Design and Development of a Sensor Fusion based Low Cost Attitude Estimator

Fahad Tanveer, Owais Talaat Waheed, Atiq-ur-Rehman
AOCS Section, Satellite Research & Development Centre-Karachi,(SUPARCO) Pakistan
Email: fahadtanveer84, owaistalaat, atiq2k@gmail.com

Abstract—this paper discusses the attitude estimation problem using low cost Micro-Electro-Mechanical System (MEMS) inertial sensors, for which a complementary sensor fusion solution is proposed using the Kalman filter. The system hardware includes three axis gyroscope, accelerometer and magnetometer along with a fixed point DSP controller. The 3-Degrees of freedom (3-DOF) quaternion based algorithm had been initially developed and tested in MATLAB/Simulink. A rapid prototyping approach has been adopted using the MATLAB Real Time Workshop Embedded Coder for direct code generation for the target system. The calibration and testing of the system has been done on a rate table using Real Time Data Exchange (RTDX) for calibrating parameters. The developed system updates at 50Hz with accuracy better than 0.5° which conforms to simulated results.

Keywords— Attitude Estimation, Micro-Electro-Mechanical System (MEMS), inertial sensors, sensor fusion, Kalman filter, gyroscope, accelerometer and magnetometer, 3-Degrees of freedom (3DOF), quaternion

1 INTRODUCTION

The availability of cheap MEMS inertial sensors and high MIPS DSPs has enabled the development of low cost strap down attitude estimator modules. However at present MEMS sensors have more noise and bias drifts than their traditional expensive counterparts, resulting in the accumulation of a reasonable error over time. Fortunately the problem can be mitigated by using complementary sensors along with data fusion schemes to obtain a more optimal attitude estimate. Similar work has been carried out such as in [4], [5] & [6]. Moreover commercially available modules such as those provided by XSENS [8] employ the same techniques and achieve accuracies less than 0.5 degrees.

The goal of this research was to develop a module that can be customized for use in different applications and test different algorithms for increasing performance. In this context a rapid prototyping approach has been adopted in order to reduce development and testing time, considerably reducing cost. This paper presents in detail the implementation of an extended Kalman filter using full quaternion for attitude estimation.

2 SENSOR FUSION

Sensor fusion involves collecting data from various sensors and applying data fusion techniques to obtain optimal measurements. For the case under consideration we predict the gyroscope bias by using an additional attitude estimate obtained via 3 axis magnetometer and accelerometer measurements. To address this sensor fusion problem the kalman filter solution is proposed, however due to the involvement of a non-linear system of equations, the Extended Kalman filter version has been employed.

3 ATTITUDE DYNAMICS

For the dynamics of the attitude of a rigid body, let $\vec{\omega} = [\omega_x\ \omega_y\ \omega_z]^T$ are defined as the instantaneous rotation rates about the body axes x', y', z' respectively.

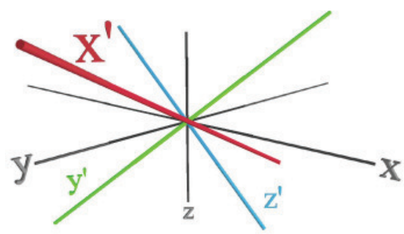

Figure 1 - Reference axes x, y, z and body axes x', y', z'

Than the derivative of the attitude Euler angels is related to $\vec{\omega}$ as follows:

$$\begin{pmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)\sec(\theta) & \cos(\phi)\sec(\theta) \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \quad (1)$$

Where φ, θ, ψ are the roll, pitch and yaw angles respectively and follow the sequence yaw around z', than pitch around y' and finally a roll around x'.
The above equation has to be integrated numerically but is computationally intensive because of the presence of transcendental functions. Moreover there are singularities at $\theta = \pm\frac{\pi}{2}$ which causes a very rapid growth of errors.

The solution to this problem is by employing quaternion dynamics where the attitude quaternion is expressed as

$$q = q_0 + \boldsymbol{i}q_1 + \boldsymbol{j}q_2 + \boldsymbol{k}q_3 = \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} \quad (2)$$

Then attitude dynamics in terms of quaternion can be expressed as

$$\dot{q} = \frac{1}{2}\begin{pmatrix} 0 & -\vec{\omega}^T \\ \vec{\omega} & -\vec{\omega}\times \end{pmatrix}\begin{pmatrix} q_0 \\ \vec{q} \end{pmatrix} \quad (3)$$

The equation is much computationally simple and has no singularities. Moreover at low integration times Δt, assuming constant body rates, the equation is linear and can be solved in closed form yielding:

$$q(t_0 + \Delta t) = Aq(t_0) \quad (4)$$

Where

$$A = \begin{pmatrix} \cos\left[\frac{|w|\Delta t}{2}\right] & -\frac{\omega_1}{|\omega|}\sin\left[\frac{|\omega|\Delta t}{2}\right] & -\frac{\omega_2}{|\omega|}\sin\left[\frac{|\omega|\Delta t}{2}\right] & -\frac{\omega_3}{|\omega|}\sin\left[\frac{|\omega|\Delta t}{2}\right] \\ \frac{\omega_1}{|\omega|}\sin\left[\frac{|\omega|\Delta t}{2}\right] & \cos\left[\frac{|w|\Delta t}{2}\right] & \frac{\omega_3}{|\omega|}\sin\left[\frac{|\omega|\Delta t}{2}\right] & -\frac{\omega_2}{|\omega|}\sin\left[\frac{|\omega|\Delta t}{2}\right] \\ \frac{\omega_2}{|\omega|}\sin\left[\frac{|\omega|\Delta t}{2}\right] & -\frac{\omega_3}{|\omega|}\sin\left[\frac{|\omega|\Delta t}{2}\right] & \cos\left[\frac{|w|\Delta t}{2}\right] & \frac{\omega_1}{|\omega|}\sin\left[\frac{|\omega|\Delta t}{2}\right] \\ \frac{\omega_3}{|\omega|}\sin\left[\frac{|\omega|\Delta t}{2}\right] & \frac{\omega_2}{|\omega|}\sin\left[\frac{|\omega|\Delta t}{2}\right] & -\frac{\omega_1}{|\omega|}\sin\left[\frac{|\omega|\Delta t}{2}\right] & \cos\left[\frac{|w|\Delta t}{2}\right] \end{pmatrix}$$

and $$|\omega| = \sqrt{{\omega_1}^2 + {\omega_2}^2 + {\omega_3}^2}$$

4 THE GYROSCOPE BIAS DRIFTS

In a strapdown Inertial Measurement Unit (IMU) orientation is obtained by integrating the angular rate output of the gyroscopes given the initial orientation. However due to the bias drifts in the gyroscope, error integrates over time. The phenomenon is displayed below.

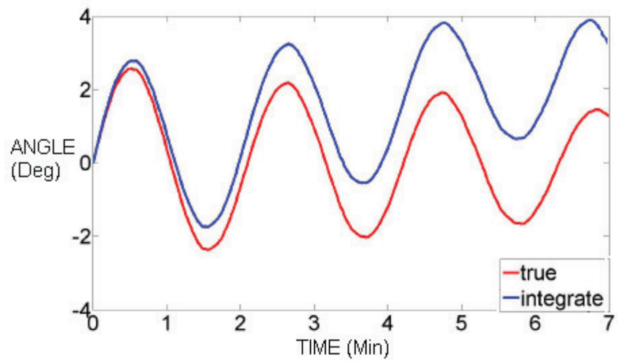

Figure 2 - Rate integration from a gyro with bias

Mathematically the rate measurement from gyros can be given as

$$\vec{z}_\omega = \begin{pmatrix} \omega_1 + b_1 \\ \omega_2 + b_2 \\ \omega_3 + b_3 \end{pmatrix} \quad (5)$$

Where b denotes biases in each axis. The bias error is considerably higher in cheap MEMS based gyroscopes as compared to their expensive traditional counterparts.

The same error can however be eliminated by updating and correcting the bias error using complementary sensors and employing a suitable sensor fusion technique, such as the Kalman filter.

5 THE EXTENDED KALMAN FILTER

The Kalman Filter was initially developed for linear systems in 1960 by R.E Kalman[1] and is actually a dynamically-weighted recursive least-squares algorithm. One way to apply the Kalman filter to a non-linear system of equations is to use the extended Kalman filter where we linearise the system dynamics and the measurement function around the expected state $\vec{\mathbf{x}}_\mathbf{k}(-)$ and then apply the Kalman filter as normal. Table-I lists the implementation equations and is adapted from the reference in [2]. For our problem we need to derive the state transfer and measurement functions and the noise covariance matrices.

6 THE COMPLEMENTARY SENSORS

A 3-axis Magnetometer and a 3-axis accelerometer have been employed as complementary sensors for attitude estimation. Attitude estimate from these two sensors does not drift with time as compared to the integrated rates from the gyros but is corrupted with high frequency noise.

TABLE 1 - THE EXTENDED KALMAN FILTER EQUATIONS

System dynamics	$\vec{x}_k = f_{k-1}(\vec{x}_{k-1}) + \vec{\omega}_{k-1}$ $\vec{\omega}_k \sim N(0, Q_k)$
Measurement	$\vec{z}_k = h_k(\vec{x}_k) + \vec{v}_{k-1}$ $\vec{v}_k \sim N(0, R_k)$
Covariance	$P_k = ([\vec{x}_k - \hat{x}_k][\vec{x}_k - \hat{x}_k]^T)$
State propagation	$\hat{x}_k(-) = f_{k-1}(\hat{x}_{k-1}(+))$
Dynamics linearization	$\emptyset_{k-1} = \frac{\partial f_{k-1}}{\partial x}\Big\vert_{x=\hat{x}_{k-1}(+)}$
Predicted measurement	$\hat{z}_k = h_k(\hat{x}_k(-))$
Measurement linearization	$H_k = \frac{\partial h_k}{\partial x}\Big\vert_{x=\hat{x}_k(-)}$
State covariance propagation	$P_k(-) = \emptyset_{k-1}P_{k-1}(+)\emptyset_{k-1}^T + Q_{k-1}$
Feedback gain	$K_k = P_k(-)H_k^T(H_kP_k(-)H_k^T + R_k)^{-1}$
State update	$\hat{x}_k(+) = \hat{x}_k(-) + K_k(\vec{z}_k - \hat{z}_k)$
State covariance update	$P_k(+) = (I - K_kH_k)P_k(-)$

The accelerometer measures acceleration due to gravity and inertial acceleration in the sensor frame. This is shown as below:

$$\vec{z}_{accels} = T\begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} - g \end{pmatrix} \quad (6)$$

However if an on-board GPS is present or inertial acceleration is negligible as compared to the value of g, than the value of g could be mapped on to the accelerometer sense axes by the following:

$$\vec{\boldsymbol{z}}_{accels} = \begin{pmatrix} (2q_0^2-1)+2q_1^2 & 2q_1q_2+2q_0q_3 & 2q_1q_3-2q_0q_2 \\ 2q_1q_2-2q_0q_3 & (2q_0^2-1)+2q_2^2 & 2q_2q_3+2q_0q_1 \\ 2q_1q_3+2q_0q_2 & 2q_2q_3-2q_0q_1 & (2q_0^2-1)+2q_3^2 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ -g \end{pmatrix} \quad (7)$$

Where the 3x3 matrix is the rotation matrix from the local frame to the sensor frame. This can be simplified further yielding

$$\vec{z}_{accels} = \begin{pmatrix} -2g(-q_0q_2+q_1q_3) \\ -2g(q_0q_1+q_2q_3) \\ -g(-1+2q_0^2+2q_3^2) \end{pmatrix} \quad (8)$$

Similarly magnetometers measure the local magnetic field in the sensor frame, given as

$$\vec{z}_{mags} = T \begin{pmatrix} B_x \\ B_y \\ B_z \end{pmatrix} \quad (9)$$

Where T is the same rotaion matrix as above. The same can be simplified yielding:

$$\vec{z}_{mags} = \begin{pmatrix} B_x(-1+2q_0^2+2q_1^2)+2B_y(q_1q_2+q_0q_3)+2B_z(-q_0q_2+q_1q_3) \\ B_y(-1+2q_0^2+2q_2^2)+2B_x(q_1q_2-q_0q_3)+2B_z(q_0q_1+q_2q_3) \\ B_z(-1+2q_0^2+2q_3^2)+2B_x(q_0q_2+q_1q_3)+2B_y(-q_0q_1+q_2q_3) \end{pmatrix} \quad (10)$$

7 THE IMPLEMENTATION EQUATIONS

The state for the system is given as

$$\vec{x} = [q\ \vec{\omega}\ \vec{b}] \quad (11)$$

Where body rates and biases are assumed to be random walk processes. The state quaternion is initialized by user input or by using a deterministic vector based attitude determination method using the accelerometer and magnetometer inputs. The state transfer function 'f' is than given as

$$\begin{pmatrix} q_{k+1} \\ \vec{w}_{k+1} \\ \vec{b}_{k+1} \end{pmatrix} = \begin{pmatrix} Aq_k \\ \vec{w}_k \\ \vec{b}_k \end{pmatrix} \quad (12)$$

The linearised dynamics 'Φ' can be found by taking the partial derivative of state transfer function, yielding a 10x10 matrix.

For the measurement equation we use the mathematical models of all 3 sensors and is given as

$$h(\vec{x}) = \begin{pmatrix} \vec{z}_{accels} \\ \vec{z}_{mags} \\ \vec{z}_{\omega} \end{pmatrix} \quad (13)$$

This is then linearised by taking the partial derivative yielding

$$H = \begin{pmatrix} 2gq_2 & -2gq_3 & 2gq_0 & -2gq_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2gq_1 & -2gq_0 & -2gq_3 & -2gq_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ -4gq_0 & 0 & 0 & -4gq_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4B_xq_0-2B_zq_2+2B_yq_3 & 4B_xq_1-2B_yq_2+2B_zq_3 & -2B_zq_0+2B_yq_1 & 2B_yq_0+2B_zq_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4B_yq_0+2B_zq_1-2B_xq_3 & 2B_zq_0+2B_xq_2 & 2B_xq_1+4B_yq_2+2B_zq_3 & -2B_xq_0+2B_zq_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4B_zq_0-2B_yq_1+2B_xq_2 & -2B_yq_0+2B_xq_3 & 2B_xq_0+2B_yq_3 & 2B_xq_1-2B_yq_2+4B_zq_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \quad (14$$

Now the process noise covariance is given as

$$Q = \begin{pmatrix} \sigma_q^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_q^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_q^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_q^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_\omega^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_\omega^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_\omega^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_b^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_b^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_b^2 \end{pmatrix} \quad (15)$$

Where σ_q is the expected quaternion system noise, σ_ω and σ_b are the random walk on the body rates and gyro biases respectively.

Similarly the sensor or measurement noise covariance is given by

$$R = \begin{pmatrix} \sigma_{accel}^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{accel}^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{accel}^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{mag}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{mag}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{mag}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_\omega^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_\omega^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_\omega^2 \end{pmatrix} \quad (16)$$

Where σ_{accel} , σ_{mag} and σ_ω are the standard deviations of the expected noise of the accelerometers, magnetometers and rate gyros respectively.

8 MATLAB/SIMULINK IMPLEMENTATION

A model based, rapid prototyping approach had been adopted for the project. For the said purpose Simulink was selected for algorithm testing and source code generation for the target system. This enabled to avoid the tedious source code development and testing phases, minimizing the overall prototyping time.

The main Filter algorithm was written in Embedded Matlab block so as to facilitate code generation for the target system using the real time embedded coder [7]. The remaining blocks of the final model consisted of signal conditioning blocks which applied the scale factors, offsets and alignment matrices on the raw sensors data. In addition there were the target support package blocks; the target initialization, the ADC and the Serial communication blocks. Figure 3 illustrates the Simulink implementation.

The first step in the process was the verification of the developed algorithm. This was achieved using a calibrated XSENS motion sensor, on Simulink itself. The Matlab interface routine provided with XSENS SDK was customised for the application to run on Simulink in a Matlab Function call block. The imported data stream consisted of raw data from all 3 sensors' and the computed Euler angles at 50Hz. The algorithm was applied on the sensors data in real time and the results were compared with the Euler angles computed by the motion sensor itself. This provided a check for the correctness of our algorithm.

Once the algorithm was tested in simulink, it was now possible to embed the algorithm in the DSP and perform a processor-in-the-loop test. Our selected DSP was fixed point, however with the help of emulated floating point libraries it is possible to run code in floating point but we had to do this in single precision format compromising our accuracy. However once the algorithm was embedded and the results compared, only negligible difference was found.

9 THE SYSTEM HARDWARE

The developed hardware consists of a DSP based processing module upon which a sensor interface board is mounted consisting of all 3 sensors. These sensors provide analogue data which after passing through a signal conditioning network is delivered to the 12-bit ADC channels of the processing module. The algorithm is updated at a frequency of 50Hz with the gyros sampled every time. However accelerometers and magnetometers can be sampled at a much lower frequency of around 5Hz.

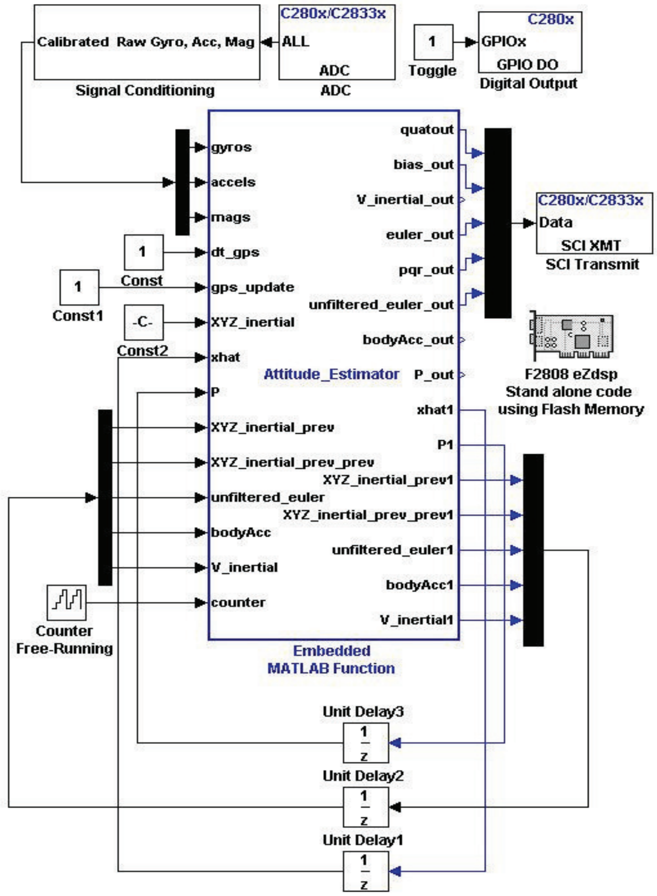

Figure 3 - Simulink Implementation of the embeddable model

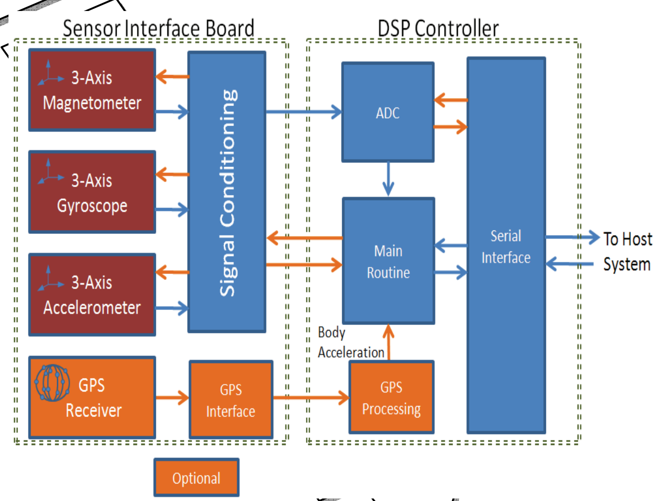

Figure 4 - Hardware Block Diagram

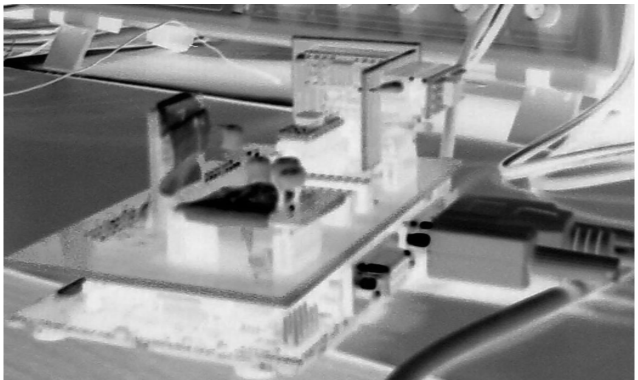

Figure 5 - A picture of the initial prototype

Data exchange between the embedded system and the PC is through a serial interface, however initial debugging and calibration has been done using the RTDX protocol via a JTAG debugger.

10 SYSTEM CALIBRATION AND TESTING

The system calibration has been done using a Hardware-in-the-loop (HIL) test bed employing a rate table. However before that could be done we also had to estimate the process and measurement noise covariance matrices for our developed system. The estimation of the measurement noise covariance matrix was done by analysing the static sensor data and calculating the standard deviation. They were estimated to be , and . The process noise coefficients were dependent upon our overall system requirements. Initially they were estimated and later tuned for better results and were found to be , and .

The next thing to do was to calibrate individual sensor in terms of scale factors, offsets and alignment matrices (for removing non-orthogonality). This was achieved using a helmholtz coil for the magnetometer, rate table for gyros and tilt table for accelerometers.

The algorithm also requires the ambient magnetic field unit vector and the value of g in the local co-ordinate frame which we have selected as the North-East-Down (NED) frame. These values are calculated and embedded into the system.

Finally the whole system is calibrated and tested using a Simulink based HIL testbed and visualization tool. The complete hardware is placed on the rate table and is given a known rotation profile. The estimated attitude is than transferred via a custom serial protocol to the simulink testbed, upon which errors can be calculated. If required calibration parameters can be changed in real time using the RTDX display panel accelerating the system calibration.

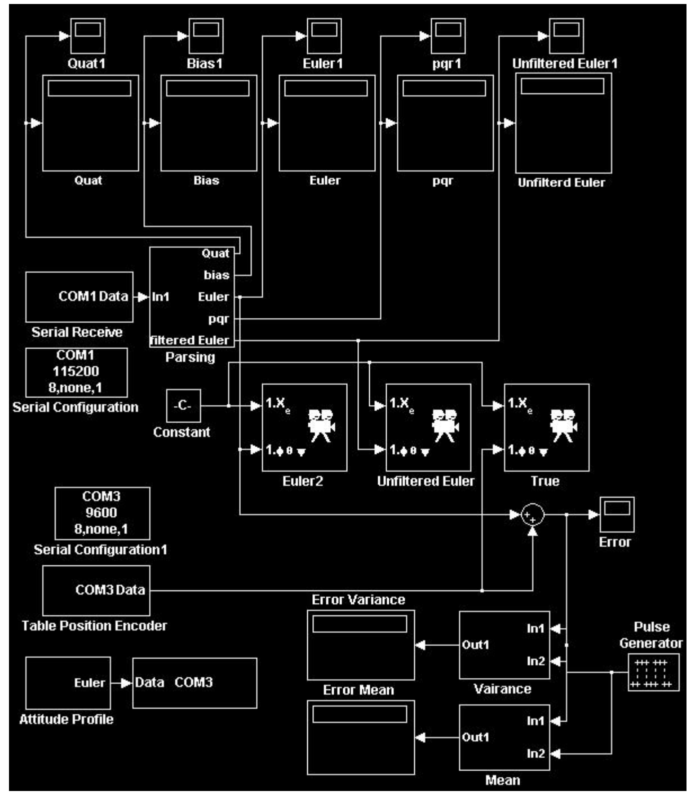

Figure 6 - Simulink Test Bench

MATLAB animation blocks have been customized and are used as a visualization tool in the test bench. This is a very helpful add-on and helped especially in the early calibration and algorithm verification phase, where errors were huge and it was much simpler to interpret the attitude from a 3D objects orientation in the figure rather than inferring graphs of the 3 Euler angles.

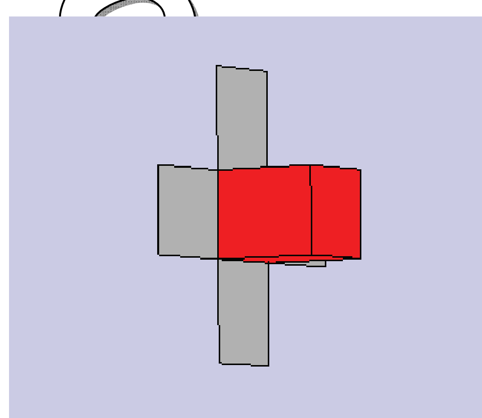

Figure 7 - The 3D Visualization

11 PERFORMANCE RESULTS

The system's static accuracy and was found to be 0.29 degrees. Initially for dynamic movements of over 20deg/sec the estimated output was seen to considerably lag with delayed settling. This issue was however solved by fine tuning the process and measurement noise coefficients. The following graph illustrates the performance results. The errors above 0.3 degrees show regions of movement.

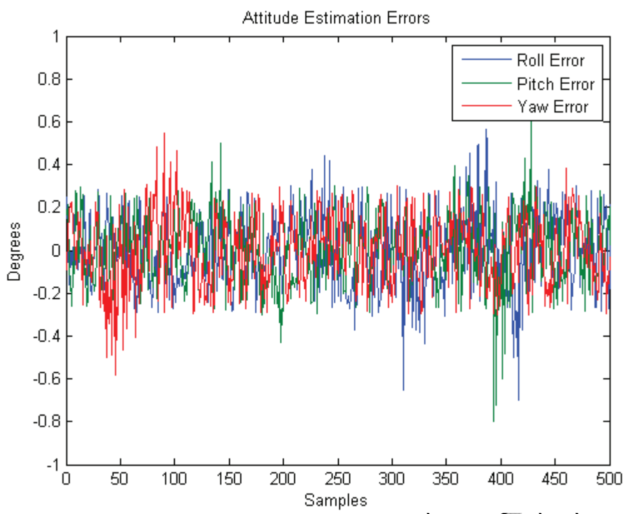

Figure 8 - Attitude Estimation Errors

PROPOSED FUTURE WORK

The paper demonstrated the use of a full quaternion filter for attitude estimation. However it is still much complex and limits the maximum update rate to only 50Hz. Other less computationally intensive algorithms does exist such as the error quaternion implementation which will be able to increase the update rates to even 100Hz. This will considerably improve the dynamic accuracy.

REFERENCES

[1] R. E. Kalman, A new approach to linear Filtering and prediction problems," Transactions of the ASME Journal of Basic Engineering, no. 82 (Series D), pp. 35-45, 1960.

[2] M. S. Grewal and A. P. Andrews, Kalman Filtering: Theory and Practice Using MATLAB. Wiley-Interscience, January 2001.

[3] J. B. Kuipers, Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace and Virtual Reality. Princeton University Press, August 2002.

[4] Tayebi, A. McGilvray, S. Roberts, A. Moallem, “Attitude estimation and stabilization of a rigid body using low-cost sensor” Decision and Control, 2007 46th IEEE Conference, pp. 6424 – 6429, Dec 2007

[5] Batista, P. Silvestre, C. Oliveira, P. Cardeira, B., “Low-cost Attitude and Heading Reference System: Filter design and experimental evaluation”, Robotics and Automation (ICRA), 2010 IEEE International Conference, pp. 2624-2629, May 2010.

[6] David Jurman, Marko Jankovec, Roman Kamnik and Marko Topič, “Calibration and data fusion solution for the miniature attitude and heading reference system”, Sensors and Actuators A: Physical Volume 138, Issue 2, , Pages 411-420,August 2007

[7] http://www.mathworks.com/products/rtwembedded/

[8] http://www.xsens.com/