_i}$)
end for
$\bar{e} = \frac{1}{2n} \sum_{i=1}^{2n} \bar{e_i}$
$e = rottoquat(\bar{e})$
$\bar{q}_{t+1} = e\bar{q}_t$
$\bar{\omega} = \frac{1}{2n} \sum_{i=1}^{2n} \bar{\omega_i}$
return $\bar{q}$, vector of $\bar{e_i}$,

Now we have the transformed sigma points, i.e., sigma points after the use of the process model which models the non-linearity of the system. The mean of the transformed sigma points is computed using gradient descent where an initial estimate is assumed and then computation is done by iterating a mean attitude error until convergence. This process is called intrinsic gradient descent from which the computed mean is:

$$\bar{x_k} = \begin{pmatrix} \bar{q} \\ \bar{\omega} \end{pmatrix} \tag{35}$$

The mean-centered transformed sigma disturbances $\mathcal{W}_i'$ are computed as:

$$\mathcal{W}_i' = \begin{pmatrix} q_{\mathcal{Y}i}(\bar{q})^{-1} \\ \omega_{\mathcal{Y}i} - \bar{\omega} \end{pmatrix} \tag{36}$$

But the first term above is already calculated from gradient descent as a vector of $\bar{e_i}$.
The covariance estimate $\bar{P}$ of the process model is:

$$\bar{P_k} = \frac{\Sigma \mathcal{W}_i' \mathcal{W}_i'^T}{2n} \tag{37}$$

With the mean and covariance of the transformed sigma points computed the process model has been now propagated.

B. Measurement update:

Next is the measurement model, where the estimated system measurements are calculated from the transformed sigma points.

$$Z_i = H(\mathcal{Y}_i, 0) \tag{38}$$

$$Z_i = \begin{pmatrix} \mathcal{Y}_q^{-1} g \mathcal{Y}_q \\ \mathcal{Y}_\omega \end{pmatrix}_i \tag{39}$$

The mean measurement estimate is computed as:

$$\bar{z_k} = \frac{\Sigma Z_i}{2n} \tag{40}$$

Additionally we calculate a term called innovation v_k which is the difference between the actual 6x1 vector of the accelerometer and gyroscope measurements stacked together and the above calculated mean measurement estimate:

$$v_k = z_k - \bar{z}_k \tag{41}$$

The covariance of measurement estimates (P_{zz}), innovation covariance (P_{vv}) and cross correlation matrix (P_{xz}) are calculated as:

$$P_{zz} = \frac{\Sigma [Z_i - \bar{z}_k][Z_i - \bar{z}_k]^T}{2n} \tag{42}$$

$$P_{xz} = \frac{\Sigma [\mathcal{W}_i'][\bar{Z} - \bar{z}_i]^T}{2n} \tag{43}$$

$$P_{vv} = P_{zz} + R \tag{44}$$

The Kalman gain is then computed as:

$$K = P_{xz} P_{vv}^{-1} \tag{45}$$

The final state estimate and state covariance can now be computed as:

$$P_k = \bar{P}_k - K P_{vv} K^T \tag{46}$$

$$\hat{x_k} = \hat{\bar{x}_k} + K(v_k) \tag{47}$$

The Q and R matrices were initialized as follows:

$$Q = \begin{pmatrix} 106 & 0 & 0 & 0 & 0 & 0 \\ 0 & 106 & 0 & 0 & 0 & 0 \\ 0 & 0 & 106 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 \end{pmatrix} \tag{48}$$

$$R = \begin{pmatrix} 8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1 \end{pmatrix} \tag{49}$$

VIII. Results

The resulting plots for the training data is shown in figures 1, 2, 3, 4, 5, and 6 below. The plots for test data is shown in 7, 8, 9, and 10. Additionally the plots for only accelerometer, only gyroscope, complimentary filter, and madgwick filter are plotted seperately to avoid clutter here: 11, 12, 13, 14, 15, 16, 17, 18, 19, 20.

IX. Conclusion

After looking at the plots for different training data it can be seen that the individual estimates from the gyroscope suffer with drift in the long term due to the numerical integration. The estimates from accelerometer are prone to noises and sudden movements. The complementary filter reduces the error from the sensors to some extent. The performance of the all the three filters are better for roll and pitch compared to yaw. Coming to the Madgwick filter since we are getting better accelerometer estimates we have better performance

than complimentary filter. After tuning the β parameter the Madgwick filter appears to follow the ground truth more closely. This filter is computationally inexpensive and less complex to implement.

In the Unscented Kalman Filter(UKF) it can be observed that it performs well and almost as good as the ground truth data, but the Madgwick filter follows the ground truth data better than the UKF. This might be because of the number of iterations of the intrinsic gradient descent for convergence leading to lower performance compared to just one step to converge in the case of Madgwick filter. The UKF might also need more tuning, although we tried to tune the values of the matrices Q and R as much as possible.

The link for the 'rotplot' videos are here: link1, link2, link3, link4, link5, and link6.

REFERENCES

[1] R. W. Beard, "Quadrotor dynamics and control rev 0.1," 2008. [Online]. Available: https://api.semanticscholar.org/CorpusID:60131240

[2] 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.

[3] 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.

[4] N. J. Sanket, "Orienation tracking based panorama stitching using unscented kalman filter." [Online]. Available: https://github.com/NitinJSanket/ESE650Project2/blob/master/Report/ESE650Project2.pdf

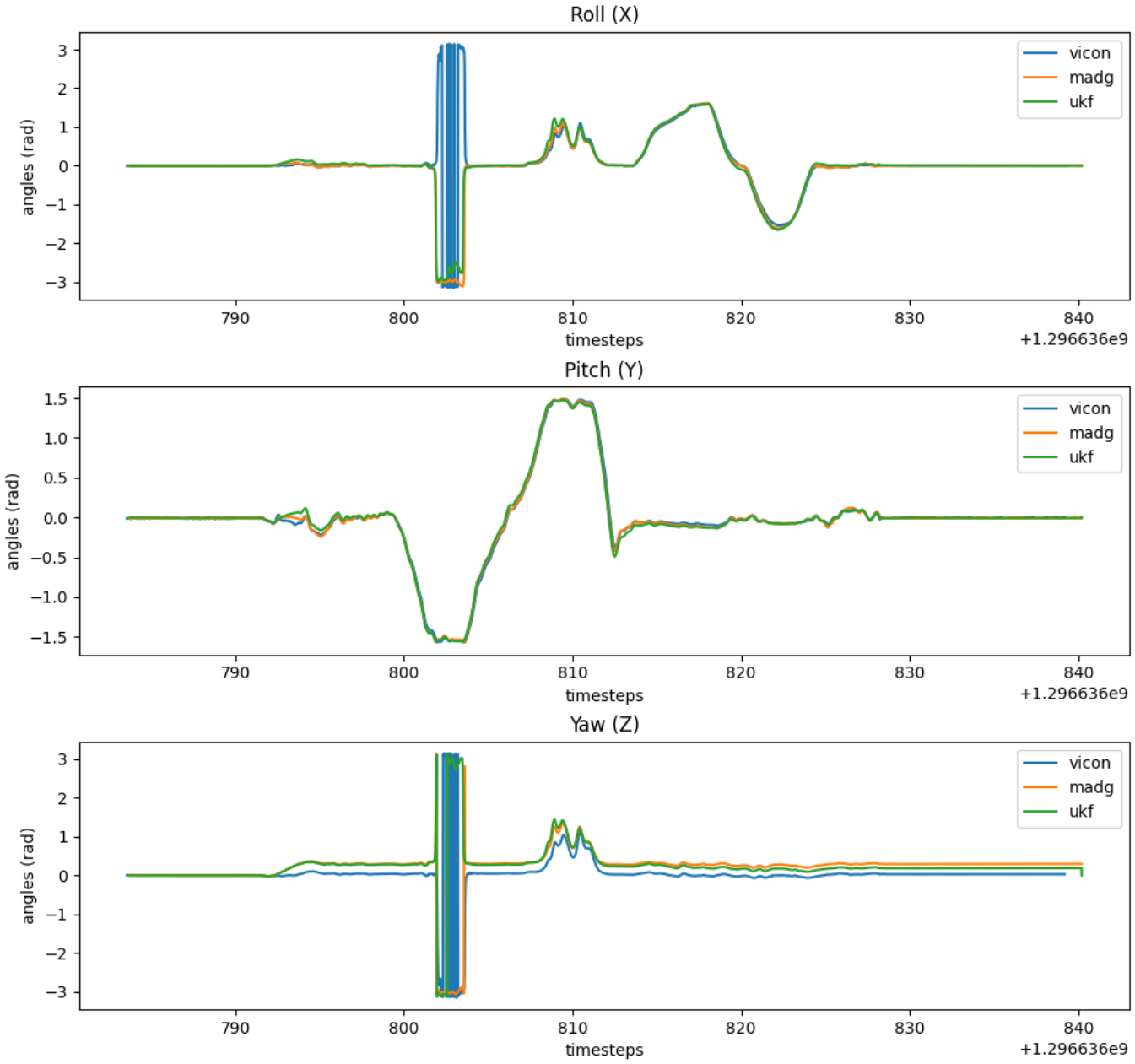

Fig. 1. Euler angle plots for train data 1.

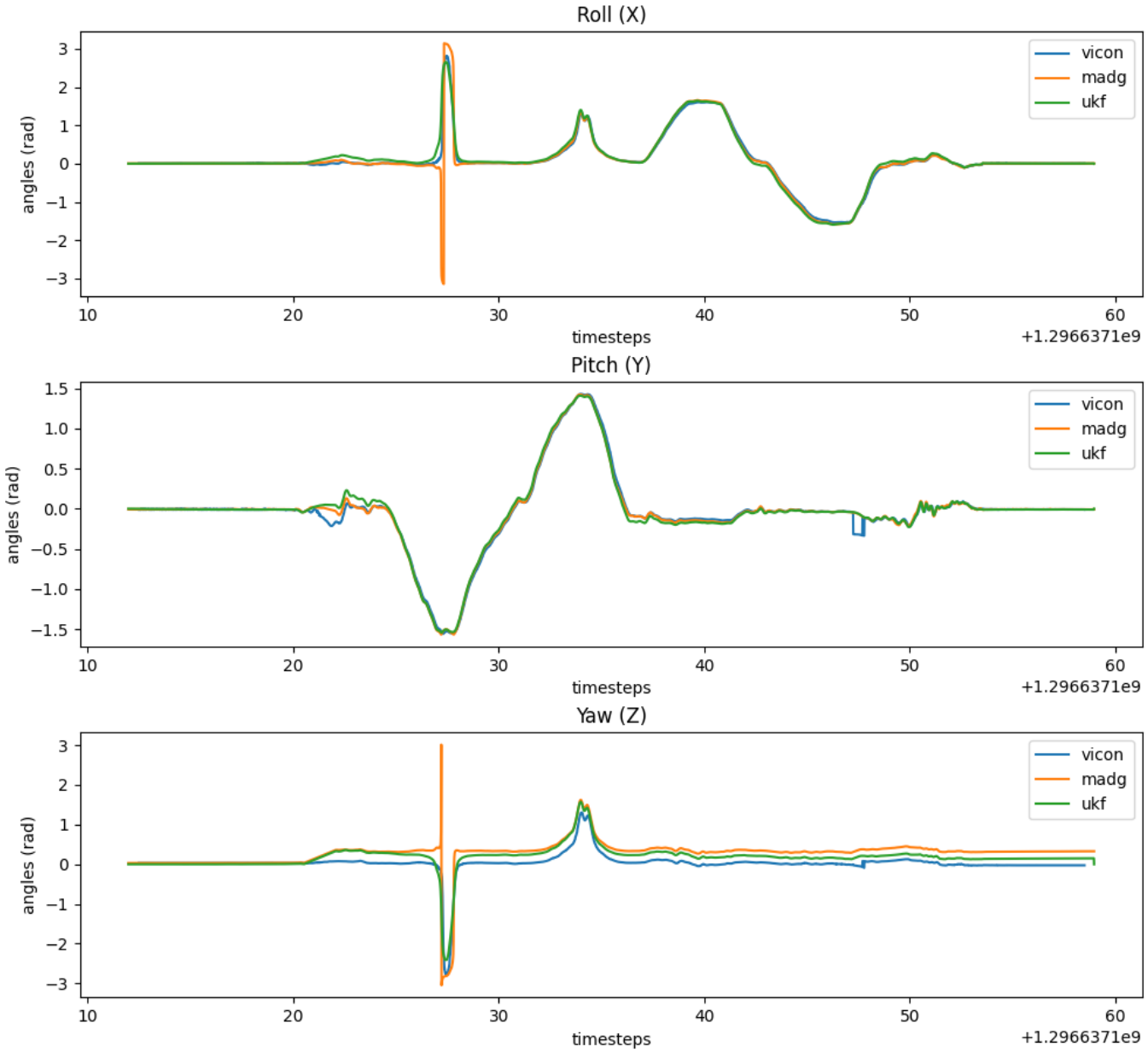

Fig. 2. Euler angle plots for train data 2.

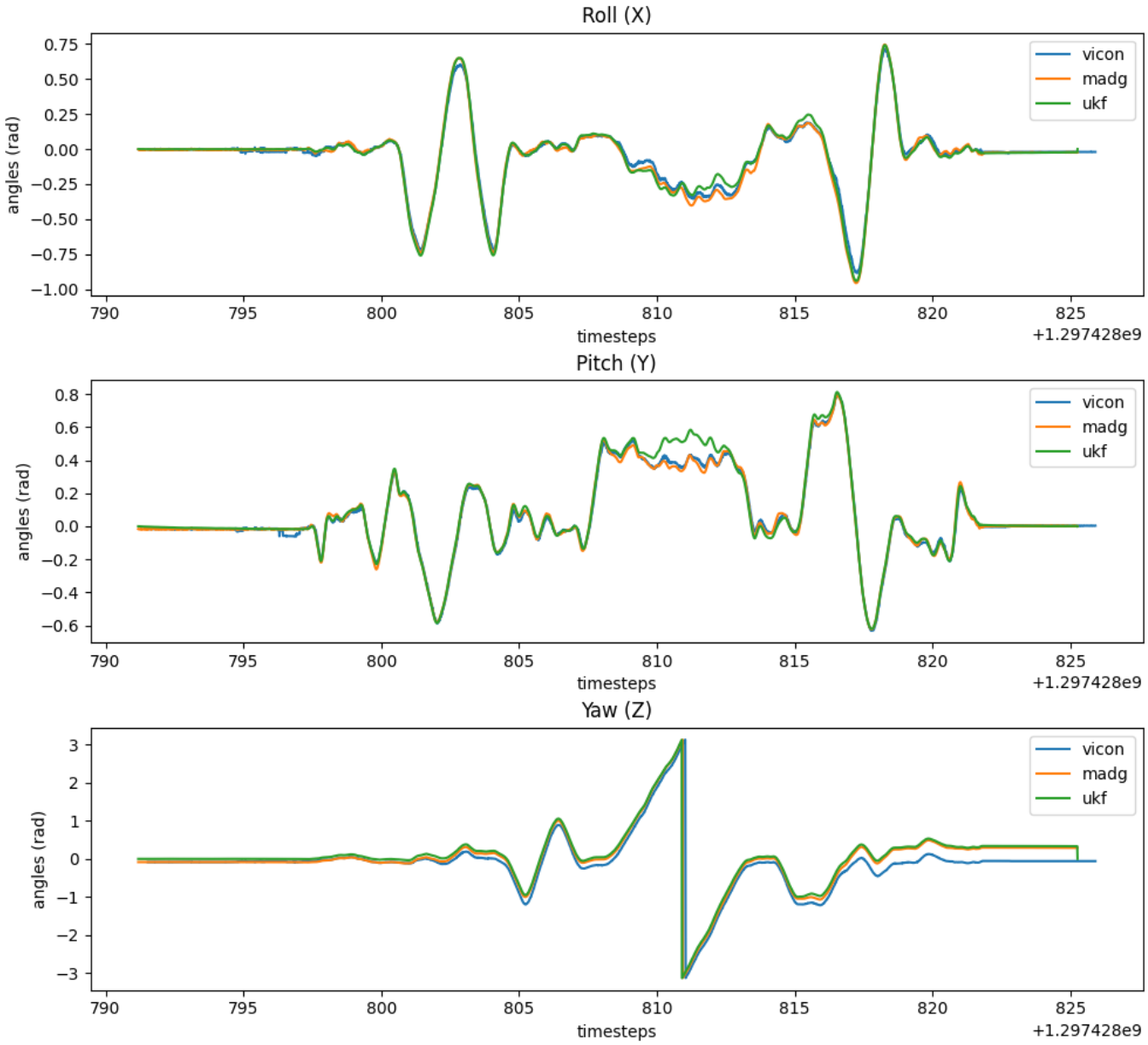

Fig. 3. Euler angle plots for train data 3.

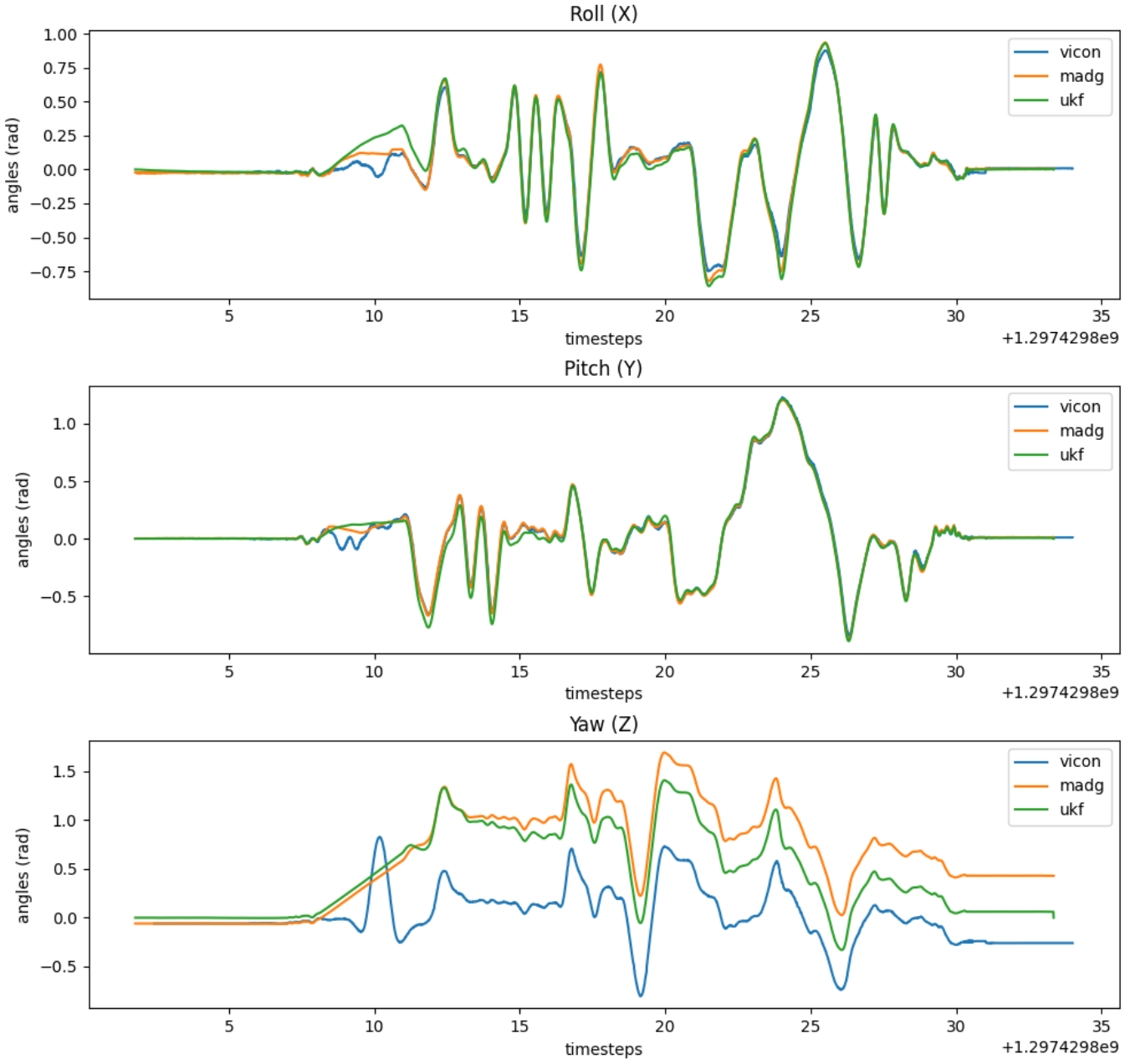

Fig. 4. Euler angle plots for train data 4.

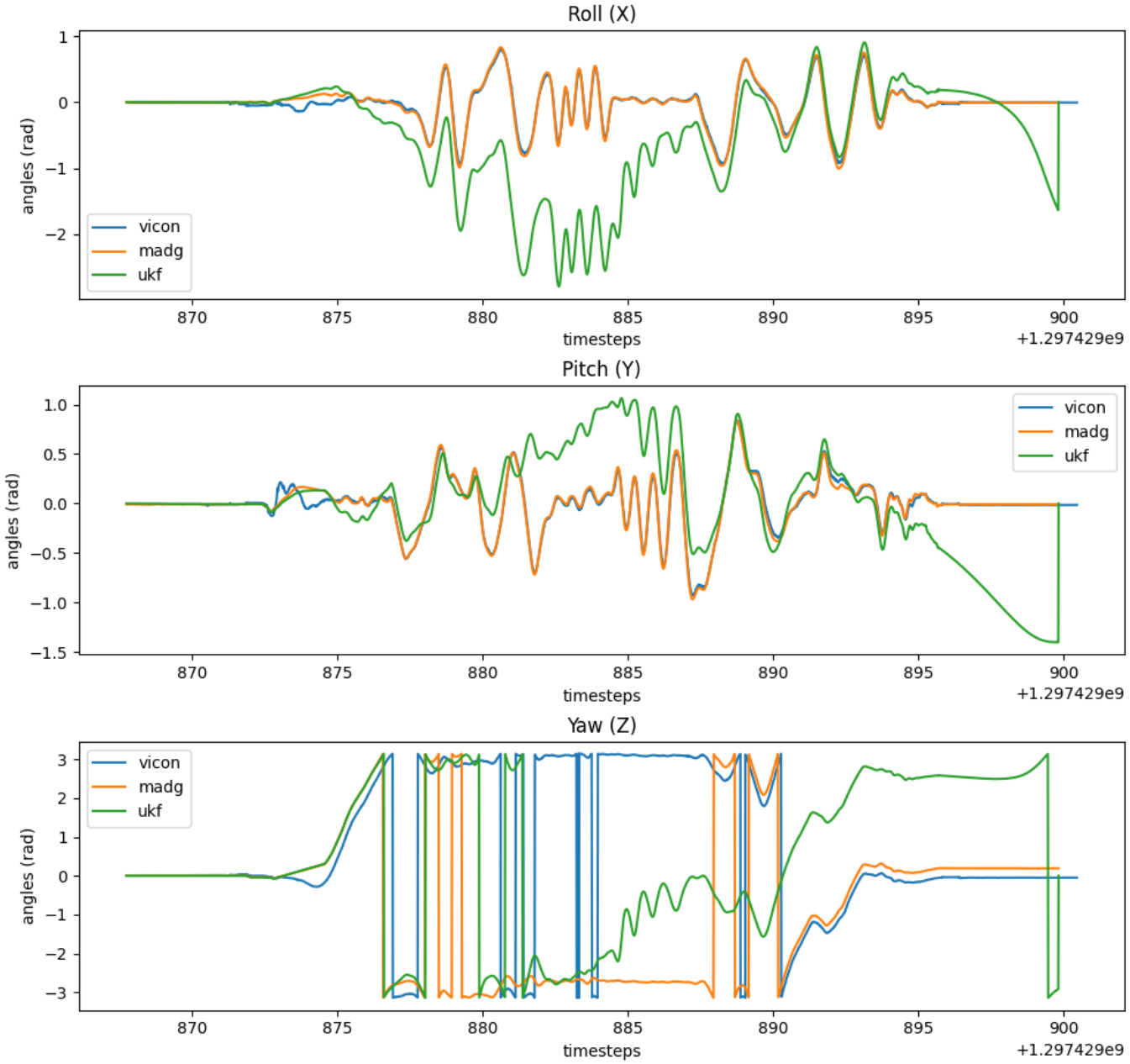

Fig. 5. Euler angle plots for train data 5.

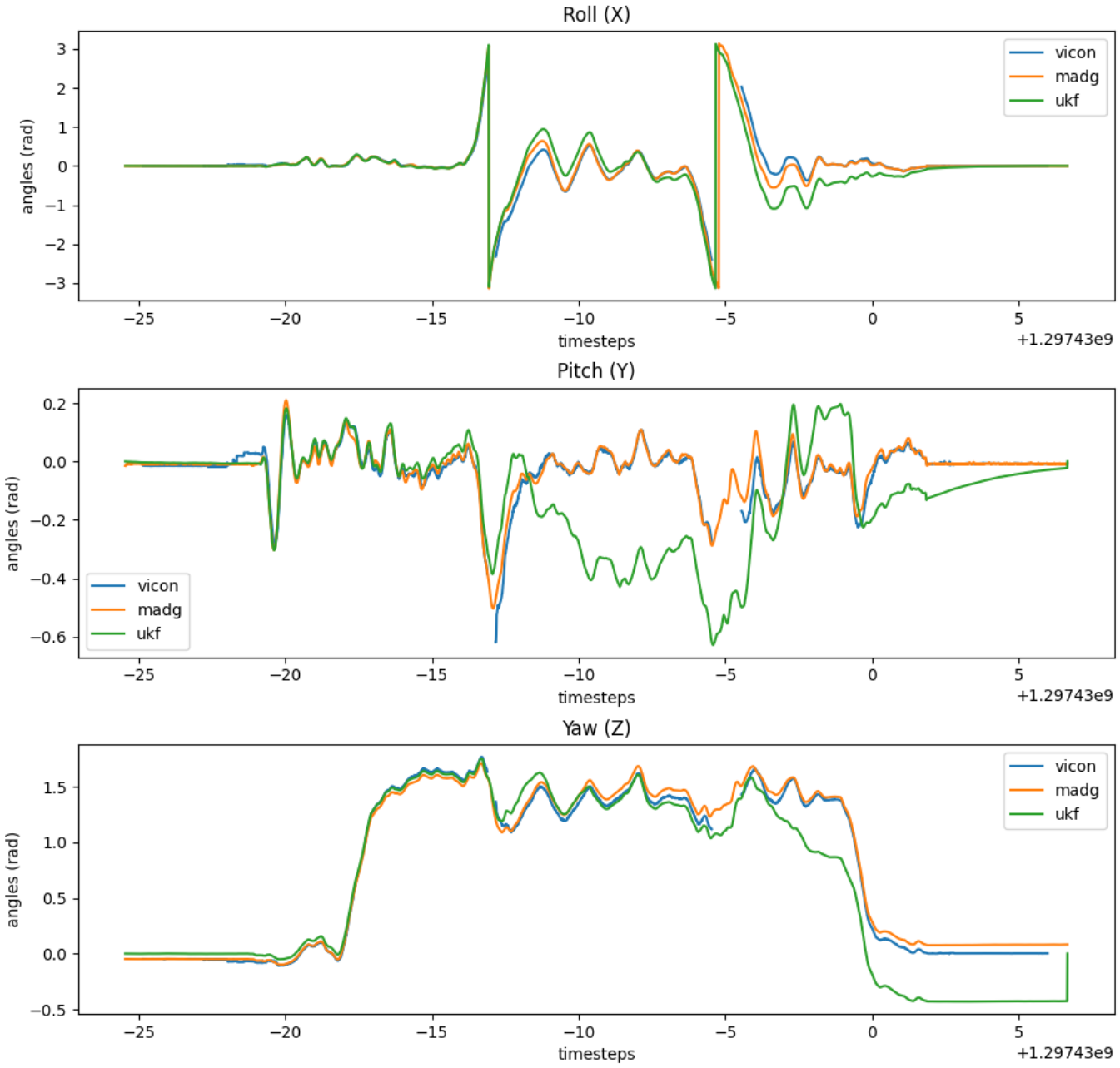

Fig. 6. Euler angle plots for train data 6.

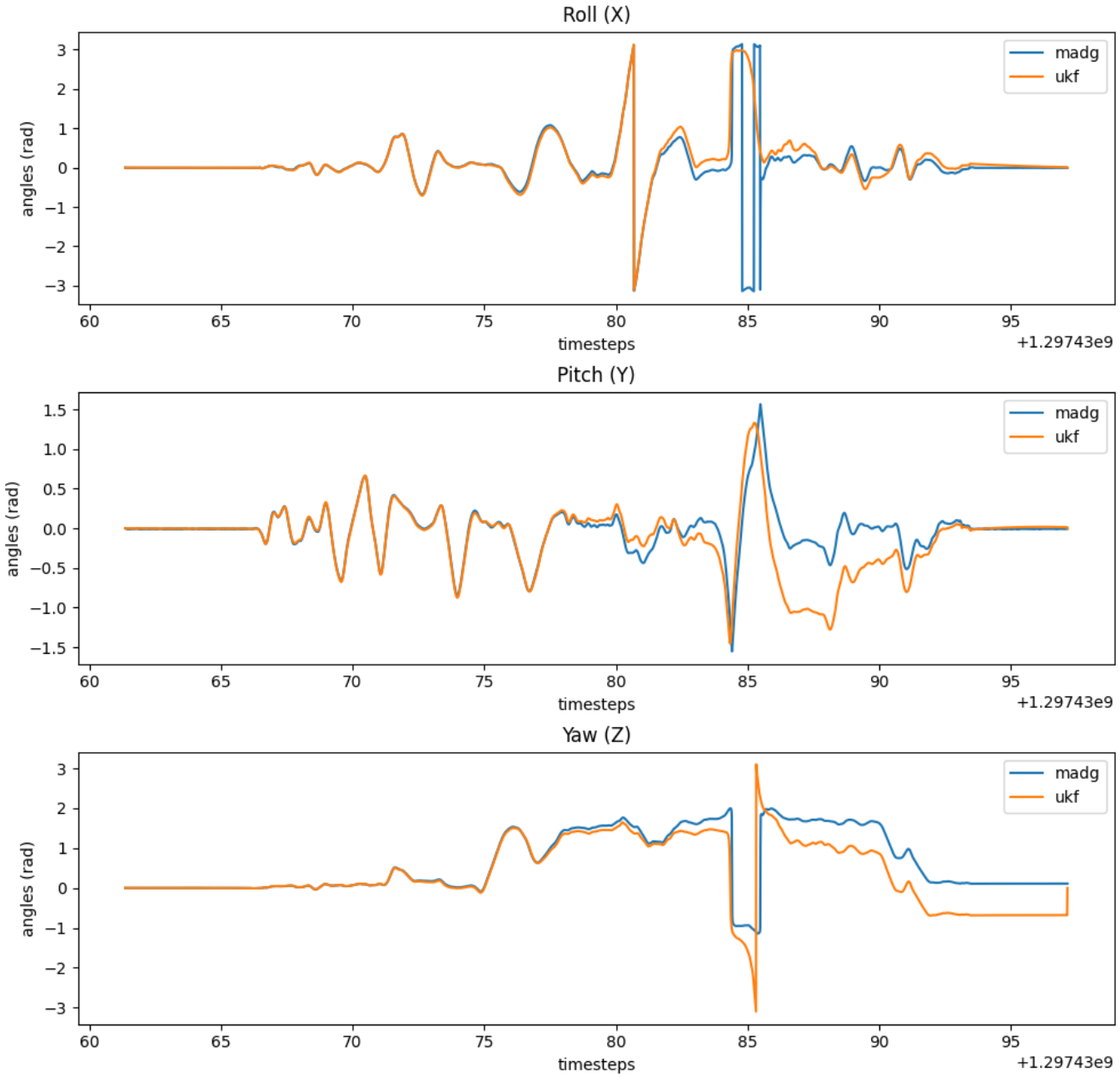

Fig. 7. Euler angle plots for test data 7.

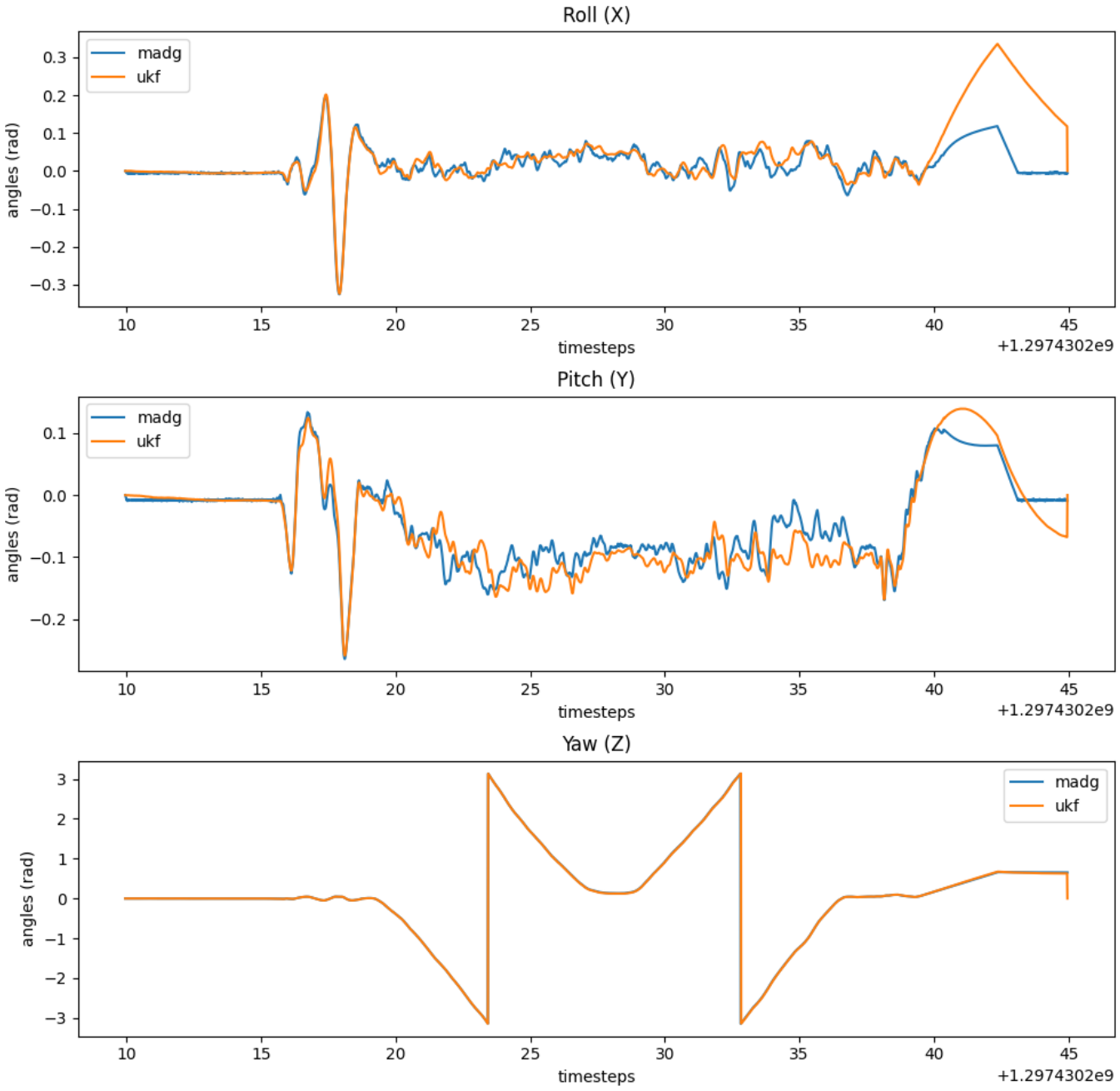

Fig. 8. Euler angle plots for test data 8.

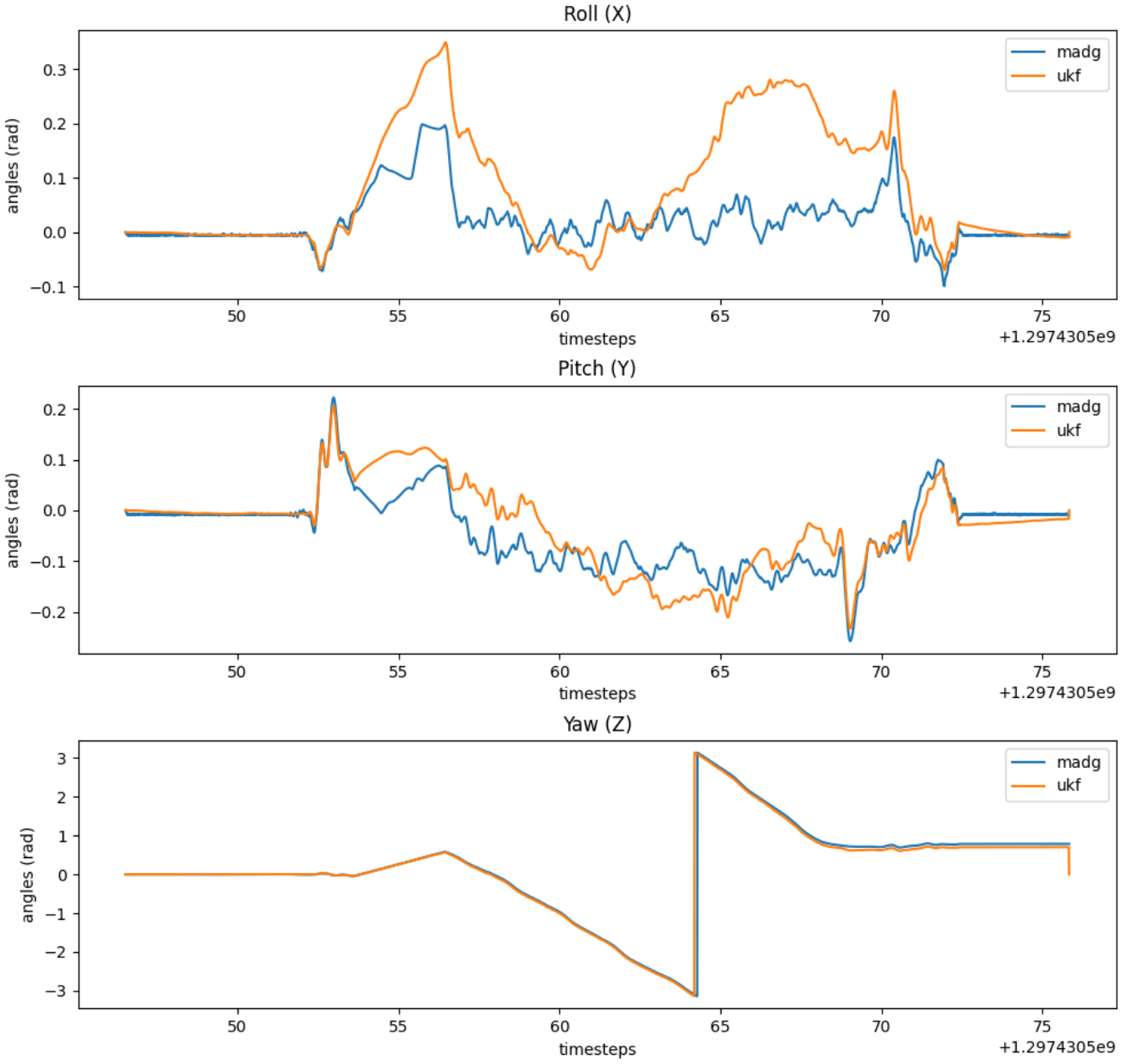

Fig. 9. Euler angle plots for test data 9.

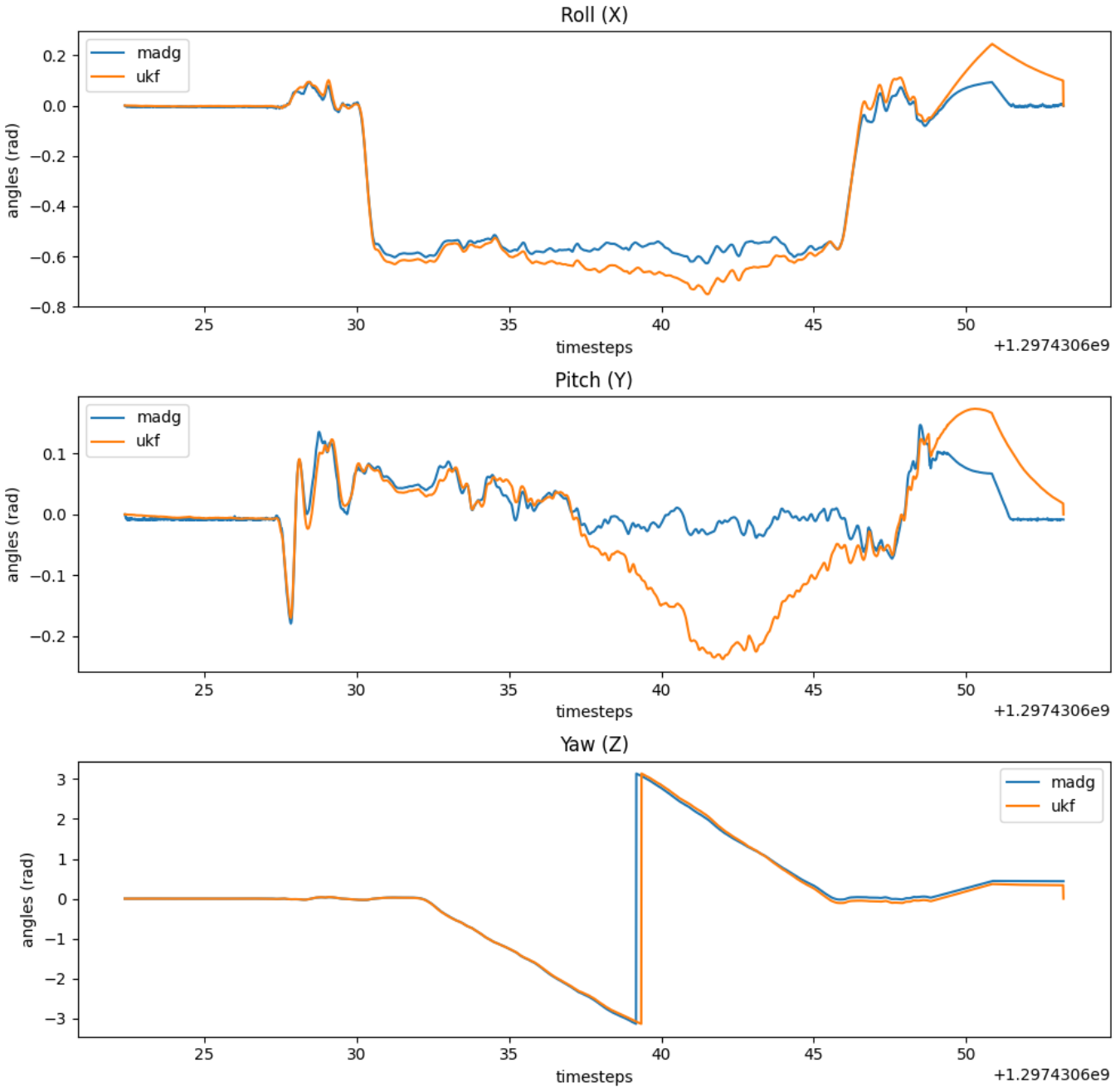

Fig. 10. Euler angle plots for test data 10.

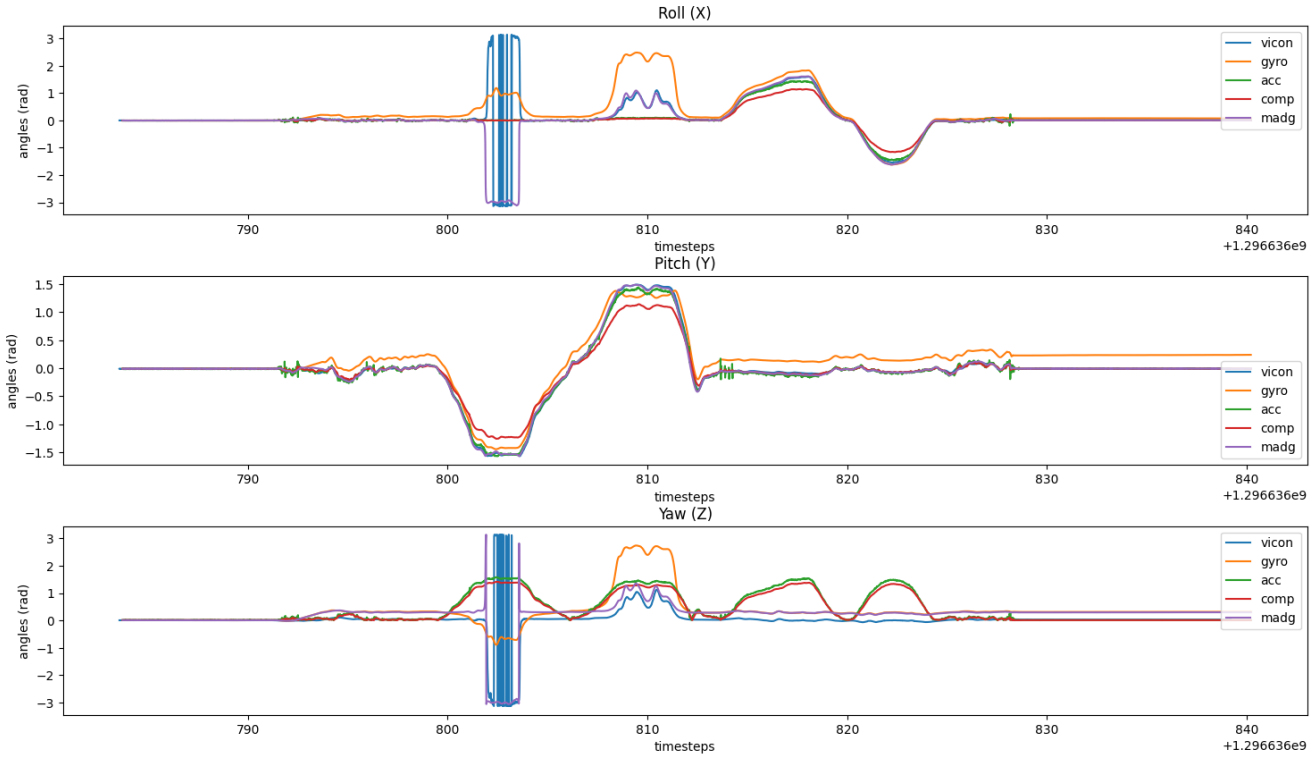

Fig. 11. Euler angle plots (acc,gyro,comp,madg) for train data 1.

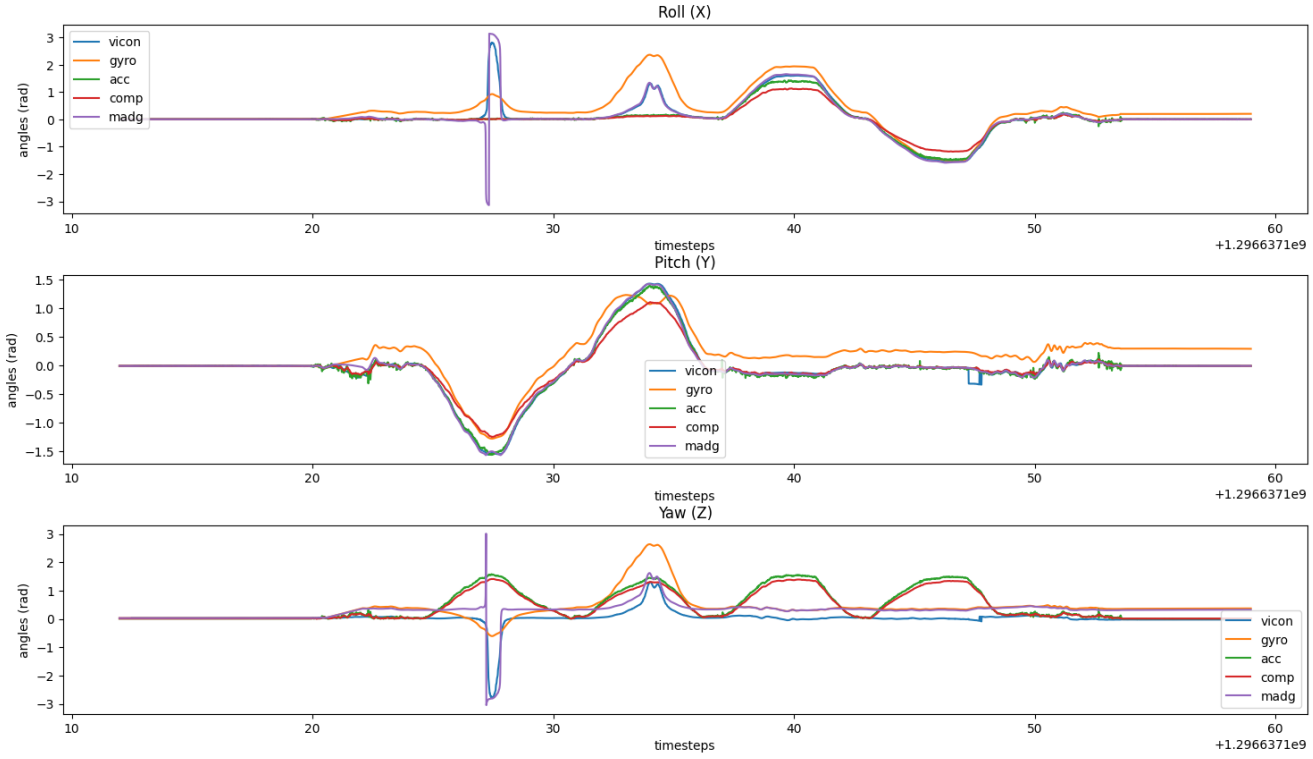

Fig. 12. Euler angle plots (acc,gyro,comp,madg) for train data 2.

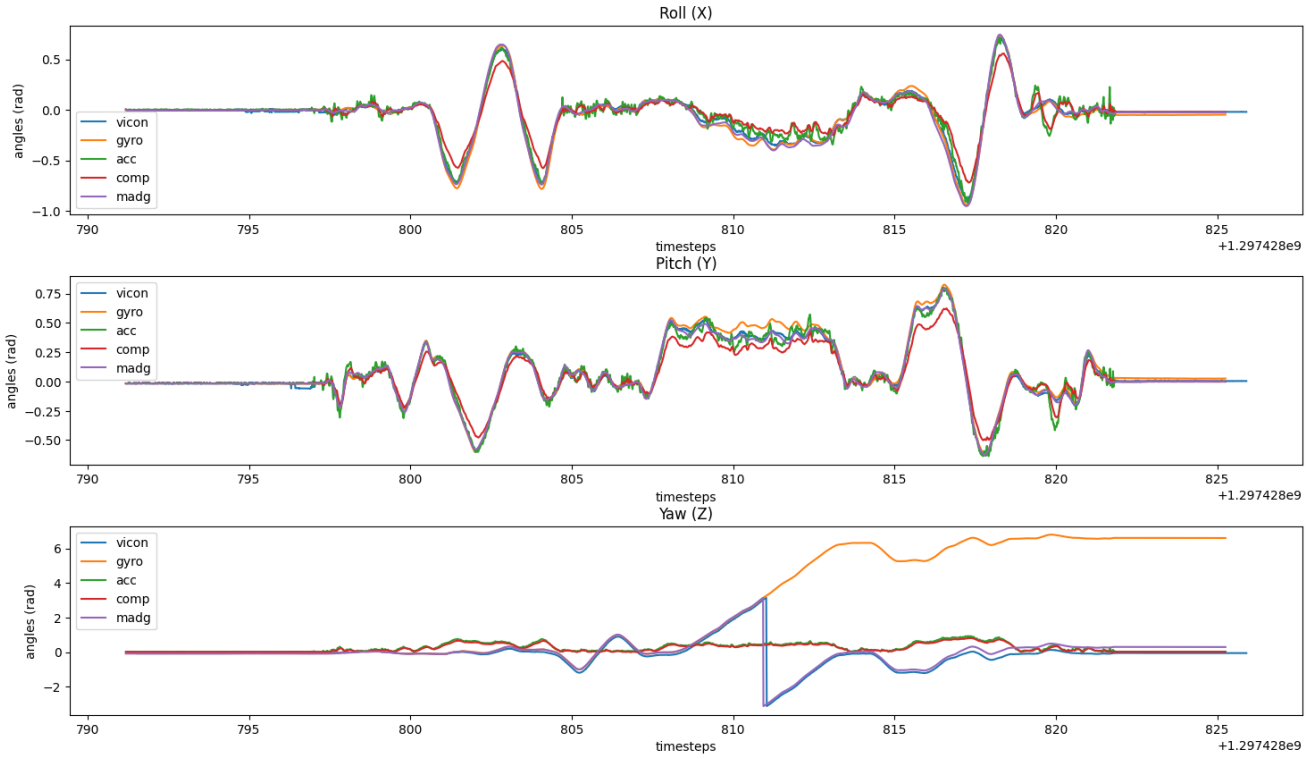

Fig. 13. Euler angle plots (acc,gyro,comp,madg) for train data 3.

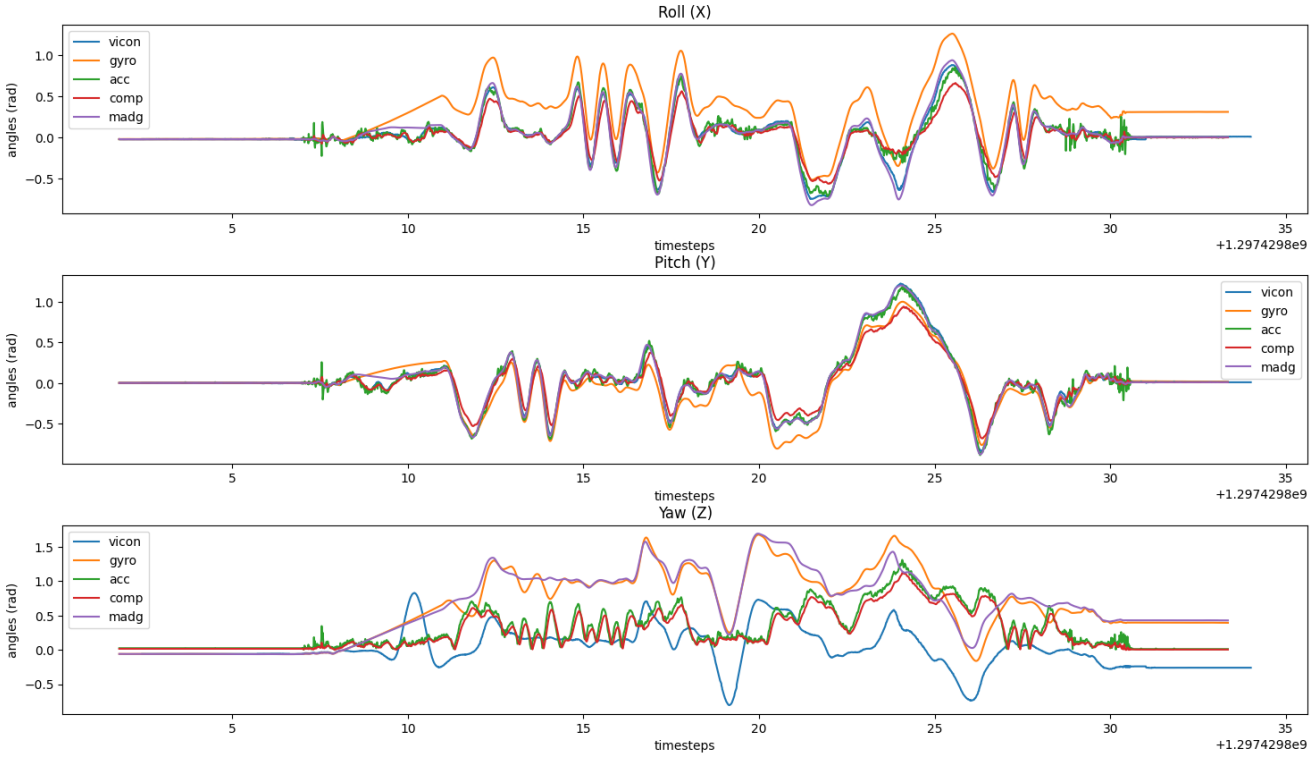

Fig. 14. Euler angle plots (acc,gyro,comp,madg) for train data 4.

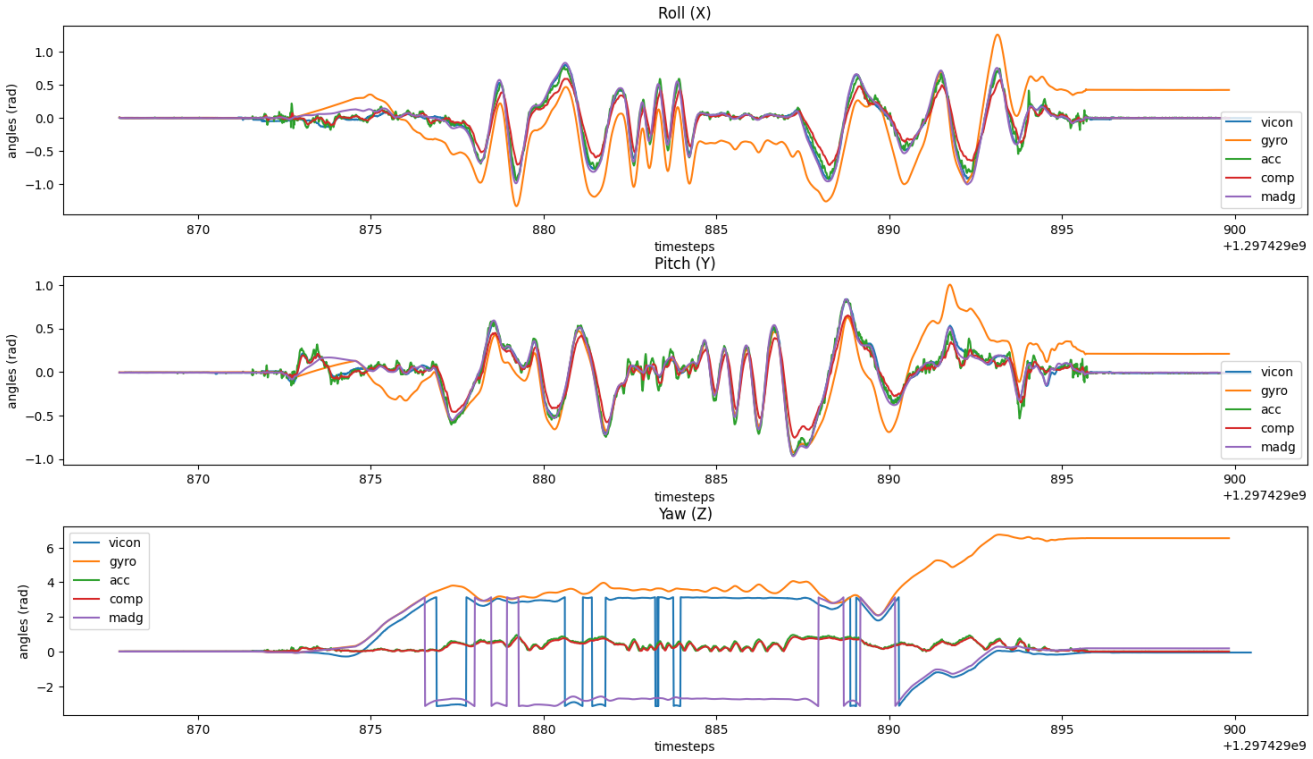

Fig. 15. Euler angle plots (acc,gyro,comp,madg) for train data 5.

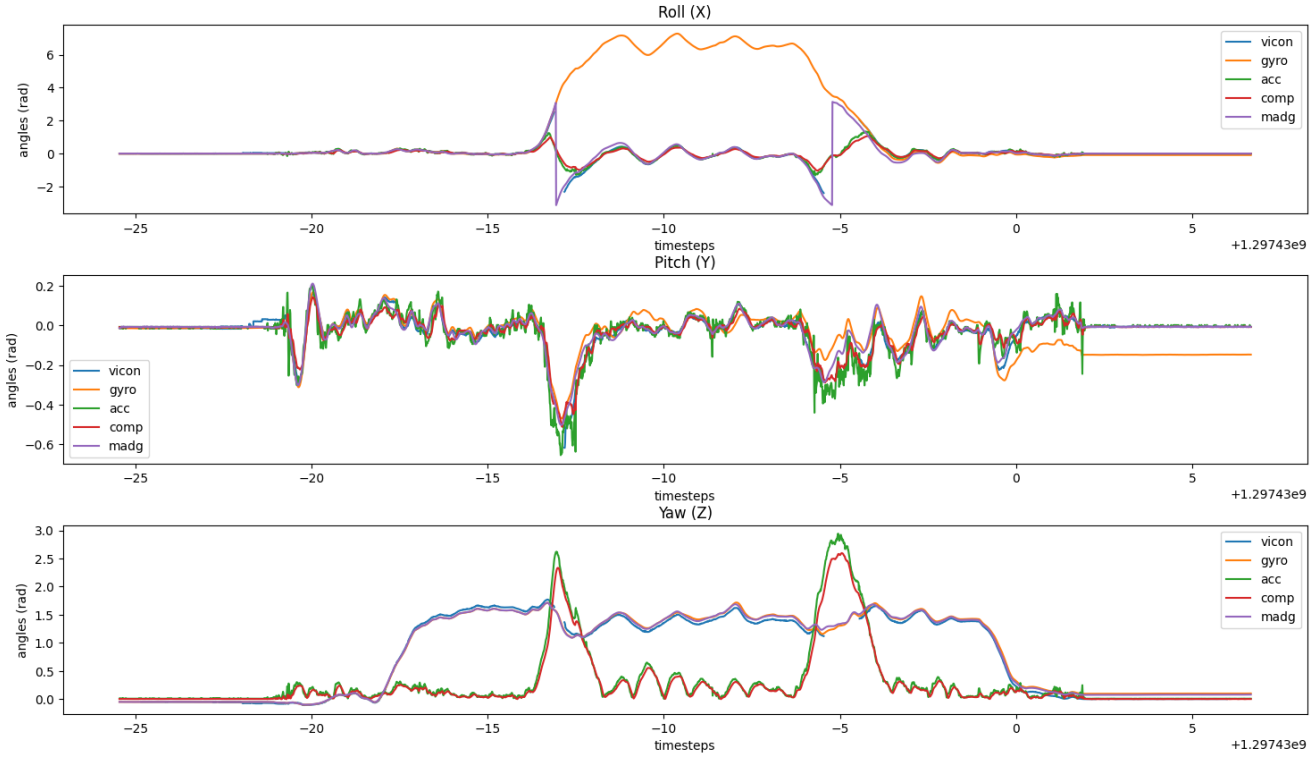

Fig. 16. Euler angle plots (acc,gyro,comp,madg) for train data 6.

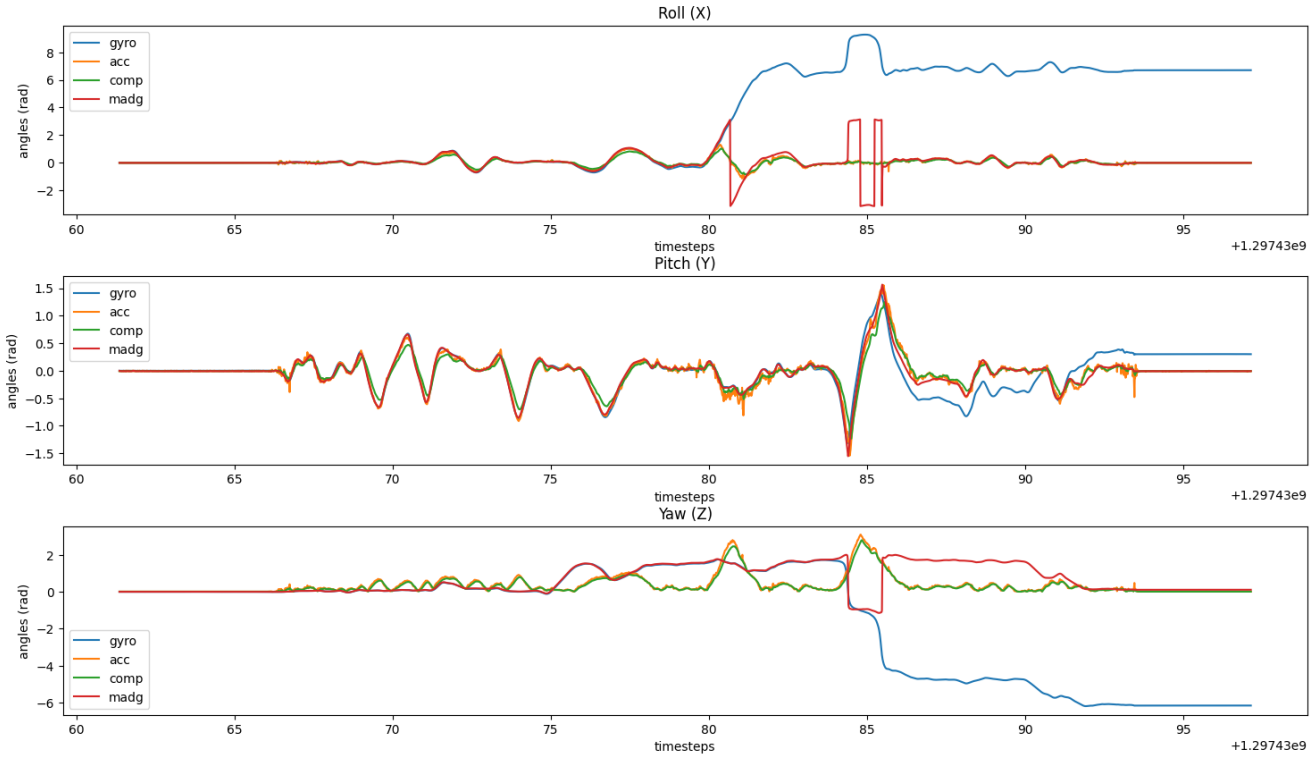

Fig. 17. Euler angle plots (acc,gyro,comp,madg) for test data 7.

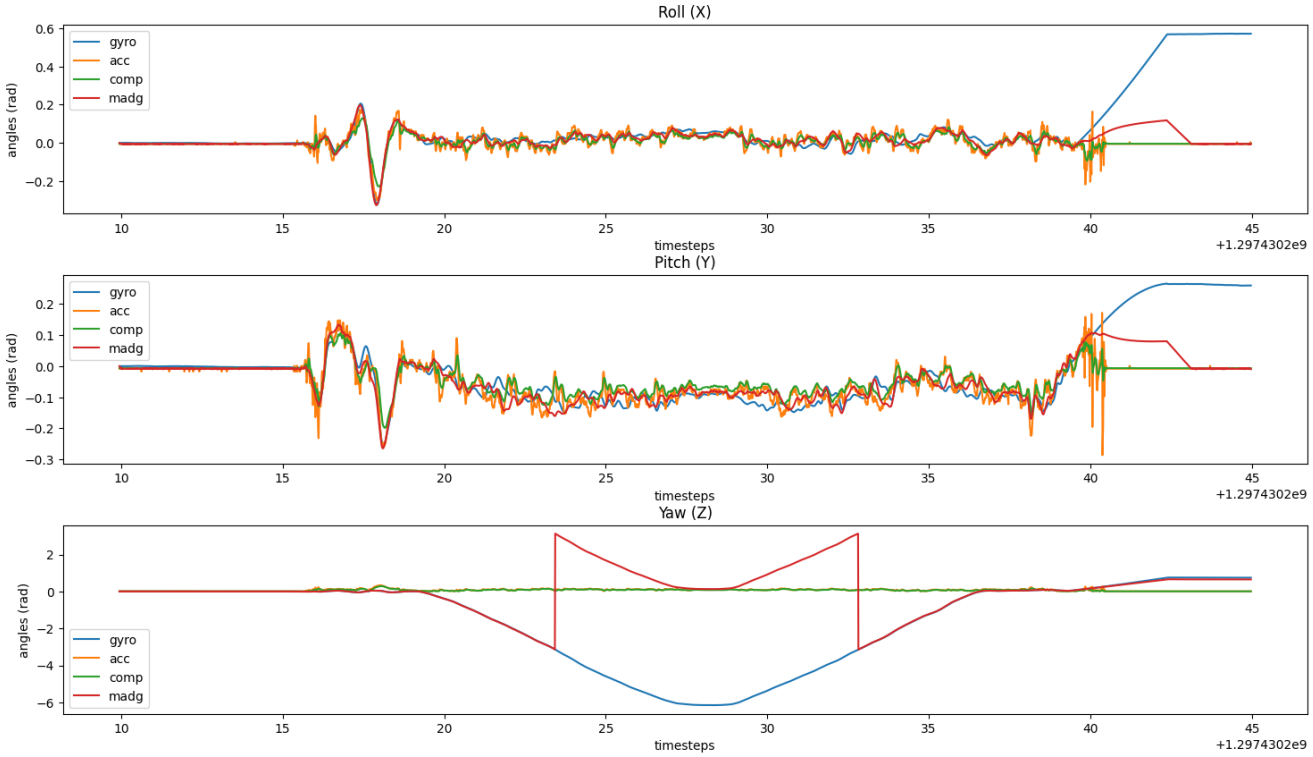

Fig. 18. Euler angle plots (acc,gyro,comp,madg) for test data 8.

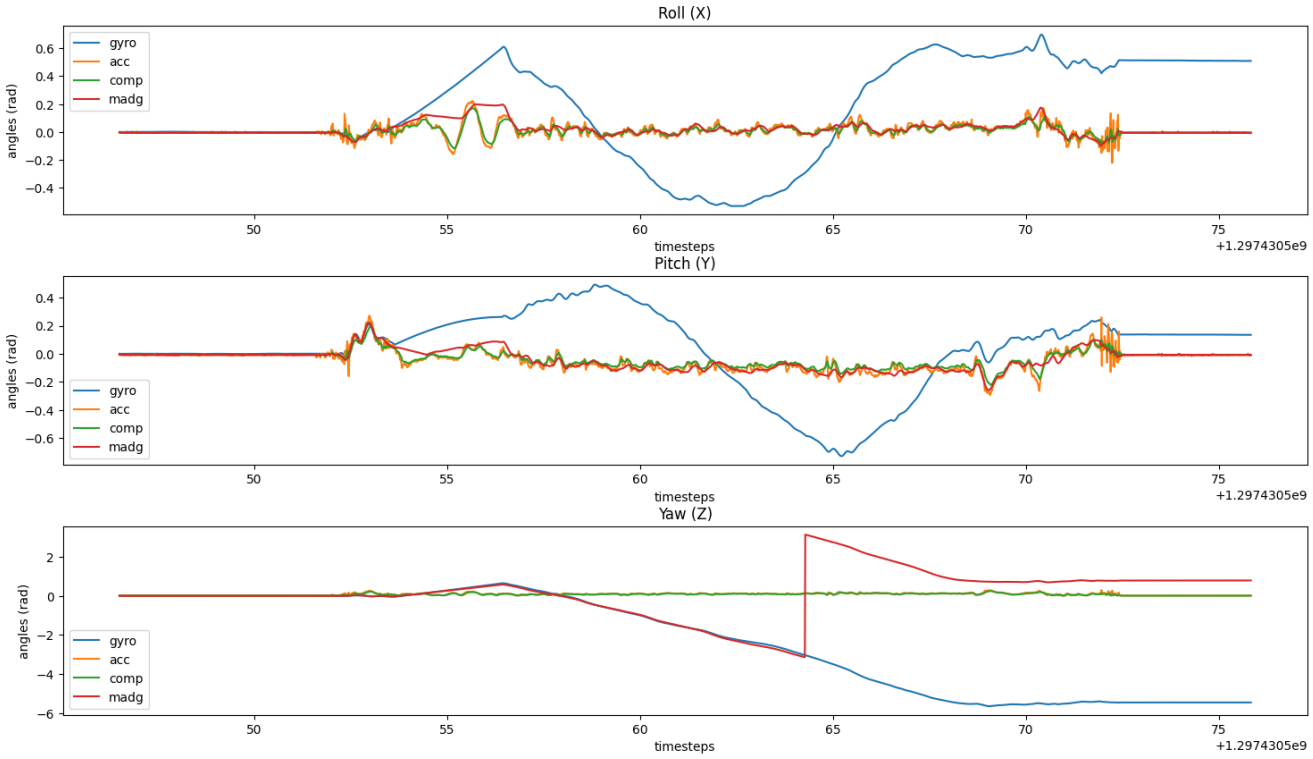

Fig. 19. Euler angle plots (acc,gyro,comp,madg) for test data 9.

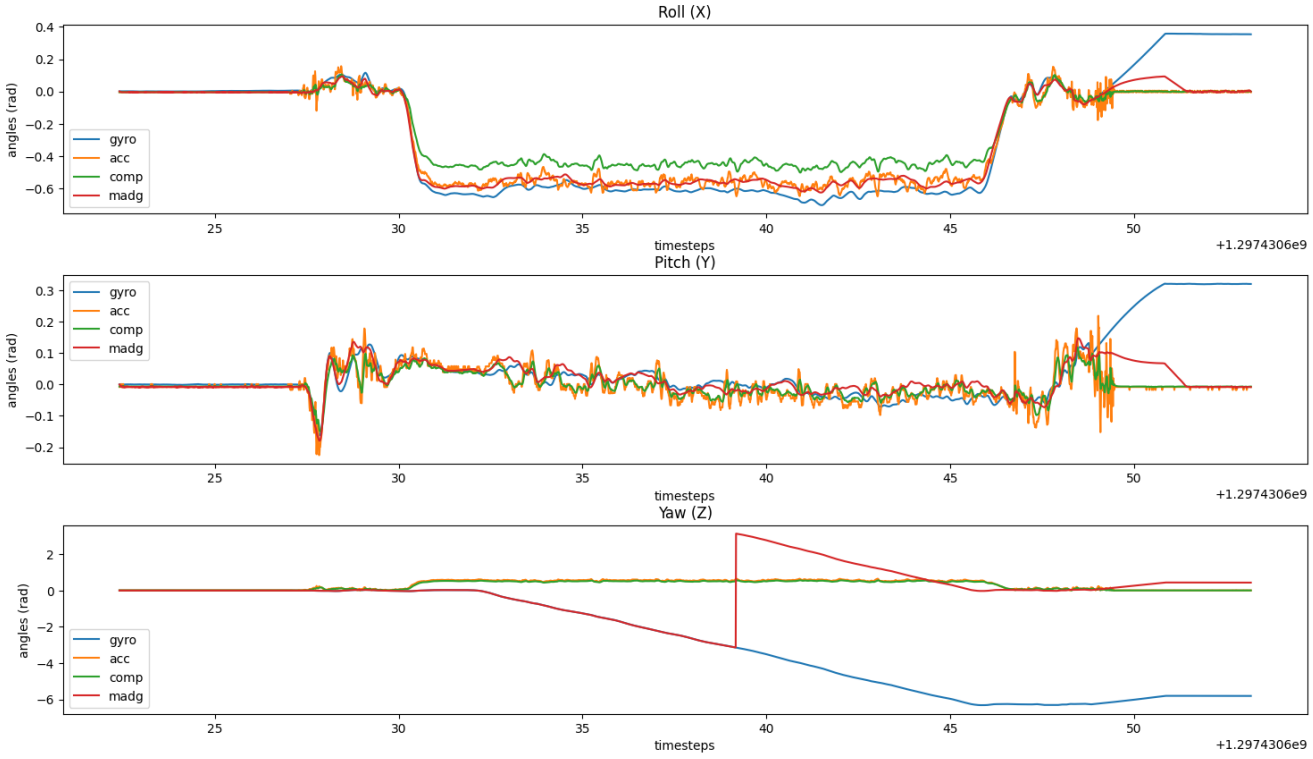

Fig. 20. Euler angle plots (acc,gyro,comp,madg) for test data 10.