Paper

Software algorithms in air data attitude heading reference systems

Michael E. Greene and Victor Trent

The authors

Michael E. Greene is a Professor at Auburn University, Auburn, Alabama, USA
Victor Trent is a Chief Engineer at Archangel Systems, Inc., Auburn, Alabama, USA

Keywords

Programming and algorithm theory, Fuzzy logic

Abstract

An introduction to air data attitude heading reference systems (ADAHRS) is presented along with the developments and discussion of software algorithms used in such systems. Both Kalman filtering and fuzzy logic adaptive signal processing are presented and discussed. Quaternions are used in this paper as the basis for the discussion of both algorithms. Flight results from Archangel Systems AHR150 ADAHRS are also presented and discussed.

Electronic access

The Emerald Research Register for this journal is available at
http://www.emeraldinsight.com/researchregister

The current issue and full text archive of this journal is available at
http://www.emeraldinsight.com/0002-2667.htm

Aircraft Engineering and Aerospace Technology
Volume 75 · Number 5 · 2003 · pp. 470–476
 · ISSN 0002-2667
DOI 10.1108/00022660310492546

Introduction

Air data attitude heading reference systems (ADAHRS) are used extensively in aircraft navigation and control systems as sources for inertial data to drive primary flight displays, navigation displays and autopilots. The combination of attitude heading data with air data results in a more robust solution and is becoming the standard core of most modern aircraft (Archangel, 2003; Avidyne, 2003; Meggitt, 2003). ADAHRS are used throughout aviation from single engine propeller general aviation craft to multi-engine, turbine business aircraft to helicopters to air transport craft. ADAHRS are increasingly being utilized in military retrofit projects worldwide.

A typical ADAHRS is the Archangel System AHR150 as shown in Figure 1. This system is composed of two boxes: an inertial sensor unit (ISU) and a remote magnetic sensor unit (MSU). The ISU (AHR150-1) contains strapdown MEMS gyros, MEMS accelerometers as well as pitot and static pressure sensors. Company proprietary mathematical algorithms are used to extract and output the attitude, angular rates, angle and acceleration. Multiple processors are used to ensure against unannounced failures of the unit.

Higher-end ISU's replace the MEMS gyros with fiber optic gyros (FOG) for increased accuracy, but at the cost of increased weight and price. Typical units output data on ARINC 429, ARINC 629 or MIL1553 busses as well as RS422 busses.

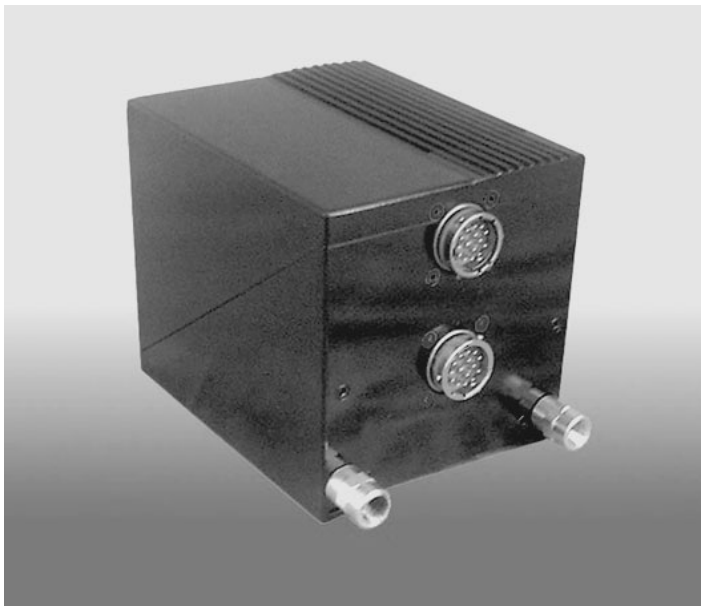
The MSU contains either two or three-axis sensors for the earth's magnetic field and is used to slave the ADAHRS heading to magnetic. Both coil type and magneto-resistive sensors are used in MSU's. Some units have analog interfaces (Goodrich Sensors, 2003), while newer ones (Archangel, 2003) use digital interfaces such as RS422.

Archangel Systems' three-axis MSU (AHR150-2) uses solid state magneto-resistive sensors with a local processor so that the unit outputs a serial data stream with error checking codes. This unit also contains longitudinal and lateral accelerometers, which are used in the field calibration of the device.

Most ADAHRS use an algorithm known as indirect Kalman filtering (Brown, 1983; Roumeliotis and Bekey, 1999; Roumeliotis

Aircraft Engineering and Aerospace Technology
Volume 75 · Number 5 · 2003 · 470–476

Figure 1 AHR150-1



et al., 1999) to determine the changes in gyro bias from predetermined temperature characteristics as well as to propagate the errors in the inertial angles. The errors in inertial angle can be propagated directly, through a direction cosine matrix or through quaternions. Updates to the Kalman filtering process are obtained using accelerometer data corrected by air data.

The AHR150 is different in that it uses a proprietary algorithm known as fuzzy logic adaptive signal processing (FLASP) to propagate the errors in the inertial angles directly as well as the changes in the gyro bias characteristics. Updates to the FLASP process are, like Kalman filtering, obtained using accelerometer data corrected by air data.

Measurement system

The measurement system of the ADAHRS is in the body frame while the desired output angles and rates are in the inertial frame. A North, East, Down coordinate system is used. The relationship between the inertial and body frames is shown in 2 and can be used to derive the measured body rates in terms of the Euler angle rates (Table I and Figure 2).

$$\omega_x = \dot{\Phi} - \dot{\Psi}\sin\Theta \quad (1)$$

$$\omega_y = \dot{\Theta}\cos\Phi + \dot{\Psi}\cos\Theta\sin\Phi \quad (2)$$

$$\omega_z = -\dot{\Theta}\sin\Phi + \dot{\Psi}\cos\Theta\cos\Phi \quad (3)$$

Table I Variable definitions

Name	Definition
ω_x	Body roll rate
ω_y	Body pitch rate
ω_z	Body yaw rate
U	Body X velocity (longitudinal)
V	Body Y velocity (lateral)
W	Body Z velocity (vertical)
$\dot{\Phi}$	Inertial roll rate
$\dot{\Theta}$	Inertial pitch rate
$\dot{\Psi}$	Inertial yaw rate
α	Angle of attack (in body axis)
β	Side slip angle (in body axis)
V_T	Total velocity

At the CG of the aircraft the X, Y and Z axes measured accelerations are due to linear acceleration terms, the cross product of rotation velocities and linear velocities and gravity terms.

$$ax_{cg} = \dot{U} + \omega_y W - \omega_z V + g\sin\Theta \quad (4)$$

$$ay_{cg} = \dot{V} + \omega_z U - \omega_x W - g\cos\Theta\sin\Phi \quad (5)$$

$$az_{cg} = \dot{W} + \omega_x V - \omega_y U - g\cos\Theta\cos\Phi \quad (6)$$

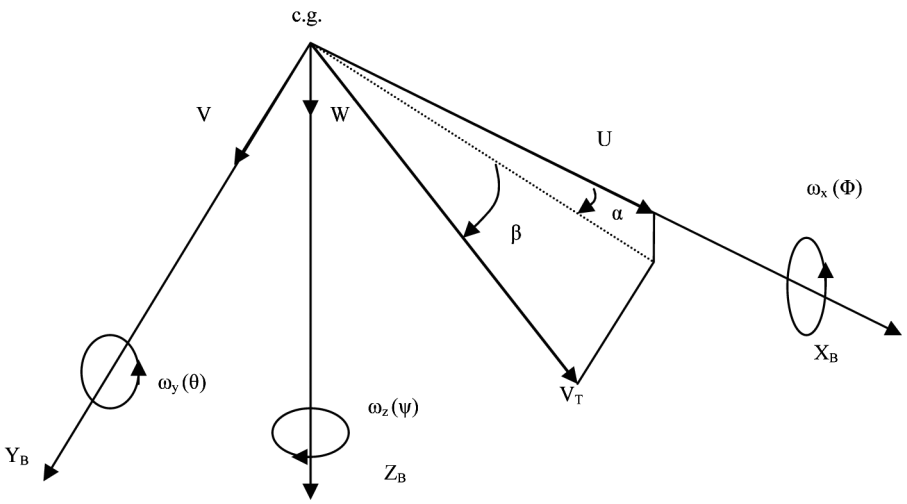
In an ADAHRS, since air data is available, the parameters of U, V and W can be measured or estimated. This allows an independent (although noisy) calculation of the pitch and roll angles using acceleration measurements. These calculations along with the yaw angle from a magnetic unit are used to form errors. The errors thus formed are typically used to remove gyro bias errors in MEMS gyros.

Quaternion representations

Since Euler angles become undefined at certain points (such as pitch of 90°), other representations such as the direction cosine matrix or quaternions are used. Due to the lower overhead involved, quaternions are typically used. An excellent reference for an overview of quaternions is presented by Chou (1992). In order to obtain the quaternion from an Euler angle representation, the following procedure is followed. First, a quaternion is created that corresponds to each of the three rotations. In yaw, the quaternion becomes:

$$\vec{q}_\Psi = \left[\cos\left(\frac{\Psi}{2}\right), 0, 0, \quad -\sin\left(\frac{\Psi}{2}\right)\vec{k}\right]^{\mathrm{T}} \quad (7)$$

Figure 2 Coordinate axes



In pitch, the quaternion becomes:

$$\vec{q}_\Theta = \left[\cos\left(\frac{\Theta}{2}\right), 0, \quad -\sin\left(\frac{\Theta}{2}\right)\vec{j}, 0\right]^{\mathrm{T}} \tag{8}$$

In roll, the quaternion becomes:

$$\vec{q}_\Phi = \left[\cos\left(\frac{\Phi}{2}\right), \quad -\sin\left(\frac{\Phi}{2}\right)\vec{i}, 0, 0\right]^{\mathrm{T}} \tag{9}$$

A vector in the inertial frame, using these three quaternion rotations, becomes a vector in the body frame:

$$\begin{aligned}\hat{b} &= \vec{q}_\Psi \otimes \left(\vec{q}_\Theta \otimes \left(\vec{q}_\Phi \otimes \hat{a} \otimes \vec{q}_\Phi^{\star}\right) \otimes \vec{q}_\Theta\right) \otimes \vec{q}_\Psi^{\star} \\ &= \left(\vec{q}_\Psi \otimes \vec{q}_\Theta \otimes \vec{q}_\Phi\right) \otimes \hat{a} \otimes \left(\vec{q}_\Psi \otimes \vec{q}_\Theta \otimes \vec{q}_\Phi\right)^{\star} \\ &= \left[\left(q_0^2 - \vec{q}^{\mathrm{T}}\vec{q}\right)I_3 + 2\left(\vec{q}\vec{q}^{\mathrm{T}} + q_0\tilde{q}\right)\right]\hat{a}\end{aligned} \tag{10}$$

where

$$\tilde{q} = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix} \tag{11}$$

and

$$\vec{q} = [q_0, q_1, q_2, q_3]^{\mathrm{T}} = [q_0, \vec{q}]^{\mathrm{T}} \tag{12}$$

$$\vec{q}^{\star} = [q_0, -q_1, -q_2, -q_3]^{\mathrm{T}} = [q_0, -\vec{q}]^{\mathrm{T}} \tag{13}$$

The final matrix relating the vector **a** to the vector **b** in equation (10) is the DCM, C, which relates vectors in the inertial space to vectors in the body space. The expression for C becomes:

$$C = \begin{bmatrix} 1-2\left(q_2^2+q_3^2\right) & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & 1-2\left(q_1^2+q_3^2\right) & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & 1-2\left(q_1^2+q_2^2\right) \end{bmatrix} \tag{14}$$

The quaternions can be propagated according to:

$$\dot{\vec{q}} = \frac{1}{2}\Omega(\omega_{bi})\vec{q}(t) \tag{15}$$

where

$$\Omega(\omega_{bi}) = \begin{bmatrix} 0 & \omega_x & \omega_y & \omega_z \\ -\omega_x & 0 & \omega_z & -\omega_y \\ -\omega_y & -\omega_z & 0 & \omega_x \\ -\omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \tag{16}$$

is a skew symmetric matrix composed of the body rotational velocities as measured by the sensors (Roumeliotis and Bekey, 1999).

The quaternion of equation (10) is defined as the rotation to the inertial frame with respect to the body frame. Equation (15) can be numerically integrated for each sample period and the Euler angles (roll, pitch and yaw) are given as:

$$\begin{aligned}\Phi &= \arctan\left(\frac{C[2,3]}{C[3,3]}\right) \\ &= \arctan\left(\frac{2(q_2q_3 + q_0q_1)}{1 - 2\left(q_1^2 + q_2^2\right)}\right)\end{aligned} \tag{17}$$

$$\begin{aligned}\Theta &= -\arcsin\left(C[1,3]\right) \\ &= -\arcsin\left(2(q_1q_3 - q_0q_2)\right)\end{aligned} \tag{18}$$

Aircraft Engineering and Aerospace Technology
Volume 75 · Number 5 · 2003 · 470–476

$$\Psi = \arctan\left(\frac{C[1,2]}{C[1,1]}\right) = \arctan\left(\frac{2(q_1q_2 + q_0q_3)}{1 - 2\left(q_2^2 + q_3^2\right)}\right). \quad (19)$$

Quaternion errors

After removal of air data terms, pitch is calculated from equation (4) and roll from equations (5) and (6). For this discussion, these variables will be called "Inertial" pitch and "Inertial" roll. Along with yaw as determined from the MSU, a quaternion with the subscript, I, will be formed according to equation (7) through (13).

The quaternion error can then be calculated as:

$$\vec{\varepsilon}_q = [\varepsilon_{q0} \;\; \varepsilon_{q1} \;\; \varepsilon_{q2} \;\; \varepsilon_{q3}]^{\mathrm{T}} = q_{\mathrm{I}} \otimes q_G^{\star} \quad (20)$$

It can be shown (Roumeliotis and Bekey, 1999) that the time rate of change of the quaternion error is given as:

$$\frac{\mathrm{d}}{\mathrm{d}t}\vec{\varepsilon} = S(\omega_{bi}) - \frac{1}{2}(\vec{\varepsilon} + n) \quad (21)$$

where

$$S(\omega_{bi}) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \quad (22)$$

The vector composed of quaternion error is actually given by:

$$\vec{\varepsilon} = [\varepsilon_{q1} \;\; \varepsilon_{q2} \;\; \varepsilon_{q3}]^{\mathrm{T}} \quad (23)$$

Assuming small quaternion error (small rotational differences between the true quaternion rotation and the estimated rotation), the real part of the quaternion error, ε_{q0}, is approximately 1 and:

$$\dot{\varepsilon}_{q0} = 0 \quad (24)$$

Indirect Kalman filtering

Kalman filtering is a model-based process for obtaining state estimations. In this case, the filter estimates the quaternion errors as well as the gyro drift. Several excellent references covering Kalman filters are available (Brown, 1983; Gelb, 1974; Grewal and Andrews, 1993; Kalman, 1960). For the estimator to be designed, the system must first be put in the standard state variable format driven by Gaussian noise called process noise and have measures corrupted by independent Gaussian noise called measurement noise.

$$x_{k+1} = \phi_k x_k + B_k w_k \quad (25)$$

$$z_k = Hx_k + v_k \quad (26)$$

The matrix Φ_{k-1} is updated for each sample period using the values of ω_x, ω_y and ω_z, which were determined during the previous sample period.

The state vector in the system is usually extended with a process to model the Markovian nature of gyro noise

$$\begin{bmatrix} \dot{\varepsilon}_{\omega x} \\ \dot{\varepsilon}_{\omega y} \\ \dot{\varepsilon}_{\omega z} \end{bmatrix} = \begin{bmatrix} -\beta_{\omega x} & 0 & 0 \\ 0 & -\beta_{\omega y} & 0 \\ 0 & 0 & -\beta_{\omega z} \end{bmatrix} \times \begin{bmatrix} \varepsilon_{\omega x} \\ \varepsilon_{\omega y} \\ \varepsilon_{\omega z} \end{bmatrix} + \begin{bmatrix} w_4 \\ w_5 \\ w_6 \end{bmatrix} \quad (27)$$

and the constant bias terms (Brown, 1983):

$$\begin{bmatrix} \dot{\varepsilon}_{\omega xc} \\ \dot{\varepsilon}_{\omega yc} \\ \dot{\varepsilon}_{\omega yz} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \varepsilon_{\omega x} \\ \varepsilon_{\omega y} \\ \varepsilon_{\omega z} \end{bmatrix} \quad (28)$$

The Kalman filter process consists of three steps (Brown, 1983). The first is projecting the state vector ahead and computing an *a priori* covariance matrix

$$\hat{x}_k^- = f\left(\hat{x}_{k-1}^+, k-1\right) \quad (29)$$

$$\hat{z}_k = H\hat{x}_k^- \quad (30)$$

$$P_k^- = \Phi_{k-1}P_{k-1}^+\Phi_{k-1}^{\mathrm{T}} + B_{k-1}Q_{k-1}B_{k-1}^{\mathrm{T}} \quad (31)$$

Step two is to compute the Kalman gain as:

$$K_k = P_k^- H_k^{\mathrm{T}}\left(H_k P_k^- H_k^{\mathrm{T}} + R_k\right)^{-1} \quad (32)$$

Step three is to update the state vector with the measurement and project the covariance forward.

$$\hat{x}_k^+ = \hat{x}_k^- + K_k(z_k - \hat{z}_k) \quad (33)$$

$$P_k^+ = (I - K_k H_k)P_k^- \quad (34)$$

Once the estimates of the quaternion errors are obtained, the output quaternion can be rotated by the error as:

Aircraft Engineering and Aerospace Technology
Volume 75 · Number 5 · 2003 · 470–476

$$\vec{q}_T = \hat{\varepsilon}_q \otimes K_F \otimes q_G \tag{35}$$

where K_F is a gain matrix used to scale the quaternion corrections. These gains may be adjusted to affect the system performance. Likewise, the estimates for gyro biases are subtracted from the measurements before usage.

Although the Kalman filter routine is widely used, problems do exist. It should be noted that a matrix inverse is required for each step of the filter, which is computationally expensive and can cause convergence problems. Furthermore, convergence of the filter is not guaranteed.

FLASP

Another approach to removing gyro errors is FLASP. FLASP, like fuzzy logic from which it was derived, is a more intuitive process than Kalman filtering. Once again the quaternion errors and gyro biases are calculated by the algorithm and used in an adaptive loop to remove their effects.

The fuzzy estimator consists of a fuzzification process, an inference mechanism, a rule base and a defuzzification process (Mendel, 1995). The fuzzification process assigns a degree of membership to the inputs over the Universe of Discourse. Referring to Figure 3, we see that if the error in Euler angle k is zero, the degree of certainty, μ_0 (center membership function), is 1 and all others are 0. As the error changes, the degree of certainty changes and the other μ have non-zero values. Thus, the errors are encoded by the degree of certainty that they are between certain error bounds.

The values for the error bounds (E_1, E_2) were determined using center clustering techniques (Passino and Yurkovich, 1998) on actual flight data. Likewise, input membership functions are determined for the change in error.

For five error input membership functions and five change in error input membership functions, 25 rules result as show in Table II.

Any membership function with a non- zero degree of certainty is said to be "on" and the corresponding rule is also active. If both the error and change in error were small enough to be within the smallest error bounds ($-E_1$ to $+E_1$ in Figure 3), the linguistic rule is considered on:

If e is 0 and change in e is 0 then *correction is* 0

The certainty of the premise, i, is given by:

$$\mu_i = \min(\mu_{e0}, \mu_{\Delta e0}) \tag{36}$$

In general, the rules are given as:

$$\text{if } \tilde{\mu}_{ei} \text{ is } A^j_{ei} \text{ and } \tilde{\mu}_{\Delta ei} \text{ is } A^k_{\Delta el} \text{ then } \varepsilon_i = g_i(\cdot) \text{ and } \dot{\varepsilon}_i = h_i(\cdot) \tag{37}$$

The symbol, "·", simply indicates argument.

In FLASP, the quaternion error is first reduced to the error in Euler angles:

$$\vec{\varepsilon}_q \rightarrow \{\varepsilon_\phi, \varepsilon_\theta, \varepsilon_\varphi\} \tag{38}$$

The rules of Table II are then applied to each Euler angle error. The output correction for each Euler angle and Euler rate is calculated using a center-of-gravity method:

$$\hat{\varepsilon}_{\text{Eulerangle}} = \frac{\sum_{i=1}^{R} g_i \mu_i}{\sum_{i=1}^{R} \mu_i} \tag{39}$$

and

$$\dot{\hat{\varepsilon}}_{\text{Eulerangle}} = \frac{\sum_{i=1}^{R} h_i \mu_i}{\sum_{i=1}^{R} \mu_i} \tag{40}$$

Corrections to the body rates can then be determined directly from equation (40) using the DCM.

Figure 3 Input membership functions

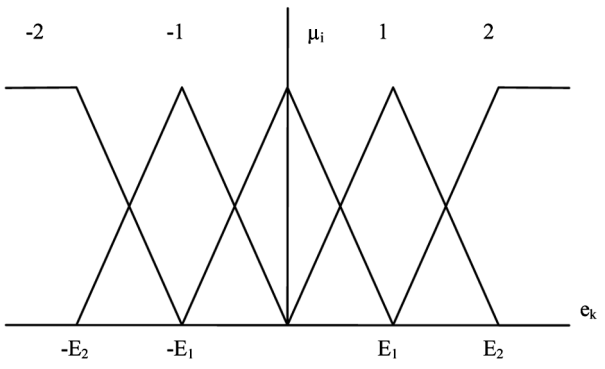


Table II Rule table

	Change in error				
Output	**2**	**1**	**0**	**−1**	**−2**
Error e					
2	2	2	2	1	0
1	2	2	1	0	−1
0	2	1	0	−1	−2
−1	1	0	−1	−2	−2
−2	0	−1	−2	−2	−2

To apply quaternion corrections, the estimated error quaternion must be reconstructed:

$$\vec{\varepsilon}_q \leftarrow \{\varepsilon_\phi, \varepsilon_\theta, \varepsilon_\varphi\} \tag{41}$$

As with the Kalman filter, these corrections are applied according to equation (35).

Flight test results using FLASP and the AHR150

The first maneuver used is called a "step turn". In a step turn, a constant Euler roll angle is applied to the plane for 1 min, and then it is rapidly increased, in a stepwise fashion, through a large range of Euler roll angles in 2° increments. The effect of a step turn on heading should be a continuous increase or decrease in heading without any reversal. Figure 3 shows a graph of the Euler roll angle measurements recorded during this maneuver by both the AHR150 and a Litton 90-100 INS.

Figure 4 shows the INS roll, the AHR150 roll, and the difference between the two or the roll error. The INS roll and AHR150 roll use the axis on the left and the difference between the two or the roll error uses the axis on the right. The worst error in the Euler roll angle is about 1.8°. The large, quick spikes in the roll error indicate a problem in the synchronization as the data were produced and recorded by different systems.

Figure 4 clearly demonstrates the advantage of an ADAHRS over an AHRS using only a gravity reference. After 10 min of continuous turning, the AHR150 is still tracking the INS output. These results are not obtainable using vertical gyros or gravity referenced AHRS units.

The results for pitch during the same maneuver are shown in Figure 5. For both pitch and roll, the RMS errors for this maneuver were under 0.5° and yaw was under 1.2°.

Conclusions

While similar flight results to the FLASP can be obtained using a Kalman filter, the operational software overhead is considerable. In our own

Figure 4 Roll angle flight results

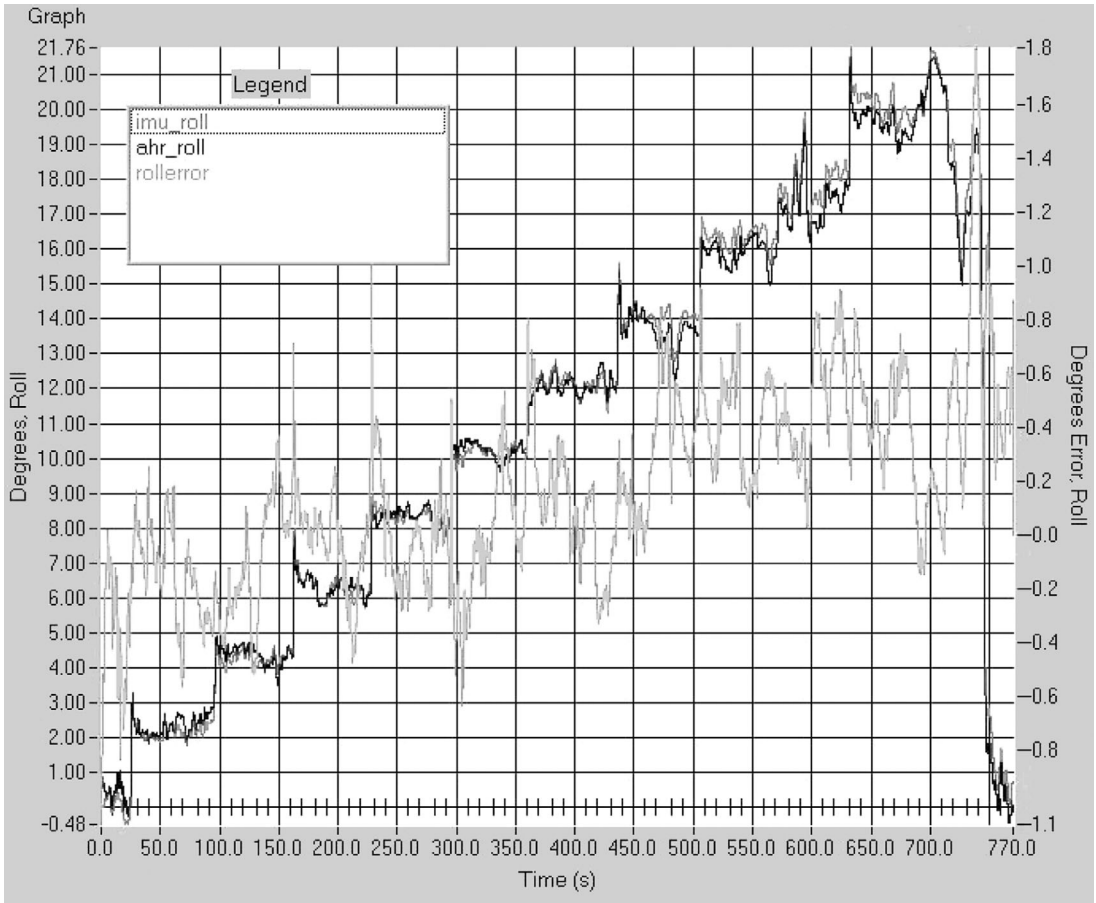
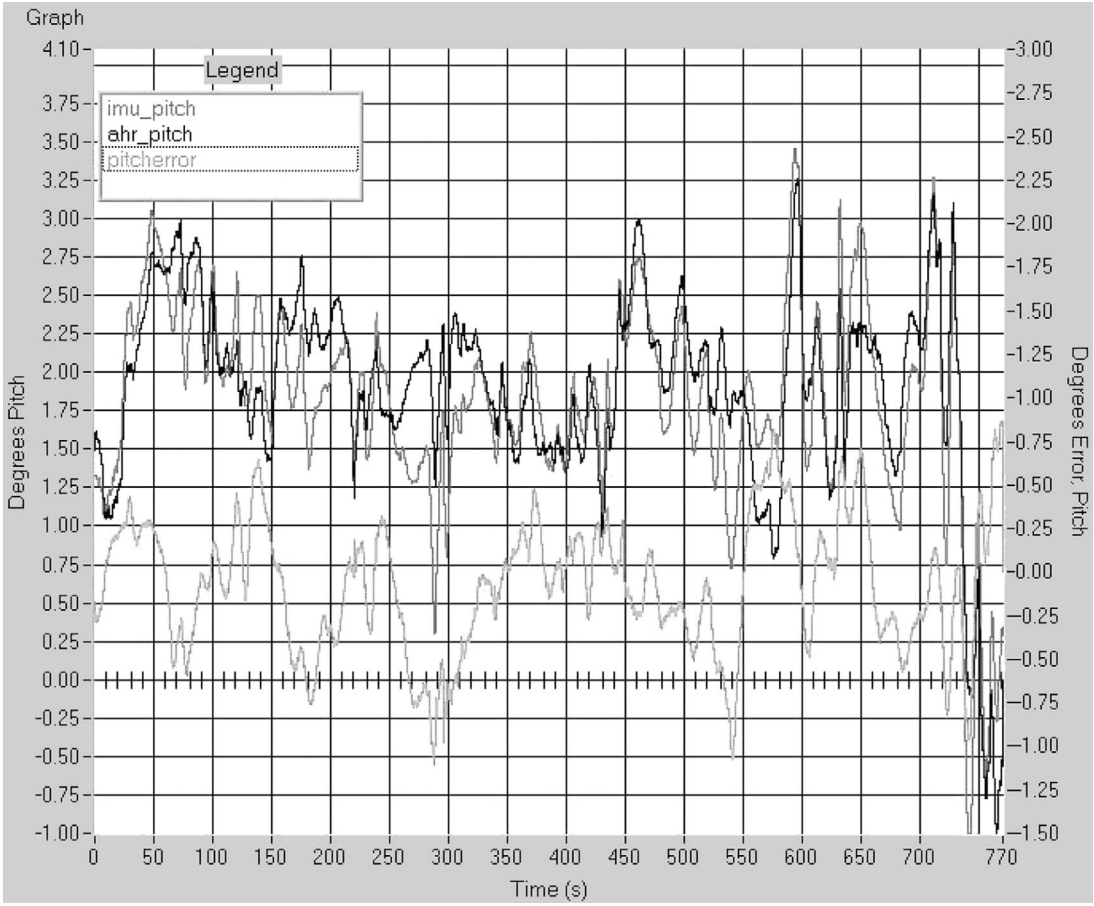


Figure 5 Pitch angle flight results



tests, the Kalman filter took 3.5 ms to run per iteration while FLASP took under 1 ms per iteration on a Texas Instrument C33 DSP with a clock speed of 60 MHz. Similarly, the Kalman filter code required memory of nearly 10,000 words, while the FLASP was under 3,000 words. Both requirements were driven in the Kalman filter by the matrix inversion, which is absent in the FLASP.

Relieved of memory and time constraints, the FLASP system can devote more resources to error checking, redundant calculation schemes and built-in-test algorithms, thus enabling a more robust system without increasing costs.

References

Archangel Systems (2003), http://www.archangel.com/aerospaceproducts.htm

Avidyne (2003), http://www.avidyne.com

Brown, R.G. (1983), *Introduction to Random Signal Analysis and Kalman Filtering*, ISBN 0-471-08732-7, Wiley, New York.

Chou, J.C.K. (1992), "Quaternion kinematic and dynamic differential equations", *IEEE Transactions on Robotics and Automation*, Vol. 8 No. 1, pp. 53-64.

Gelb, A. (1974), *Applied Optimal Estimation*, ISBN 0 262 70008-5, MIT press, Cambridge, Mass., p. 374.

Goodrich Systems (2003), http://www.sensors.goodrich.com

Grewal, M.S. and Andrews, A.P. (1993), *Kalman Filtering Theory and Practice*, Prentice-Hall, Englewood Cliffs, New Jersey.

Kalman, R.E. (1960), "A new approach to linear filtering and prediction problems", *Transaction of the ASME, Series D, Journal Basic Engineering*, Vol. 82, pp. 35-45.

Meggitt (2003), http://222.meggitt.com

Mendel, J.M. (1995), "Fuzzy logic systems for engineering: a tutorial", *Proceedings of the IEEE*, Special Issue on Fuzzy Logic in Engineering Applications, Vol. 83 No. 3, pp. 345-77.

Passino, M. and Yurkovich, S. (1998), *Fuzzy Control*, ISBN 0-201-18074-X, Addison-Wesley Longman, Inc., Menlo Park, California.

Roumeliotis, S.I. and Bekey, G.A. (1999), "3D localization for a mars rover prototype", *5th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, (i-SAIRAS '99) 1-3 June 1999, ESTEC, Noordwijk, The Netherlands, pp. 441-8.

Roumeliotis, S.I., Sukhatme, G.S. and Bekey, G.A. (1999), "Smoother based 3D attitude estimation for mobile robot localization", *Proc. 1999 IEEE International Conference on Robotics and Automation*, 10-15 May 1999, Detroit, Michigan, pp. 1979-86.