ENAE788M Project 1a
Team Bouncing Rainbow Zebras

Erik Holum
Graduate Student
University of Maryland
Email: eholum@gmail.com

Edward Carney
Graduate Student
University of Maryland
Email: carneyedwardj@gmail.com

Derek Thompson
Graduate Student
University of Maryland
Email: derekbt@yahoo.com

***Abstract*—This project present results from determining orientation of a platform from gyroscope and accelerometer measurements. Results are presented using pure accelerometer readings, pure gyroscope readings, a complementary filter which combines gyroscopic and accelerometer filters, and an implementation of a Madgwick filter. Results are compared to data captured by a Vicon imaging system, and links to videos animating the orientation through time are provided. The Madgwick filter can be seen to outperform measurements from only accelerometer or gyroscope readings as it most closely follows the Vicon data.**

I. Introduction

The purpose of this project is to estimate the orientation of a platform based on data collected from a mounted 6-Degree of Freedom Inertial Measurement Unit (6 DoF IMU). This was done using four different types of filters, an only linear acceleration filter, an only gyroscopic filter, a complementary filter, and a Madgwick filter. In all of cases, accelerometer bias and scale data were provided; gyroscope bias was determined for each data set based on the first three hundred gyroscope measurements. These parameters allowed conversion of raw data values to SI units. The filters were executed on 5 training sets and 4 previously unseen sets. For the training data sets, corresponding 'truth' orientation data was collected from a Vicon motion capture system and was available for comparison against estimated values.

In this report, we will discuss the implementation details of each filter. Then present results comparing the computed attitude estimates using each filter type. Relevant lessons and observations are discussed.

II. Scaling and Removing Bias

This section describes the methods for estimating and removing bias and scaling the raw data values collected from the IMU to SI units for the accelerometer and gyroscope.

A. Accelerometer

To interpret the raw IMU accelerometer readings, the data was multiplied by the given scale factor, s_a and added to the given accelerometer bias b_a for each axis of the accelerometer. This resulted in the measured acceleration as a function of gravitational acceleration.

$$\hat{\tilde{a}}_i = ((a_i \times s_{a,i}) + b_{a,i}) \tag{1}$$

This was done for each axis $i \in \{x, y, z\}$. The values for both b_a and s_a were provided and were constant for each set of data considered here.

B. Gyroscope

The data from the gyroscopes were also processed with a known scale factor s_g, but with a computed bias factor b_g. The scale factor was given from a IMU data sheet as,

$$s_g = \frac{3300}{1023} \times \frac{\pi}{180} \times 0.3 \tag{2}$$

The gyroscope bias was calculated from the average of the first 200 gyroscope measurements. The gyroscope is assumed to be steady for the first 200 measurements of each data set to determine initial angular rates. For each axis, the bias was determined as

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

where $k = 200$. These terms were used to calculate the desired rad/s angular rate, $\tilde{\omega}$, from the raw gyroscope data using the equation below.

$$\tilde{\omega_i} = s_{g,i} \times (\omega_i \times b_{g,i}) \tag{4}$$

Similar to the accelerometer readings, this was done for each axis $i \in \{x, y, z\}$. Unlike the accelerometer data, the bias value b_g was computed for each data set.

III. Interpreting the data

The data was initially read into the the script through the MATLAB files and converted into arrays using the supporting NumPy library in Python. The raw Vicon, IMU, and IMU parameter data was stored for each individual test. In order to compare the computed attitude during testing, the Vicon data was converted from the provided rotation matrix format to Roll, Pitch, and Yaw Euler angles. The rotation matrix provided by the Vicon data is represented as follows,

$$R = \begin{bmatrix} r_{1,1} & r_{1,2} & r_{1,3} \\ r_{2,1} & r_{2,2} & r_{2,3} \\ r_{3,1} & r_{3,2} & r_{3,3} \end{bmatrix},$$

Euler angles were computed via the following conversions:

$$Roll, \phi = tan^{-1}(r_{3,2}/r_{3,3}) \tag{5}$$

$$Pitch, \theta = tan^{-1}(-r_{3,1}/\sqrt{r_{3,2}^2 + r_{3,3}^2}) \tag{6}$$

$$Yaw, \psi = tan^{-1}(r_{2,1}/r_{1,1}) \tag{7}$$

These values formed the truth data against which the estimated values could be compared.

IV. Orientation from Accelerometer

Estimated orientation values were obtained from the accelerometer data by determining the relative acceleration vector compared to the known orientation of the gravity vector. The following trigonometric relationships were used to estimate orientation in this way:

$$Roll, \phi = tan^{-1}(a_y/\sqrt{a_x^2 + a_z^2}) \tag{8}$$

$$Pitch, \theta = -tan^{-1}(a_x/\sqrt{a_y^2 + a_z^2}) \tag{9}$$

$$Yaw, \psi = tan^{-1}(\sqrt{a_x^2 + a_y^2}/a_z) \tag{10}$$

However, it is important to note that due to the symmetry of the gravity vector about the z-axis, this method of attitude estimation is inherently inaccurate for Yaw measurements. Additionally, it is not possible to separate acceleration caused by rapid motion from that due to gravity, thus this method does not provide accurate estimates for rapid motion over short periods of time.

V. Orientation from Gyroscope

The IMU outputs the angular rates $\bar{\omega}_i = [\dot{\phi}, \dot{\theta}, \dot{\psi}]^T$ in the sensor frame, therefore to calculate the orientation $\bar{x} = [\phi, \theta, \psi]^T$ at time $t = t_1$ from the rotation data, the angular rates can be integrated over time.

$$\bar{x}_{t_1} = \int_{t_0}^{t_1} \bar{\omega}\, dt \tag{11}$$

This can be estimated numerically over the discrete periods of gyroscope data to determine the state $\bar{x}_{t_j}$ by taking the last estimated position, $\bar{x}_{t_{j-1}}$, and propagating it forward with the current angular rates $\bar{\omega}_{t_j}$, with the time since the last measurement $t_j - t_{j-1}$.

$$\bar{x}_{t_j} = \bar{x}_{t_{j-1}} + \bar{\omega}_{t_j} \times (t_j - t_{j-1}) \tag{12}$$

This method generally performs well, but offers no way to compensate for noise in the IMU readings, therefore the estimates tend to drift and become more inaccurate with time.

VI. Orientation using Complementary Filter

After calculating the attitude from the accelerometer data and the gyroscope data separately, the resulting attitudes can be combined to improve the attitude estimation. The complementary filter takes a fixed weighted sum of both components to estimate the attitude. The weight of each component is determined from the constant α indicating the weight of the estimated attitude from the IMU acceleration data $x_{e\bar{s}t,a}$ versus the IMU gyroscope data $x_{e\bar{s}t,g}$. .

$$\bar{x}_{est,comp} = \alpha \times \bar{x}_{est,a} + s(1-\alpha) \times \bar{x}_{est,g} \tag{13}$$

To further filter the data a low pass filter is added to the accelerometer data to reduce error from noise in the data. The low pass filter takes a weighted sum of the predicted attitude and the current attitude in order to estimate a better filtered estimate. γ was chosen as 0.2 through testing.

$$x_{t+1} = (1-\gamma)\tilde{x}_{t+1} + \gamma x_{t-1} \tag{14}$$

This combined approach varies in quality. For Roll and Pitch measurements it generally outperforms the orientation estimates from the accelerometer or gyroscope readings alone. However, the quality of the Yaw measurements are still negatively impacted by the estimates from only accelerometer data. Additionally, similar to the gyroscope, these estimates also suffer from large drift as time goes on due to the weighting coefficient being set to a fixed value. Through trial and error, the estimates for any one case could be tuned to be more accurate by modifying the value of γ, but this tuned result tended to lead towards higher levels of inaccuracy in other cases.

VII. Orientation from Madgwick Filter

The same acceleration and angular rate data used in the above estimations was then used in an implementation of the Madgwick filter. This filter first requires converting both sets of data to quaternions, then determining the rate of change in the quaternion values based on the angular rates from the gyroscope and the rate of change based on a modified gradient descent method with the acceleration data (the gradient descent method makes several key assumptions to reduce the number of steps required for convergence to one). These two rates of change are then combined to achieve a net estimated rate of change for the quaternion values and numerical integration is preformed over the timestep with the net rate. The final quaternion estimate is then converted back to Euler angles.

Conversions from Euler angles to quaternions and vice versa, as well as quaternion multiplication, are accomplished using the ROS tf library for transformations, which provides numerous useful built-in functions to handle conversions and operations with quaternions. Significantly, we found the order of the quaternion values in for this library required that several of the operations documented in the original paper for the Madgwick filter [1] be modified. Namely, the paper assumes a quaternion representation of $\bar{q} = [q_w, q_x, q_y, q_z]^T$, whereas the tf library maintains the convention $\bar{q} = [q_x, q_y, q_z, q_w]^T$.

The initial orientation at $t_0 = 0$ was assumed to be $\bar{x}_{t_0} = [0, 0, 0]^T$ which corresponds to an initial quaternion value of $\bar{q}_{t_0} = [0, 0, 0, 1]^T$. This value formed the initial orientation estimate $\hat{q}_{est,t_i}$ where $\hat{q} := ||\bar{q}||$. From this, the rate of change estimate from the gyroscope measurements was determined as

$$\dot{\bar{q}}_{est,t_{i+1},g} = \frac{1}{2}\hat{q}_{est,t_i} \otimes [\bar{\omega}, 0]^T \tag{15}$$

It is then possible to determine the direction of the estimated error of the net $\bar{q}_{est,t_i}$ caused by the gyroscope and accelerometer measurements, and subtract this from the net value. This directional error value $\dot{\hat{q}}_{\epsilon,t_i}$ can be computed as

$$\dot{\hat{q}}_{\epsilon,t_{i+1}} = \frac{\nabla f}{||\nabla f||} \tag{16}$$

where f and ∇f can be computed using the following equations where $\hat{g}$ represents the normalized gravity vector.

$$f = f(\hat{q}_{est,t_i}, \hat{g}, \bar{a}_{t_{i+1}}) = \begin{bmatrix} 2(q_1q_3 - q_4q_2) - a_x \\ 2(q_4q_1 + q_2q_3) - a_y \\ 2(0.5 - q_1^2 - q_2^2) - a_z \end{bmatrix}, \tag{17}$$

$$\nabla f = J^T f, \tag{18}$$

$$J = J(\hat{q}_{est,t_i}, \hat{g}, \bar{a}_{t_{i+1}}) = \begin{bmatrix} -2q_2 & 2q_3 & -2q_4 & 2q_1 \\ 2q_1 & 2q_4 & 2q_3 & 2q_2 \\ 0 & -4q_1 & -4q_2 & 0 \end{bmatrix}, \tag{19}$$

It is then possible to combine these values with a fixed estimate β for the magnitude of the gyroscope error to yield a final estimate of the rate of change in the quaternion manifold.

$$\dot{\bar{q}}_{est,t_{i+1}} = \dot{\bar{q}}_{est,t_{i+1},g} - \beta\dot{\hat{q}}_{\epsilon,t_{i+1}} \tag{20}$$

Through trail and inspection, the value of the β for our work was set to 0.01. This rate of change was then used to numerically approximate over the timestep $t_{i+1} - t_i$ in a similar way to the gyroscope estimation.

$$\bar{q}_{est,t_{i+1}} = \bar{q}_{est,t_i} + \dot{\bar{q}}_{est,t_{i+1}}(t_{i+1} - t_i) \tag{21}$$

Once a suitable value for β was determined the Madgwick filter can be seen to outperform all other attitude estimation methods explored here. A significant part of this is due to the fact that the Madgwick filter attempts to determine the direction of the error in the rate change estimation and subtract the error magnitude only in that direction. This approach is more adaptive than any of the other approaches explored here and therefore results in the overall best tracking ability. However, this filter still lacks any understanding of the system dynamics and can still be seen to lose accuracy for rapid platform motions. Although once the rapid motions stop, the estimations tend to return to a more accurate state.

VIII. Results

A. Training Data Sets

Euler angles for the training data sets are shown in Figures 1, 2, 3, 4, 5, and 6.

B. Test Data Sets

Euler angles for the real test data sets are shown in Figures 7, 8, 9, and 10.

IX. Videos

We have uploaded our video results to YouTube, and provide links to each test set here. Note that the 'real' sets will only have 4 plots.

A. Training Data Videos

1) https://www.youtube.com/watch?v=RrQRfxPl1Zo
2) https://www.youtube.com/watch?v=sOW8vC5bjB4
3) https://www.youtube.com/watch?v=J5DwQtJTMSs
4) https://www.youtube.com/watch?v=VNjoiwr2MLE
5) https://www.youtube.com/watch?v=aaCrD4lyRvk
6) https://www.youtube.com/watch?v=$_5q8VNZvpzw$

B. Real Data Videos

1) https://www.youtube.com/watch?v=EVzozAoE$_{Sc}$
2) https://www.youtube.com/watch?v=2TCLJWEsbxc
3) https://www.youtube.com/watch?v=0gi7TIfEq6g
4) https://www.youtube.com/watch?v=581fbyzaDD0

X. Important Lessons Learned

1) β Sensitivity: The Madgwick filter is highly sensitive to the value of the magnitude of the gyroscope error β. If this value is too large or small, the orientation estimations can be negatively impacted to a high degree and can result in worse estimations than simpler estimation methods (e.g. gyroscope data only). However, once an appropriate value was determined, the Madgwick filter outperformed all other filter types considered here.

2) Debugging: We found out the hard way that it is critically important to have an understanding of exactly what measurement data is being produced by the IMU. We jumped into visualizing data, and spent more time than we should have debugging other issues when we simply had an error in our conversion to physical units. Also, while the math and filter functions themselves were relatively straightforward to understand, the intricacies of scipy, numpy, ROS tf, and other libraries required many iterations with multiple eyes on the code until we finally got each filter functioning properly. Having the Vicon data plotted and available proved to be invaluable for rapidly testing and checking new changes.

3) Learning New Libraries: We experimented with different methods of converting Matlab plot files to video files using the OpenCV library. Ultimately, we were unable to get it functioning properly. We instead simply used matplotlibs FuncAnimation utility to generate each plot frame by frame and convert it to an .mp4 file. While the method was simpler, the actually processing time was between 5 and 10 minutes per video. Down the road we believe it would be beneficial to develop of better understanding of the OpenCV python interface.

XI. Conclusion

It is clear from the test and actual data presented here that the Madgwick filter's approach to dynamically determination and remove the estimated gyroscope error in the direction of the gyroscope error significantly outperforms orientation estimation methods that rely on fixed weightings. However, it can still be seen that in some cases, such as training data set 3, rapid changes in the attitude can still result in poor state estimations.

Acknowledgment

The authors would like to thank the professors for this course, Nitin J. Sanket and Chahat Deep Singh, as well as Dr. Inderjit Chopra.

References

[1] S. O. H. Madgwick, *An efficient orientation filter for inertial and inertial/magnetic sensor arrays*. University of Bristol Rep., 2010

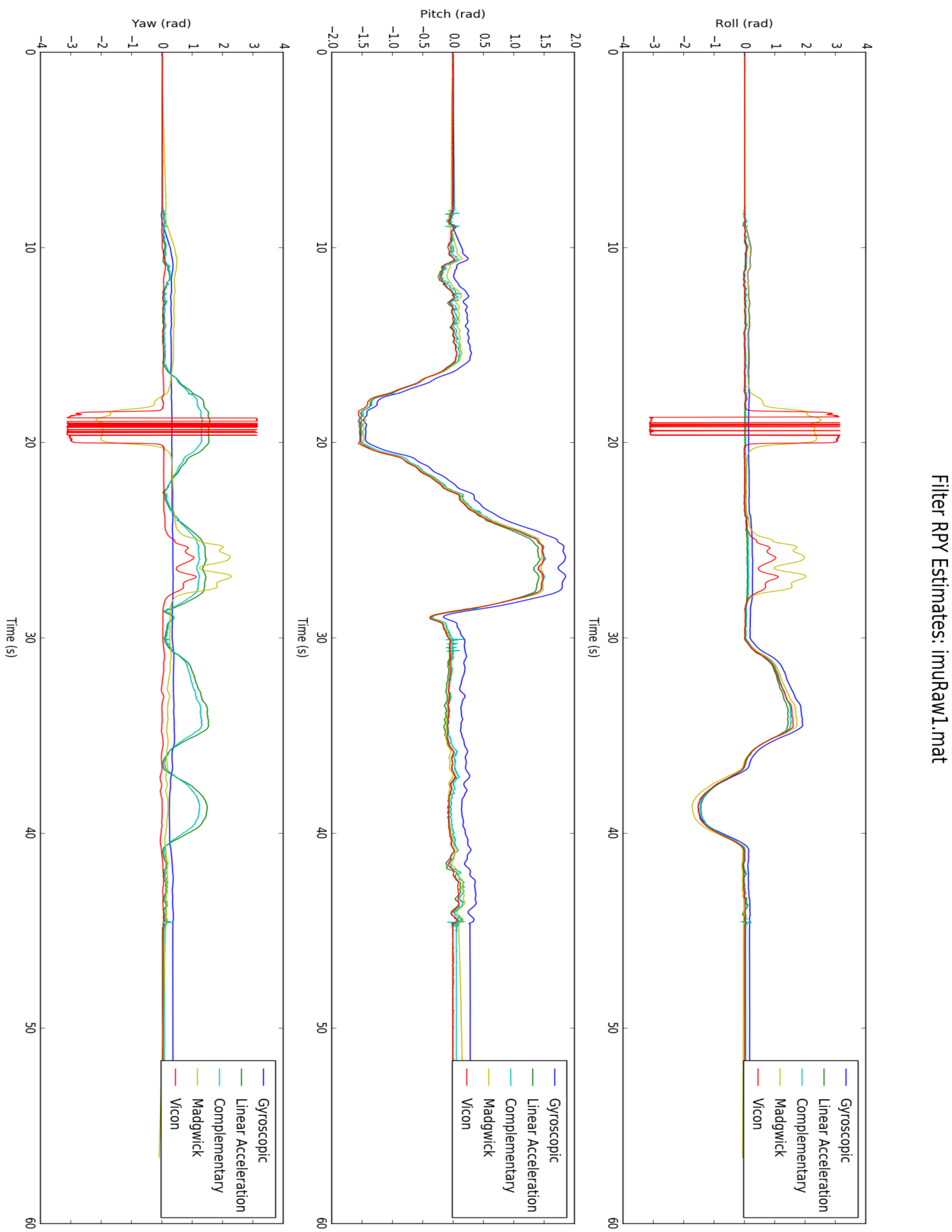

Fig. 1. Euler angles for training data set 1.

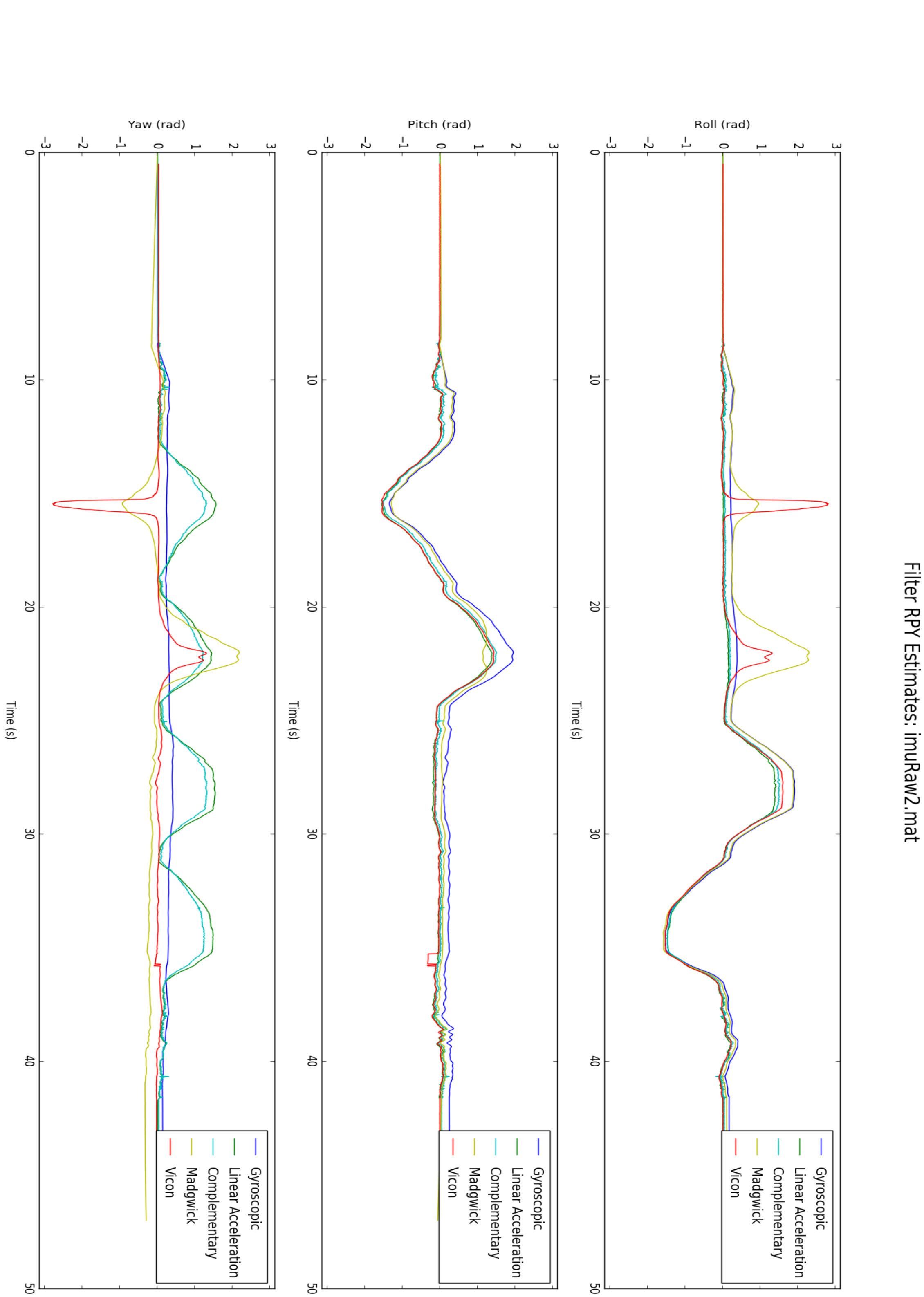

Fig. 2. Euler angles for training data set 2.

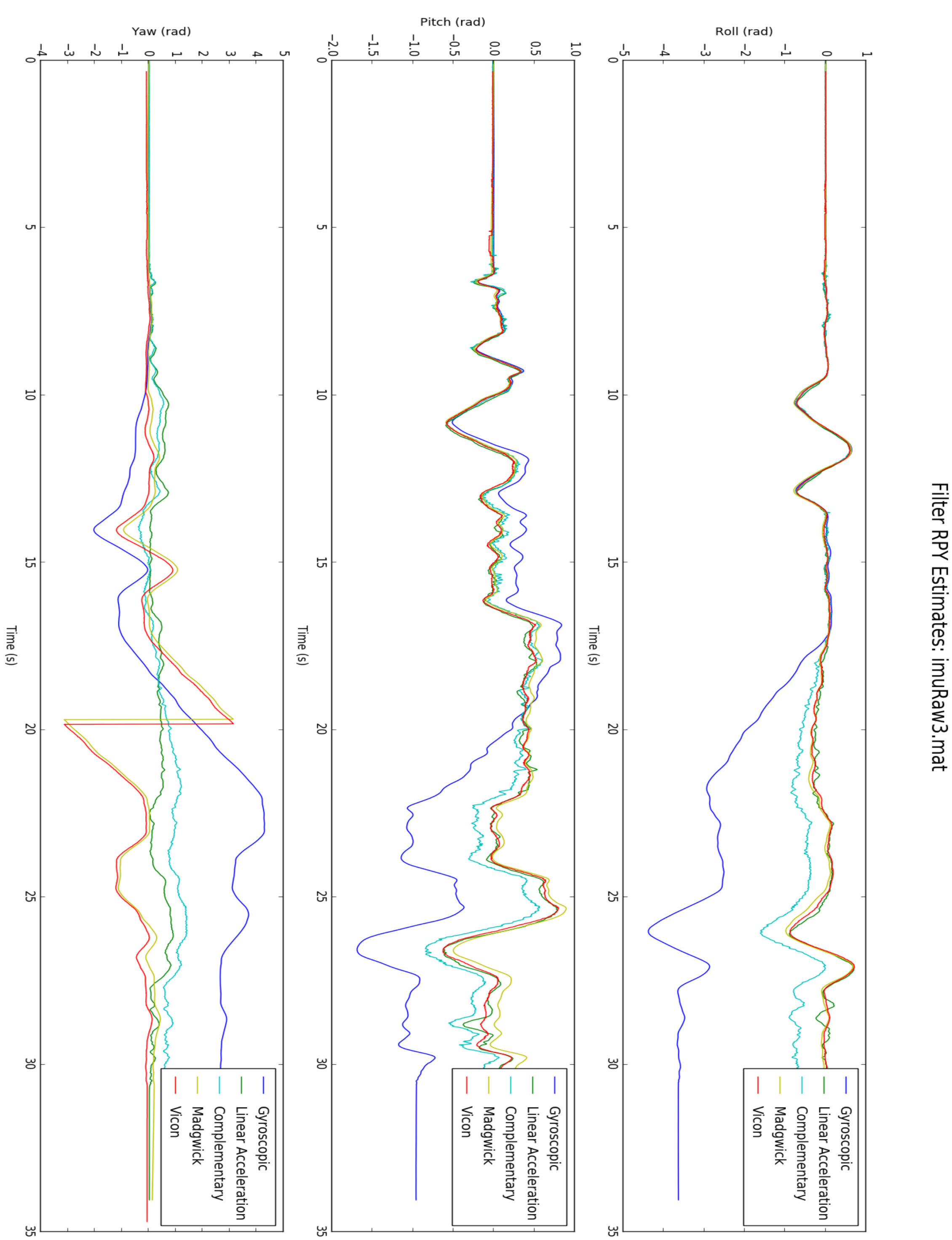

Fig. 3. Euler angles for training data set 3.

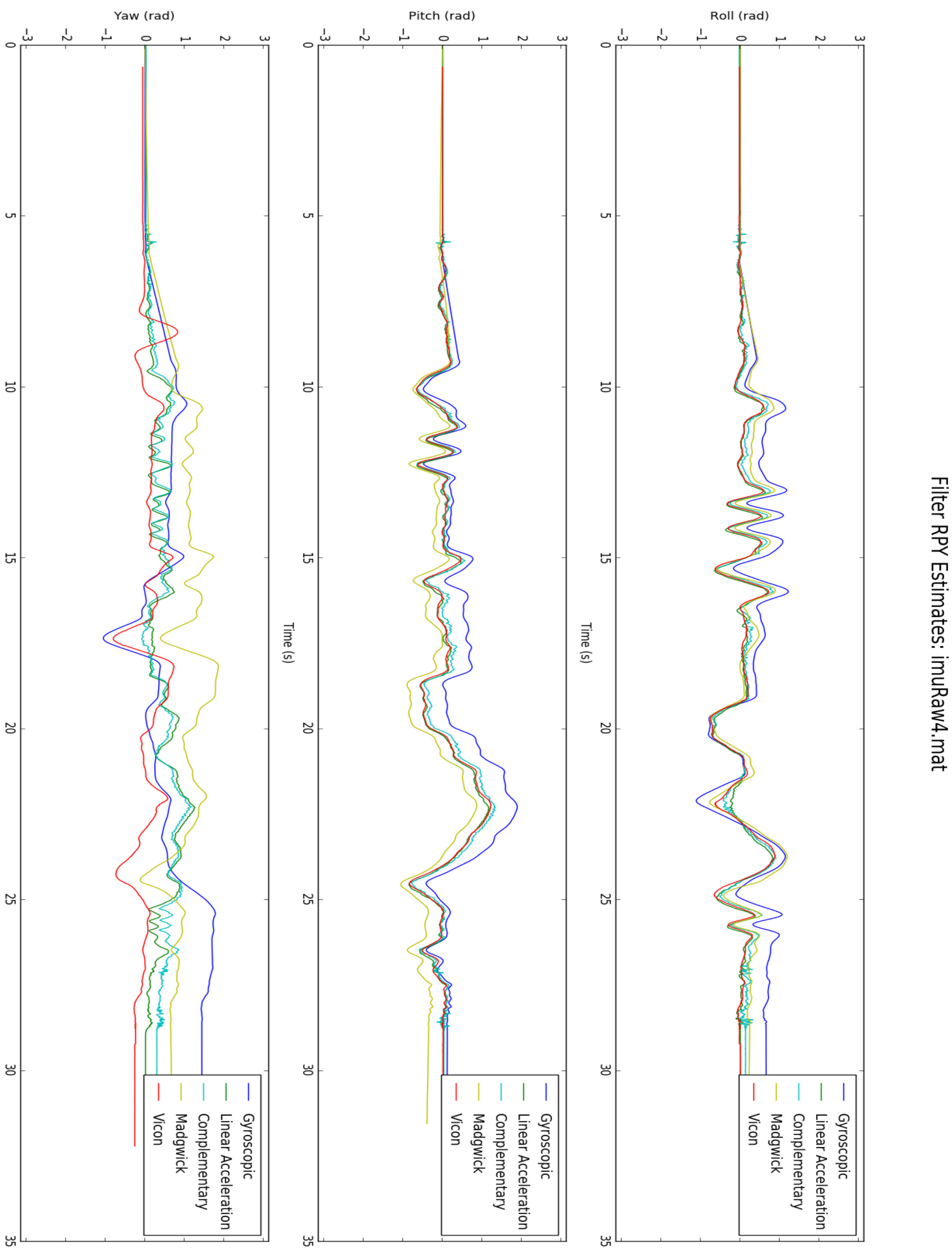

Fig. 4. Euler angles for training data set 4.

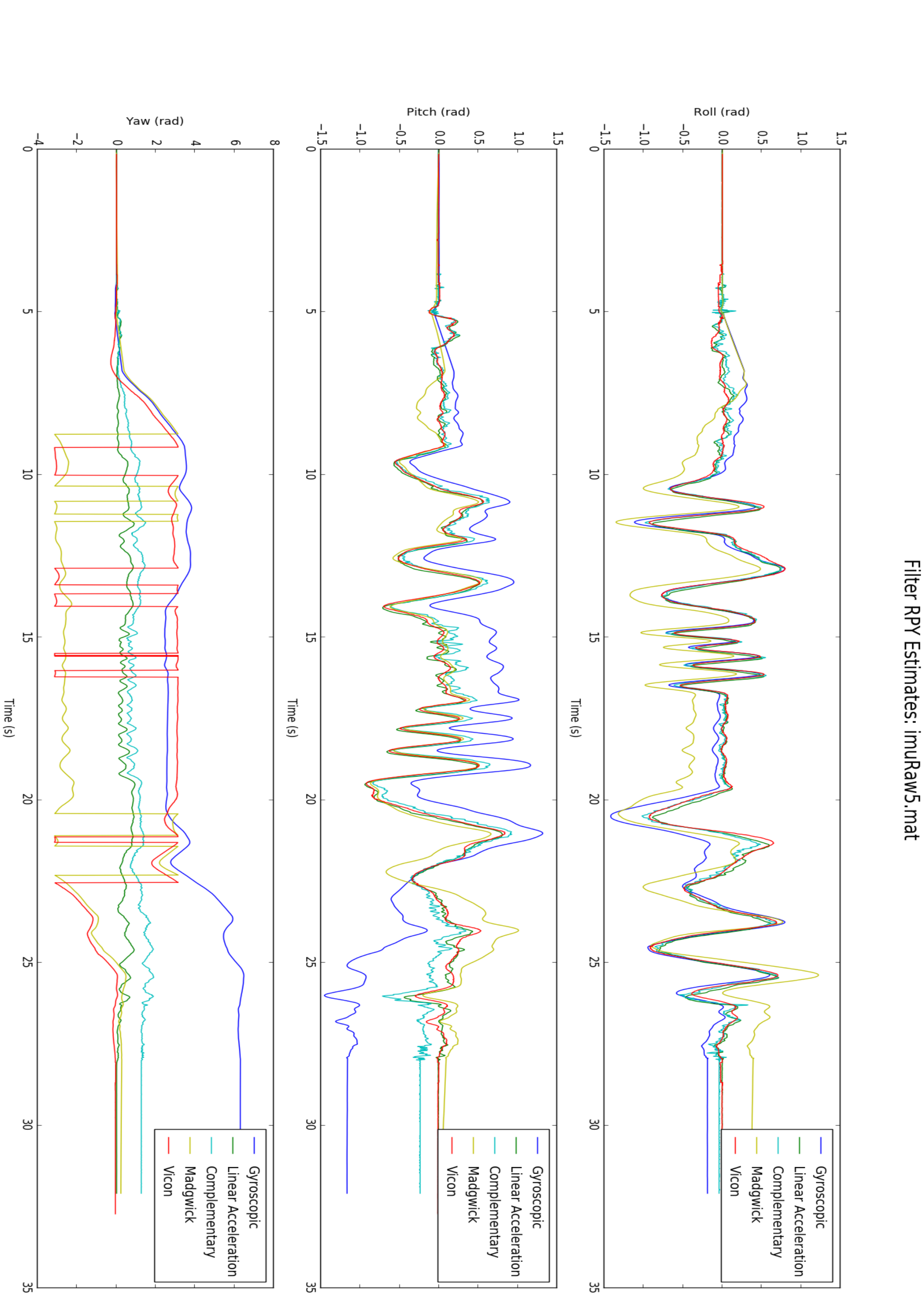

Fig. 5. Euler angles for training data set 5.

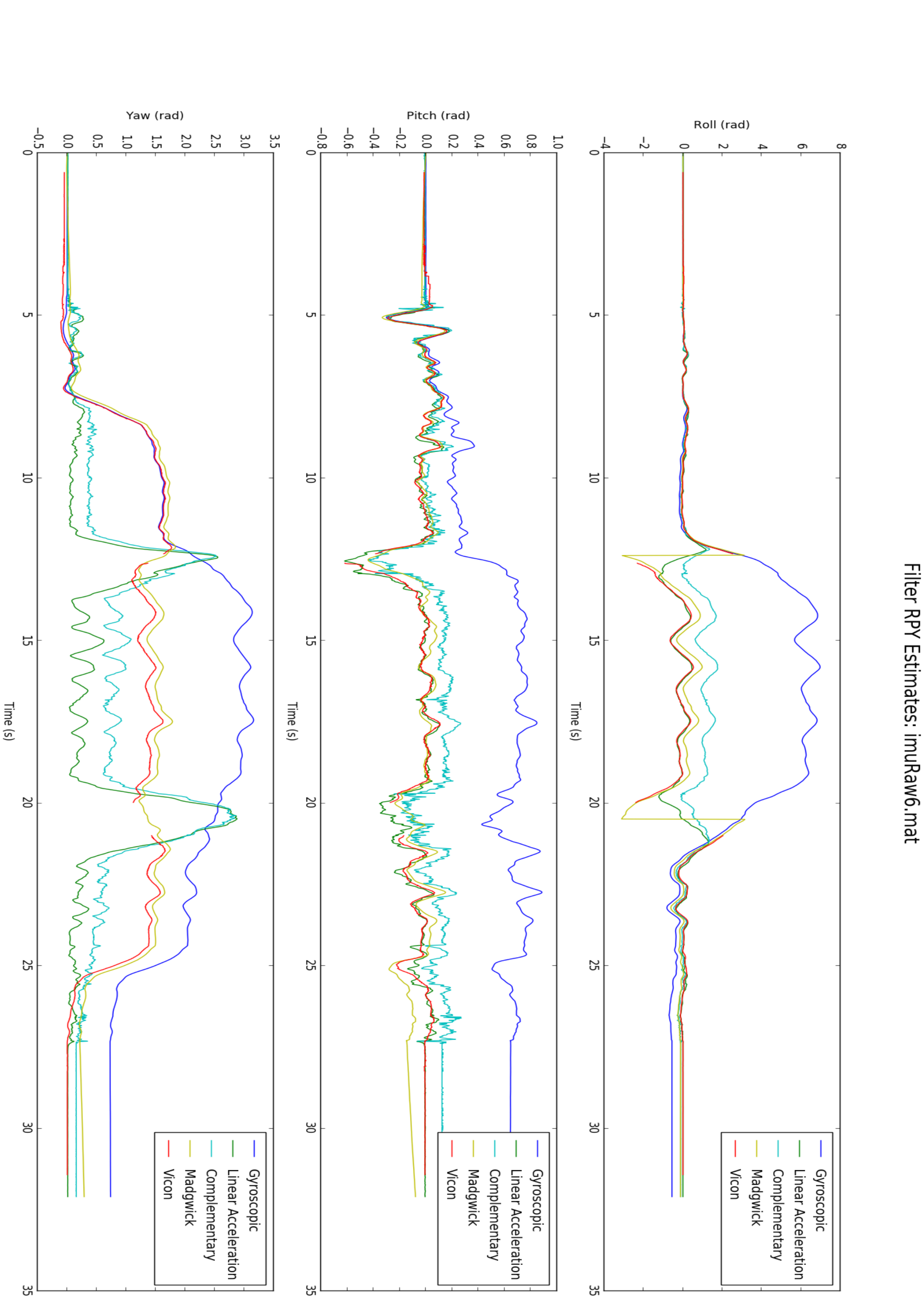

Fig. 6. Euler angles for training data set 6.

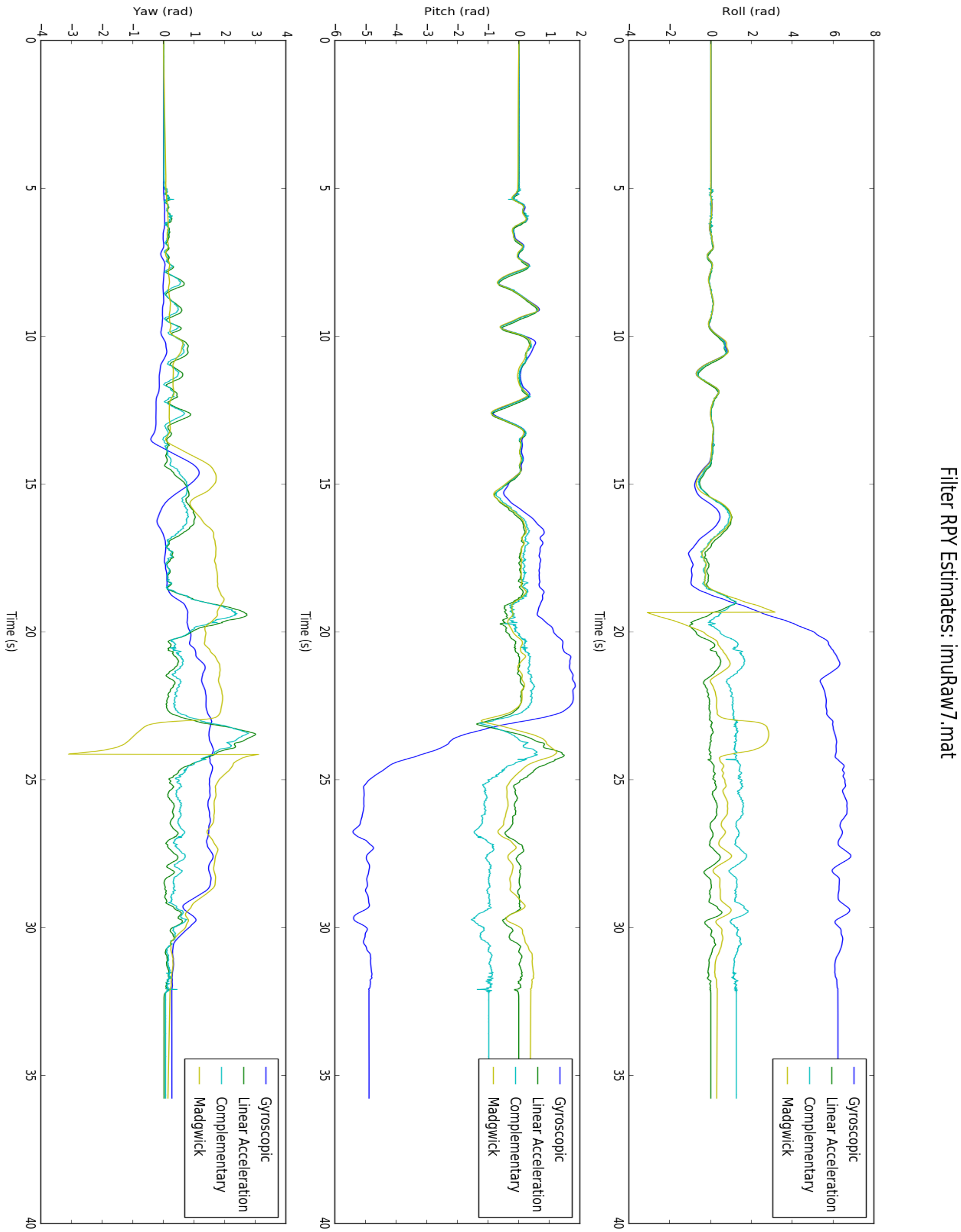

Fig. 7. Euler angles for real data set 1.

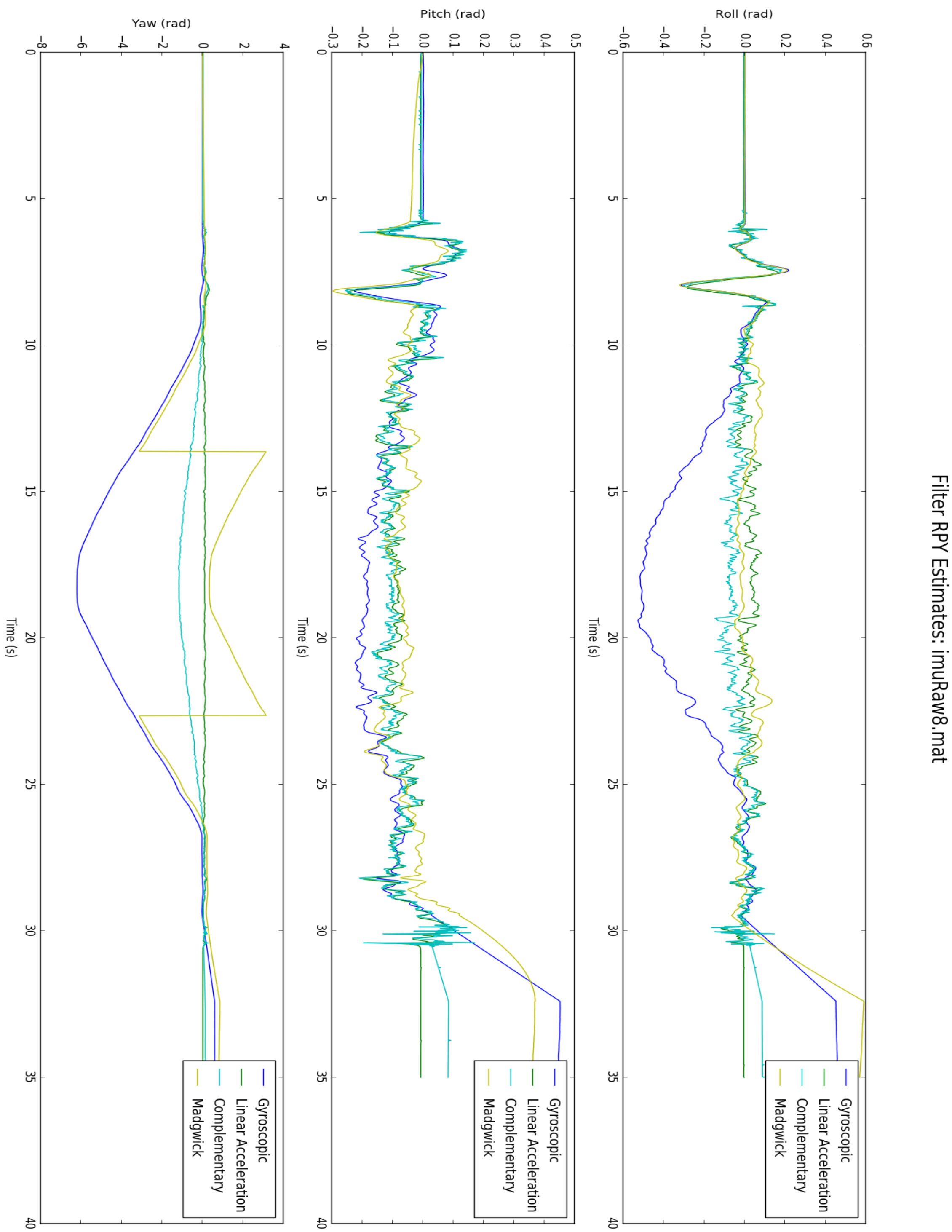

Fig. 8. Euler angles for real data set 2.

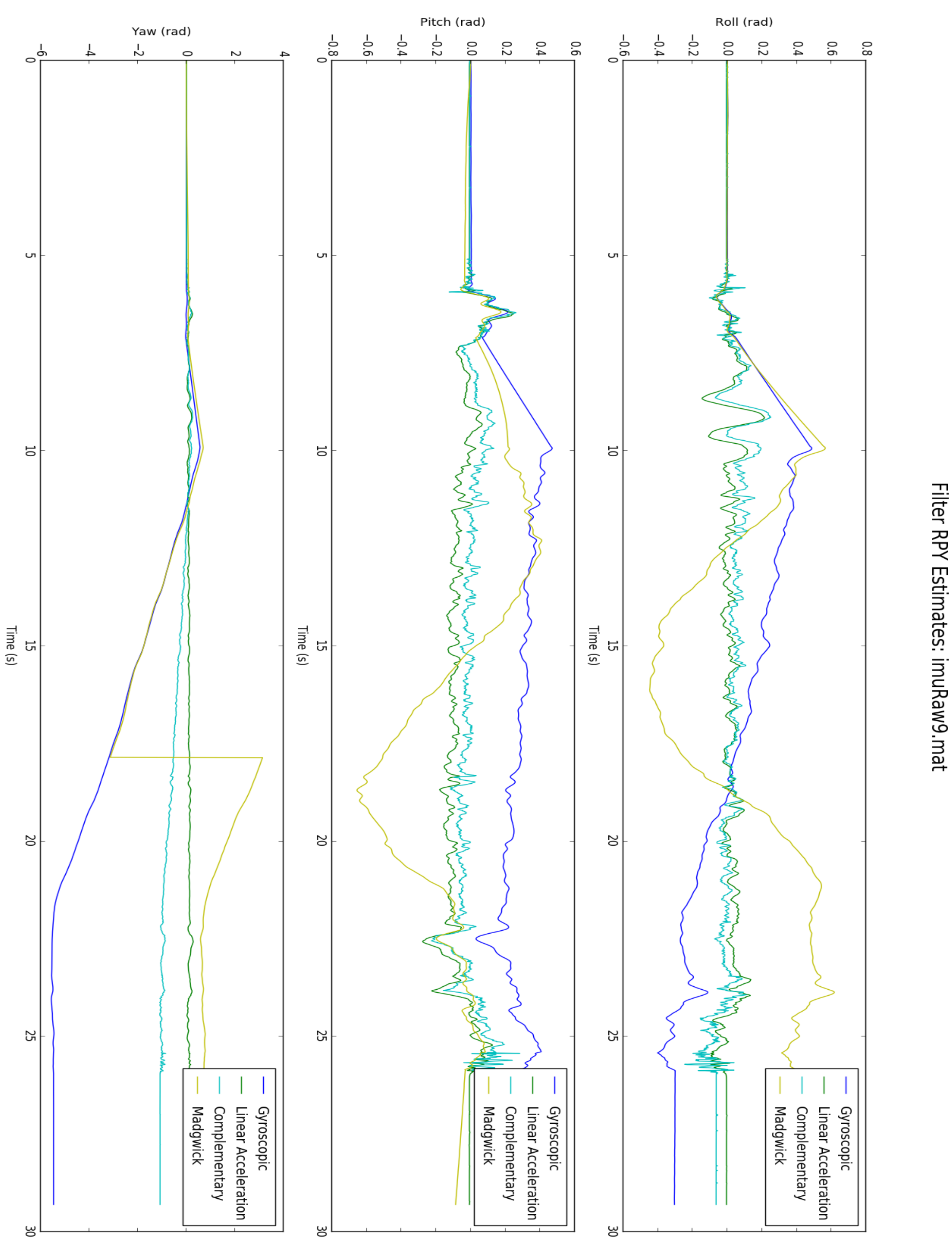

Fig. 9. Euler angles for real data set 3.

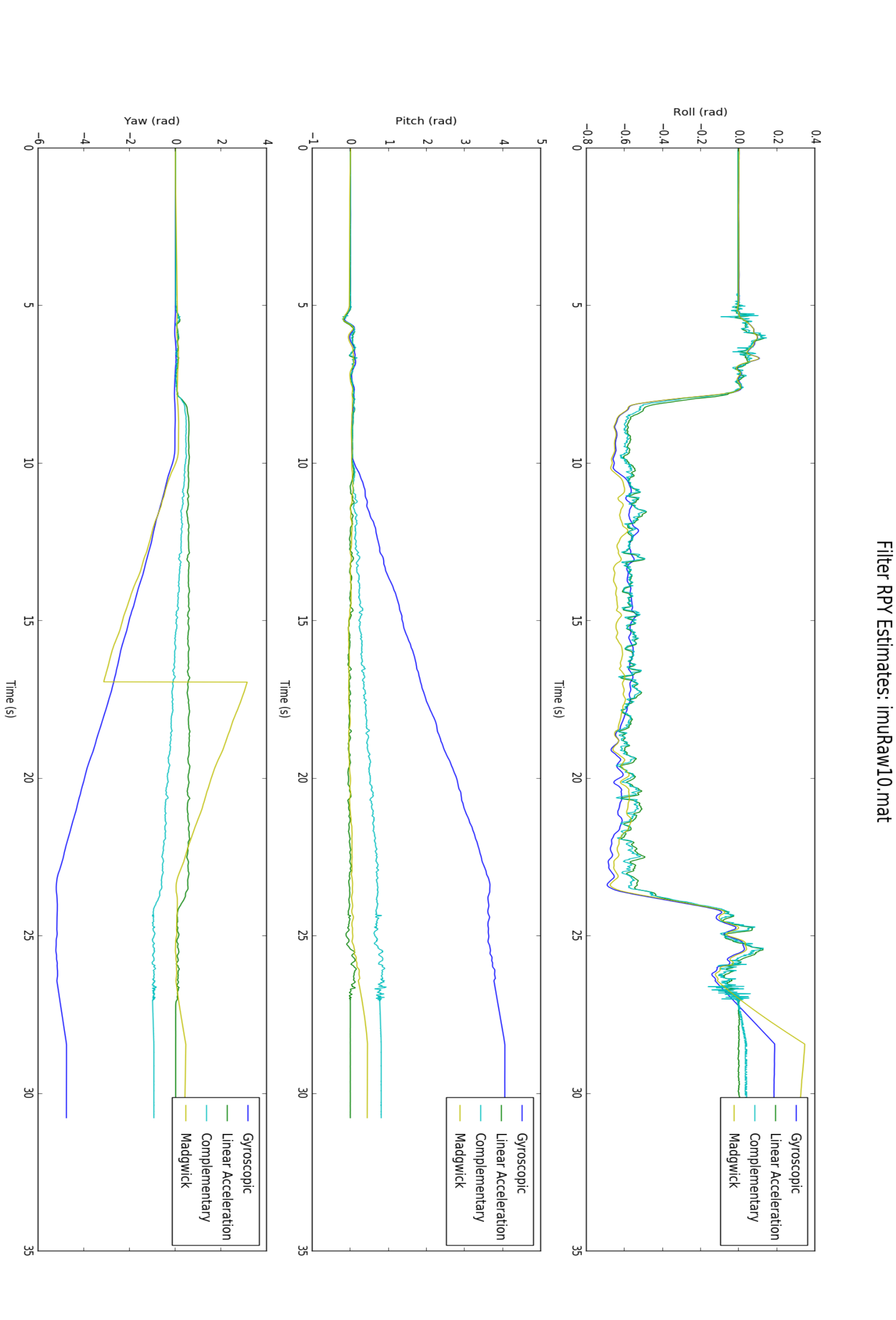

Fig. 10. Euler angles for real data set 4.