Sensor Stability of a Low-Cost Attitude Sensor Suitable for Micro Air Vehicles

Gabriel Hugh Elkaim and Christopher C. Foster
University of California, Santa Cruz, CA

BIOGRAPHY

Gabriel Hugh Elkaim is an Assistant Professor in the Computer Engineering Department at the University of California, Santa Cruz, in Santa Cruz, CA. He received his B.S. degree in Mechanical/Aerospace Engineering from Princeton University, Princeton, NJ, in 1990, the M.S. and Ph.D. Degrees from Stanford University, Stanford, CA, in Aeronautics and Astronautics, in 1995 and 2002 respectively. In 2003, he joined the faculty of the Jack Baskin School of Engineering, at the UC Santa Cruz. His research interests include control systems, sensor fusion, GPS, system identification, and autonomous vehicle systems. His research focuses on intelligent autonomous vehicles, with an emphasis on robust guidance, navigation, and control strategies. Specifically, he has founded the Autonomous Systems Lab at UC Santa Cruz, and is currently developing an autonomous wing-sailed marine surface vehicle and off-road autonomous ground vehicles.

Christopher C. Foster received his B.S. degree in Computer Engineering from Santa Clara University, Santa Clara CA, in 2003 and is finishing his M.S. at University of California, Santa Cruz, in 2007.

ABSTRACT

This paper expands on previous work to introduce a high quality, but low cost attitude and heading reference system (AHRS), that is physically small, lightweight, and low-power. Previous Work on the sensor has shown that for non-accelerating environments, this small low-cost sensor has a short term accuracy of approximately $\frac{1}{4}^\circ$ in pitch, roll, and yaw. We extend the previous two-step non-linear calibration algorithm to account for non-orthogonality in the sensor axes, and also correlate the measured temperature of the sensor against the long term bias stability. Further bias stability is investigated using the Allan Variance, demonstrating that with temperature correction, the static attitude solution has approximately $\frac{1}{2}$ the standard deviation as without temperature correction. When the data is averaged to a 1Hz data rate, the 24 hour standard deviations vary from 0.017° to 0.06° for pitch, roll, and yaw.

1 INTRODUCTION

The recent interest in high performance, Micro Aerial Vehicles (MAVs) has necessitated the design of compact, accurate and inexpensive attitude determination systems. MAVs are designed to be disposable and are very small in size and weight (largest dimensions no greater than 15 cm) [17, 19]. At these scales, the avionics and sensor payloads can represent a significant fraction of the overall vehicle dimension and weight. To address this need many inexpensive rate-gyro based attitude determination systems have been developed [14, 20, 21]. However, inexpensive and miniature solid-state rate gyros (<$1000 per axis) tend to be low-performance sensors which have outputs subject to wide band noise and rate instabilities on the order of 10 to 100 °/hr [12, 11]. In order to determine attitude, the rate gyro outputs must be integrated to give attitude and this leads to unbounded attitude errors. Thus, successful implementation of an attitude determination system that relies solely on rate gyros requires the use of sensors with exceptionally accurate and stable outputs. These types of gyros would have output errors less than 0.1 °/hr and tend to be (1) prohibitively expensive, (2) have high power consumption, and/or (3) be physically too large for many miniature vehicle applications.

An alternative to relying on accurate and expensive rate gyros is to devise a system which fuses miniature, low performance (and low cost and low power) gyros with a gyro-free aiding system using a complementary filter architecture. The gyro-free aiding system provides either one of two types of information. It can provide (1) a noisy but un-

biased direct attitude measurement periodically (*i.e.*, a low bandwidth attitude update), or (2) it can provide indirect measurements from which attitude can be extracted using an observer. The aiding system's measurements are used to arrest error growth due the integration of the gyro biases and also to estimate these gyro biases in real time. A few examples of aiding system which provide direct measurements include multi-antenna GPS attitude determination systems [1], accelerometer- or inclinometer-based leveling systems [11] and the novel pseudo-attitude system described in [15].

Current low-cost solutions to guidance, navigation, and control (GNC) of a small unmanned vehicle (underwater, ground, marine surface, or flight) still cost several thousands of dollars, and offer relatively poor performance at that price point. Tactical grade inertial navigation solutions (INS) run in the several tens of thousands of dollars, and often come with data restrictions due to arms treaties (ITAR). Given that navigation is, at its most basic, the enabling technology for unmanned systems, and especially Micro Air Vehicles (MAVs), a lower cost (lower power, and smaller size) solution becomes an enabling technology for the proliferation of these systems. The design philosophy is to reduce expenditures on sensors (which are expensive), and augment the system with better processing (which is cheap). While at this point we have not completed the full up gyro based attitude sensor, this paper describes the hardware, software, and methodology to provide a direct measurement of attitude using magnetometers and accelerometers.

The sensor consists of a three axis magnetometer, and three axis accelerometer, and a single axis rate gyroscope. Using novel calibration algorithms to correct for bias and scale factor errors, as well as non-orthogonalities in the sensor axes, we are able to get superior performance using inexpensive sensors. The attitude is computed using a quaternion based estimation filter that linearizes the attitude in the Navigation frame. Our hardware/firmware combination achieves post-calibration results with standard deviations under 0.7 milli-g's and 0.6 milli-Gauss on the accelerometers and magnetometers, respectively. Combined with the quaternion attitude estimation algorithms, we are getting static performance with standard deviations of 0.12 degrees in pitch, 0.04 degrees in roll, and 0.08 degrees in yaw, with min to max deviations of less than 0.6 degrees on all axes.

This work extends previous efforts [6] by including an analysis of sensor parameter stability and correlation between these parameters and temperature. Allan variances of the sensor suite are presented, and it is demonstrated that even better results are obtained when correcting the bias offset (or null shift) term for temperature effects. Note that much of the hardware and software sections is identical

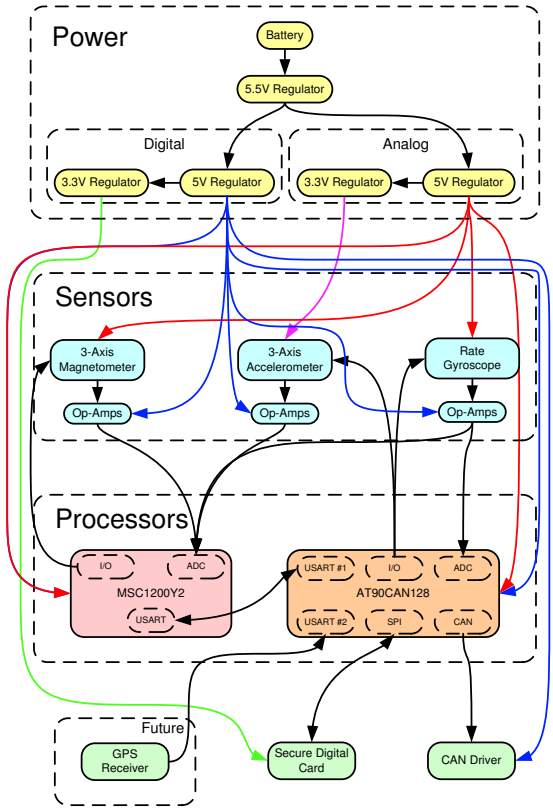

Figure 1: Functional block diagram of the hardware presented in this paper.

to that presented in [6], and is presented here for completeness.

2 HARDWARE

The hardware used in the sensor head is capable of sampling each axis of the three axis magnetometer and three axis accelerometer at 100 Hz., and at 24 bits per sample. Figure 1 shows a functional block diagram of the hardware components.

A total of ten data points are generated in this implementation and are classified into two categories. Eight of these data points are classified as primary sensors, including the six originally specified parameters plus values from a rate gyroscope monitoring rotations about the z-axis and a temperature sensor embedded in one of the processors. The final two sensors, the two secondary sensors, are the rate gyroscope's internal temperature and internal 2.5 volt reference.

The primary sensors, except for the processor temperature, represent the most important data points and the ones most likely to change rapidly. Primary sensors are sampled at

100 SPS with 24-bits of resolution. The processor temperature is included as a primary sensor only because it is internal to the processor that is gathering the other primary data points, and it can not be monitored externally. Secondary sensors are less likely to change rapidly and are of less significance, so are monitored at only 10-bits of resolution by a separate ADC.

2.1 Power

For mobility, the core power source for this project is batteries. A single switching regulator with an input range of 12.5 volts to 8.0 volts outputs 5.5 volts to two linear 5 volt regulators. Two regulators are used to isolate the analog components from the digital components. There are two devices, one digital, one analog, that require 3.3 volts DC, so these components are powered from two separate linear regulators that are connected to the corresponding analog and digital 5 volt regulators.

2.2 STMircoelectronics LIS3L06AL Accelerometer

The LIS3L06AL from STMicroelectronics is a three axis MEMS accelerometer in a single 8-terminal Ceramic Land Grid Array (LGA) package for less than $15.00. This chip outputs three analog voltages, each between zero and the 3.3 volt supply, corresponding to accelerations in the x, y, and z directions. This sensor has two sensitivity modes, the first with a measurement range of $\pm$2g's on each axis and the second with a range of $\pm$6g's on each and is sensitive to both static accelerations such as gravity and dynamic acceleration such as vibrations. The LIS3L06AL detects accelerations by monitoring the position of a mass suspended within the chip and includes a self test feature.

Before leaving the chip, each signal is passed through an onboard resistor allowing for an external capacitor to be added between each output and ground to form a passive low-pass RC filter. For the purpose of this project it was decided that a minimum bandwidth of 20 Hz. would be sufficient for acceleration in the x, y, and z directions so a 0.047 μF capacitor is used. Due to component tolerances, the resulting bandwidth is between 24.43 Hz. and 40.51 Hz. Finally, the signals pass through separate Operational Amplifiers (OpAmps) with a nominal gain of 1.499. This amplifies a full scale output from the accelerometer of 3.3 volts to between 4.91 volts and 4.98 volts for the analog to digital converter.

2.3 Analog Devices ADXRS150 Yaw Rate Gyroscope

The ADXRS150 from Analog Devices is MEMS gyroscope in a 32 terminal Ball Grid Array (BGA) package for under $50.00. This chip outputs three analog voltages between zero and the supply voltage that represent the observed rotation rate in the plane of the chip, the temperature of the chip, and the internal 2.5 volt reference. The rate output of the sensor has a range of $\pm$150 degrees per second of rotation and is determined by monitoring the Coriolis acceleration.

In addition to outputting the rate of rotation, the internal temperature, and the chip's internal 2.5 Volt reference, the ADXRS150 also provides a self test feature. The rotational rate output is amplified on the chip through an active low-pass filter, the capacitor of which is external to the chip and can be set by the user. In this case, a 0.015 nF capacitor is used for a nominal 58.9 kHz bandwidth, but due to component tolerances, has a range of between 55.5 Hz and 62.7 Hz. Finally, after leaving the chip, the signals go through their own unity gain OpAmps to boost the available current to the ADC.

2.4 Honeywell HMC1043 Magnetometer

Honeywell's HMC1043 is a three axis magnetometer in a 16-pin Leadless Plastic Chip Carrier (LPCC) package for less than $41.00. Each axes of the magnetometer consists of a four element Wheatstone bridge that is sensitive to magnetic fields within the range of $\pm$6 gauss. As the magnetic fields change, the magneto-resistive elements will buildup a memory. To wipe the elements clean, the magnetometer has a pair Set/Reset straps built into the chip.

External circuitry including capacitors and dual channel MOSFET switches are used to generate both set and reset pulses sufficient to clear all three axes of the HMC1043 magnetometer. As the current drawn out of the two outputs must be negligible for the Wheatstone bridges to work correctly, the output pins from each magnetometer are connected directly to OpAmps. The high impedance inputs of an OpAmp result in insignificant current flow out of the magnetometers. Furthermore, running each OpAmp in a differential configuration allows for the two outputs for each axis to be subtracted at the same stage while also boosting the available current for the ADC conversions.

2.5 Processors

There are two 8-bit processors in use for this project, both running on the same circuit board and connected to each other through a serial buss running at 115,200 baud. This

section will discuss the rolls of these two processors only to the level required to explain the hardware used. Further details of the programming will be discussed in next section on software.

2.5.1 Texas Instruments MSC1200Y2

The MSC1200Y2 from Texas instruments is an 8051-based microprocessor with an 9-input Delta-Sigma ADC, 4kB of flash memory, 256 Bytes of SRAM, various I/O pins, and a Universal Synchronous/Asynchronous Receiver/Transmitter (USART) in a 48 pin Thin Quad Flat Pack (TQFP) package for less than $10.00. Although an internal 2.5 volt reference is available, a high precision external reference is used instead to increase overall performance. Programming of the MSC1200Y2 is performed through the non-standard use of serial port requiring special on-board hardware and custom software.

The hardware features that are used here include the serial programming capability, USART for communications, two 8-pin I/O ports, and the Delta-Sigma ADC. This processor includes a single Delta-Sigma ADC that can sample nine inputs (eight external inputs, one internal) through a multiplexor. The task for the MSC1200Y2 is to read the primary sensor analog inputs, do the Analog to Digital Conversions at 24-bits, send alternating set and reset pluses to the magnetometers though a pair of I/O ports, and finally output the ADC results across the processor's USART.

Do to the required sample rate, this task load results in an approximate 45% utilization of the processor. Synchronization of the various steps performed by this processor is critical so it was decided that it was not feasible to perform all the data manipulation and data storage on this single device. For this reason, a second processor is attached to the serial output of the MSC1200Y2 to collect, process, and store the readings.

2.5.2 Atmel AT90CAN128

Atmel's AT90CAN128 is an AVR series microcontroller with 128 kB of flash memory, 4 kB of SRAM, 53 I/O pins, an 8-port 10-bit ADC, dual USARTs, and a CAN interface in a 64 pin TQFP package for under $17.00. The hardware features that are used here are both USARTs, one for programming and the other for serial communication with the MSC1200Y2, the SPI buss to connect to the SD card, one I/O port for buttons and Light Emitting Diodes (LEDs) to indicate the SD card's status, and one I/O port to connect to the accelerometer scale pin and the self test pins on both the accelerometer and rate gyroscope. The CAN interface pins are connected to an external CAN driver and DB-9 port for future implementations.

The tasks dedicated to this processor are to collect data over the serial port from the MSC1200Y2, perform any data manipulation that may be needed, and finally save the data to an external Secure Digital (SD) card. Additionally, the required hardware is in place for connection to a GPS receiver, communication over a CAN interface, and performing ADC conversions of the secondary sensors in the future. Three external buttons and LEDs are provided for user interaction. In the current implementation, one button is used for both the mounting and un-mounting of a SD card's file system and the LEDs are used to initiate the SD card status and activity.

2.6 Data Storage

For persistent data storage, a Secure Digital card is attached to the AT90CAN128. Communication between the processor and SD card are conducted via an SPI interface. Because SD cards are 3.3V devices and are interfaced with a 5V device, 1kΩ resistors are placed between the SD card and AT90CAN128 on any signal going to the SD card. This results in a maximum, worst case current to through the SD card of 1.7mA, low enough not to damage the card. The maximum data rate available between the AtMega128 and the SD card is limited by the maximum speed of the processor's SPI bus, in this case a clock speed of 3.6 MHz.

3 SOFTWARE

The software has been implemented to enable the firmware to monitor and record, at the highest resolution possible and at a rate of 100 Samples Per Second (SPS), both the earth's magnetic field and gravity on the x, y, and z axes. In order to achieve this using low cost hardware, the software was implemented in a modular fashion.

There are three software components to this project, the code running on the MSC1200Y2, the code running on the AT90CAN128, and finally the Matlab code used to read the SD card and perform post analysis. Just as with the hardware, the software is designed to support reading all ten data points, even through only six are being used here. Figure 2 shows a functional block diagram of the software components presented here.

3.1 MSC1200Y2 Code

As mentioned earlier, the task for the MSC1200Y2 is to sample the primary data points at 24-bits and transmit those

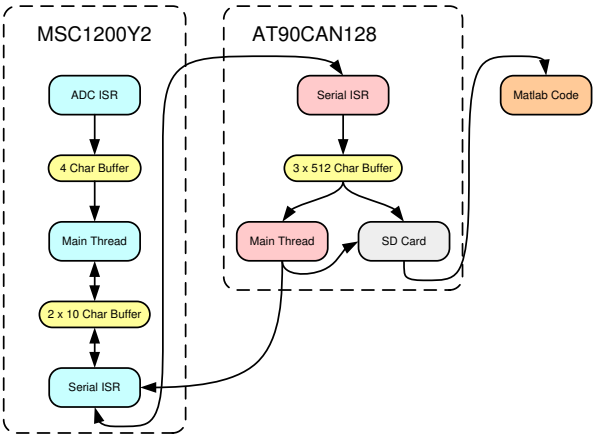

Figure 2: Functional block diagram of the software presented in this paper.

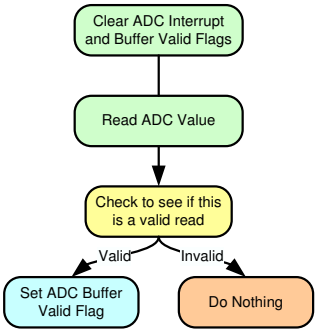

Figure 3: MSC1200Y2 ADC ISR flow diagram.

values across the USART while sending alternating set and reset pulses to the magnetometer. To perform these tasks, three separate threads run on the processor with context switches handled by Interrupt Service Routines (ISRs). The lowest priority thread is the main loop, the medium priority thread is the USART ISR, and the highest priority thread is the ADC ISR. To reach the level of performance required, the most critical aspect to the code running on the MSC1200Y2 is the timing.

Every time an ADC conversion is complete, an interrupt is triggered. It is the ADC ISR's task to service that interrupt by clearing the interrupt flags, reading the ADC value, deciding if it is a valid conversion, and if so, setting the ADC buffer valid flag appropriately. Figure 3 shows the flow diagram for the ADC ISR.

Once the data is collected, it must be transmitted. 100 times per second a line of data is generated consisting of a 32-bit counter, the 9 ADC values each at 24-bits, and an 8-bit spacer between each of the 10 values, for a total of 328 bits. Serial data is transmitted in 8-bit increments with a start and stop bit added by the USART for a 20% protocol overhead. The serial communications between the MSC1200Y2 and AT90CAN128 is run at 115,200 bits per second (bps) even though less than 40,000 bps of data is transmitted. By transmitting data much faster than is actually required, we have both reduced the time needed to transmit each line and left space for more data in the future.

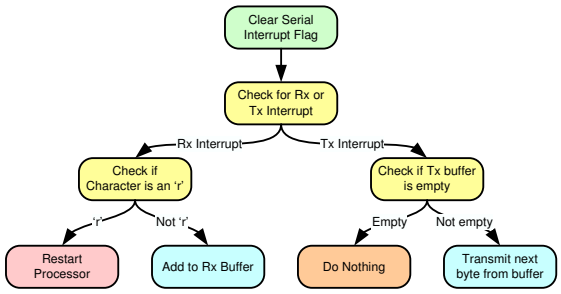

Figure 4: MAC1200Y2 serial ISR flow diagram.

Every time the USART receives a byte or is done transmitting a byte, a serial interrupt is triggered. It is the roll of the Serial ISR to move new bytes from the transmit buffer to the USART buffer whenever a byte transmit is complete, and move received bytes into the receive buffer when bytes are received. Additionally, there is a special character, a lower case "r", that when received by the serial ISR will cause a reboot command to be generated, restarting the processor. Figure 4 shows the flow diagram for the Serial ISR.

The main thread acts as the glue between the Serial ISR and ADC ISR. The task assigned to the main thread is to first initialize the processor and all of the functions and then to loop over the conversions, that is switching the ADC input and starting the conversions, sending set/reset pulses, and once a valid conversion is complete, moving values from the ADC buffer to the serial transmit buffer. During the initialization processes, the first character received is used to calibrate the USART with the correct data rate and parameters. After calibrated, configuration settings such as ADC filter mode and raw sampling rate are sent over the serial line from to the MSC1200Y2.

As much effort as possible has been put into equally spacing all the sensor readings, placing everything that is not time critical in the waiting time between conversions. As soon as a conversion is complete the next conversion is started, and only then is the value from the previous conversion moved from the ADC buffer to the serial buffer. The order of magnetometer and non-magnetometer sensors have been staggered such that there is enough time to fire a set or reset pulse and wait for sensor's output to stabilize before each magnetometer reading is started. Figure 5

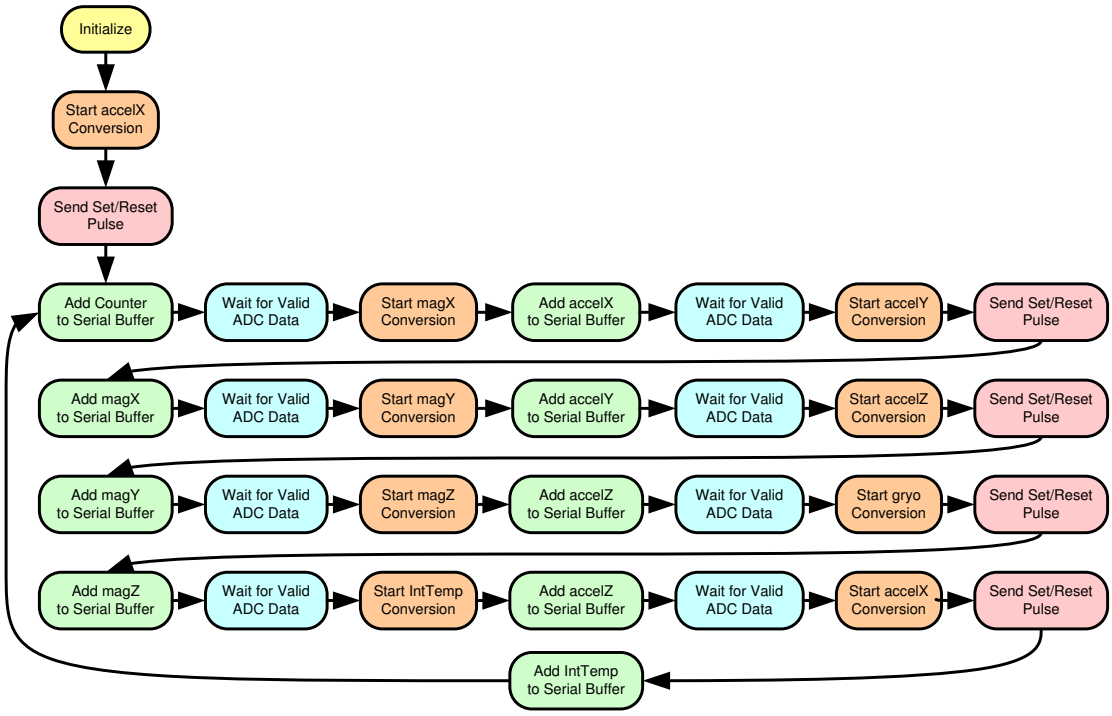

Figure 5: MAC1200Y2 main thread flow diagram.

shows the flow diagram of the main thread.

All of the code written for the MSC1200Y2 is original for this project except for compiler include files, register definition files, and an ADC read utility written in assembly and provided by Texas Instruments. The Integrated Development Environment (IDE) and compiler used is a custom build from Raisonance specifically for the MSC series. The software used to interface with the custom hardware on the board and program the processor is a tool also provided by Texas Instruments.

3.2 AT90CAN128 Code

The task assigned to the AT90CAN128 software is to mount and initialize an SD card, read initialization parameters from the SD card and send them to the MSC1200Y2, collect data from the MSC1200Y2 over the serial port, perform computations on the data, and finally save the data to the SD card. When the AT90CAN128 is powered up it initializes itself and then waits for a button to be pressed indicating that the SD card has been inserted and is ready to be mounted. Once the button is pressed the SD card is mounted and the AT90CAN128 confirms that the SD card is formatted in FAT16. FAT16 is used because it is the easiest file system to implement while maintaining support for volumes up to 2 GB in size and allowing the card to be mounted on any compatible PC.

For a correct implementation of a FAT16 file system, every time the file grows to fill the current cluster, the file allocation table must be searched for the next available cluster, and that cluster added to the existing chain. Because the file allocation table is more than one sector long, multiple sectors may need to be read and written to find the next available cluster and add it to the existing chain. This procedure results in inconsistent sector write times and is unacceptable for our time sensitive approach. To avoid these sporadic sector write wait times, this implementation makes use of a single continuous scratch file and data is written in 512 byte sectors starting at the beginning of the scratch file and continuing until the end.

The AVR library provided by ProcyonEngineering allows for USART transmits and receives to be handled independently. As transmissions from the AtMega128 to the MSC1200Y2 are done only at the very beginning and very end, and are not time critical, a polled-waiting loop is used. Receiving data on the other hand is very time critical as data can show up in the receive buffer at any time and must be removed before another byte arrives and overwrites the buffer. A serial receive ISR is used to move new bytes from the receive buffer into the data buffer as soon as they arrive. Figure 6 shows flow diagrams for both the serial transmit polled-waiting loop and the serial receive ISR.

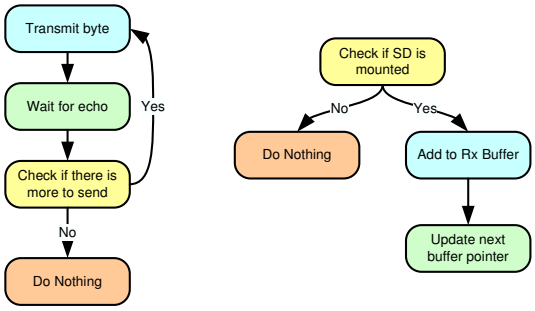

Figure 6: AT90CAN128 flow diagrams for the serial transmit loop (left) and serial receive ISR (right).

A set of three 512 byte buffers are maintained for the serial data received by the AT90CAN128. The receive ISR writes directly into these buffers and the main loop monitors their status. As soon as one of the buffers is full, the SD write function is called to write the buffer contents to the next sector on the SD card. The SD write function is provided by ProcyonEngineering's AVR library.

When the un-mount button is pressed, the serial receive function stops filling the buffers, the last partial buffer is sent to the SD card, both the directory table and file allocation tables are updated to reflect the correct file size, the remaining unused linked clusters become the new scratch file, and a restart character is sent to the MSC1200Y2 so that it will be ready to start over collecting data. Figure 7 shows the flow diagram of the AT90CAN128's main loop.

In the final implementation, 1,641 bytes of SRAM are used for data leaving the remaining 2,455 bytes for stack space and future implementations. The majority of the code for the AT90CAN128 is original except for configuration and helper functions provided by ProcyonEngineering and files included in the WinAVR compiler. Files included in a Circuit Cellar article on FAT16 implementation were used as a guide [13]. Programming of the AT90CAN128 is performed through special hardware on the board and AVRDude software.

3.3 Matlab Data Import Code

Because the data is stored on the SD card's FAT16 file system, the SD card can be removed, placed into a computer, mounted, and data files copied. Data import code can then read the data files into Matlab for post processing.

4 ATTITUDE ESTIMATION

Using the hardware and software detailed in the sections above, the measurements are used to estimate the vehicle attitude. Attitude is the term used to describe a rigid body's orientation in three dimensional space. In a more general sense, it is the description of the relative orientation of two coordinate frames. In vehicle guidance, navigation, and control (GNC) applications near Earth's surface (e.g, applications involving airplanes, marine vessels, etc.) the two coordinate frames of interest are sometimes referred to as the body and navigation reference frames. The body frame is rigidly attached to and moves with the vehicle. The navigation frame is normally a locally level (or tangent) coordinate frame. That is, it has an origin attached to Earth's surface and located directly below the vehicle's current position. Its x-y-z axes are lined-up with North, East and Down (along the local vertical) directions, respectively. Attitude determination systems are used to measure or estimate the relative orientation of these two frames. The information generated by attitude determination systems is indispensable in many GNC applications. A few examples of applications requiring attitude information include pilot-in-the-loop control of manned aircraft, accurate payload pointing on remote sensing platforms, and autonomous navigation and guidance of uninhabited aerial, ground, and marine vehicles.

5 ATTITUDE ESTIMATION

Attitude estimation is the process of determining the three dimensional orientation of an object based on some set of *noisy* measurements such as to minimize the estimation error. Within attitude estimation are two related issues: the first is the parametrization of attitude, and the second is the actual estimation filter. There are several different parameterizations of attitude, each with its own advantages and disadvantages. Briefly, the most common are the Direction Cosine Matrix (DCM), Euler Angles, Rodrigues Parameters, and Quaternions. Note that an excellent treatment of each of these appears in [16].

The Direction Cosine Matrix is the rotation matrix that rotates a vector (or transforms it) from one coordinate frame to the other. It is a 3×3 matrix that is orthonormal (that is, its transpose and its inverse are the same). It is called the direction cosine matrix because $R_{ij} = \cos(\alpha_{ij})$. While there are 9 numbers making up the DCM, only three are independent, the other 6 are constrained by the orthonomality of the matrix.

Euler angles are the classic aircraft angles, usually described as yaw, pitch, and roll (the $[3-1-2]$ Euler angle set), though these are certainly not the only set of pos-

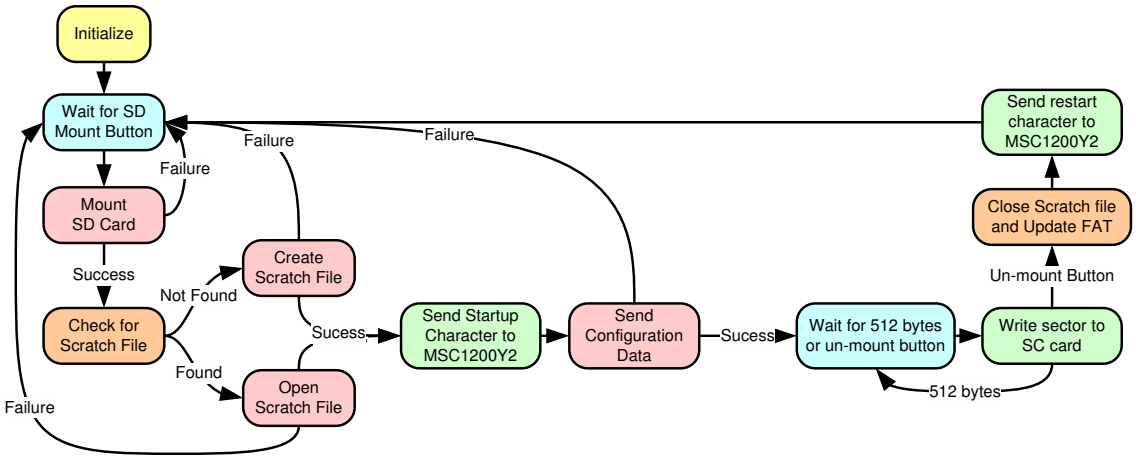

Figure 7: AT90CAN128 main loop flow diagram.

sible Euler angles. The advantage of Euler angles is that only three parameters are required, however, the construction of the rotation matrix requires the evaluation of transcendental functions, which can be costly in computational terms depending on the platform. Another advantage of the Euler angles is that they are immediately intuitive to a human observer, whereas none of the other parameterizations are. However, a disadvantage of Euler angles is the singularity that occurs when pitch, θ, reaches 90° and the remaining angles become poorly defined. This is the so-called "gimbal lock" problem, which is much less of an issue with strapdown attitude systems than it was with mechanical gimbals.

The Rodrigues parameters (also known as the Gibbs vector), is defined as $p = \text{atan}(\phi/2)$, where a is the unit vector about which the rotation occurs, and ϕ is the angle of rotation. Note that the Rodrigues parameters also have a mathematical singularity (which is, in fact, common to all three-parameter attitude parameterizations, see [2]).

Lastly, Quaternions are a four parameter representation of attitude, and thus do not suffer from the mathematical singularity. Quaternions represent rotation as a hyper complex number, that is the three dimensional analog of a complex number. The Quaternion is defined as:

$$\mathbf{q} = \left[\begin{array}{c|c} q_0 & \vec{q} \end{array} \right] \tag{1}$$

where

$$q_0 = \cos(\phi/2) \tag{2}$$

and

$$\vec{q} = \text{asin}(\phi/2) \tag{3}$$

Note that all of the transformations and conversions with Quaternions are accomplished with simple multiplies and additions.

In this work, we specifically solve for attitude in the Quaternion domain, by solving Wahba's problem for the measurements of Earth's magnetic and gravitational fields. The basic formulation of attitude determination from two or more non-collinear, non-zero vectors measured in the body frame and known in the navigation frame is referred to as Wahba's problem, based on her 1966 formulation [22]. While there exist a variety of solutions for Wahba's probelm (see for instance, [3]), our solution is based on a linearization of the attitude estimation problem in the navigation frame (as opposed to the body frame). The great advantage of this formulation is that it results in a linear time invariant measurement equation, easily solvable by either an iterated least squares or Kalman filter solution.

While the complete treatment of this attitude estimation algorithm is beyond the scope of this work, see [4, **?**, 9]. It is important to note that in this application, the two vector quantities being measured are Earth's magnetic and gravitational fields. Note, the on-board accelerometers do not, however, measure only the gravitational acceleration. Rather, the accelerometers measure the quantity $\vec{g} - \vec{a}$, the specific force on the accelerometers. Note that in the MAV application, we use the numerical derivative of successive GPS measurements to estimate $\vec{a}$ in the navigation frame. For the experimental set-up in this work, the sensor was stationary, and thus we need no further input as there is no platform acceleration.

The basic steps of the attitude filter are as follows (again, see [4, **?**, 9] for details):

1. Initialize the attitude quaternion estimate to:

$$\hat{q} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T \tag{4}$$

and the error quaternion to

$$q_e = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T \tag{5}$$

2. Use the attitude estimate, $\hat{q}$, to map the measured magnetic field to the navigation frame. That is:

$$\hat{h}^n = \hat{q} \otimes h^b \otimes \hat{q}^* \tag{6}$$

where $\otimes$ is quaternion multiplication and $\hat{q}^*$ is the quaternion complement of $\hat{q}$.

3. Map the accelerometer measurements to the navigation frame as above.

4. Formulate the errors in the navigation from by subtracting the estimated values from the known ones:

$$\delta\hat{h}^n = \vec{h}^n - \hat{h}^n \tag{7}$$

and

$$\delta\hat{a}^n = \vec{a}^n - \hat{a}^n \tag{8}$$

5. Formulate the measurement matrix directly from the body measured quantities:

$$H = \begin{bmatrix} -2\left[\vec{h}^b\times\right] \\ -2\left[\vec{a}^b\times\right] \end{bmatrix} \tag{9}$$

6. Solve the Normal equations (i.e. by using the Moore-Penrose pseudo-inverse):

$$H^\dagger = \left[H^T H\right]^{-1} H^T \tag{10}$$

and use it to compute the new error quaternion.

7. The new error quaternion is computed from:

$$q_e = \alpha H^\dagger \begin{bmatrix} \delta\hat{h}^n \\ \delta\hat{a}^n \end{bmatrix} \tag{11}$$

where α is a smoothing parameter chosen between 0 and 1.

8. Update the quaternion estimate using quaternion multiplication:

$$\hat{q}(+) = \hat{q}(-) \otimes q_e \tag{12}$$

note that this is rotating $\hat{q}$ by the small correction q_e.

9. Repeat from step 2 until converged. Also note that the previous value for $\hat{q}$ can be used to initialize the algoritm the next time through.

Note that there are several other formulations that break down into roughly three distinct types: (1) Exact solutions which cannot easily accommodate errors in the sensors, (2) batch least squares approaches (of which the above is a member), and (3) filtering methods which cast the problem in a traditional observer form (for further discussion of the attitude estimation, see [8]).

Using our experimental hardware, as previously published in [6], we have demonstrated a short term (1 minute) attitude noise of less than 0.13° in pitch, roll, and yaw, and a long term (1 hour) attitude drift of less than 0.25°. Likewise, a static reading was preformed, the sensor reoriented, and then moved back to the original position, and showed less than 0.25° of offset from the original attitude.

6 SENSOR CALIBRATION

While the main contribution of this work is a detailed study of the sensor bias stability, the method for calibration is worthy of mention. In this work, we use a technique for calibrating the accelerometers and magnetometers directly from rotation using a two-step algorithm described in [7, 10]. The algorithm stems from the observation that when rotating the perfect sensor around through all angles, then the measurements plotted would trace out a circle for a 2D sensor, and a sphere for a 3D sensor.

The basic measurement equation for a given axis on a sensor is:

$$h_{meas} = \frac{1}{sf} h_{true} + b + \omega_n \tag{13}$$

where h_{meas} is the measured output, h_{true} is the true measurement, b is the bias or offset, and ω_n is the wide band noise on the sensor. In the two axis case, when the body fixed sensor is rotated around a circle, its components should be such that when plotted they have a center point of (0,0) and a radius of the value of the magnitude of the magnetic field. Bias errors will cause the circle to be shifted off of the origin, and scale factor errors will distort the circle into an ellipse.

The algorithm for calibration is a non-linear two step algorithm which first does a least squares estimation of a set of parameters which are then manipulated algebraically to extract the scale factor and bias errors. Note than non-orthogonality of the measurement axes will cause a distortion to the ellipse as well, and this can also be accounted for. The center of the ellipse is the bias error for both axes, and the semi-major and semi-minor axes of the ellipse are the scale factors.

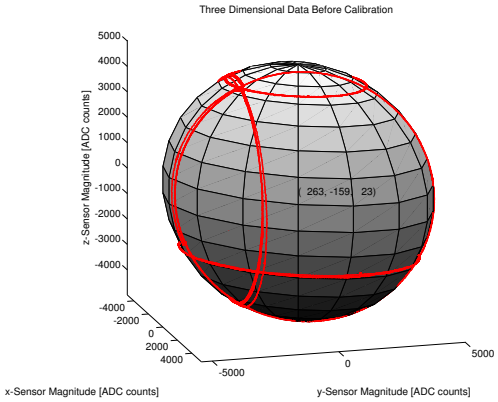

Figure 8: Experimental three axis magnetometer pre-calibration data displayed on a sphere of the true magnetic field.

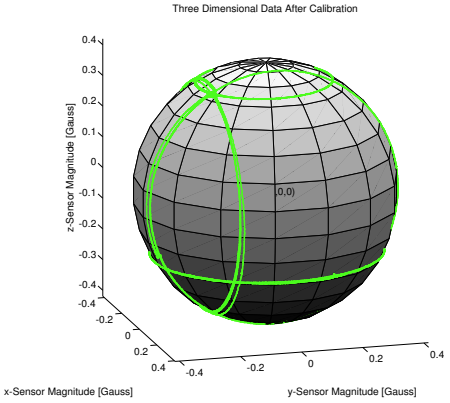

Figure 9: Experimental three axis magnetometer data displayed on a sphere of the true magnetic field, after it has been calibrated.

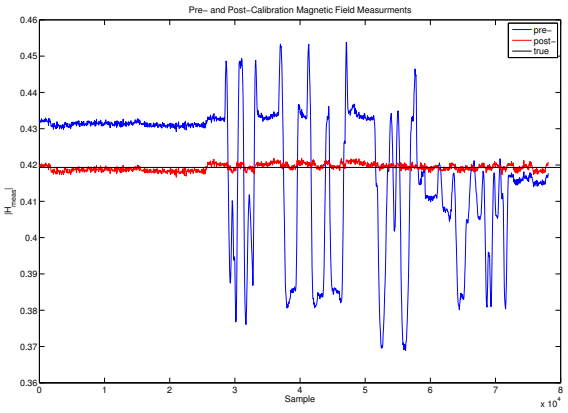

Figure 10: Pre- and Post-Calibration magnetic field magnitude as measured by the 3-axis magnetometer. Standard deviations change from 18.3mG to 0.93mG with calibration.

A major extension to this calibration methodology which includes the non-orthogonality of the sensor axes induced both by physical misalignment and soft/hard iron interactions on the magnetometers is detailed in [5]. Using this extended calibration methodology, which requires only rotation of the MAV in order to extract the calibration constants of biases, scale factors, and non-orthogonality in the sensor axes. Note that this method does not require an external reference in order to perform the calibration. The caveat of this is that the calibration constants are in the sensor axes, not in the vehicle body axes. In order to perform that rotation, an external measurement of attitude is required.

Also, there is no correction for misalignments between the magnetometer and accelerometer traids. This can be corrected, and will be the subject of a forthcoming paper.

Using the calibration methodology detailed in [5], the sensor is rotated in order to extract the parameters. The pre- and post-calibration of the magnetometers can be seen in Figures 8 and 9, respectively. Lastly, the pre- and post-calibration magnitude of the magnetic field is plotted as a function of time for the experimental data in Figure [**?**]. This pre-calibration data has a standard deviation of 18.4 mGauss, and a post-calibration standard deviation of 0.93 mGauss.

7 BIAS STABILITY

All of the previous sections have demonstrated that we have achieved a small, lightweight, and low-cost sensor platform that is quite capable of producing high quality attitude measurements. However, as the plots presented in our previous work ([6]) demonstrate, even on a single hour time scale,

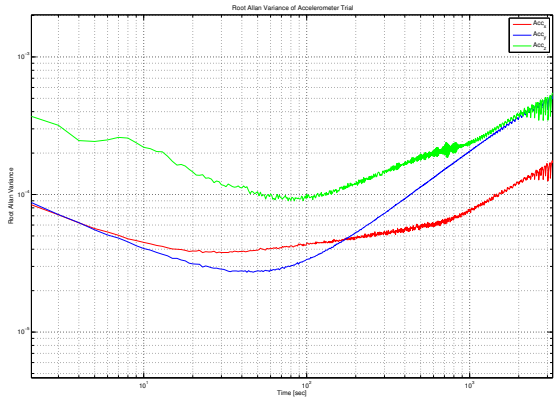

Figure 11: The Allan Variance of the x−, y−, and z−axis MEMs accelerometer, data taken at 100 Hz for 24 hours.

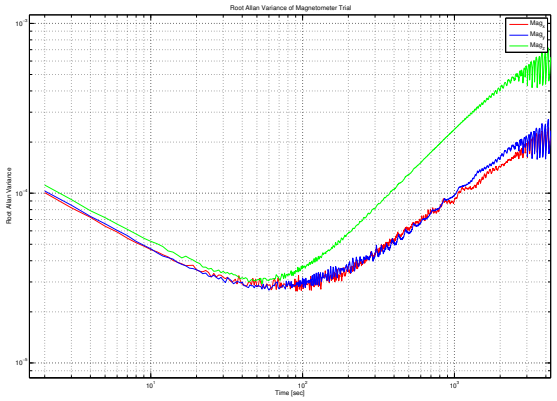

Figure 12: The Allan Variance of the x−, y−, and z−axis magnetometer, data taken at 100 Hz for 24 hours.

the bias can be seen to drift significantly. Indeed, outside of the short term (white) noise, the attitude variation can be seen to be bias drift. Close inspection of the recorded temperature (sampled at the MSC1201 chip) can be shown to be correlated with the boas drift.

With these observations in mind, long term data to determine the bias stability of the sensor platform: (1) long term data was recorded (24 hours worth) and the Allan Variance was used to characterize the sensor noise, (2) the bias versus temperature was examined, and an empirical (linear) relationship was derived, and (3) the sensor package was recalibrated several times in order to see how much the calibration constants changed over the 24 hours.

7.1 Allan Variance

The Allan Variance is based on the work of David Allan characterizing the stability of clocks and oscillators while working at Hewlett Packard. The advantage of the Allan Variance over normal variance methods is that it converges to a finite value for most types of noise, whereas the conventional variance does not. Flicker noise and random walk are two examples of noise types that do not converge under normal variance estimates, but do under the Allan Variance. For a detailed explanation of the Allan Variance method as applied to sensor characterization, see [18].

Long term data was taken with the sensor at rest in a static orientation. The data was taken at the full data rate (100 Hz per sensor) for over 24 hours. The plot of the root Allan Variance is shown in Figure 11 for the three accelerometers, and Figure 12 for the magnetometers. From these plots, we can see that white noise dominates the response out to an integration time of approximately 40 seconds. Beyond that, we see much more correlated noise (and thus averaging will

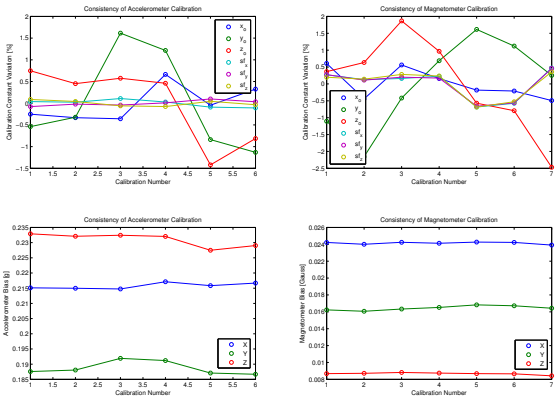

Figure 13: Variation of calibration constants, recalibration performed several times of the course of 48 hours. Left figures are accelerometers and the Right ones are the magnetometers.

not increase sensor performance).

7.2 Periodic Recalibration

As another method for determining the sensor stability of our sensor suite, the sensor was periodically recalibrated at distinct times, several hours apart, over the span of 48 hours while the sensor was powered and logging data. The data is presented in aggregate form in Figure 13, with the upper figures showing the variation as a percentage (which can be seen to remain under 2%) and the lower figures showing the bias directly in g′s or in Gauss. As can be seen from the figure, the total variation in bias is less than 0.01g′s in the accelerometers, and less than 0.001Gauss in the magnetometers.

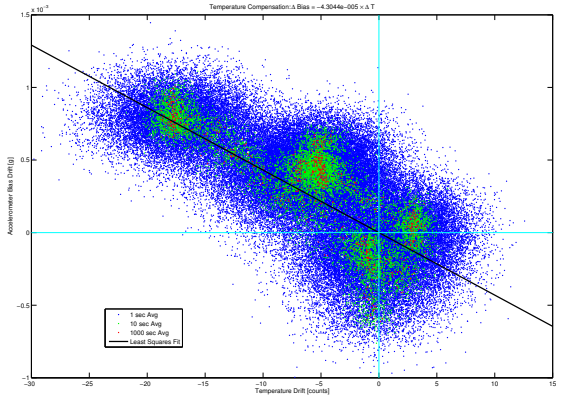

Figure 14: Bias Drift vs. Temperature for the Accelerometer x-axis sensor. Data is averaged for 1,10, and 100 seconds, and a linear least squares fit is used to determine the best line.

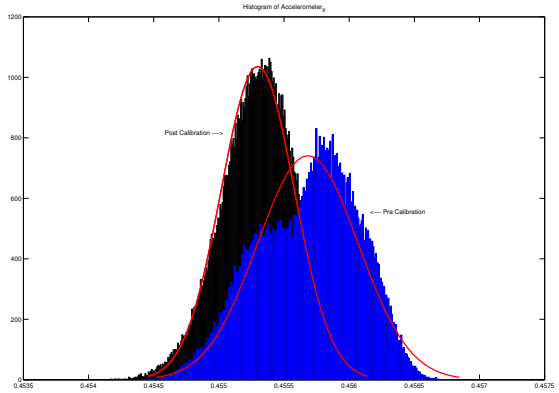

Figure 15: Histograms of the Accelerometer x−axis measurements before and after temperature compensation.

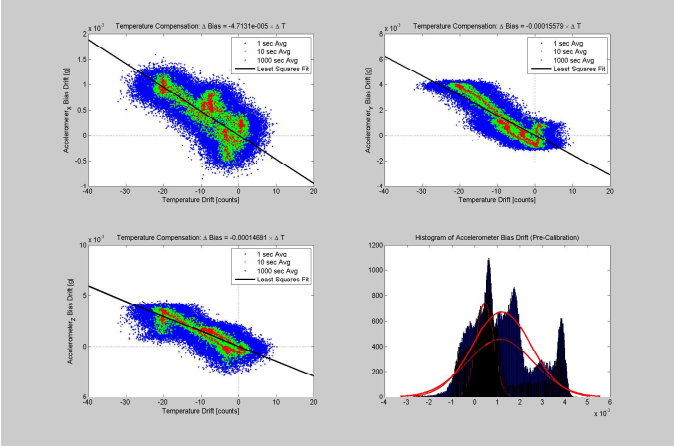

Figure 16: Accelerometer Temperature Compensation.

7.3 Linear Temperature Compensation

The thought is that for most low-cost, inexpensive sensors, the primary cause of bias drift is temperature (note that scale factors and non-linearities are also functions of temperature, but that these are smaller effects). In order to remove the effect of temperature, we begin by revisiting our model of a generic sensor given in Eq. 9, repeated here.

$$h_{meas} = \frac{1}{sf} h_{true} + b + \omega_n \tag{14}$$

Our most basic model holds that b is constant. However, we extend this to include the variation in temperature of the bias constant by adding an equation for b:

$$b = b_0 + b(T) + \omega_b \tag{15}$$

That is, b now has a component that is a function of temperature (which is measured at the microcontroller on the sensor board). In order to determine the relation between temperature and bias drift, the data was averaged into one second bins (this was done to make the data more manageable, and from the Allan Variance plots, should not distort the data), and the initial point removed from the sensor as well as the temperature outputs, which are then plotted against each other. This can be seen for the x−axis of the accelerometer in Figure 14. For this work, we use only a simple linear model as our function of temperature; that is $b(T) = K \times T$, where we determine the gain K using a least squares fit.

We perform this same calculation for both the accelerometers and magnetometers, on every axis, and find the gain for each axis. We then remove the temperature related portion of the bias from the sensor, based on the temperature measurement from the microcontroller, and the initial bias reading from the sensor. Figure 15 shows a histogram of the measured data from the x−axis accelerometer before and after the temperature compensation. What is immediately seen is that the post-compensation histogram is very close to an ideal Gaussian distribution, whereas the pre-compensation histogram is not even close.

This was done for every axis of both the accelerometers and magnetometers, and the data is presented in Figures **??**, 17, 18, and 19 along with the histogramps for the pre- and post-compensated data.

Lastly, in order to determine the benefits of this kind of temperature compensation in the attitude domain, we ran the data through the attitude estimation algorithm (both the uncompensated and the temperature compensated). While

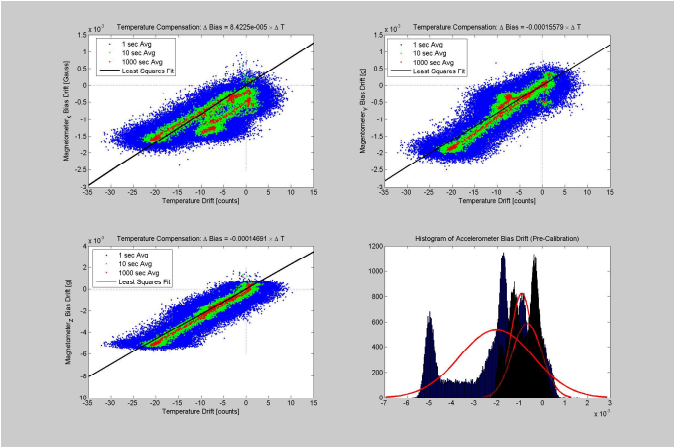

Figure 17: Magnetometer Temperature Compensation.

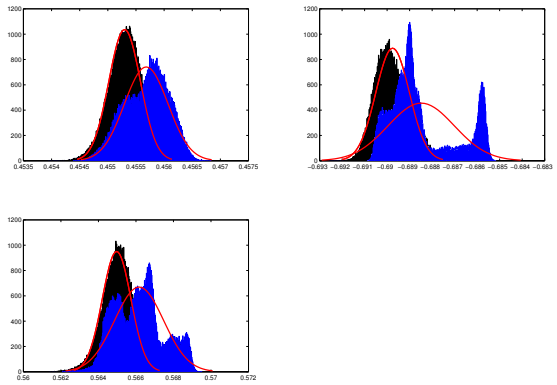

Figure 18: Accelerometer histograms, pre- and post-compensation.

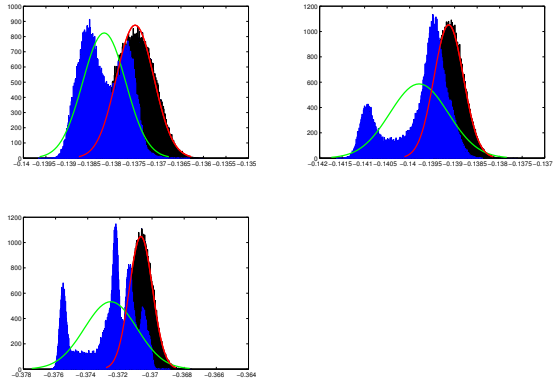

Figure 19: Magnetometer histograms, pre- and post-compensation.

the exact attitude of the sensor suite is unknown (no truth measurement was available), we can, however, comment on the stability of the computed attitude solution. Again, in order to keep the data size manageable, we averaged the data over each second (thus, our standard deviation should be $1/\sqrt{(n)}$ better than the normal attitude solution, where $n = 100$. The pre- and post-compensation histograms are presented in Figure 20, which shows improvement in the attitude estimate, and a reduction in the standard deviations of the recorded attitude solution.

This data is also recorded in Table 1, which summarizes the improvement in the attitude estimation achieved by temperature compensating with a simple linear temperature fit to the bias drift. Note that the attitude estimate prior to temperature compensation was $[121.6327° - 27.081° - 50.5425°]$ for yaw, pitch, and roll. After temperature compensation, the attitude estimate was $[121.6939° - 27.0504° - 50.6771°]$. The standard deviation (for 1 second averaging) was $[0.08479°0.030164°0.12353°]$ pre-compensation, and $[0.051829°0.017386°0.059834°]$ post-compensation. Corrections for sample averaging have been included in the table.

Note that it is quite clear from the data that a linear temperature compensation does not remove all of the error terms. Indeed, there is a great deal left to explore in the future regarding this sensor package.

8 CONCLUSION

In this work we have demonstrated the continuing development of a sensor suite suitable for attitude determination in a physically small, low-cost, and low-power package. By utilizing a very high bit $\Delta\Sigma$ ADC, a good precision voltage reference, and careful noise isolation, we achieve high performance at a low cost. Looking at the Allan Variance, we see that the noise is uncorrelated for integration times of less than 40 seconds. We specifically look at the bias stability of this sensor, and using a simple linear temperature fit, we are able to reduce the attitude noise by a factor of two. Over a 24 hour period, we report an attitude standard deviation of $[0.0848°0.0302°0.1235°]$ post temperature compensation, when the sensors samples are averaged in bins of 100 samples (net 1 Hz rate). This accuracy and stability are quite adequate for flight control and stabilization of Micro Air Vehicles.

ACKNOWLEDGEMENT

The authors would like to acknowledge the University Affiliated Research Center (UARC) Aligned Research Program (ARP) Grant that has funded the work contained

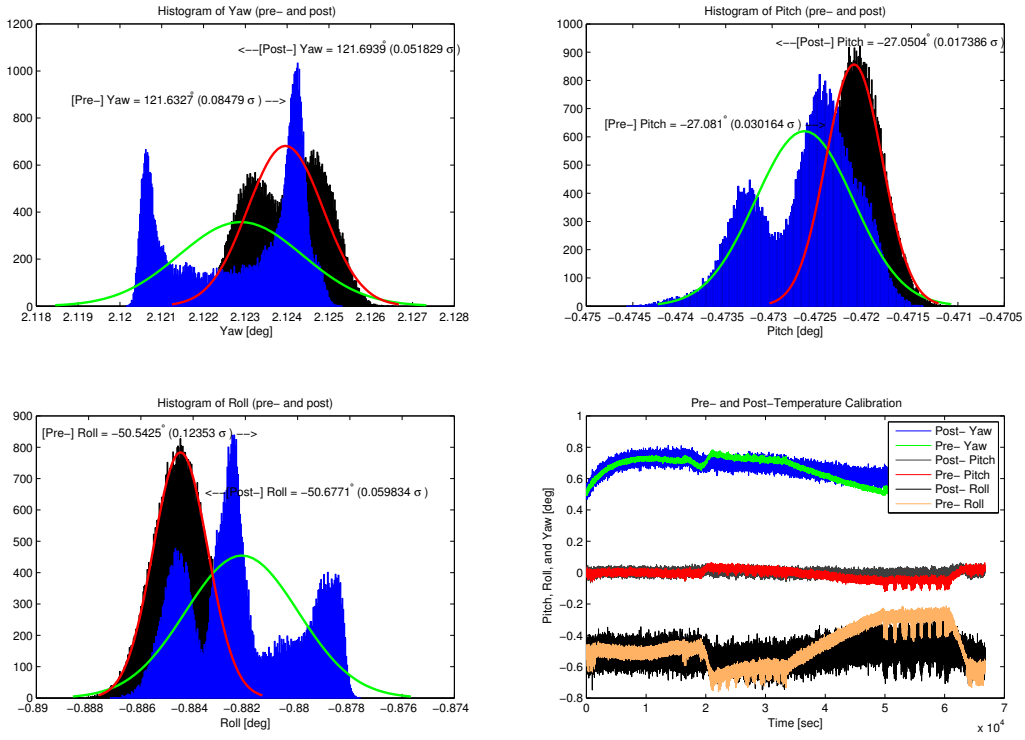

Figure 20: Attitude histograms, pre- and post-compensation.

	Attitude		$1-\sigma$ (1 Hz)		$1-\sigma$ (10 Hz)		$1-\sigma$ (100 Hz)	
	Pre-	Post-	Pre-	Post-	Pre-	Post-	Pre-	Post-
yaw (ψ)	121.6327°	121.6939°	0.08479°	0.051829°	0.26813°	0.163898°	0.8479°	0.5183°
pitch (θ)	−27.081°	−27.0504°	0.030164°	0.017386°	0.09539°	0.054979°	0.3016°	0.1739°
roll (ϕ)	−50.5425°	−50.6771°	0.12353°	0.059834°	0.39064°	0.189211°	1.2353°	0.5983°

Table 1: Attitude and Standard Deviations for Pre- and Post-Temperature Compensation

within this paper. Also, we would like to acknowledge the long and fruitful collaboration on this topic between the authors and Dr. Demoz Gebre-Egziabher of the University of Minnesota, without which this project would have long since faltered.

References

[1] Clark E. Cohen. *Attitude Determination Using GPS*. PhD thesis, Stanford University, Stanford, California 94305, December 1992.

[2] J. L. Crassidis and F. L. Markley. Unscented filtering for spacecraft attitude estimation. *AIAA Journal of Guidance, Control, and Dynamics*, 26(4):536–542, 2003.

[3] G. Creamer. Spacecraft Attitude Determination Using Gyros and Quaternion Measurements. *The Journal of Astronautical Sciences*, 44(3):357 – 371, July - September 1996.

[4] G. H. Elkaim. *System Identification for Precision Control of a WingSailed GPS-Guided Catamaran*. PhD thesis, Stanford University, Stanford, CA, 2001.

[5] G. H. Elkaim and C. C. Foster. Extension of a nonlinear, two-step calibration methodology to include non-orthogonal sensor axes. *IEEE Transactions on Aerospace Electronic Systems*, in press, 2007.

[6] C. C. Foster and G. H. Elkaim. Development of the Metasensor: A Low-Cost Attitude Heading Reference System for use in Autonomous Vehicles. In *Institute of Navigation ION-GNSS Conference, Fort Worth, TX*. ION, 2006.

[7] D. Gebre-Egziabher and G. H. Elkaim. Calibration of strapdown magnetometers in magnetic field domain. *ASCE Journal of Aerospace Engineering*, 19(2):1–16, 2006.

[8] D. Gebre-Egziabher and G. H. Elkaim. Mav attitude determination from observations of earths magnetic and gravity field vectors. *IEEE Transactions on Aerospace Electronic Systems*, page in press, 2007.

[9] D. Gebre-Egziabher, G. H. Elkaim, J. D. Powell, and B. W. Parkinson. A Gyro-Free, Quaternion Based Attitude Determination System Suitable for Implementation Using Low-Cost Sensors. In *Proceedings of the IEEE Position Location and Navigation Symposium, PLANS 2000*, pages 185 – 192. IEEE, 2000.

[10] D. Gebre-Egziabher, G. H. Elkaim, J. D. Powell, and B. W. Parkinson. A non-linear, two-step estimation algorithm for calibrating solid-state strapdown magnetometers. In *8th International St. Petersburg Conference on Navigation Systems, St. Petersburg, Russia*. IEEE/AIAA, 2001.

[11] D. Gebre-Egziabher, R. Hayward, and J. D. Powell. Design of multi-sensor attitude determination systems. *IEEE Journal of Aerospace Electronic Systems*, 40(2):627 – 643, 2004.

[12] Demoz Gebre-Egziabher. *Design and Performance Analysis of a Low-Cost Aided-Dead Reckoning Navigation System*. PhD thesis, Department of Aeronautics and Astronautics, Stanford University, Stanford, California 94305, December 2001.

[13] W Hue I Sham and P Rizun. Portable FAT Library for MCU Applications. *Circuit Cellar*, pages 18–26, March 2005.

[14] Microbotics Inc. *MIDG II Specifications*, August 2005.

[15] R. P. Kornfeld, R. J. Hansman, and J. J. Dyest. Single Antenna GPS Based Attitude Determination. In *Proceedings of the Institute of Navigation National Technical Meeting ION-NTM, Long Beach CA, January 1998*, pages 345 – 354. ION, 1998.

[16] Jack B. Kuipers. *Quaternions and Rotation Sequences*. Princeton University Press, Princeton, New Jersey, 1999.

[17] J. M. McMichael and M. S. Francis. Micro air vehicles - toward a new dimension in flight. *Defense Advanced Research Projects Agency (DARPA) Briefing*, http://www.darpa.mil/tto/MAV/mavauvsi.html.

[18] L.C Ng and D.J. Pines.

[19] Office of the Secretary of Defense. Unmanned aircraft systems roadmap 2005 - 2030. *http://www.acq.osd.mil/usd/uavroadmap.pdf*, December 2002.

[20] Xsens Technologies. *MT9 Inertial 3D Motion Tracker Specification Sheet*.

[21] Cloud Cap Technology. *Crista Inertial Measurement Unit, Interface/Operation Document*.

[22] Grace Wahba. Problem 65-1 (Solution). *SIAM, Review*, 8:384 – 386, 1966.