Attitude Estimation Using an Unscented Kalman Filter

Ilya Semenov
University of Maryland
isemenov@umd.edu

Tim Kurtiak
University of Maryland
tkurtiak@terpmail.umd.edu

Nathan Witztum
University of Maryland
nwitztum@terpmail.umd.edu

***Abstract*—This report presents the theory and application of an Unscented Kalman Filter (UKF) for attitude estimation. Raw data from an accelerometer and gyroscope are processed using the UKF, and the resulting attitude estimations are compared to actual orientation as measured by a visual orientation (VICON) system.**

I. Problem Statement

This project aims to demonstrate the accuracy of different attitude estimation methods: Madgwick filter and Unscented Kalman Filter (UKF). Raw sensor data from an ArduIMU+ V2 6DOF IMU is processed and compared to VICON motion capture system data recorded simultaneously. The test data for this project was provided courtesy of the ESE 650: Learning In Robotics course at the University of Pennsylvania

II. Processing Sensor Data

Raw IMU data must be processed in order to gain usable information about the system orientation.

First, gyroscope readings must be scaled to units of radians per second according to equation 1 below [3]. Additionally, gyroscope bias is removed. Gyroscope bias, b_g, is calculated by assuming the IMU is at rest for the first 200 readings after it begins logging data. An average of the gyroscope readings is taken in each axis as the bias for that axis.

$$\underline{w} = \frac{3300}{1023} \cdot \frac{\pi}{180} \cdot 0.3 \cdot (\underline{w}_{raw} - \underline{b}_g) \tag{1}$$

Raw accelerometer data is processed using equation 2 below. This equation differs from the provided accelerometer processing equation as we found that the raw accelerometer reports accelerations in G's and not in m/s^2. As such, an appropriate equation was reverse engineered to process accelerometer data in a realistic magnitude.

$$\underline{a} = (9.81\frac{m}{s^2})(\underline{a}_{raw} \cdot \underline{s}_a + \underline{b}_a) \tag{2}$$

Accelerometer data may further be processed via a low pass filter to remove noise. Additionally, the gyroscope data can be high pass filtered in order to reduce low frequency errors. However, we elected to proceed without filtering to illustrate the pros and cons of each attitude estimation algorithm discussed.

III. Madgwick Filter Method

The Madgwick filter combines both accelerometer and gyroscope data to calculate attitude [1]. The algorithm integrates gyroscope rates to capture fast movements as well as the gyroscope method discussed earlier, but also uses the accelerometer to eliminate gyroscopic drift and maintain an accurate and calibrated attitude estimate. The Madgwick filter utilizes quaternions to represent attitude without singularities known as gimbal lock. The filter uses the current state $\underline{q}_t$, where $\underline{q}_t$ represents the quaternion state at time t, along with gyroscope data $\underline{w}_{t+1}$ to find $\dot{\underline{q}}_t$. It is assumed that the initial quaternion attitude state q_0 is known. We first estimate the rate of change based on the gyroscope measurements by equation 3 below:

$$\dot{\underline{q}}_\omega = \frac{1}{2}\underline{q}_t \otimes \begin{bmatrix} 0 \\ \underline{w} \end{bmatrix} \tag{3}$$

Next, the accelerometer data is integrated using a solution to the minimization of the error between the gravity vector and the measured acceleration vector [4]. Let $\underline{q}_t = [p_1, p_2, p_3, p_4]^T$ and $\underline{a}_{t+1} = [a_x, a_y, a_z]^T$ The math has been simplified for implementation below:

$$\nabla f = \begin{bmatrix} -2q_3 & 2q_2 & 0 \\ 2q_4 & 2q_1 & -4q_2 \\ -2q_1 & 2q_4 & -4q_3 \\ 2q_2 & 2q_3 & 0 \end{bmatrix} \begin{bmatrix} 2(q_2q_4 - q_1q_3) - a_x \\ 2(q_1q_2 + q_3q_4) - a_y \\ 2(.5 - q_2^2 - q_3^2) - a_z \end{bmatrix} \tag{4}$$

$$\Delta\underline{q}_a = -\beta\frac{\nabla f}{\|\nabla f\|} \tag{5}$$

β is a tunable parameter which controls the magnitude of accelerometer data to trust in order to counter gyroscope error. For our implementation, a value of $\beta = 0.1$ was used which resulted in a good attitude estimation with minimal overshoot.

Lastly, the accelerometer and the gyroscope rates are combined and integrated over the sampling time Δt to generate an attitude increment which is applied to the previous attitude estimate. The attitude state is thus updated to reflect the gyroscope and accelerometer updates as per equation 6.

$$\underline{q}_{t+1} = \underline{q}_t + (1-\gamma) \cdot \dot{q_\omega} \cdot \Delta t + \gamma \cdot \Delta\underline{q}_a \tag{6}$$

where γ is the accelerometer weight factor. In our implementation, a value of $\gamma = 0.1$ is used.

IV. Unscented Kalman Filter Method

The UKF combines both accelerometer and gyroscope data to calculate attitude [2]. The algorithm is based on the probabilistic approach of the original Kalman filter, but is modified to account for the fact that the system contains inherent nonlinearities. In addition, the UKF utilizes quaternions to represent attitude, avoiding the issue of singularities, also known as gimbal lock.

The UKF algorithm begins by constructing the state vector:

$$x = \begin{pmatrix} q \\ \overrightarrow{\omega} \end{pmatrix} \tag{7}$$

which has seven components - four from q and three from $\overrightarrow{\omega}$. However, the UKF assumes that in reality the state vector will be disturbed resulting in:

$$\tilde{x_k} = \begin{pmatrix} \tilde{q_k} \\ \overrightarrow{\omega}_k \end{pmatrix} = \begin{pmatrix} q_k q_w \\ \overrightarrow{\omega}_k + \overrightarrow{w}_\omega \end{pmatrix} \tag{8}$$

where q_k and w_ω represent the disturbances in the quaternion state and the angular velocity state respectively at point k. Finally, the process model is given by:

$$x_{k+1} = \begin{pmatrix} q_k q_w q_\Delta \\ \overrightarrow{\omega}_k + \overrightarrow{w}_\omega \end{pmatrix} \tag{9}$$

where q_Δ is the differential rotation that occurred during the time interval Δt between points k and $k+1$. In other words, the UKF *predicts* the evolution of the state vector.

In the beginning of every UKF recursion, the previous estimates of the state vector $\hat{x}_{k-1}$ and its covariance P_{k-1} are known. Then, if Q is the covariance of the process noise, the square root matrix, S is found by:

$$S = \sqrt{P_{k-1} + Q} \tag{10}$$

and the disturbance of the sigma points, $W_{i,i+n}$, is:

$$W_{i,i+n} = \text{columns}\left(\pm\sqrt{2n(P_{k-1} + Q)}\right) \tag{11}$$

where n is the number of independent variables in the state. It is worth noting that the Matrix Q has to be assumed or otherwise found and is one of the tunable variables of the filter. Tuning a 6x6 matrix can be quite difficult, and as seen in this report we did not have great success finding the optimal values.

The sigma points are then found by:

$$X_i = \hat{x}_{k-1} + W_i \tag{12}$$

It is important to note that W_i is a grouping of 1x6 vectors. To create sigma points which are 1x7 vectors one must transform the angles w_k (the first 3 components) into quaternions by assuming the unit vector of those angles is a rotation axes $\hat{w_k}$, and the magnitude is the rotation angle α. Then the quaternion representation is $q_k = [cos(\alpha/2), \hat{w_k} * sin(\alpha/2)]$. The updated sigma points are computed by applying the process model:

$$Y_i = \begin{bmatrix} q_{k-1} q_W q_\Delta \\ \overrightarrow{\omega}_{k-1} + \overrightarrow{\omega}_W \end{bmatrix} \tag{13}$$

To find the mean $\bar{y_i}$ of the state vectors Y_i a barycentric mean is insufficient due to the properties of quaternions. Thus each quaternion must first be rotated by the inverse of the current mean quaternion estimate resulting in an error quaternion, which represents the rotation needed to reach the mean quaternion. This error quaternion is then transformed into it's rotation vector representation by reversing process used when completing equation 12. The the barycentric mean $\overrightarrow{e}$ of all error vectors is found:

$$\overrightarrow{e} = \frac{1}{2n}\sum_{i=1}^{2n} \overrightarrow{e}_i \tag{14}$$

The corresponding quaternion e can be used to calculate a better estimate for the next iteration step:

$$\overline{q}_{t+1} = e\overline{q}_t \tag{15}$$

The rest of the state vector is simple:

$$\overline{\omega} = \frac{1}{2n}\sum_{i=1}^{2n} \overline{\omega}_i \tag{16}$$

The notation from [2] is adopted for the next several steps for brevity.

Next, the covariance of the Y_i state vectors is found:

$$\overline{P}_k - = \frac{1}{2n}\sum_{i=1}^{2n} (y_k - \bar{y_i})(y_k - \bar{y_i})^T \tag{17}$$

Again it's not as simple as subtracting between 7x1 vectors. However, the error quaternion difference between each vector in Y_i and $\bar{y_i}$ was found in the computation of the mean, and so can be reused here along with a simple difference for the rates.

The measurement model is applied to the transformed state vectors Y_i to find the expected readings from the sensors Z. The measured angular rates are trivial, but the measured accelerations are:

$$\overrightarrow{z}_{acc} = q_k g g_k{}^{-1} \tag{18}$$

Where $g = [0, 0, 0, -9.81]$ The covariance P_{zz} of Z is unchanged from the usual definition:

$$P_{zz} = \frac{1}{2n}\sum_{i=1}^{2n} \Phi_i {\Phi_i}^T \tag{19}$$

Where Φ is each 6x1 vector in Z minus the barycentric mean. The cross-covariance is:

$$P_{xz} = \frac{1}{2n}\sum_{i=1}^{2n} y_i - \bar{y_i}{\Phi_i}^T \tag{20}$$

Also, the innovation term is given by:

$$\nu_k = z_k - \overline{z}_k \tag{21}$$

Where z_k are elements of Z and the innovation covariance is calculated using:

$$P_{\nu\nu} = P_{zz} + R \tag{22}$$

Finally, the Kalman gain is calculated as:

$$K_k = P_{xz} P_{\nu\nu}^{-1} \tag{23}$$

the updated state as:

$$\hat{x}_k = \hat{\overline{x}}_k + K_k \nu_k \tag{24}$$

and the updated state covariance as:

$$P_k = \overline{P}_k - K_k P_{\nu\nu} {K_k}^T \tag{25}$$

V. Results

Figures 1 through 6 below show Z-Y-X Euler angle orientation for each of the 6 test data sets comparing UKF, Madgwick, and VICON reported attitudes.

Additionally, test data sets 7 through 10 are shown in Figures 7 through 10. These data sets do not have VICON data as comparison.

The results of the first 6 datasets were also animated and recorded here:

Dataset 1: https://youtu.be/wmey2GYD8D0
Dataset 2: https://youtu.be/X_lSR0j9nYo
Dataset 3: https://youtu.be/2ZyL6xvgTow
Dataset 4: https://youtu.be/I0rjA6dUlp4
Dataset 5: https://youtu.be/NnzCwk6TWDM
Dataset 6: https://youtu.be/tN2z_LEeOEk
Dataset 7: https://youtu.be/03gPwkNdllk
Dataset 8: https://youtu.be/N1ppREns7Pg
Dataset 9: https://youtu.be/3pOi8cpWjgc
Dataset 10: https://youtu.be/EFKbMtSasuc

VI. Lessons Learned

While the UKF has powerful properties for generalization to many systems there are a few key drawbacks. First is the non-intuitive nature of implementing random variables makes tuning the Q and R quite difficult. In this work it was assumed that $R = diag([Ra, Ra, Ra, Rg, Rg, Rg])$ and $Q = diag([Qa, Qa, Qa, Qg, Qg, Qg])$. The values of Ra, Rg, Qa, Qg were first attempted to be found via trial and error to some success. Then an "autotuning" loop was implemented that tested a combination of values and attempted to find the best gains. This helped but still proved imperfect. Given more time the UKF could be tuned to a better level and performance can be improved. The fact that this was not achieved speaks to one of the drawbacks of this filter. Additionally, for the given problem a regular Kalman Filter of Extended Kalman Filter could likely be used, as attitude change with time is nearly linear.

The work done by Kraft in [2] was groundbreaking, however the use of quaternions complicates many of the definitions of variables, making simple tasks such as averaging require a gradient descent algorithm. This may reduce some of the claimed computational speed benefits claimed by this filter.

Regardless, the results are shown below.

In figure 1 it can be seen that the UKF preforms adequately in pitch and roll but poorly in yaw. This trend is continued for all figures 1-5. Figure 6 shows that the UKF reacts poorly when a dataset has inconsistent timewise distribution. Because the tuning is clearly not optimized it is hard to determine if these problems would be apparent with proper Q and R matricies. In practice it was hard to get a good match for yaw when tuning.

Figures 7-10 do not include Vicon data, however some conclusions can still be drawn. As before it seems that yaw is the worst parameter for UKF, not lining up with the Magwick approximation as closely as do pitch and roll. Figure 7 demonstrates how the tuning of the filter can be sensitive to small differences, the filter completely breaks down in the conditions shown here. Similarly figures 8-10 demonstrate an increased level of noise in the UKF, again a tuning issue.

The Madgwick filter does not accrue error over time like the gyroscope method, but still tracks fast changes in orientation well. In certain data sets, such as the one shown in Figure 6, the Madgwick filter is affected by high frequency noise. Additionally, it was susceptible to small amounts of yaw axis drift over time due to the low ability of the accelerometer to correct yaw axis errors.

It should be noted that there exists a constant phase shift between the VICON attitudes and all IMU derived attitudes. The soft-synchronization appears to present a constant lag between the VICON and IMU data timestamps. While this problem was ignored during our analysis, an automatic syncing of the timestamps may be completed for better plots and data presentation.

Throughout this project the limitations of Euler angles in describing attitude orientation have repeatedly presented themselves. The VICON data was given in $3x3$ rotation matrix format, allowing two possible sets of Euler angles, and assumption had to be made to pick one set. Converting body rates and quaternions to Euler angles is cumbersome and error prone. Our group spent a significant amount of time debugging such issues. Additionally, gimbal lock is possible with Euler angles, observed as large jumps in orientation angles when that angle nears $90°$, as seen in Figure 1. Overall it is much simpler to work with quaternions to avoid problems with representing Euler angles for simple Magwick filter.

VII. Conclusion

The UKF has a lot of potential when tuned properly, but this can be difficult to accomplish. For many applications a simple Magwick filter can be sufficient, or perhaps a standard Kalman Filter or Extended Kalman Filter.

References

[1] Sebastian OH Madgwick, Andrew JL Harrison, and Ravi Vaidyanathan. "Estimation of IMU and MARG orientation using a gradient descent algorithm." 2011 IEEE international conference on rehabilitation robotics. IEEE, 2011.

[2] Edgar Kraft. "A Quaternion-based Unscented Kalman Filter for Orientation Tracking." Sixth International Conference of Information Fusion. IEEE, 2003.

[3] https://prgaero.github.io/2019/proj/p1a/report

[4] *https://www.x-io.co.uk/res/doc/madgwick_internal_report.pdf*

[5] *http://www.stengel.mycpanel.princeton.edu/Quaternions.pdf*

VIII. FIGURES

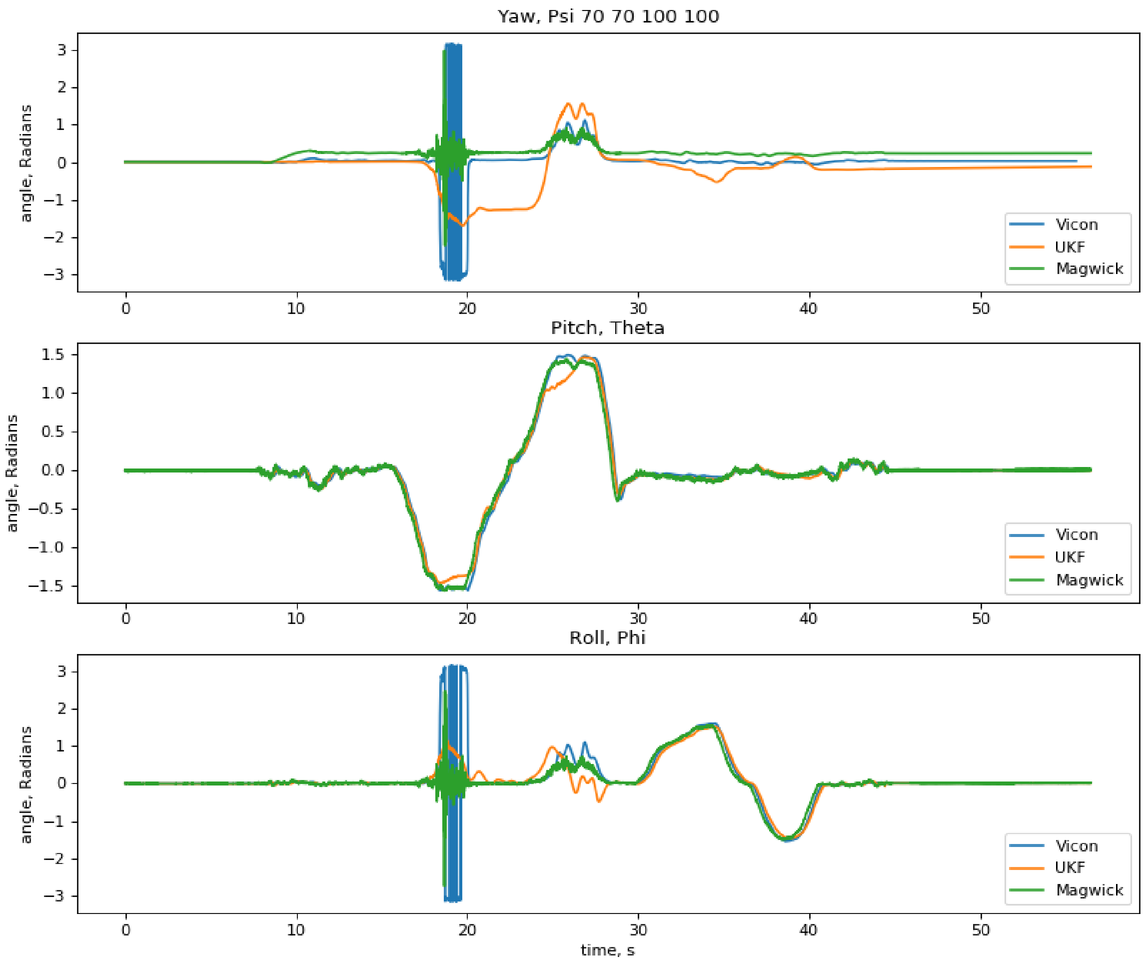

Figure 1. Euler Angle Plots for Test Data Set 1

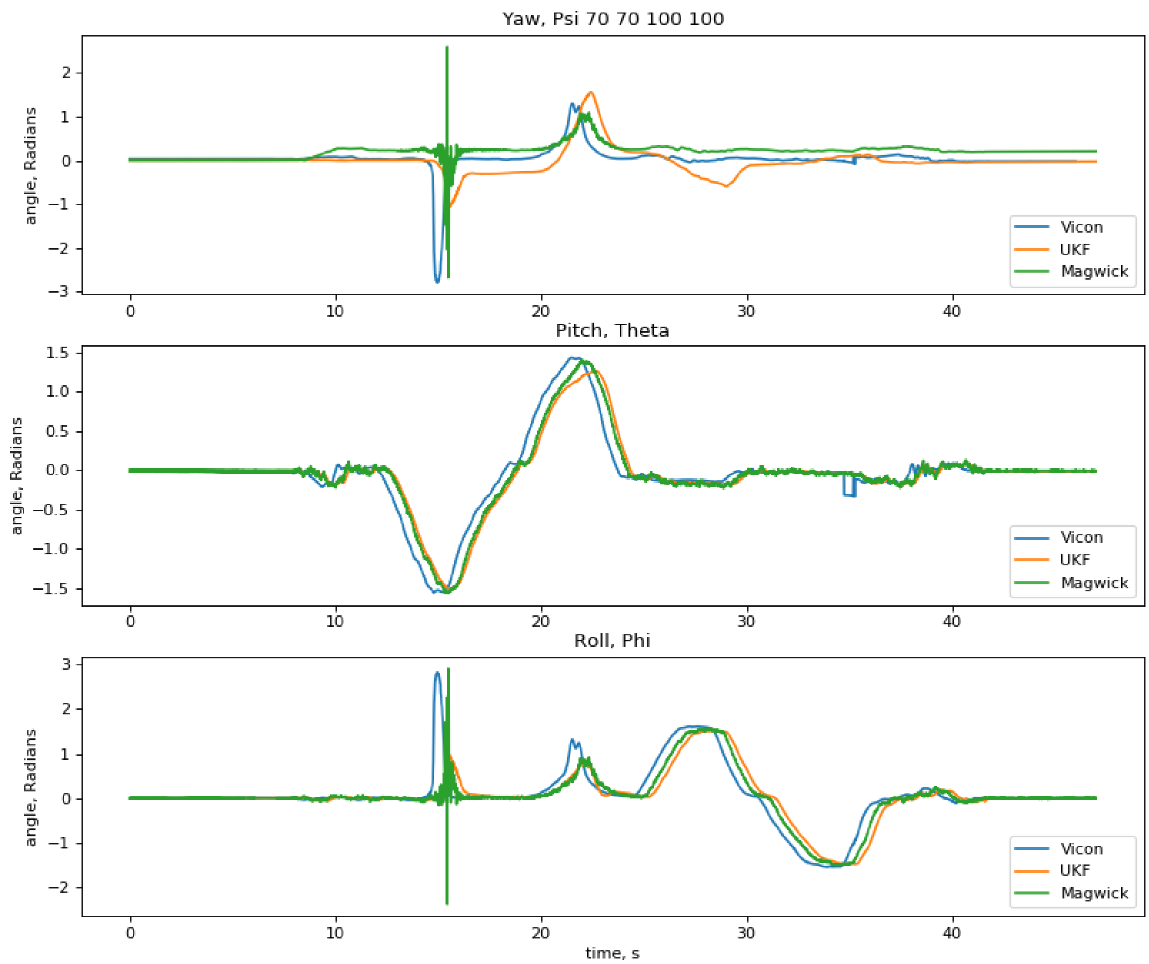

Figure 2. Euler Angle Plots for Test Data Set 2

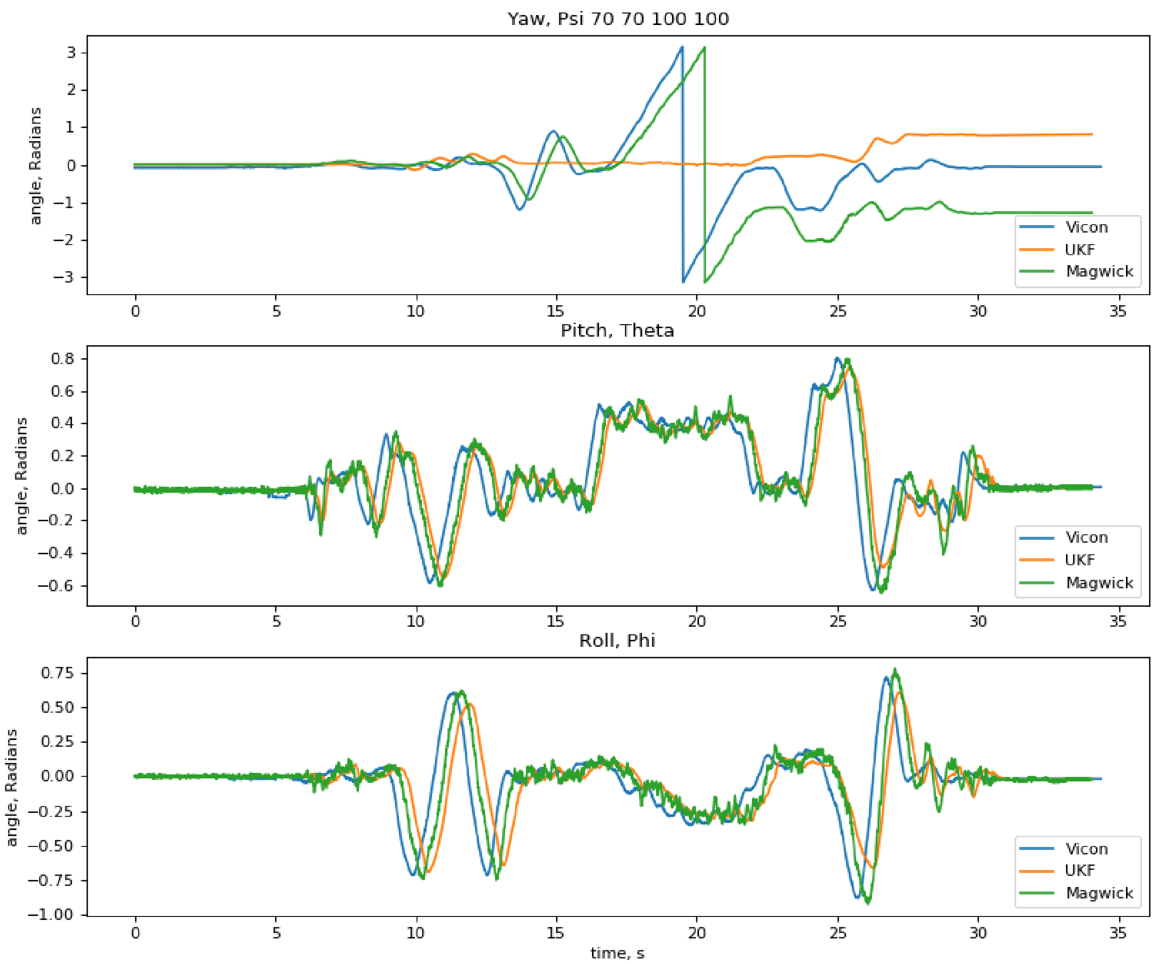

Figure 3. Euler Angle Plots for Test Data Set 3

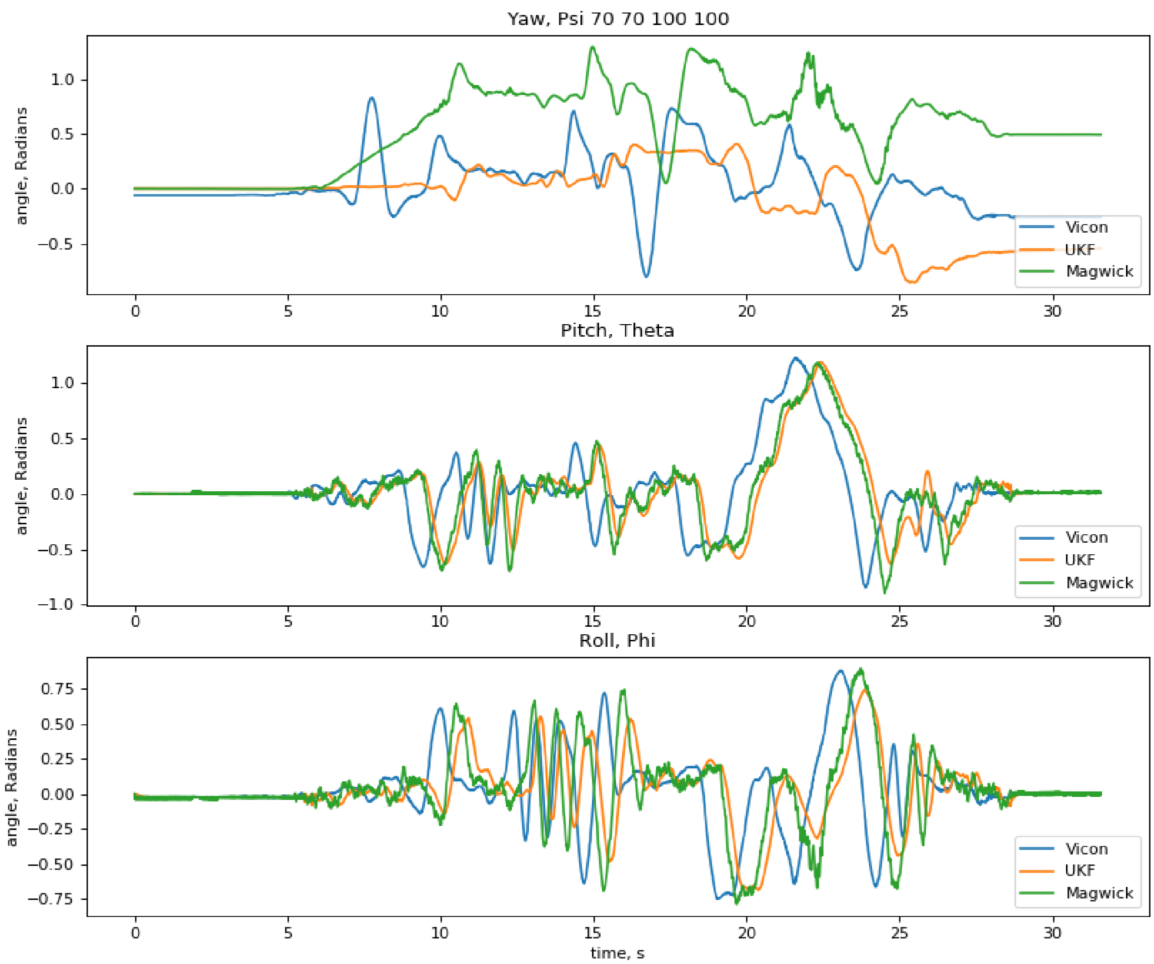

Figure 4. Euler Angle Plots for Test Data Set 4

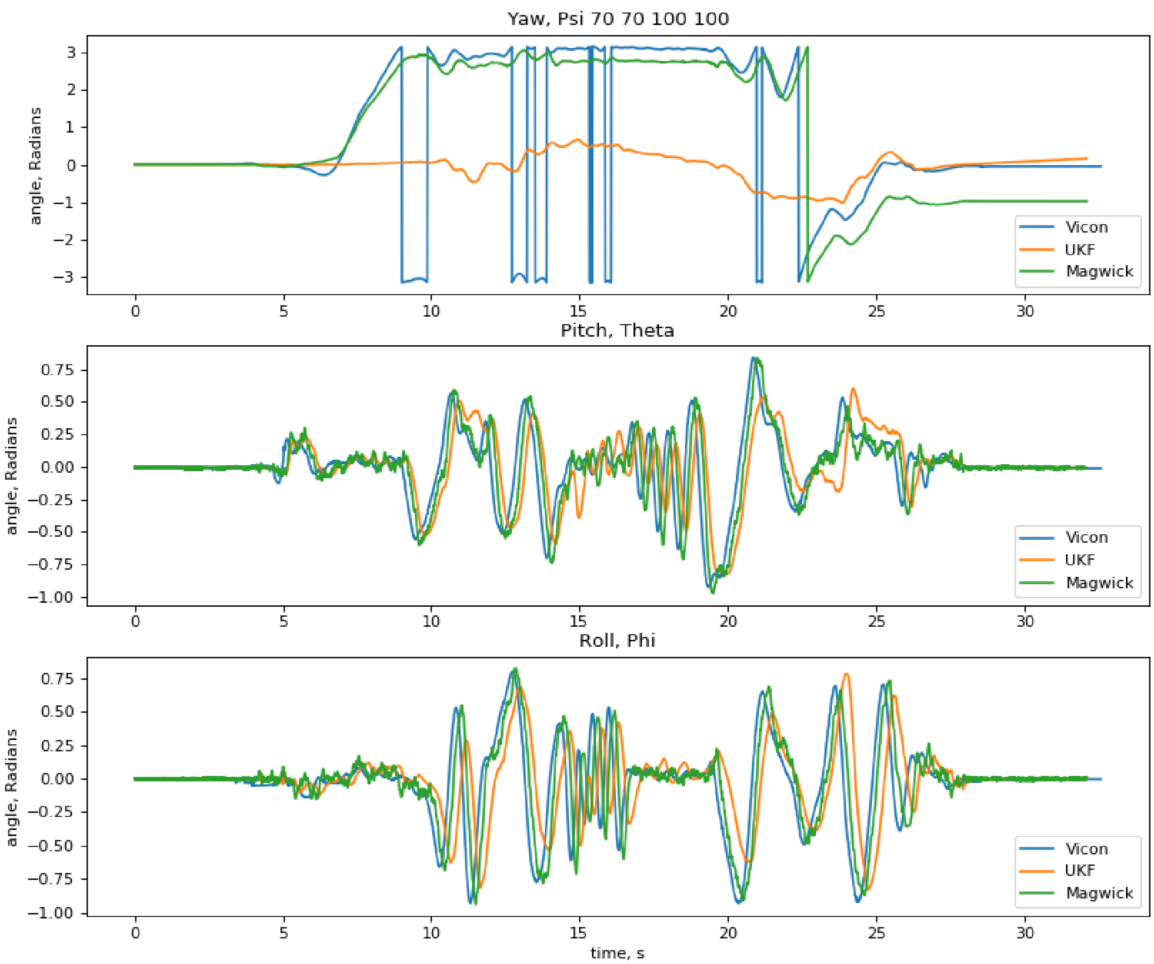

Figure 5. Euler Angle Plots for Test Data Set 5

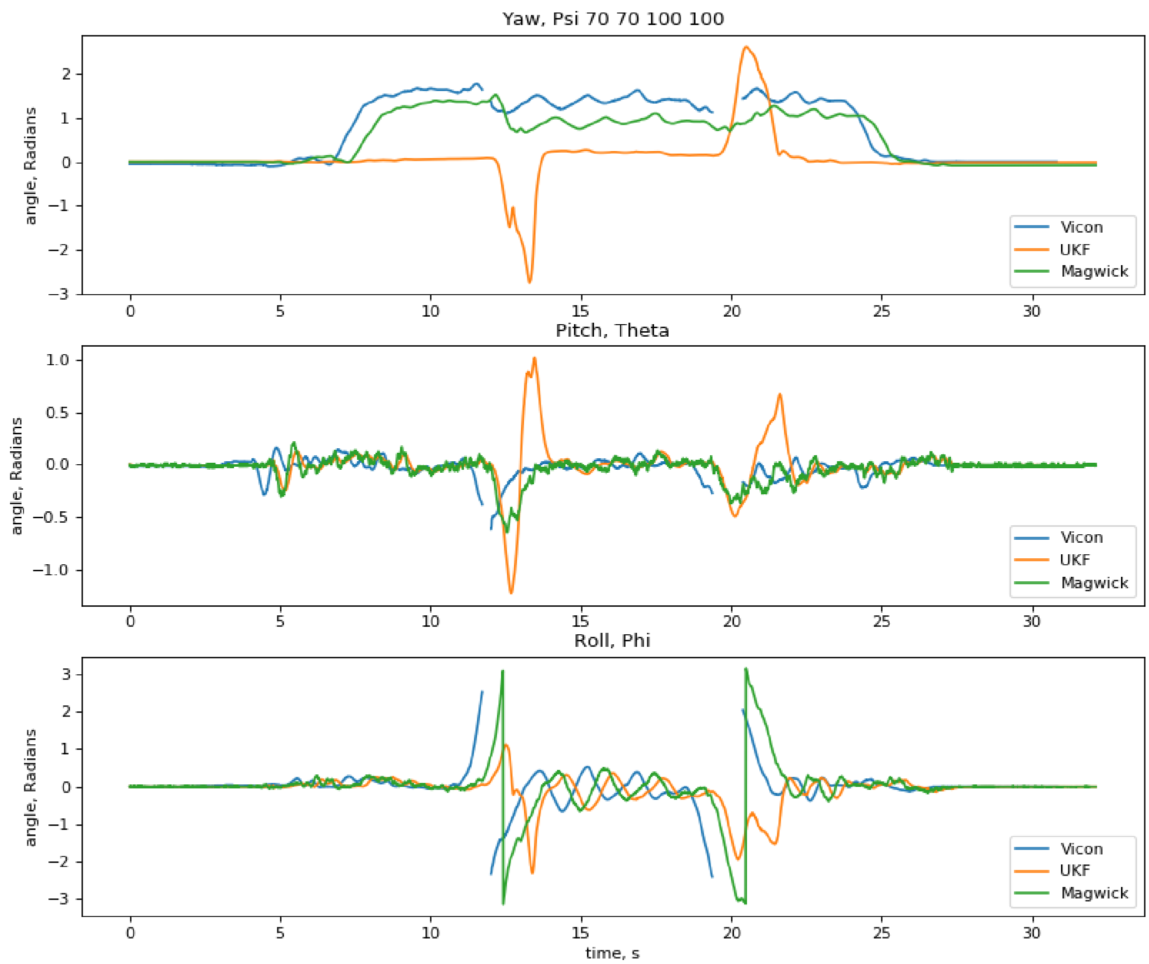

Figure 6. Euler Angle Plots for Test Data Set 6

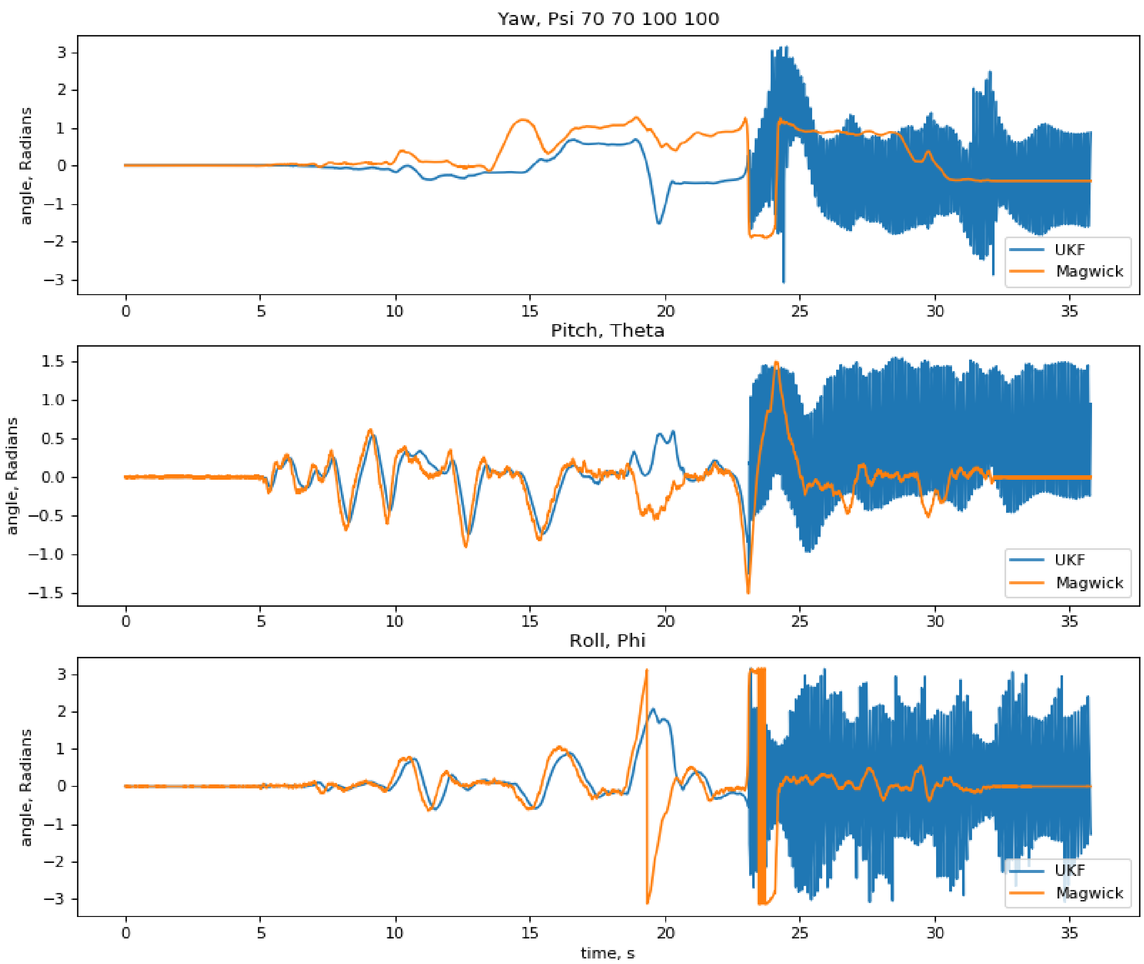

Figure 7. Euler Angle Plots for Data Set 7 - No VICON data included

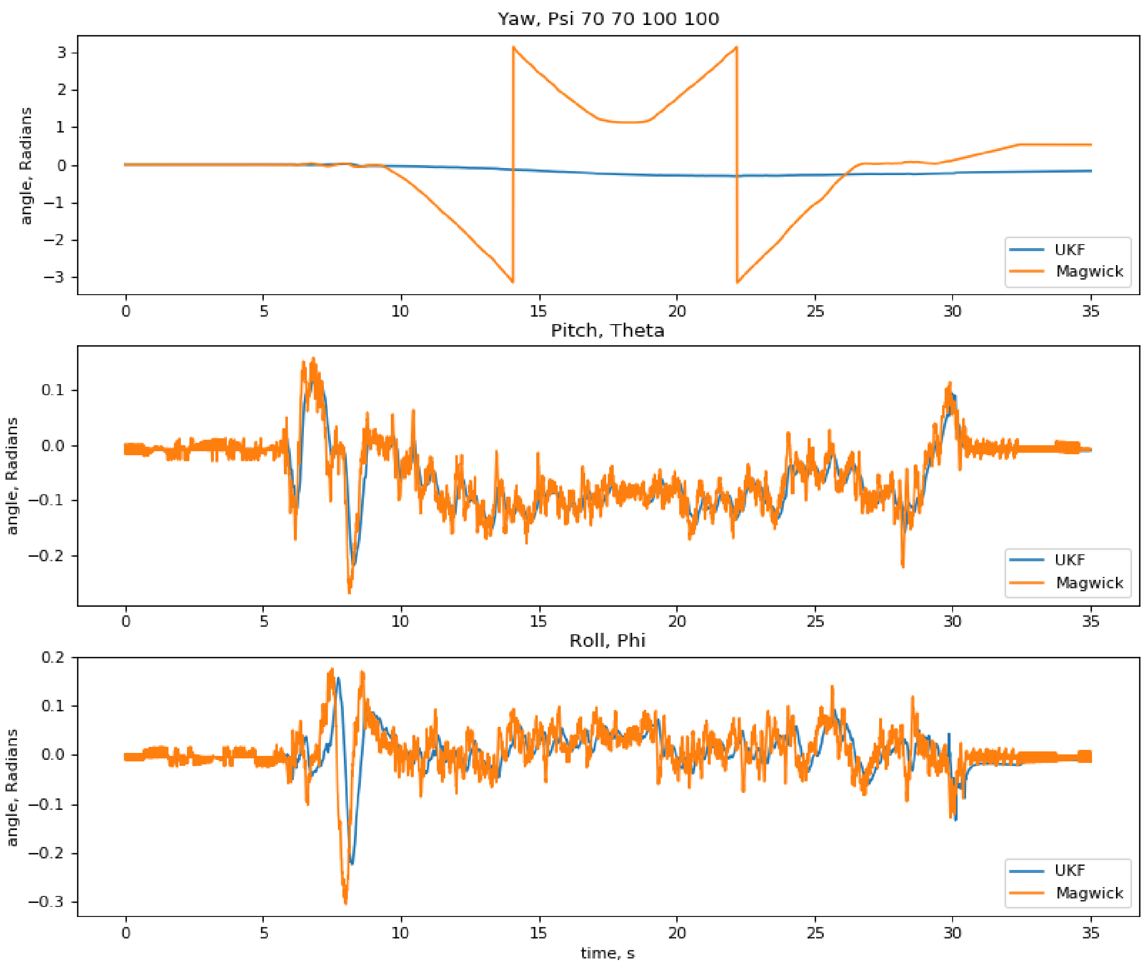

Figure 8. Euler Angle Plots for Data Set 8 - No VICON data included

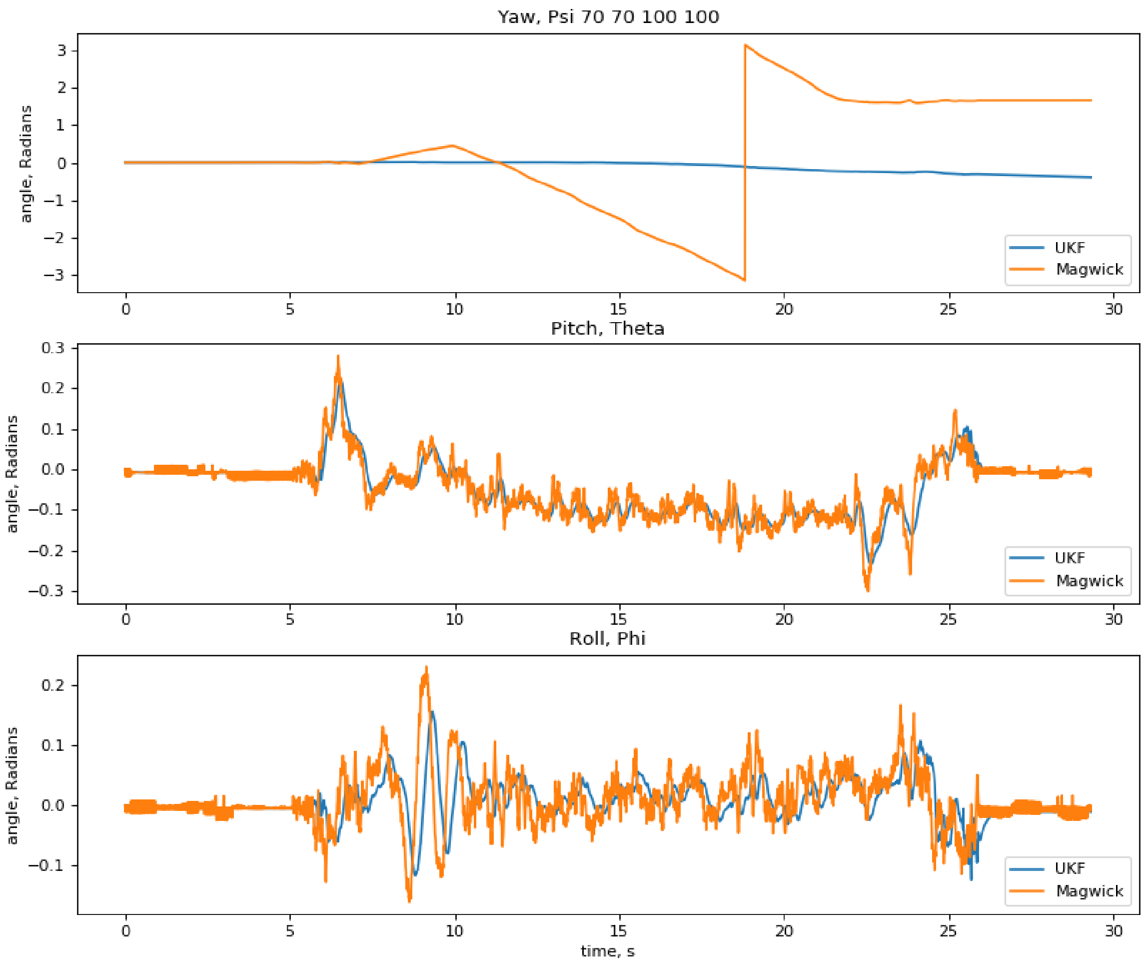

Figure 9. Euler Angle Plots for Data Set 9 - No VICON data included

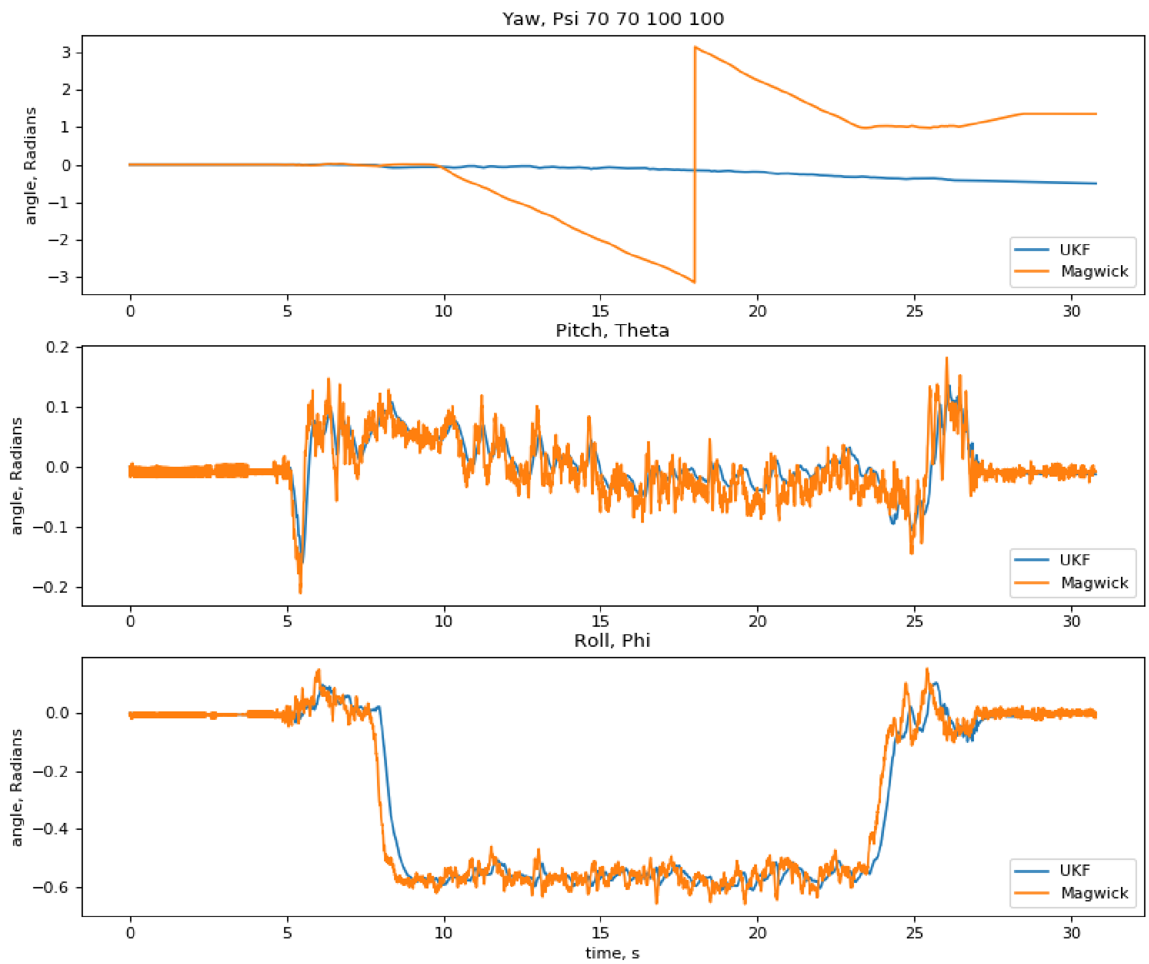

Figure 10. Euler Angle Plots for Data Set 10 - No VICON data included