Comparing Complementary filter with Madgwick filter for Attitde Estimation

Shakthibala Sivagami Balamurugan
Robotics Engineering
Worcester Polytechnic Institute
Email: sbalamurugan@wpi.edu

Aditya Patwardhan
Robotics Engineering
Worcester Polytechnic Institute
Email: apatwardhan@wpi.edu

***Abstract*—This project develops a per-axis complementary filter to reduce drift errors from both gyroscope and accelerometer measurements. The method was evaluated using a dataset containing 6-DoF IMU signals and Vicon rotation ground truth. Gyroscope integration delivers high-frequency orientation estimates but suffers from drift, while accelerometer-based tilt estimation provides low-frequency, drift-free corrections. By combining the two, a fusion parameter of $\alpha = 0.98$ produces smooth and accurate orientation trajectories—where the gyroscope captures short-term dynamics and the accelerometer ensures long-term stability. A comparison with the Madgwick filter indicates that, at $\beta = 0.06$, its performance surpasses that of the complementary filter.**

I. Introduction

In many robotics and navigation systems, accurate estimation of 3D orientation from IMU data is essential. Gyroscopes provide good short-term angular velocity measurements but drift over time due to integration errors. Accelerometers provide an absolute measure of the gravity vector but are noisy and affected by linear acceleration. To address these limitations, sensor fusion methods are employed. A complementary filter offers a simple yet effective solution by blending fast dynamics from the gyroscope with drift-free tilt from the accelerometer. In addition, more advanced nonlinear filters such as the Madgwick filter leverage a gradient-descent optimization approach to fuse gyroscope, accelerometer, and optionally magnetometer data, yielding accurate and robust orientation estimates even under dynamic motion. This study investigates both the complementary filter and the Madgwick filter, comparing their performance against ground-truth measurements to highlight their respective strengths in practical robotics applications.

II. Dataset Description

The dataset consists of MATLAB `.mat` files organized into two subfolders: `Phase1/Data/Train/IMU` and `Phase1/Data/Train/Vicon`. Each IMU file corresponds to a Vicon file with the same numeric suffix, ensuring synchronization between the two datasets. For example, the file `imuRaw1.mat` directly matches with `viconRot1.mat`, meaning they represent the same recording session from both sensors. Each IMU `.mat` file contains two essential variables: `vals` and `ts`. The variable $\texttt{vals} \in R^{6\times N}$ represents the raw IMU sensor readings, arranged in the following order:

$$\texttt{vals} = a_x a_y a_z \omega_z \omega_x \omega_y^T.$$

Here, a_x, a_y, and a_z correspond to the raw accelerometer measurements along the x, y, and z axes, respectively, while ω_x, ω_y, and ω_z represent the raw gyroscope readings about the corresponding axes. The second variable, $\texttt{ts} \in R^N$, contains the timestamps in seconds, defining the sampling instances for each recorded IMU measurement.

The Vicon `.mat` files, on the other hand, provide ground-truth orientation data for the same recording session. These files contain two key variables: $\texttt{rots} \in R^{3\times3\times N}$, which represent rotation matrices following the Z–Y–X Euler angle convention, and $\texttt{ts} \in R^N$, which holds the Vicon timestamps synchronized to its data stream. Together, these IMU and Vicon datasets allow precise comparison between estimated orientation and ground-truth orientation for further analysis and sensor fusion.

III. Bias Estimation and Calibration

Calibration is essential to convert raw IMU counts into physical units. For the accelerometer, scale factors and biases are stored in `IMUParams.mat`. The first row contains the scale vector

$$\mathbf{s} = [s_x,\, s_y,\, s_z],$$

and the second row contains the bias vector

$$\mathbf{b}_a = [b_{a,x},\, b_{a,y},\, b_{a,z}].$$

The calibrated accelerations are computed as:

$$\tilde{a}_x = (a_x s_x + b_{a,x}) \cdot 9.81 \tag{1}$$

$$\tilde{a}_y = (a_y s_y + b_{a,y}) \cdot 9.81 \tag{2}$$

$$\tilde{a}_z = (a_z s_z + b_{a,z}) \cdot 9.81 \tag{3}$$

ensuring that accelerations are expressed in m/s^2.

For the gyroscope, the bias b_g is estimated under the assumption that the IMU is stationary during the initial samples. The bias is obtained by averaging the first n measurements:

$$b_g = \frac{1}{n}\sum_{i=1}^{n} \omega_i.$$

The calibrated gyroscope readings are then converted into angular velocities (in rad/s) using the scale relationship:

$$\tilde{\omega} = \frac{3300}{1023} \cdot \frac{\pi}{180} \cdot 0.3 \cdot (\omega - b_g)\,.$$

This calibration ensures that both accelerometer and gyroscope data are physically meaningful for orientation estimation.

IV. Orientation Estimation

A. Gyroscope-Only Orientation

Gyroscope-based orientation is estimated by integrating angular velocity over time. In the per-axis complementary filter implementation, however, gyroscope integration was simplified to Euler angles. For example, the roll update is computed as

$$roll_{k+1} = roll_k + \omega_x[k+1]\,\Delta t_k,$$

with analogous expressions for pitch and yaw. This small-angle approximation is computationally simple but can introduce errors for large rotations or coupled axis motions. As shown in Fig.2, The gyroscope-only trajectory gradually diverges from the ground truth due to drift..

B. Accelerometer-Only Orientation

The accelerometer provides the direction of gravity, allowing estimation of roll and pitch. By aligning the measured acceleration vector with the known gravity vector

$$g = [0,\ 0,\ -1]^T,$$

tilt orientation is computed. Since yaw is unobservable from accelerometer data, it is adopted from the gyroscope estimate. The accelerometer-based orientation , Fig.3 tracks tilt well under static conditions but becomes highly noisy and unreliable during dynamic motion.

C. Complementary Filter Fusion

The complementary filter fuses both estimates by applying low-pass characteristics to accelerometer tilt and high-pass characteristics to gyroscope integration. In this implementation, fusion is performed in a per-axis fashion using Euler angles. The update rule is:

$$\theta_{fused}[k+1] = \alpha \cdot \theta_{gyro}[k+1] + (1-\alpha) \cdot \theta_{acc}[k+1],$$

where $\theta \in \{roll, pitch, yaw\}$, θ_{gyro} is the orientation propagated by gyroscope integration, and θ_{acc} is the accelerometer tilt estimate. Here, α is the fusion factor that weights the gyroscope contribution. For example, $\alpha = 0.98$ gives 98% weight to the gyroscope (capturing short-term dynamics) and 2% weight to the accelerometer (providing long-term drift correction).

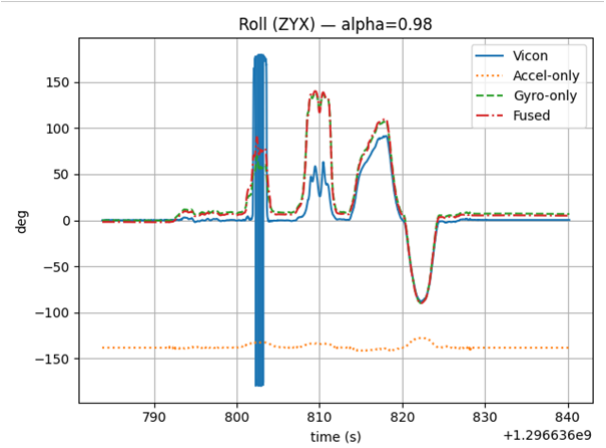

Fig. 1. Benchmarking the results

V. Software Time Synchronization

Since the IMU and Vicon operate at different sampling rates, software synchronization was necessary. For each IMU timestamp $t_{IMU}[i]$, the nearest Vicon timestamp was selected as:

$$j = \arg\min_k \left|t_{Vicon}[k] - t_{IMU}[i]\right|.$$

This was efficiently implemented using binary search with `np.searchsorted`, ensuring computational efficiency even for large datasets. The matched indices aligned the Vicon orientation sequence with the IMU timeline, enabling direct comparison. The Vicon system output, Fig.5 is used as the reference for evaluating all IMU-based orientation estimates. This synchronization ensured that orientation estimates derived from IMU data were evaluated against the temporally closest Vicon ground-truth measurements.

VI. Gimbal lock

This is a common phenomenon when working with Euler angles. As shown in Fig. 1, a discontinuity is observed in the roll trajectory, which arises due to gimbal lock in the Z–Y–X Euler representation. Gimbal lock occurs when the pitch angle approaches ±90°, causing two rotational axes to align and leading to a loss of one degree of freedom. In such a configuration, multiple physical orientations may map to the same Euler angle representation, and small smooth changes in the underlying orientation can result in abrupt jumps in the Euler angle plots. It is important to note that this discontinuity is not a failure of the complementary filter itself, nor an error in the underlying rotation estimate, but rather an artifact of the Euler angle parameterization.

VII. Madgwick Filter

The complementary filter addresses the shortcomings of both the accelerometer and gyroscope by linearly blending their outputs. However, accelerometer-only orientation estimates are often unreliable due to errors in vector decomposition, leading to poor alignment with the ground truth. Moreover, previous implementations based on Euler angles suffer

from singularities such as gimbal lock at certain orientations. To overcome these issues and obtain more robust orientation estimates, the Madgwick filter is employed. This filter makes use of quaternions to represent attitude, thereby avoiding singularities. For clarity, the scalar part of the quaternion is assumed to be the first element. In this work, the initial quaternion estimate is obtained from the average of the first 200 Vicon measurements in the training set, while for the test set it is initialized to zero. The filter implementation can be divided into three steps:

A. Accelerometer based update

The accelerometer is used to provide an absolute reference for orientation by formulating the problem as an optimization task. Specifically, a gradient-descent algorithm is applied to minimize the discrepancy between the measured acceleration vector and the gravity vector predicted by the current quaternion estimate. The objective function can be written as:

$$\hat{q} = \arg\min_{\hat{q}} f(\hat{q}, \hat{g}, \hat{a}), \tag{4}$$

where

$$f(\hat{q}, \hat{g}, \hat{a}) = \hat{q}^* \otimes \hat{g} \otimes \hat{q} - \hat{a}. \tag{5}$$

Here, $\hat{q}$ is the quaternion estimate, $\hat{q}^*$ is its conjugate, $\hat{g} = [0,\ 0,\ 0,\ 1]^T$ represents the gravity vector, and $\hat{a}$ denotes the accelerometer measurement.

The gradient of this function is given by:

$$\nabla f(\hat{q}_t, \hat{g}, \hat{a}_{t+1}) = J^T(\hat{q}_t, \hat{g}) f(\hat{q}_t, \hat{g}, \hat{a}_{t+1}), \tag{6}$$

where

$$f(\hat{q}_t, \hat{g}, \hat{a}_{t+1}) = \begin{bmatrix} 2(q_2q_4 - q_1q_3) - a_x \\ 2(q_1q_2 + q_3q_4) - a_y \\ 2(0.5 - q_2^2 - q_3^2) - a_z \end{bmatrix} \tag{7}$$

The Jacobian matrix is defined as:

$$J(\hat{q}_t, \hat{g}) = \begin{bmatrix} -2q_3 & 2q_4 & -2q_1 & 2q_2 \\ 2q_2 & 2q_1 & 2q_4 & 2q_3 \\ 0 & -4q_2 & -4q_3 & 0 \end{bmatrix}. \tag{8}$$

The correction term for the gradient descent step is:

$$q_{\nabla,t+1} = -\beta \frac{\nabla f(\hat{q}_t, \hat{g}, \hat{a}_{t+1})}{\|\nabla f(\hat{q}_t, \hat{g}, \hat{a}_{t+1})\|}, \tag{9}$$

where β is a tunable gain parameter, set to $\beta = 0.1$ for this project.

B. Gyroscope based update

The gyroscope provides angular velocity, which is used to compute the derivative of the quaternion through:

$$\dot{q}_{\omega,t+1} = \frac{1}{2}\hat{q}_t \otimes \begin{bmatrix} 0, \omega_{x,t+1} \end{bmatrix}^T \tag{10}$$

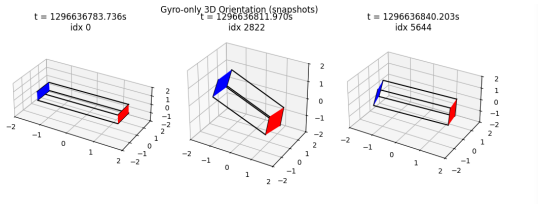

Fig. 2. 3D Orientation of Standalone Gyroscope

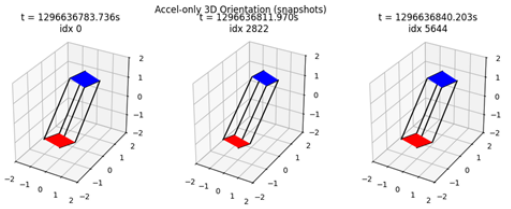

Fig. 3. 3D Orientation of Standalone Accelerometer

C. Sensor Fusion

Finally, the accelerometer correction and gyroscope integration are combined as

$$\dot{q}_{est,t+1} = \dot{q}_{\omega,t+1} + q_{\nabla,t+1}, \tag{11}$$

$$\hat{q}_{t+1} = \hat{q}_t + \dot{q}_{est,t+1}\,\Delta t, \tag{12}$$

where Δt is the time step between updates. Since the quaternion may drift away from unit length, normalization is enforced after each update:

$$\hat{q}_{t+1} \leftarrow \frac{\hat{q}_{t+1}}{\|\hat{q}_{t+1}\|}. \tag{13}$$

As shown in Fig.6, The 3D orientation with Madgwick filter is better than Complementary filter.

VIII. Results and Analysis

The complementary and Madgwick filters were tested on multiple sequences. Gyroscope-only integration produced smooth orientation trajectories but diverged from the Vicon ground truth due to drift. Accelerometer-only estimation provided correct tilt under static conditions but was noisy and unreliable during dynamic motion. The complementary filter successfully fused both signals, generating stable orientation estimates with minimal drift, while the Madgwick filter further improved responsiveness and convergence through its gradient-descent correction scheme. Euler angle plots (yaw, pitch, roll) confirmed that both fusion methods closely tracked Vicon ground truth, with the complementary filter excelling in smoothness and the Madgwick filter providing faster adaptation to rapid motion. The fused estimates, as shown in Fig. 6, demonstrate robust tracking performance with strong agreement to the ground truth across all test cases.

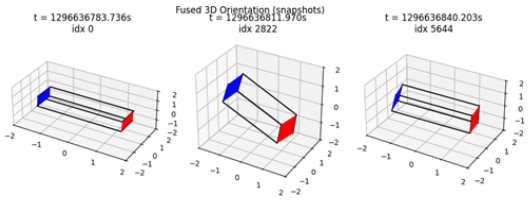

Fig. 4. 3D Orientation using Complementary filter

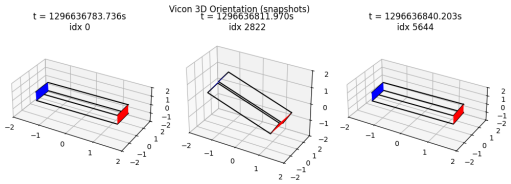

Fig. 5. 3D Orientation using Vicon Motion Capture System

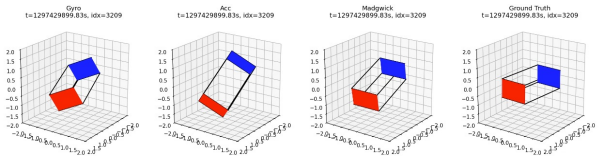

Fig. 6. 3D orientation with Madgwick filter

IX. Conclusion

This study demonstrated the effectiveness of combining a per-axis complementary filter and the Madgwick filter for orientation estimation using IMU data. The complementary filter provided smooth and drift-corrected trajectories by blending gyroscope integration with accelerometer tilt estimates with $\alpha = 0.98$, while the Madgwick filter offered faster convergence and improved robustness in dynamic conditions through gradient-descent-based sensor fusion with $\beta = 0.06$. Together, these approaches produced orientation estimates closely aligned with Vicon ground truth, with the complementary filter excelling in stability and the Madgwick filter in responsiveness.

X. AI Usage

A. Time synchronization

Prompt:I have two dict which contains 'values' and 'timestamps' as key, I want to map them by synchronizing their timestaps in software, Remember not in hardware, Give me a set of ideas arranging them according to their time complexity.

Answer:Got nearest neighbor method as the best fit.

B. Validation of the result

Prompt:I have attached three images, One only with standalone gyroscope, one with stanalone accelerometer, one with complementary filter.

Answer:The graphs are right and the complementary filter tries to align with Vicon data.

XI. Supplementary Material

The videos of the complementary filter along with its comparison against ground truth is given: bit.ly/3HWZjXb

References

[1] Wikipedia, "Slerp," [Online]. [Accessed: Aug. 22, 2025].
[2] Stack Overflow, "How to load mat files in Python and access columns individually," [Online]. [Accessed: Aug. 22, 2025].
[3] PRD UMG Teaching, "Sensor Fusion and IMU Orientation," YouTube playlist, [Online]. [Accessed: Aug. 22, 2025].
[4] N. J. Sanket, "Attitude Estimation from IMU," [Online]. [Accessed: Aug. 23, 2025].
[5] S. O. H. Madgwick, *An efficient orientation filter for inertial and inertial/magnetic sensor arrays*, Technical Report, University of Bristol, UK, April 30, 2010.

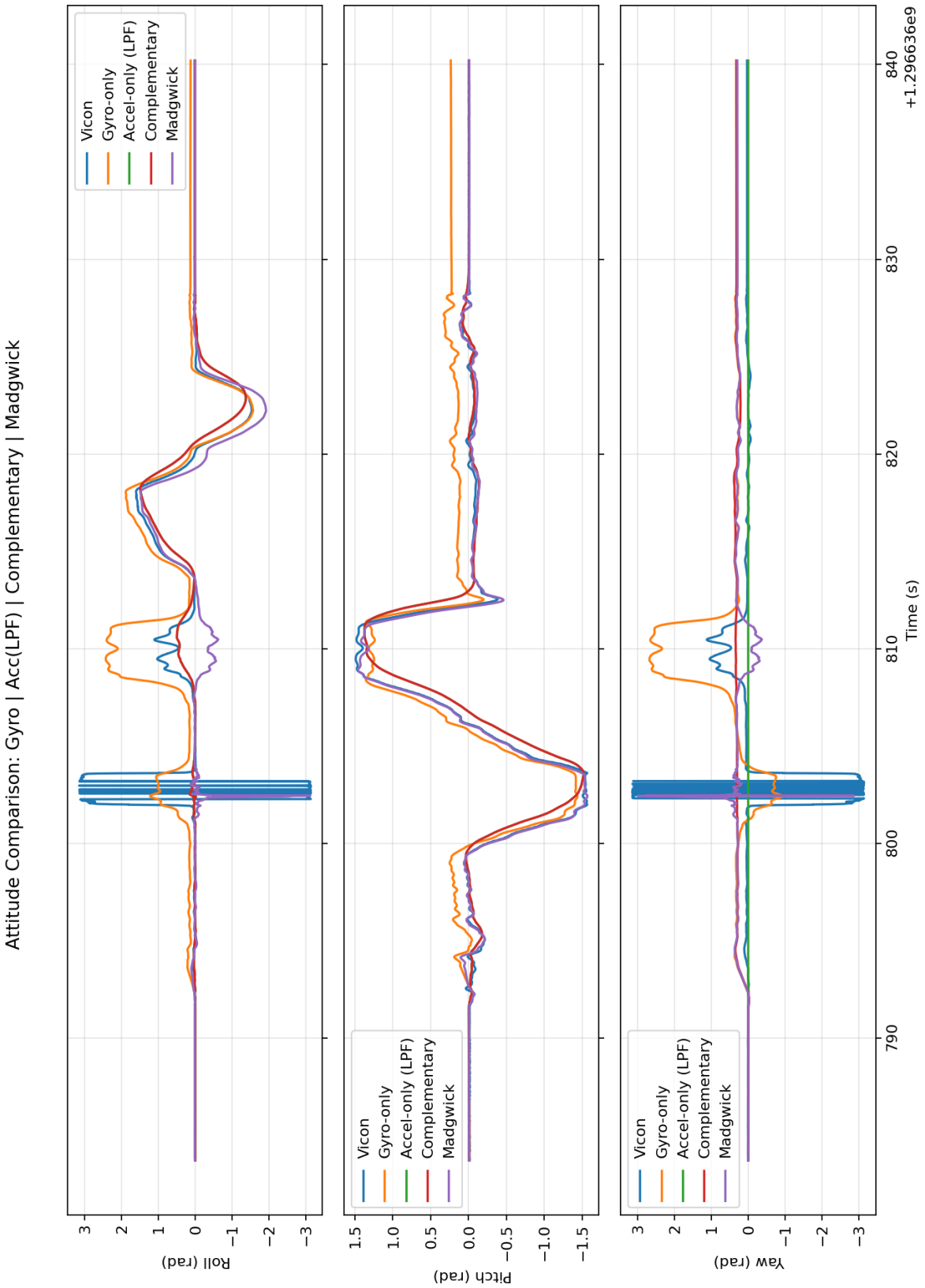

Fig. 7. Attitude estimation for dataset 1

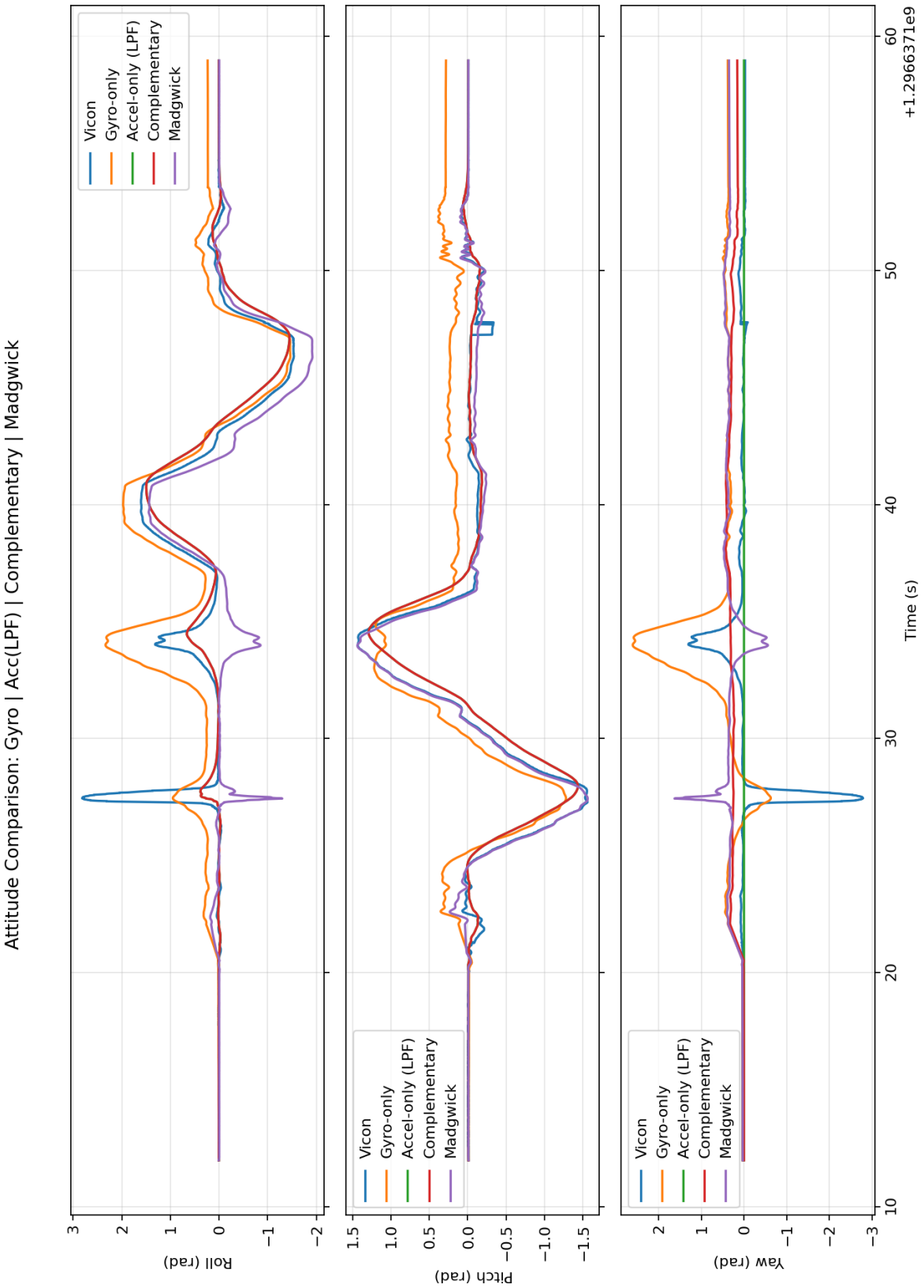

Fig. 8. Attitude estimation for dataset 2

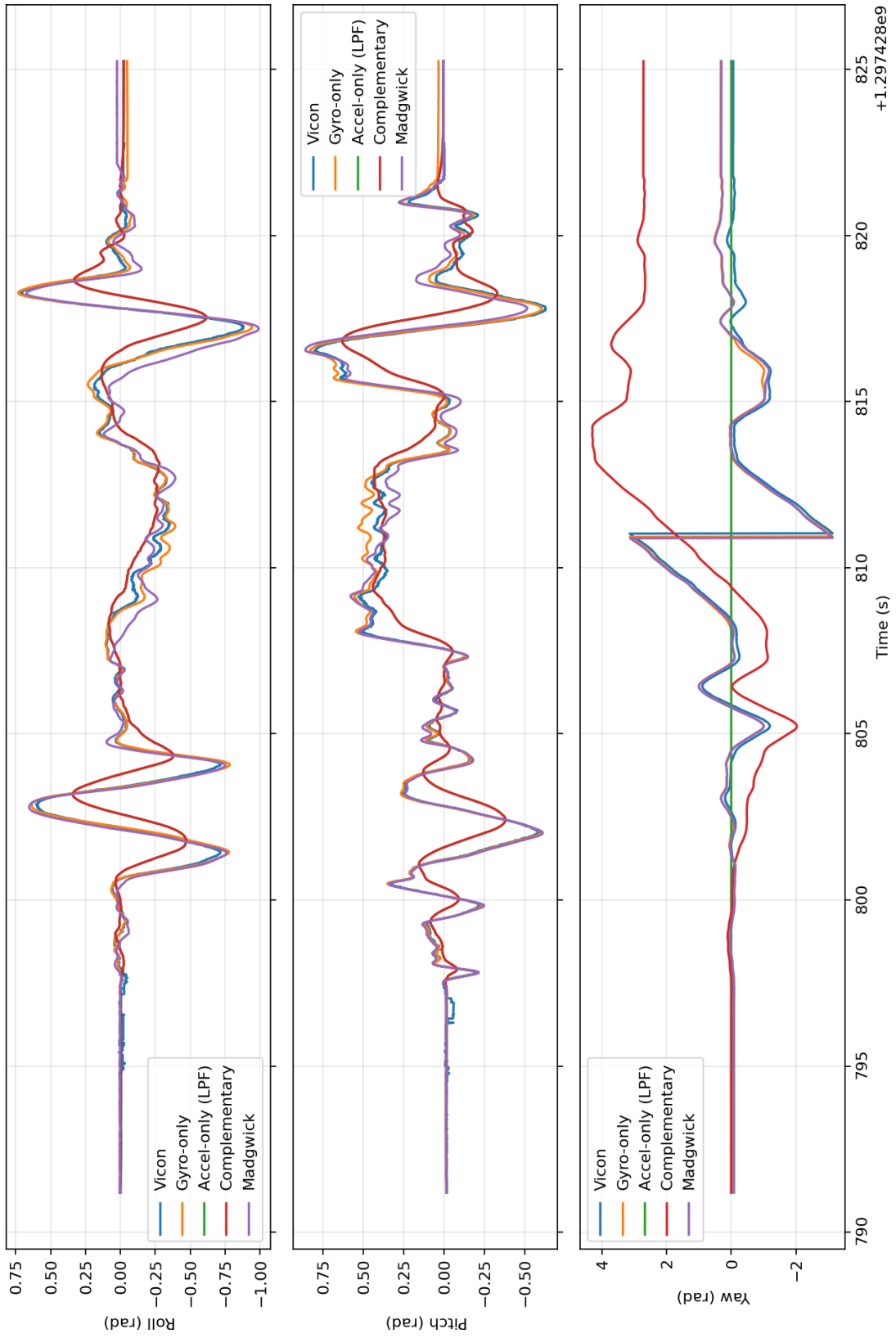

Fig. 9. Attitude estimation for dataset 3

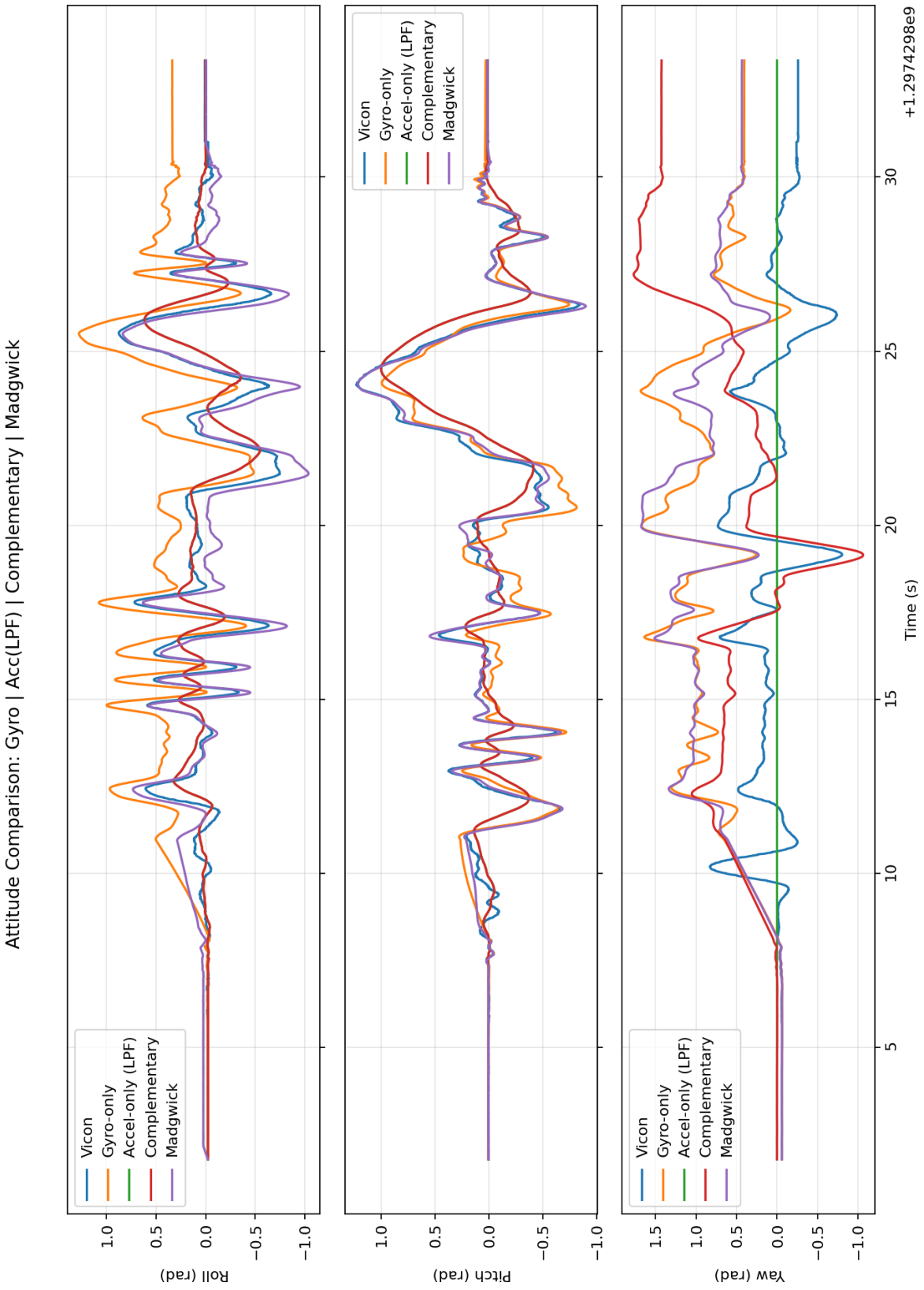

Fig. 10. Attitude estimation for dataset 4

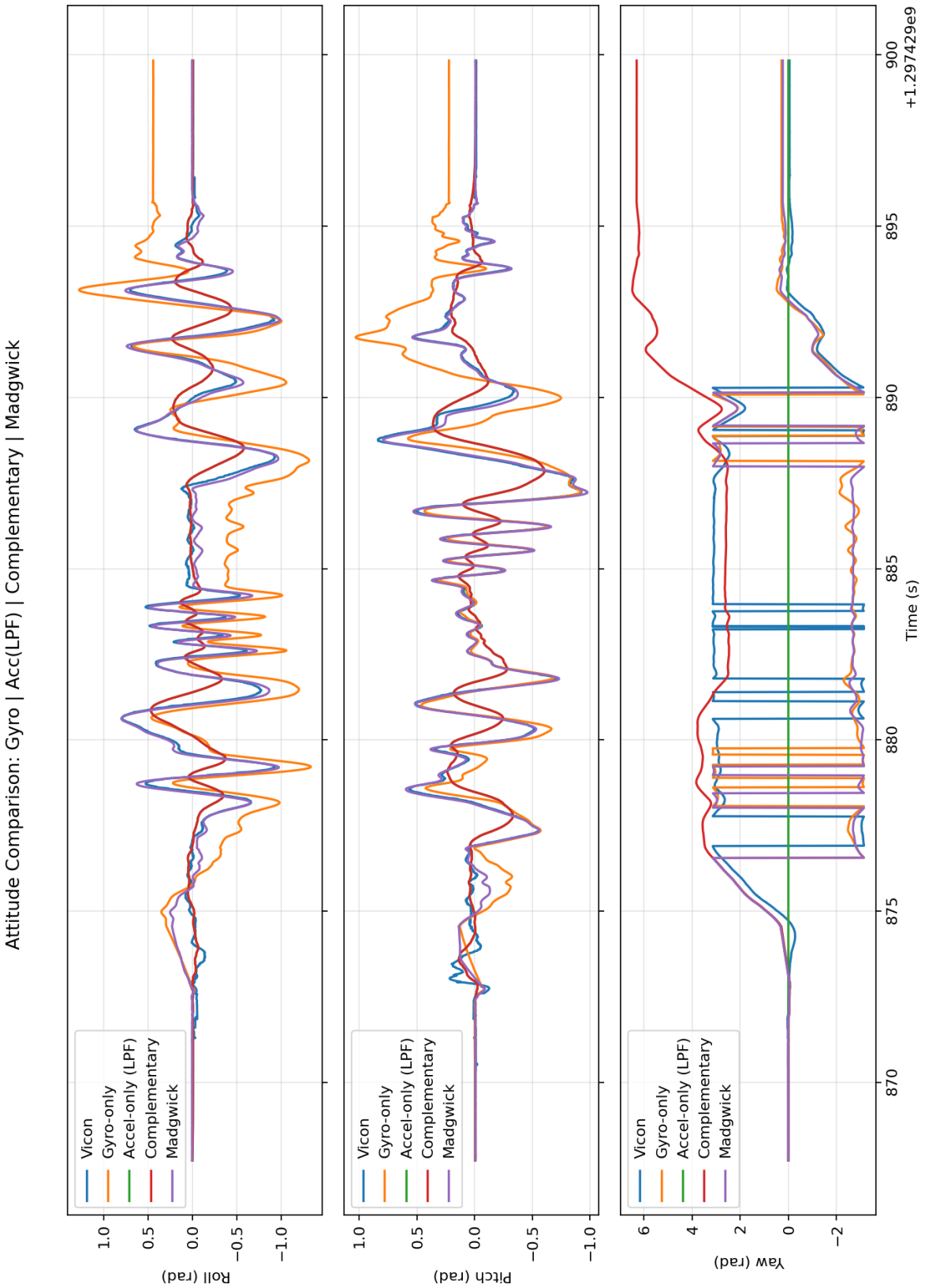

Fig. 11. Attitude estimation for dataset 5

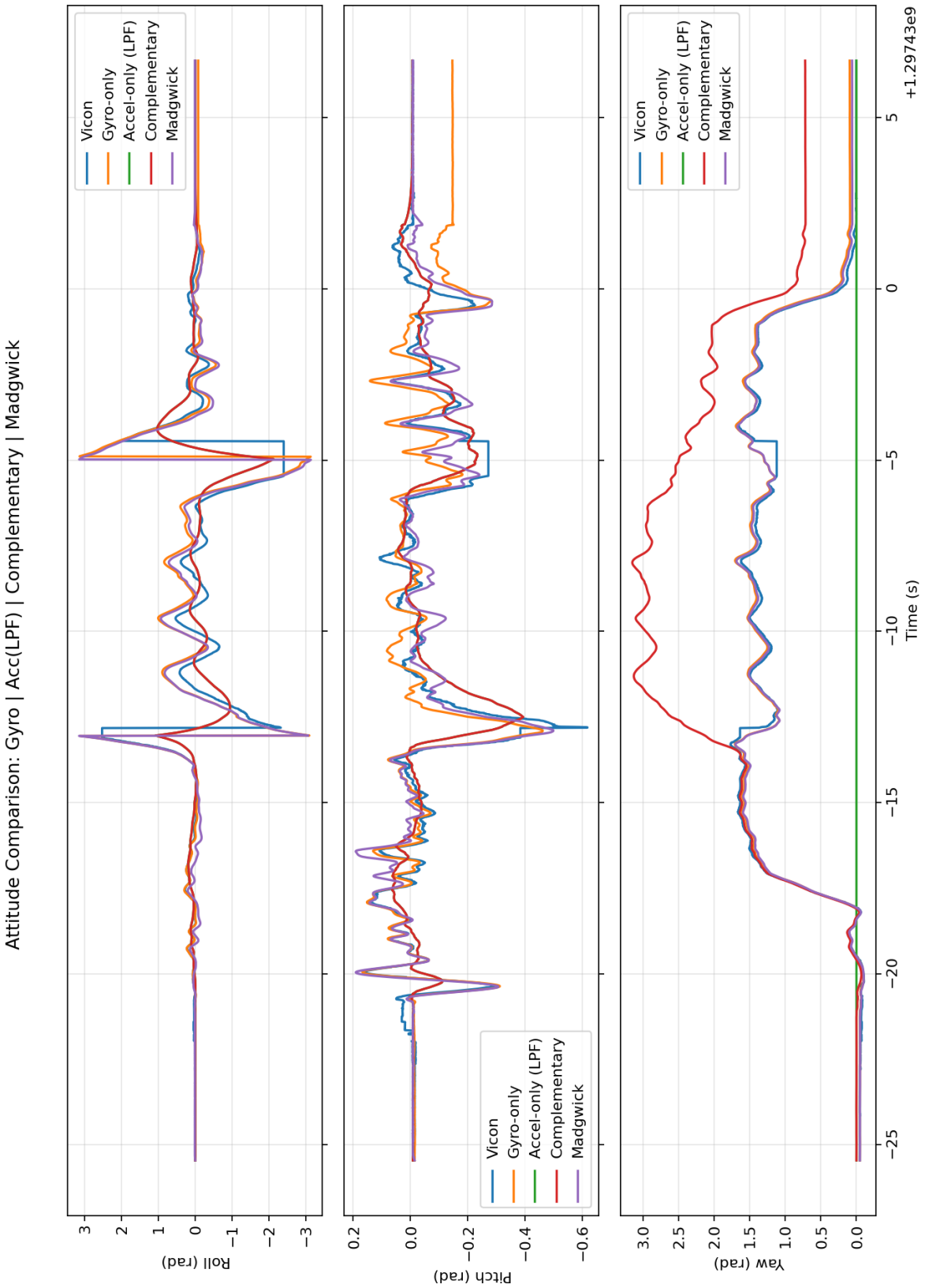

Fig. 12. Attitude estimation for dataset 6

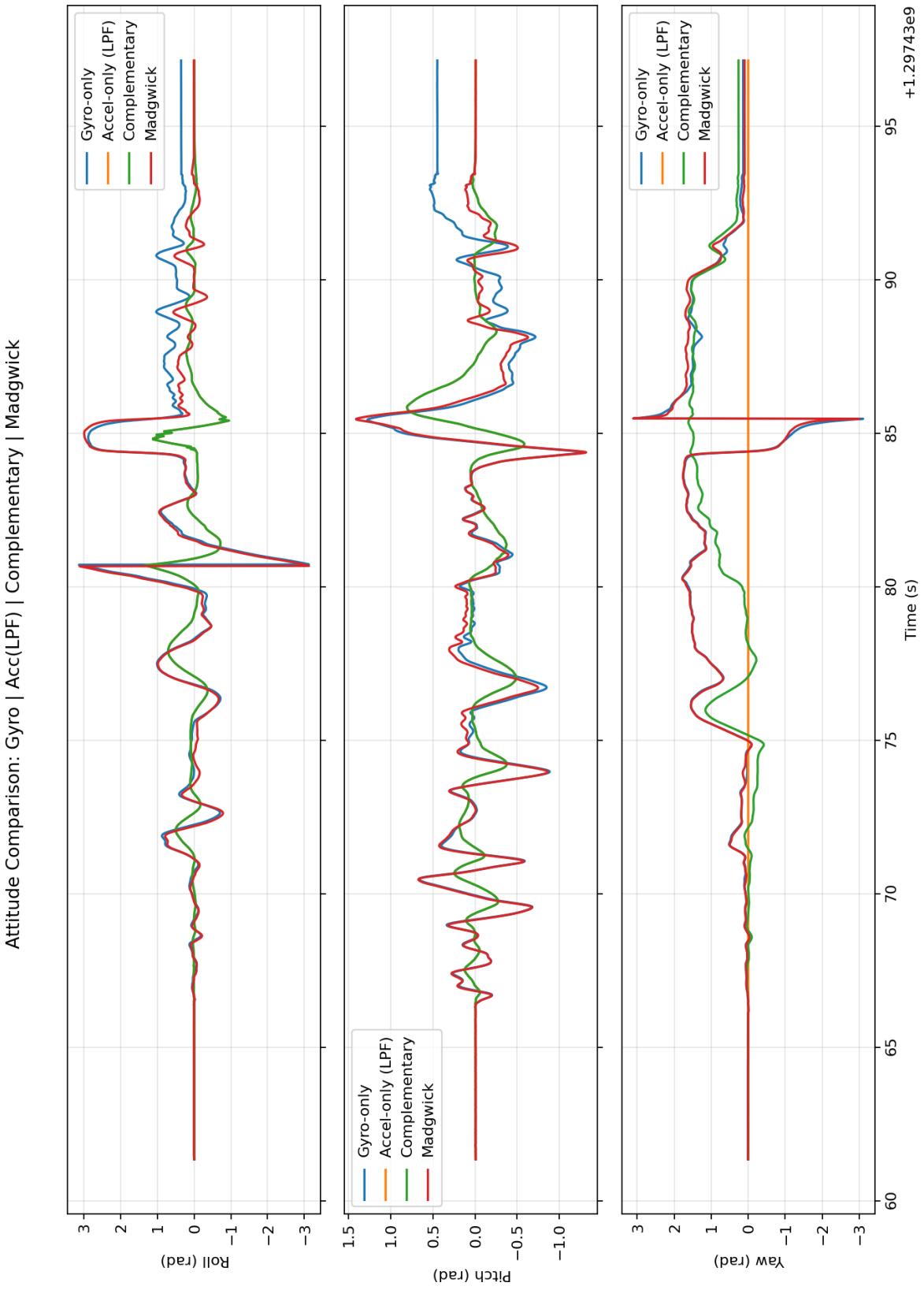

Fig. 13. Attitude estimation for test dataset 7

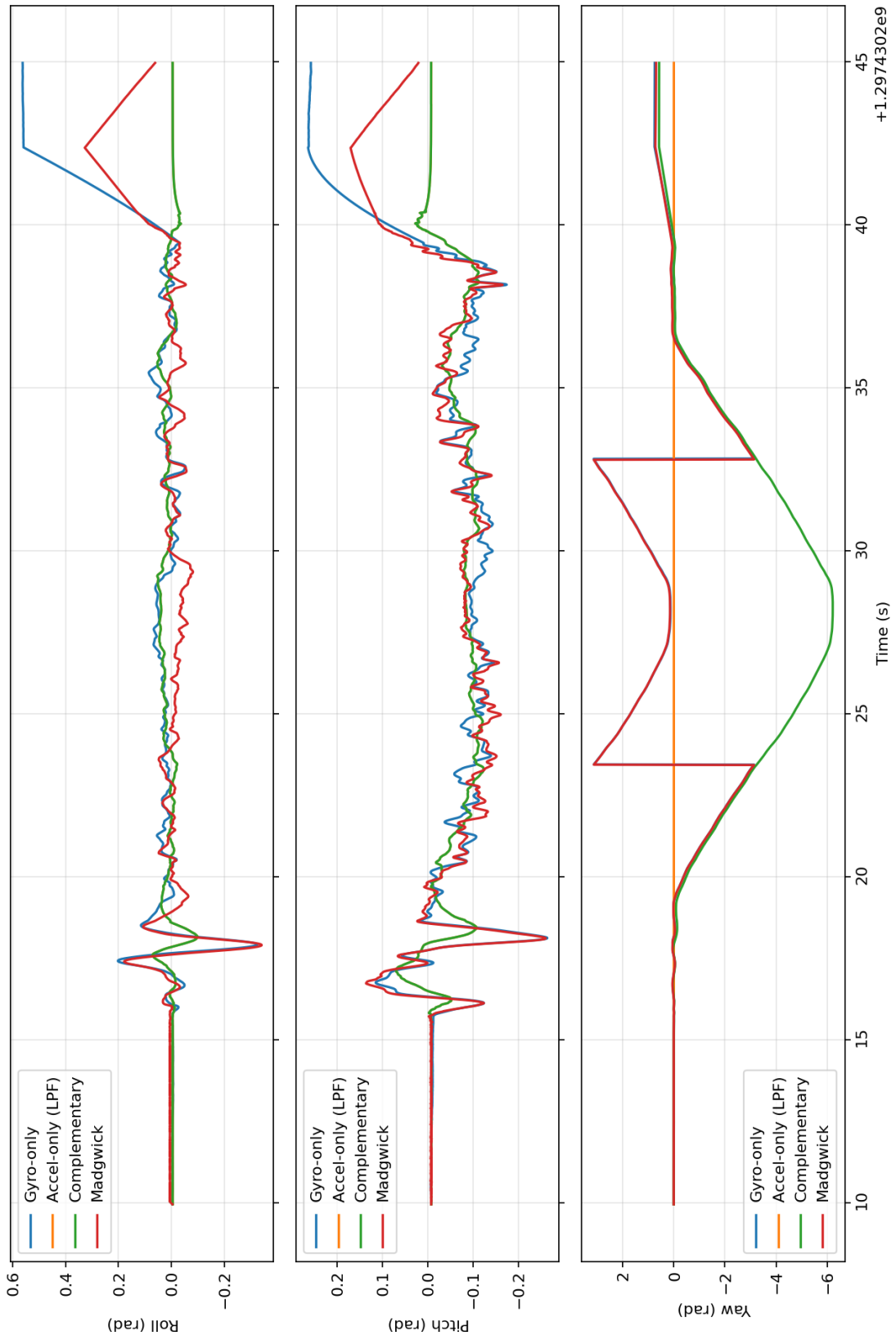

Fig. 14. Attitude estimation for test dataset 8

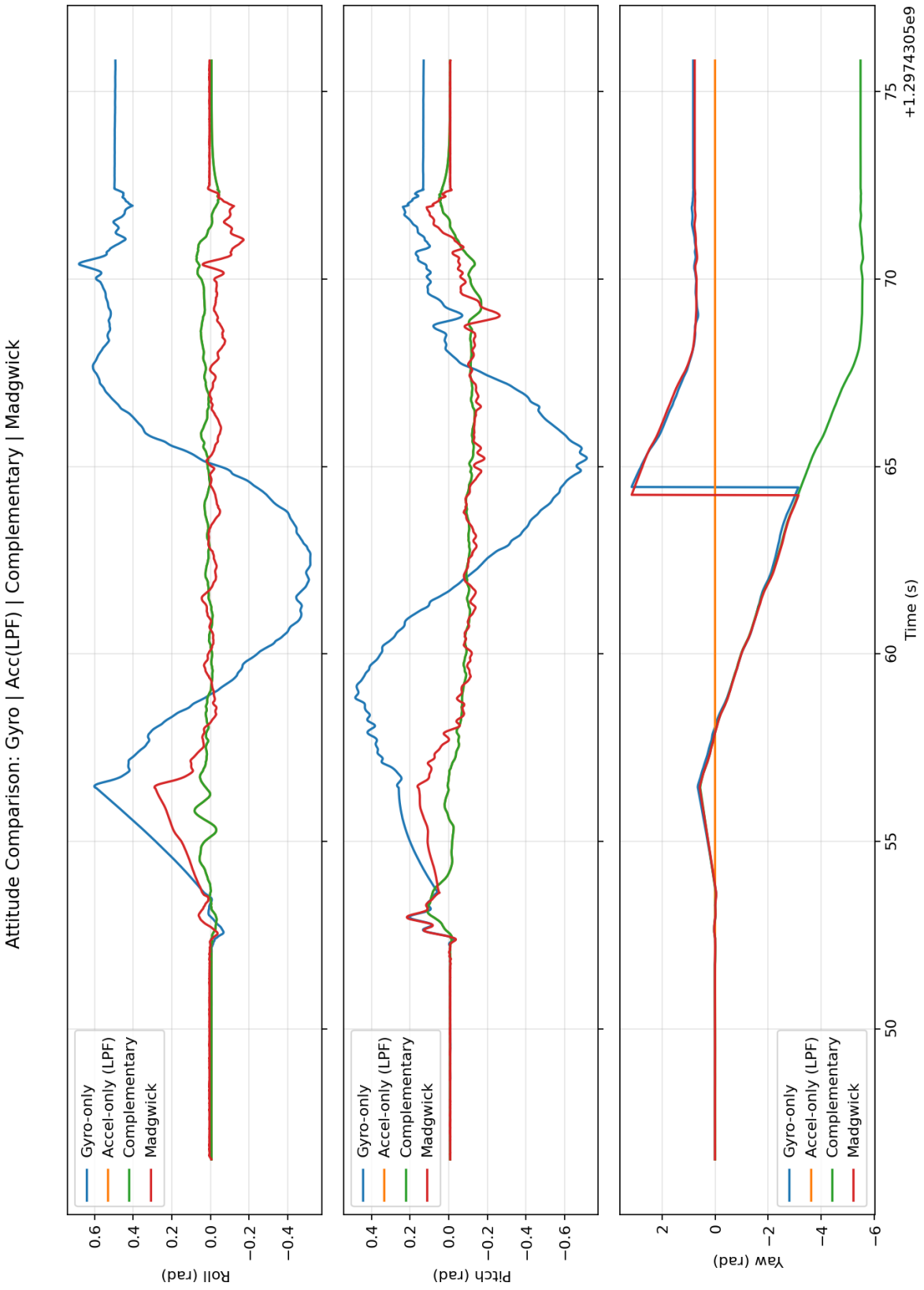

Fig. 15. Attitude estimation for test dataset 9

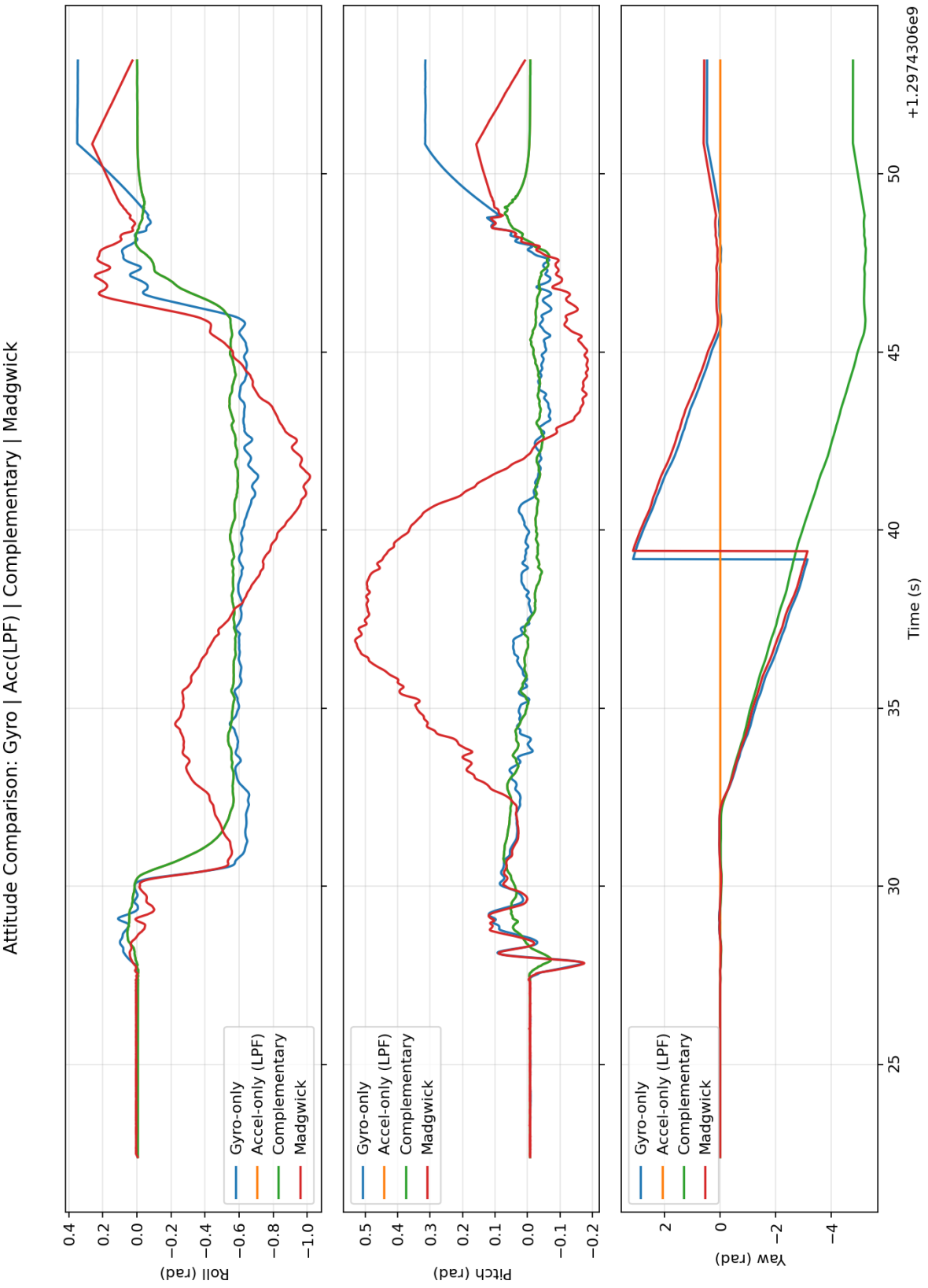

Fig. 16. Attitude estimation for test dataset 10