JOURNAL OF AEROSPACE COMPUTING, INFORMATION, AND COMMUNICATION
Vol. 6, July 2009

Field Programmable Gate Array-Based Attitude Stabilization

Michael J. Stepaniak[*]
Air Force Institute of Technology, Wright-Patterson AFB, OH 45433

and

Maarten Uijt de Haag[†] and Frank van Graas[‡]
Ohio University, Athens, OH 45701

DOI: 10.2514/1.39707

A system for determining vehicle attitude using a field programmable gate array and low-cost gyroscopes is presented. The method is intended to support the stabilization of a short duration, unmanned aerial vehicle. Using a microelectromechanical system inertial sensor for the calibration and serial interface, the algorithm sidesteps concerns related to electromagnetic interference and the impact of embedded, proprietary filters. An Allan variance analysis is used to characterize the sensor errors and predict system performance. A floating point representation using a direction cosine matrix is hosted on the field programmable gate array alongside the platform stabilization feedback loops. Although prone to drifting without additional aiding, the derived attitude has been demonstrated to be effective in stabilizing a remotely piloted quadrotor.

Nomenclature

A	angle random walk
B	bias instability
b_i	bit length of the initialization period
$\mathbf{C}$	attitude direction cosine matrix
f	sensor data rate
$\mathbf{I}$	identity matrix
K	number of intervals
M	number of samples per interval
S_r	scaling error due to rounding
T_i	initialization period

Received 31 July 2008; accepted for publication 24 March 2009.

[*] Assistant Professor, Department of Electrical and Computer Engineering, 2950 Hobson Way, Michael.Stepaniak@afit.edu, AIAA Member.

[†] Professor, Avionics Engineering Center, 321B Stocker Center, AIAA Member.

[‡] Professor, Avionics Engineering Center, 321B Stocker Center.

Greek

α	loop gain mantissa
β	loop gain exponent
$\boldsymbol{\Theta}_{\times}$	skew-symmetric form of the angle vector
θ	pitch angle
σ_A^2	Allan variance
τ_M	period spanning M samples
ϕ	roll angle
ψ	yaw angle
$\boldsymbol{\Omega}_{\times}$	skew-symmetric form of the angular rate vector
$\bar{\omega}_k$	mean angular rate for interval k
$\boldsymbol{\omega}$	angular rate vector
$\boldsymbol{\omega}_a$	angular rate due to angle random walk
$\boldsymbol{\omega}_b$	angular rate due to bias instability

Subscript

r	command reference
t	true
x	body x-axis, aligned with a motor arm
y	body y-axis, aligned with an orthogonal arm
z	body z-axis, oriented down

I. Introduction

THE Avionics Engineering Center (AEC) at Ohio University has designed a quadrotor unmanned aerial vehicle for use as a sensor testbed platform. The primary application involves a line-scanning laser range finder, shown mounted under the platform in Fig. 1, operating in an urban environment. The quadrotor configuration provides the lift necessary to carry the sensor, which weighs over six pounds with associated hardware, and can also serve as a sensor gimbal for data collection. Remotely piloting the vehicle requires stability augmentation which can be accomplished using simple classical control techniques based on attitude and angular rate feedback. Although commercially available autopilots can provide this data, they rely on the global positioning system (GPS) [1] and are therefore not suitable for AEC research involving GPS-denied environments.

A low-cost inertial system, however, can be effectively applied since the vehicle will be piloted remotely. Combining gyroscopes with accelerometers and magnetometers provides reasonable accuracy, but these microelectromechanical system (MEMS) sensors are both susceptible to electromagnetic interference and introduce an

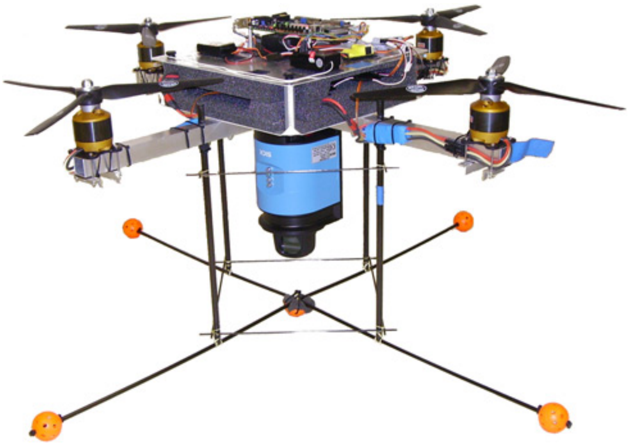

Fig. 1 AEC quadrotor.

element of uncertainty due to proprietary blending algorithms. Therefore, this paper investigates the ability to derive a satisfactory attitude solution using only the gyroscope data from a MEMS inertial sensor. The remainder of this paper characterizes the Microstrain 3DM-GX1, presents the algorithm for propagating attitude over time, describes the implementation in a field programmable gate array (FPGA), and concludes with flight test results.

II. Background

The Microstrain 3DM-GX1, Fig. 2, is a MEMS inertial unit capable of providing orientation information using an RS-232 serial port interface. The attitude solution benefits greatly from the application of a proprietary algorithm that blends the output of orthogonal gyroscopes with data from the accelerometers and magnetometers in order to minimize drift and sensitivity to inertial forces. For dynamic applications, the 3DM-GX1 typically provides two degree RMS accuracy [2]. Their level of performance and their relative low cost makes devices such as these well-suited for unmanned vehicles applications especially when aided by the GPS [3–5]. In addition, without aiding the 3DM-GX1 has been demonstrated capable of hovering a micro air vehicle autonomously for 35 s and for several minutes when assisting a pilot [6].

One limitation of the 3DM-GX1 is the sensitivity of the unit's magnetometers to electromagnetic interference and the presence of metal objects. In fact, the magnetic disturbances measured by the 3DM-GX1 have been used as an additional means of positioning [7]. Potential sources of interference on the AEC quadrotor include four high-power motors, with motor currents as high as 55 A each, the avionics associated with the flight control system, and possibly the sensor payload as well. One means of mitigating the errors on the blended navigation solution used for wheeled robots is to position the sensor as far as possible from the sources of interference [8]. Within the constraints of a small aerial vehicle, an alternate solution is to limit aiding performed by the magnetometers. For example, German researchers integrating a GPS receiver with a MEMS inertial measurement unit opted to restrict the magnetometers aiding to the yaw channel only where errors such as 20 degrees were considered "tolerable" [4].

In addition to interference, the filtering operation itself can add undesired artifacts. For example, the low pass filter used in the blending algorithm is sensitive to "sustained inertial influences, such as long duration (30 s) coordinated turns at velocity", leading an increase in error during the maneuver [9]. Also, the filter adds unspecified phase lag which has a destabilizing effect. To mitigate the interference and eliminate uncertainty associated with a proprietary algorithm, the proposed solution is to bypass the proprietary filters altogether and to perform the bias correction and attitude calculation externally in real-time on a FPGA. This approach relies solely on the gyroscopes for inertial data, minimizing the impact of electromagnetic interference, and allows implementation on a single light-weight board which is capable of hosting both the attitude algorithm and the flight control system. The remaining sections in this paper detail the implementation of the attitude algorithm, characterize the performance, and present the results noted during flight test of the AEC quadrotor sensor testbed.

Without requiring data from the accelerometers and magnetometers, only a portion of the functionality of the 3DM-GX1 was utilized. As such, this technique described in this paper could also be applied to discrete MEMS gyroscopes. However, the 3DM-GX1 sensor package was retained to take advantage of the digital interface and the gyroscope calibration. The "instantaneous" angular rate vector, to which blending has not been applied, is compensated for temperature, alignment, and G-sensitivity but not for the constant bias [10].

Fig. 2 3DM-GX1 orientation sensor.

A byproduct of bypassing the internal bias correction algorithm is an increase in maximum data rate for Euler angles from 100 Hz to 333 Hz [11]. The increased data rate benefits the fidelity of the attitude solution, though the control loop for the quadrotor is still constrained to 50 Hz by the pulse width modulated (PWM) protocol used by the radio receiver and electronic speed controllers.

III. Allan Variance Analysis

The instantaneous angular rates retain a significant bias that can be removed by averaging the angular rate during an initialization period and then subtracting this bias from subsequent measurements. The appropriate averaging interval is determined from a plot of the Allan variance versus averaging interval. The Allan variance provides the variation between mean values calculated for consecutive intervals. Based on N data points, the Allan variance is calculated as [12]

$$\sigma_A^2(\tau_M) = \frac{1}{2(K-1)} \sum_{k=1}^{K-1} (\bar{\omega}_{k+1}(M) - \bar{\omega}_k(M))^2 \tag{1}$$

where M is the number of sample points in an interval and $K = \text{floor}(N/M)$ is the integer number of distinct intervals that can be formed within the collected data. With data collected at a constant frequency, f, the length of the interval is given by $\tau_M = M/f$. The mean value of the kth interval is denoted by $\bar{\omega}_k$.

The percent error associated with the root Allan variance, or Allan standard deviation, is given in [12]

$$\%\ \text{error} = \frac{100}{\sqrt{2(K-1)}} \tag{2}$$

Because the percent error grows rapidly as the interval length approaches the sample length, care must be taken to collect sufficient data to avoid erroneous results. The data set represented in Fig. 3 was collected at 333 Hz for 6.7 h and exhibits less than ten percent error for averaging intervals shorter than seven minutes in length. The minimum point for each curve indicates the interval length for which the calculated mean will remove the bias with minimal residual error. Although the three gyros attained minimum variance at different interval lengths, for simplicity a single initialization period of 50 seconds was selected for all three gyros.

The Allan variance analysis also provides a graphical method for estimating sensor noise based on the slopes of various line segments as detailed in [12,13]. In particular for the quadrotor, the slope of −0.5 for short averaging intervals is indicative of angle random walk and the level segment for longer intervals is associated with the bias

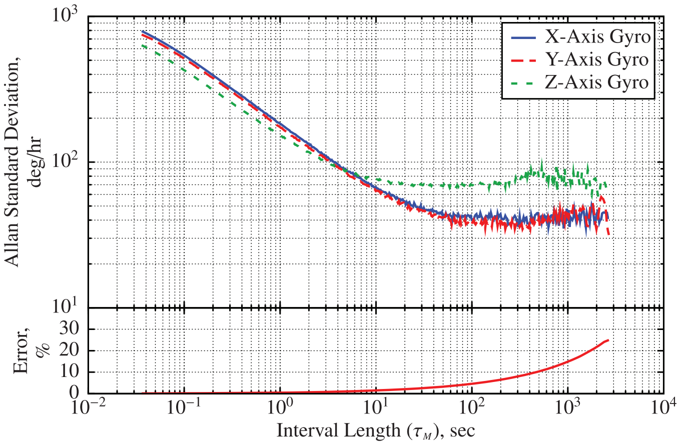

Fig. 3 Allan variance plot for the gyroscopes.

Table 1 Allan variance-derived noise coefficients

Noise source	Units	Specification	Value	
Angle random walk	$\frac{\text{deg}}{\sqrt{\text{h}}}$	3.5	x-axis	3.1
			y-axis	2.9
			z-axis	2.6
Bias instability	$\frac{\text{deg}}{\text{h}}$	72	x-axis	61
			y-axis	56
			z-axis	104

instability. The expression for the angle random walk coefficient is typically presented as [12]

$$\sigma_A = \frac{A^2}{\tau_M} \tag{3}$$

which requires one to extrapolate the sloped line to an intercept point at $\tau_M = 3600$ s. Adding a conversion factor results in an expression that can be applied directly to the curve as depicted

$$A = \sigma_A \frac{\sqrt{\tau_M}}{60} \tag{4}$$

As an example, for the x-axis gyroscope shown in Fig. 3, the standard deviation for a 0.85 s interval is 200 deg/h corresponding to a angle random walk of 3.1 deg/$\sqrt{\text{h}}$. The bias instability coefficient, B, is proportional to Allan standard deviation of the horizontal segment [12]

$$B = \sigma_A \sqrt{\frac{\pi}{2\,\ell n\, 2}} \tag{5}$$

The noise coefficients extracted from the Allan variance analysis are listed in Table 1 for each of the three gyroscopes. The measured coefficients are found to be in reasonable agreement with the specifications provided by the manufacturer, though the z-axis gyroscope exhibits a larger than expected bias instability.

IV. Deriving Attitude

The platform attitude is the orientation of a body fixed-reference system with respect to inertial space or a predefined navigation frame. The body reference frame, Fig. 4, is defined analogous to that of a typical fixed-wing aircraft with the origin located at the center of gravity; the x-axis is aligned with one arm, the z-axis pointing down, and the y-axis aligned with a second arm consistent with a right-hand coordinate system. For the planned short duration flights, the rotation of the earth can be ignored and the axes of the inertial reference frame are set to north, east, and down, corresponding to the navigation frame.

The attitude is described by three Euler angles, ψ, θ, and ϕ. Applying the *yaw-pitch-roll* convention, the Euler angles describe three successive rotations about the z-, y-, and x-axes starting at the navigation frame and ending at the body frame. Note that the second rotation occurs about an intermediate y-axis which is found by first yawing

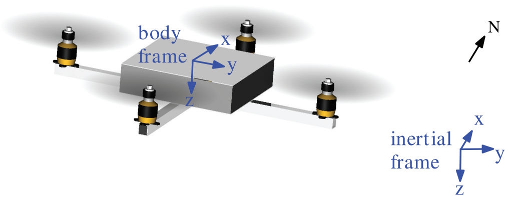

Fig. 4 Quadrotor reference frames.

about the z-axis. As a result, in general the Euler angles cannot be determined by direct integration of the body-referenced angular rates, which are measured by the strapdown inertial sensor. Instead, the attitude is propagated using a direction cosine matrix from which the Euler angles can be extracted as needed. The direction cosine matrix relating the body and inertial frames using the Euler angles is [14,15]

$$\boldsymbol{C} = \begin{bmatrix} \cos\theta\cos\psi & -\cos\theta\sin\psi + \sin\phi\sin\theta\cos\psi & \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi \\ \cos\theta\sin\psi & \cos\theta\cos\psi + \sin\phi\sin\theta\sin\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \tag{6}$$

This matrix is propagated forward in time using a first-order differential equation [14]

$$\dot{\boldsymbol{C}} = \boldsymbol{C}\boldsymbol{\Omega}_\times \tag{7}$$

where $\boldsymbol{\Omega}_\times$ is a skew-symmetric matrix containing the body angular rate vector as measured by the gyroscopes:

$$\boldsymbol{\Omega}_\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{8}$$

Because the gyroscopes' data rate is high compared to the bandwidth for the quadrotor platform motion, a first-order approximation is sufficient to propagate the attitude direction cosine matrix in discrete time

$$\begin{aligned} \boldsymbol{C}_{k+1} &= \boldsymbol{C}_k \exp\int_k^{k+1} \boldsymbol{\Omega}_\times \mathrm{d}t \\ &\approx \boldsymbol{C}_k(\boldsymbol{I} + \boldsymbol{\Theta}_\times) \end{aligned} \tag{9}$$

The Euler angles can be extracted from the direction cosine matrix using trigonometry:

$$\begin{aligned} \phi &= \arctan\frac{\boldsymbol{C}(3,2)}{\boldsymbol{C}(3,3)} \\ \theta &= -\arcsin\boldsymbol{C}(3,1) \\ \psi &= \arctan\frac{\boldsymbol{C}(2,1)}{\boldsymbol{C}(1,1)} \end{aligned} \tag{10}$$

V. Hardware Implementation

The quadrotor's flight control system is implemented in a Xilinx Spartan-3 FPGA hosted on a Digilent starter board. The size and weight of the FPGA board are suitable for use on the moderate-sized quadrotor platform. In addition, the starter board provides a serial port for interfacing with the inertial sensor and also sufficient switches and displays to facilitate setting configuration options and diagnostics such as system status reporting. The attitude algorithm is hosted on a one million gate FPGA with a system clock operating at 50 MHz. To accommodate additional planned functionality, the footprint of the attitude algorithm was minimized where possible.

A block diagram of the system is shown in Fig. 5. The five core functions hosted on the FPGA are the inertial and radio interfaces, the bias correction, an attitude update algorithm, and the control loops used for stabilization. Each of these five function are described in the following sections.

A. Inertial Interface

A generic signal interface for the inertial sensor was established to facilitate the use of other sensors in the future. This interface consists of the following five signals: angular rate measurements, $\boldsymbol{\omega}$, for three axes in radians per second; the time increment, t, in seconds since the last data set; and a boolean flag, *rdy* that is asserted when a new data set is ready. The angular rate and time signals are defined as fixed-point numbers with the number of bits declared in a configuration file. In this manner, the width of the interface can vary based on the resolution of the inertial unit

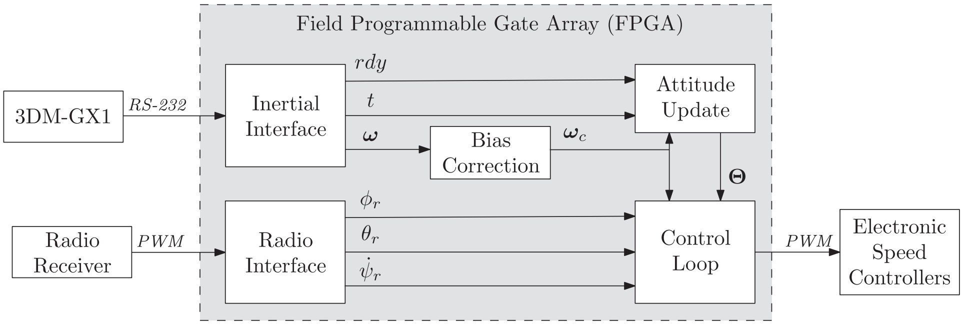

Fig. 5 FPGA block diagram.

Table 2 Rounding error in the time-scale factor

Length (bit)	Value (m)	Error (%)
8–9	3.906	30.21
10–12	2.930	2.34
13	3.052	1.73
14–15	2.991	0.31
16	3.006	0.20
17–18	2.998	0.05

and, by configuring the hardware description with adjustable length ports, the FPGA synthesis tool automatically adjusts the bias correction and attitude update sections of the FPGA accordingly.

Specifically for the Microstrain 3DM-GX1, the inertial interface establishes serial port communications and places the sensor into a continuous data transfer mode. In addition, the interface performs basic error checking and converts the 16-bit raw data counts to the appropriate units for the generic interface. The scale factors are provided by the manufacturer, and for angular rate, the scale factor is 17×2^{-16} rad/s/bit which is readily implemented in fixed-point arithmetic using a multiplication followed by a shift operation. Operating at 333 MHz, the time-scale factor for the 3DM-GX1 is 0.003 sec/bit, or equivalently $\frac{3}{125} \times 2^{-3}$ sec/bit. When implemented using a multiplier and shift, the fractional multiplier introduces rounding error which varies as a function of bit length. As shown in Table 2, selecting a scale factor word length of 10 bits or greater minimizes the degradation in accuracy, and 16 bits were used for the time-scale factor during flight test.

B. Bias Correction

The nominal 50 s initialization period, T_i, used to determine the sensor bias is adjusted so that the mean can be implemented in hardware by a summation followed by a simple shift operation. Because the Allan variance analysis exhibited a flat bias instability region for the interval of interest, a slight change in the initialization period is of no consequence. If f is the sensor data rate, then the actual length of the initialization period is given by

$$T_i' = \frac{2^{b_i}}{f} \tag{11}$$

where $b_i = \mathrm{round}(\log_2(T_i f))$. The number of discrete samples included in the summation is 2^{b_i} and the decimal is shifted over b_i bits to calculate the mean. For precision, the bias and the corrected angular rates retain all fractional bits. The values used in flight test were $b_i = 14$ and $T_i' = 49.2$ s.

C. Radio Interface

Even after removing the initial bias, the bias instability and the angle random walk in the sensor will introduce a residual nonstationary error that must be manually compensated for by the pilot. With stick inputs that only cover a fixed range, the scale factor for the reference command inputs must be applied in the FPGA that is large enough to accommodate predicted error growth without unduly raising the system gain. In practice, without additional aiding the error grows without bounds and the flight ends when a full stick input is required to maintain a level attitude.

Neglecting the coupling inherent in the Euler angles, the attitude angles can be represented in a simple form as

$$\boldsymbol{\Theta} = \int \boldsymbol{\omega} t \, \mathrm{d}t \tag{12}$$

As shown in the Allan variance analysis, the angular rate is corrupted by errors due to bias instability and angle random walk such that $\boldsymbol{\omega} = \boldsymbol{\omega}_t + \boldsymbol{\omega}_a + \boldsymbol{\omega}_b$, where $\boldsymbol{\omega}_t$ is the true angular rate, $\boldsymbol{\omega}_a$ is the error in the angular rate due to angle random walk, and $\boldsymbol{\omega}_b$ is the error due to bias instability. In addition, the time interval contains an error due to rounding such that $t = t_t(1 + S_r)$, where t_t is the true time and S_r is the scaling error due to rounding. Incorporating these errors into the expression for attitude yields

$$\begin{aligned} \boldsymbol{\Theta} &= \int (\boldsymbol{\omega}_t + \boldsymbol{\omega}_a + \boldsymbol{\omega}_b) \, \mathrm{d}(t_t(1 + S_r)) \\ &= \int (\boldsymbol{\omega}_t(1 + S_r) + (\boldsymbol{\omega}_a + \boldsymbol{\omega}_b)(1 + S_r)) \, \mathrm{d}t_t \\ &= \boldsymbol{\Theta}_t(1 + S_r) + \int (\boldsymbol{\omega}_a + \boldsymbol{\omega}_b)(1 + S_r) \, \mathrm{d}t_t \end{aligned} \tag{13}$$

where the first term represents a scaling of the true attitude and the second term is the error due to both sensor noise and rounding error. Since the quadrotor flight profile that consists of only nominal excursions from a hover, the true attitude will generally remain close to zero for pitch and roll and the affect of rounding is minimal. The second term is the residual error for which the pilot must compensate. The uncertainty of this term as a function of time is

$$\sigma_{\boldsymbol{\Theta}}(t) = (A\sqrt{t} + Bt)(1 + S_r) \tag{14}$$

Using the error coefficients from Table 1 and a 10-bit time-scale factor, a suitable command reference-scale factor can be selected that encompasses the two-sigma attitude uncertainties for roll and pitch angles. For example, from Fig. 6 a scale factor corresponding to ±13 deg accommodates five minutes of flight. This duration is sufficient for many of the profiles planned for the AEC quadrotor and longer flights are accommodated by increasing the scale factor at the expense of increasing the sensitivity to the control stick.

D. Attitude Update

Based on its resolution, the sensor was capable of reporting an incremental angular change as small as 0.78 μrad. To adequately cover the dynamic range from the incremental change to a full rotation, a nonstandard floating point representation was implemented for the attitude calculations. The exponent was set at six bits to span the dynamic range and the length of the mantissa was determined using a simulation of the FPGA algorithm; the algorithm was propagated using a brief data set collected as the inertial sensor was subjected to a series of doublets followed by

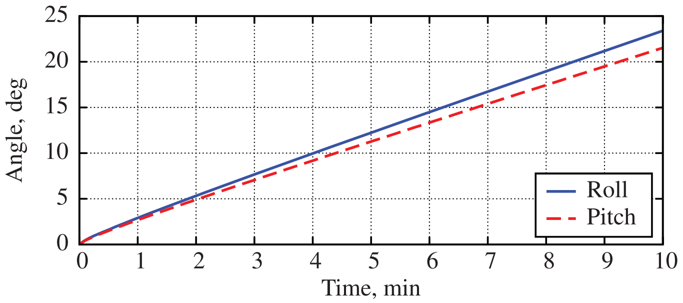

Fig. 6 Uncertainty (2σ) in roll and pitch.

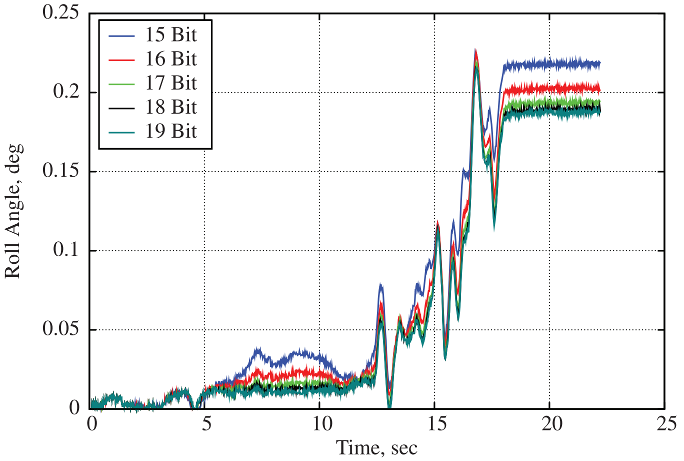

Fig. 7 Roll angle error with varying mantissa length.

a three-axis oscillation. The resulting length of the mantissa was then determined to be 18 bits, beyond which the improvement in accuracy was negligible as seen in Fig. 7 which is a representative plot of the roll angle error.

Direct coding of the 3 × 3 matrix multiplication is impractical based on the requirement for 18 additions and 27 products. The resources required for each of these operations are summarized in Table 3 in terms of FPGA slices and dedicated 18 × 18 bit multipliers. Each FPGA slice contains two four-input lookup tables and two flip-flops and the Spartan-3 chip used contains 7680 slices and 24 dedicated multipliers. For comparison, the results for a 25-bit fixed-point implementation is also included.

While the products can be readily implemented, and are in fact more compact than the corresponding fixed point representation, the floating point addition is prohibitively large. Therefore, the matrix multiplication is subdivided first into a series of dot products and then into a series of additions, each of which is carried out sequentially using the same hardware. This method, for which the pseudo-code is shown in Fig. 8, eliminates 17 additions but at the cost of a number of multiplexers to route the signals. In the case of floating-point multiplication, however, it is advantageous to use three dedicated multipliers for the dot product, as shown in the pseudo-code, rather than introduce the multiplexers. As implemented the matrix multiplication occupies 1191 slices and three multipliers, corresponding to 15% and 12% of available resources, respectively.

The pitch and roll angles required for platform stabilization are extracted from the direction cosine matrix using small angle approximations and Eq. (10). Note that the small angle approximations impact only the pitch and roll angles calculated for a specific moment in time and have no influence on the accuracy of the attitude solution carried in the direction cosine matrix, even in the case of large excursions in pitch or roll. For the quadrotor platform, pitch and roll angles are controlled to be less than eight degrees for all modes of flight and the heading is maintained solely by the pilot. Therefore, the small errors introduced by the approximations are inconsequential. For other applications it may be beneficial to implement the trigonometric functions using lookup tables or the coordinate rotation digital computer (CORDIC) algorithm [16].

Table 3 FPGA resources for floating-point and fixed-point operations

Operation	Type	Slices	18 × 18 Multipliers	Latency (ns)
Addition	Float	421	0	14.3
	Fixed	27	0	7.0
Multiplication	Float	51	1	15.1
	Fixed	53	4	15.6

```
C = matrix_multiplication(A,B)
  for r = 1 to 3
    for c = 1 to 3
      C(r,c) = dot_product(row(A,r),
                           col(B,c))
    end
  end
```

```
C = dot_product(A,B)
  product(1) = A(1)*B(1)
  product(2) = A(2)*B(2)
  product(3) = A(3)*B(3)
  sum = 0
  for i = 1 to 3
    sum = sum + product(i)
  end
```

Fig. 8 Pseudo-code for 3 × 3 matrix multiplication.

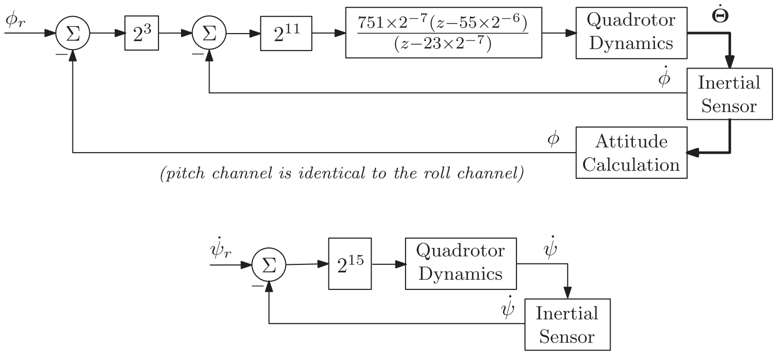

Fig. 9 Flight control system diagram.

E. Control Loop

The flight control system uses feedback loops, shown in Fig. 9, to stabilize the vehicle based on the measured angular rates and the derived attitude angles. Note that with symmetry the feedback loops for the roll channel are identical to that for the pitch channel. Loop gains and the pole/zero placement for the lead compensator were initially selected to achieve desired performance and then the values were adjusted to be of the form $\alpha \times 2^{\beta}$, where α and β are integer constants. This facilitated implementation on the FPGA in a fixed-point notation using only multipliers and shift operations. Performance with the new gains was then verified to ensure that the desired performance was not adversely impacted.

The quadrotor dynamics include the motor and propeller as well as the aerodynamics. The input quadrotor dynamics is a PWM motor command signal that is delivered to an electronic speed controller. The standard PWM signals used for the quadrotor have a variable width component of 0–1 m. With the FPGA operating at 50 MHz, the PWM interface is able to provide a command signal resolution of 1/50,000. The ability of the speed controller to respond to 20 ns changes was not ascertained and likely this response varies between different controllers. Nonetheless, an upper bound on the useful resolution of the attitude is determined by the closed loop gain which varies as a function of the PWM resolution. For example, the closed loop gain for the roll and pitch channels shown in Fig. 9 is 2^{14}. In this case, angular precision, in radians, for pitch and roll beyond 14 bits will have no impact on the motor commands. Therefore pitch and roll angles have an effective resolution of 3.5×10^{-3} deg even though the sensor itself supports a resolution of 4.5×10^{-5} deg.

VI. Results

A Monte Carlo analysis confirms that measured data matches the error model described by Eq. (14), validating the use of the Allan variance to estimate noise coefficients. Figure 10 shows the roll angle Monte Carlo plot and the predicted and calculated statistics. The pitch axis exhibits similar characteristics with yaw axis having somewhat worse performance as anticipated by the noise coefficients in Table 1. The data were collected from a stationary sensor and the Euler angles were calculated using the FPGA algorithm after compensating for the bias from the angular rates as previously described. As expected, there is significant error growth due to the bias instability even for short time periods.

Figure 11 plots the standard deviation of the Monte Carlo analysis for each channel and the device's accuracy specification for dynamic maneuvering. The dynamic accuracy was not measured due to current limitations in the data collection setup for the quadrotor. However, a Monte Carlo analysis for the static condition was completed and the one-sigma parameters were found to be in general agreement with the specifications: 0.60°, 0.56°, and 0.98° for roll, pitch, and yaw, respectively, compared to a specification of 0.5°. Furthermore, because the attitude algorithm is not based on the accelerometers, unlike the solution embedded in the sensor, the results are expected to be representative of dynamic performance.

Two conclusions can be drawn from Fig. 11. First, for the first two minutes the gyro-based solution implemented on the FPGA outperforms the dynamic-maneuvering specification. Second, the growth is slow enough that the drift can easily be corrected by the pilot. The operational limit for mission duration is then based on the scale factor applied to the command reference input. As long as the pilot can compensate for the drift using trim or manual inputs, the performance remains satisfactory.

Flight test results were consistent with predicted performance. With the FPGA providing stability augmentation, the operator was able to maintain a hover or maneuver the quadrotor without difficulty. Compensating for sensor drift using the trim switches on the radio controller was not practical, but compensating for the drift by holding a nonzero stick input was not problematic. Over time, the drift in attitude would saturate one of the controls. However, it was not necessary to preemptively land as the drift was slow enough that the vehicle could still be brought down safely.

On two occasions the quadrotor lifted off with a slight pitch or roll rate. Because even small angles induce an acceleration by tipping the lift vector into the horizontal plane, this led to an unexpected translation and the operator opted to immediately land. The cause may be attributed to a chance offshoot in the angular rates, but with the data recording currently limited this could not be conclusively verified. The post-flight data did show a large bias in the indicated channel, but this could also have been the result of the hard landing saturating the gyroscope. In addition,

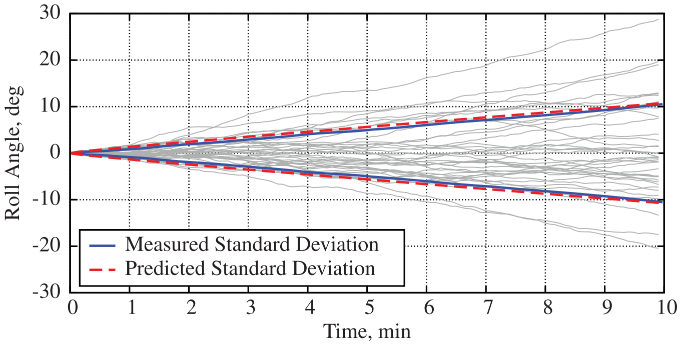

Fig. 10 Monte carlo analysis (37 runs).

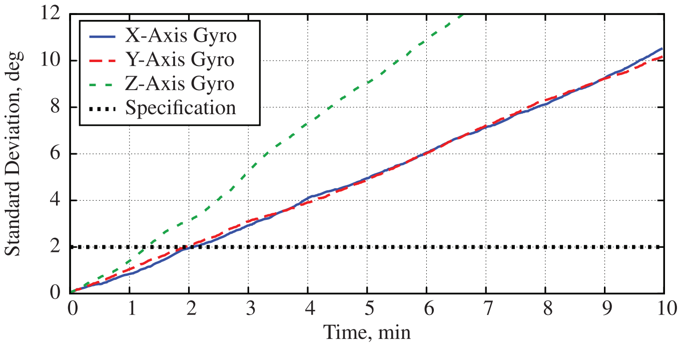

Fig. 11 Statistical comparison of attitude methods.

the motors occasionally experience a brief jitter in power on the order of a second in duration. The attitude is largely unaffected, though there is a audible increase in motor speed, and no command inputs are necessary. Possibly a spike in the sensor noise is amplified by the lead compensator only to be immediately compensated for through the feedback loop. A data telemetry system is planned that will allow an analysis of both anomalies.

The system was tested with a range of payloads, facilitating both a build-up approach to testing and simulating a variety of sensor payloads. Simulation predicted that a single set of gains would suffice to control the attitude, and this was verified in flight. Qualitative feedback from the pilot was that handling improved slightly as the payload increased from 0 to 3.1, 4.7, and 6.1 pound. Subsequent flights have been flown primarily with the 6.1 lb payload, representing the configuration pictured in Fig. 1, in preparation for data collection.

VII. Conclusions

The computational resources for the attitude algorithm can easily implemented on a low-cost FPGA board. Even allowing for a floating-point implementation and propagating the attitude using a direction cosine matrix, the FPGA board retains enough capacity to also host other functions including the control laws. In addition, although the results of the Monte Carlo simulation show that the uncertainty in the attitude solution varies greatly over time, in flight test the variation has been shown to be manageable for profiles suitable for sensors research. The completion of a telemetry downlink will provide data to support a detailed analysis of anomalies noted and will also support the real-time evaluation of the sensor payload. Some aiding would be necessary if an autonomous hover were required; however, the quadrotor has demonstrated the ability to support piloted flights with payloads representative of relatively heavy sensors such the line-scanning laser range finder.

Acknowledgment

The views expressed in this article are those of the authors and do not reflect the official policy or position of the United States Air Force, the Department of Defense, or the United States Government.

References

[1] Christophersen, H. B., Pickell, R., Neidhoefer, J. C., Koller, A. A., Suresh, K. K., and Johnson, E. N., “A Compact Guidance, Navigation, and Control System for Unmanned Aerial Vehicles,” *Journal of Aerospace Computing, Information, and Communication*, Vol. 3, May 2006, pp. 187–213.
doi: 10.2514/1.18998

[2] Microstrain, *Detailed Specifications for 3DM-GX1*, rev. 1 Microstrain, Williston, VT, July 2007.

[3] Mancini, A., Caponetti, F., Monteriù, A., Frontoni, E., Zingaretti, P., and Longhi, S., “Safe Flying for an UAV Helicopter,” *15th Mediterranean Conference on Control & Automation*, IEEE, Piscataway, NJ, Paper T34-010, 27–29 July 2007.
doi: 10.1109/MED.2007.4433946

[4] Wendel, J., Meister, O., Schlaile, C., and Trommer, G. F., “An Integrated GPS/MEMS-IMU Navigation System for an Autonomous Helicopter,” *Aerospace Science and Technology*, Vol. 10, No. 6, Sept. 2006, pp. 527–533.
doi: 10.1016/j.ast.2006.04.002

[5] Hoffmann, G. M., Huang, H., Waslander, S. L., and Tomlin, C. J., “Quadrotor Helicopter Flight Dynamics and Control: Theory and Experiment,” *AIAA Guidance, Navigation and Control Conference*, AIAA Paper 2007-6461, 20–23 Aug. 2007.

[6] Green, W. E., and Oh, P. Y., “Autonomous Hovering of a Fixed-Wing Micro Air Vehicle,” *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, IEEE, Piscataway, NJ, 15–19 May 2006. pp. 2164–2169.
doi: 10.1109/ROBOT.2006.1642024

[7] Vissière, D., Martin, A., and Petit, N., “Using Magnetic Disturbances to Improve IMU-Based Position Estimation,” *Proceedings of the European Control Conference* (*CDROM*), European Union Control Association, 2–5 July 2007, pp. 2853–2858.

[8] Cruz, D., McClintock, J., Perteet, B., Orqueda, O. A., Cao, U., and Fierro, R., “Decentralized Cooperative Control,” *IEEE Control Systems Magazine*, Vol. 27, No. 3, June 2007, pp. 58–78.
doi: 10.1109/MCS.2007.365004

[9] Churchill, D. L., and Arms, S. W., *3DM-GX1 Gyro Enhanced Orientation Sensor FAQ’s*, Microstrain, Williston, VT, 14 June 2005.

[10] Microstrain, *3DM-GX1 Data Communications Protocol*, Ver. 3.1.00, Microstrain, Williston, VT, 5 July 2005.

[11] Microstrain, *3DM-GX1 Orientation Sensor: Fastest Data Output Rates*, Microstrain, Williston, VT, 2006.

[12] Ng, L. C., and Pines, D. J., “Characterization of Ring Laser Gyro Performance Using the Allan Variance Method,” *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 1, Jan.–Feb. 1997, pp. 211–214.
doi: 10.2514/2.4026

[13] El-Sheimy, N., Hou, H., and Niu, X., “Analysis and Modeling of Inertial Sensors Using Allan Variance,” *IEEE Transactions on Instrumentation and Measurements*, Vol. 57, No. 1, Jan. 2008, pp. 140–149.
doi: 10.1109/TIM.2007.908635

[14] Titterton, D. H., and Weston, J. L., *Strapdown Inertial Navigation Technology*, 2nd ed., Vol. 207, *Progress in Astronautics and Aeronautics*, AIAA, Reston, VA, 2004.

[15] Nelson, R. C., *Flight Stability and Automatic Control*, McGraw-Hill, New York, NY, 1989.

[16] Andraka, R., “A Survey of CORDIC Algorithms for FPGA Based Computers,” *Proceedings of the Sixth ACM/SIGDA International Symposium on Field-Programmable Gate Arrays* (*FPGA’98*), ACM, New York, NY, 22–24 Feb. 1998, pp. 191–200.

James Hargrave
Associate Editor