Initial Results from an In-Situ Environmental Monitoring Marine Mammal Tag

Gabriel Hugh Elkaim*, Eric B. Decker*, Guy Oliver†, and Brent Wright‡
*Autonomous Systems Lab, Computer Engineering, University of California, Santa Cruz, Santa Cruz, CA 95064 USA
†Institute for Marine Science, University of California, Santa Cruz, Santa Cruz, CA 95064 USA
‡Boulder Creek, CA 95006 USA

***Abstract*— Current understanding of the behavior of marine mammals (pinnipeds) is quite limited by the observation technology used. Surface tracking using geolocation or Argos satellite tags have shown that these mammals range much farther than previously thought. Relatively simple time/depth recorders (TDRs) have shown that they dive to depths of over 1000 meters for over one hour.**

In order to further the understanding of these aquatic creatures, a smaller, more capable tag is being developed that can be deployed for longer durations and with increased capability.

The MAMMARK tag measures approximately 2.5 x 4 cm., uses a low-power microcontroller, and multiplexes a set of sensors through a high resolution analog-to-digital converter. The sensor suite consists of temperature, depth, speed, salinity, three axes of magnetic field, three axes of acceleration, and GPS. GPS measurements are, of course, only available at the surface. Quick measurements are enabled by keeping the GPS receiver in "hot-start" mode enabling satellite reacquisition in as little as one second upon return to the surface.

The three-axis magnetometer and accelerometer are used to construct the attitude of the pinniped using a quaternion-based Whaba's problem solution. Fusing the attitude data with velocity measurements and GPS position data at the surface, a dead reckoning filter is used to generate the full pinneped trajectory.

This work extends previous development efforts on the MAMMARK tag by creating a unified model for sensor errors based on the Allan Variance and autocorrelation methods. Bias drift is modeled as a first order Gauss-Markov process (exponentially correlated). Long term static (bench) accelerometer data is used to develop this unified model, showing short and long term stability.

Data taken from the three axis accelerometers while tumbling the sensor in order to calibrate the null shift, scale factor, and cross-coupling errors using a two-step non-linear estimation algorithm.

Lastly, bench tests of power consumption are performed and used to estimate probable MAMMARK tag longevity while attached to the tagged pinneped.

I. Introduction

This paper describes ongoing development of a low-cost, small, dead reckoning sensor, pictured in Fig. 1, used to acquire data on diving sea mammals. Previous development efforts are described in [7] and [9], with a more formal treatment of the Navigation Filtering presented in [8]. As such, this paper has largely the same background and hardware/software descriptions as these earlier works, which are included here for completeness. This work describes ongoing changes to the hardware and software and presents some early results from bench tests of the actual sensors.

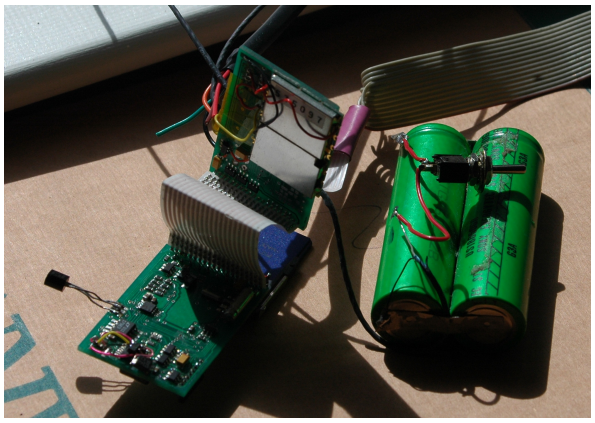

Fig. 1. The Prototype Pinniped Sensor Tag. This tag is designed to be attached to the skin of marine mammals and record their velocity, orientation, and depth, along with the environmental salinity and temperature.

Marine mammals are inherently difficult to study. The cetaceans (whales, dolphins, and porpoises) are totally aquatic and even the amphibious pinnipeds (seals and sea lions) spend most of their lives at sea. Biologists can only catch a glimpse of them as they surface and so have turned to technological solutions to study these animals at sea. The most extensively studied is the northern elephant seal, *Mirounga angustirostris*, with much of this work performed at the elephant seal rookery at Año Nuevo State Reserve, 65 kilometers north of Monterey.

In the early 1980's time-depth-recorders (TDRs), which record changes in water pressure over time, were first attached to elephant seals. Instruments were deployed on seals by gluing them to the seal's pelage just prior to their departure on a foraging trip and were recovered 2.5–8 months later when the seals returned to the rookery. The initial results revealed dives that were incredibly long, phenomenally deep and continuous 24 hours a day, day after day, week after week [3]. Mean dive duration of adult females was 22.1 minutes followed by a surface interval of 2.3 minutes [4]. One female in a 10 hour period made 10 dives, 7 of which exceeded an hour, with the longest lasting 97 minutes, and each of these dives was followed by a surface interval of 3 minutes or less. Modal dive depths ranged from 350 to 600 m with a maximum depth exceeding 1600 m.

By adding a photocell to the TDRs, locations could be

1-4244-1537-3/08/$25.00 ©2008 IEEE

calculated by determining the day length, which revealed latitude, and the offset of the times of sunrise and sunset from the place where they were originally tagged, which revealed longitude [14]. This method is accurate to within approximately 100 km.

For elephant seals, this system of geolocation was adequate to describe their long-range movements throughout the north-eastern Pacific. It showed that they undertake two complete foraging migrations each year [6] and that the sexes segregate on their foraging migrations and employ different foraging strategies [5]. Adult males forage off the continental shelf, especially along the Aleutian Islands, and pursue benthic fish, rays, skates, and cephalopods. Females move well offshore and into the pelagic zone where they forage in the upper 1000 m of the water column. Their daily pattern of diving—deep in the day and shallower at night—tracks the diurnal vertical migration of the community of organisms known as the deep scattering layer upon which they feed.

Improvements in the ability to track the seal's movements occurred in the mid 1990s with the development of transmitters, which could be detected by the polar orbiting Service Argos/NOAA satellites. When a satellite was above the horizon and a seal equipped with an Argos transmitter surfaced, an uplink occurred and location could be calculated. The more uplinks which occurred in a single surfacing, the greater the accuracy of the location, and since the number of uplinks per fix is known, a location quality (LQ) could be determined. Because elephant seals are underwater about 90% of the time they are at sea, most seals only had one to four "good" locations a day and over 90% of these were Argos LQ 0, A, or B. These range from an accuracy of 9 km $\pm$ 16 km for LQ 0 hits to 48 km $\pm$ 71 km for LQ B hits [5], which is a considerable improvement over geolocation. Additional advantages of the Argos tags is that they can be tracked in near-real time and approximate locations of mortality can be determined if the transmitter stops transmitting and the seal is never seen again or if the transmitter appears to be moving as if on a ship and the seal is never seen again.

Additional sensors added to the TDRs in the 1990s included thermisters, velocity meters, hydrophones, video cameras and heart rate monitors. Suddenly biologists were data rich as the number of instrumented elephant seals soared past 200. The range of insights into the biology of these seals was fascinating. The data revealed that the seals, while diving, were employing a variety of behavioral [2] and physiological [22] "tricks" enabling them to have a lower metabolic rate while diving than when sleeping on the beach!

But biologists are constantly impatient for technological advances to occur, and they can construct overly complex contraptions in their attempts to learn more about their animals. Two areas where improvements were sought were in the accuracy and frequency of surface locations and the ability to record the 3-dimensional movements of the seal between its surfacing locations. Several MAP tags that married a GPS receiver to a TDR, a velocity meter, and a 3-dimensional digital compass were constructed. It was conceptually successful [17],

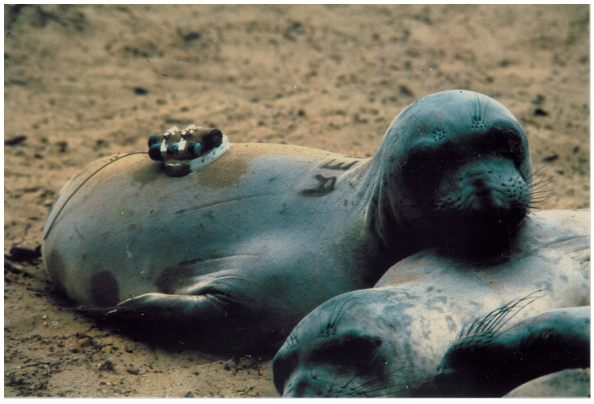

Fig. 2. A young elephant seal with a modern time data recorder (TDR) affixed to its body. This TDR is approximately 4 times as large as the prototype MAMMARK tag.

but was a 14-pound behemoth requiring 12 d-cell batteries to power it and was only deployed on one translocated seal [18], which never returned to Año Nuevo.

Tags were getting increasingly larger and more expensive. Only the best-funded researchers could afford to deploy the newer tags and even they were limited in how many they could afford to deploy because of costs ranging from several thousands to tens of thousands of dollars. Clearly, there was a need for a newly designed tag with the capabilities of the MAP tag while shrinking its size and cost. It will be mounted on top of the seal's head so that when the seal surfaces, the GPS antenna will rapidly shed water and have maximum exposure to the sky. The tag needs to contain at least two external environment sensors, temperature and salinity, which will allow identification of water masses. The electronics will need to be potted to protect them from salt water and to allow them to withstand up to 3000 psi of pressure. To minimize disturbance and work for the seal, the cross-sectional area of the new tag should be less than 5% of the cross-sectional area of a seal's head. To be affordable and deployable in large numbers target cost needs to be between $500-700. The new MAMMARK tag is intended to fulfill these requirements.

The paper is organized as follows: Section I presents the background and motivations for developing the MAMMARK tag. Section II presents an overview of recent changes to the hardware and software of the system, Section III details the physical hardware design. Section IV presents a detailed software description. Section V presents a unified sensor error model, and how it is used to determine the bias drift from actual data. Section VII analyzes static bench data from the accelerometer and temperature probes. Section VI provides an overview of the two-step non-linear calibration methodology, and Section VIII demonstrates the results of applying this method to the tumbled accelerometer data. Lastly, conclusions and future work are presented in Sections IX and X respectively.

II. Recent Modifications

Technological changes have prompted reevaluating the hardware and software components that form the basis for the MAMMARK tag. On the hardware side, the GPS and radio chipsets were changed. On the software side, the custom operating system was abandoned in favor of using TinyOS 2.x as the structural underpinnings of the system.

A. Hardware Changes

The first component reevaluated was the GPS receiver. A Trimble Lassen iQ GPS receiver was used on the origianl tag. The receiver specification are for a hot/warm/cold start of 10/38/50 seconds. Empirical data suggested hot start times closer to 15 seconds. As the GPS unit is attached to the head of the animal and will potentially have only a short interval of time to acquire a fix before the animal re-submerges on its next dive. Thus hot start time is a critical parameter for tag performance. New GPS chips (such as designs based on the Sirf–III chipset [15]) have an order of magnitude better hot start performance, with advertized hot/warm/cold times of $<$ 1s/35s/35s). These times have been verified empirically and hot start times significantly better than 1 sec have been observed under trying conditions (simulated water environment, indoors).

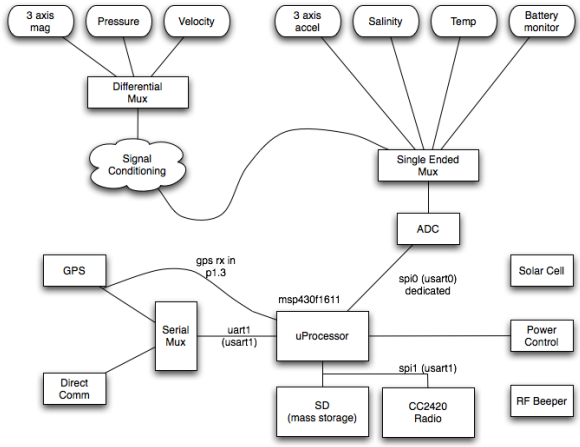

Fig. 3. Revised Tag Hardware Block Diagram. The system has been designed to be small and very low power in order to record data in-situ for long duration on *very* limited battery power.

To protect against the harsh salt water environment the entire tag is potted in marine epoxy. Performance of the GPS subsystem will be dependent on the effects this epoxy has on the antenna system. These effects will need to be evaluated in order to determine what, if any, methods exist to mitigate the attenuation effects of the epoxy.

The original hardware structure of the tag also presented some challenges. As can be seen from Fig. 4, there are several shared components. The SD card and serial communications (UART1) share USART1 on the processor and three serial modules share the UART1 port. The shared components requires multiple levels of arbitration which greatly complicates the software design. The previous software revision ran into this problem and necessitated structural changes. This prompted reevaluating the approach and led to adopting TinyOS 2 [16] for the structural underpinnings (see Section II-B).

GPS communication was discovered to be problematic. In order to conserve power, the GPS is normally switched off (the GPS RAM is powered via a battery connection to preserve the state and enable hot-start). On power up, the GPS takes an unspecified amount of time before periodic updates begin to be transmitted. This required switching the USART/UART hardware to listening to the GPS for a significant amount of time, effectively blocking other peripherals from being accessed (including the SD card).

The solution was to move the RX pin of the GPS to its own input line that can generate a microcontroller interrupt. This allows the processor to independently process GPS input once the GPS configuration has been set up following initial power on. By moving the RX pin to an interrupt, the system can effectively ignore the GPS data stream until it is working, and then process the data directly. During GPS initilization bidirectional communications is required to properly set up the GPS. This is provided via UART1 and a serial multiplexer and is only active during startup. This will not conflict with other peripherals sharing the buses.

The last major component changed was the RF module. The original radio consisted of a Radiotronix Wi.232DTS module, which simulates a hardwired RS-232 serial link via a radio link. After analysis of the communications needs, it became clear that a communications stack implementing reliable and networked communication was needed. This will allow future robust mesh wireless sensor networking for data and control mechanisms. This capability would initially be used when the returning animals are on the beach within range of a base station. As the MAMMARK tag is deployed to different types of animals this new capability becomes more critical. A radio was selected in conjunction with software changes that would provide a robust communications stack with minimized development effort (leveraging the TinyOS 2 framework). A chipset that has good power to distance metrics and is supported by existing software is the TI/Chipcon CC2420 [20].

B. Software Changes

In our experiences developing the original software, we found the previous structure had limitations and was imposing considerable constraints on system flexibility. Specifically, while maintaining ruthless power management in order to maximize tag life, we were lead to increasingly complex arbitration schemes for limited resources that ultimately proved to be unworkable. In an effort to find a better software structure that supports reliability, portability, and code reuse, we settled on TinyOS 2 [16].

TinyOS 2 is the second iteration of a mature, capable, and well supported community supported operating system that is especially targeted for low power wireless sensor networks.

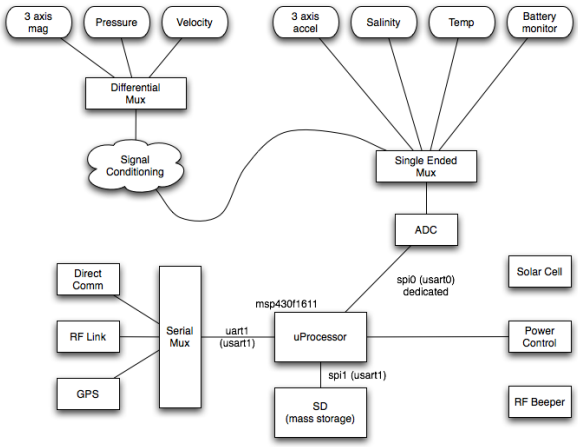

Fig. 4. Previous hardware block diagram.

As such it includes many features that can be leveraged for the MAMMARK Tag, including communication stacks, device drivers, and a well documented core API.

Leveraging the utility of TinyOS has led to a significant rewrite while porting the original software. The result is a streamlined architecture with increased functionality.

III. MAMMARK HARDWARE

Fig. 1 shows the physical prototype MAMMARK tag, and Figs. 3 and 4 show block diagrams of the current and previous hardware subsystems. As much of the original hardware has been retained on the new prototype, much of the description that follows is again included from [7], [8], and [9]. We include it here for completeness.

The core of the hardware is a TI MSP430 ultra-low power microcontroller [19]. The microcontroller includes several on-chip peripherals: SPI controllers, serial communication, clock control, watchdog mechanisms, DMA controllers, timers, 48KB of flash for code, and 10KB of RAM.

As all of the sensors are analog in nature, the main interface between the microcontroller and the sensors is the analog-to-digital converter (ADC) subsystem. While the MSP430 has an onboard ADC capable of converting analog signals with 12 bits of precision, it was felt that this was insufficiently precise to achieve the desired performance. Instead an external 16 bit ADC is attached via one of the SPI channels. In order to sample all of the various sensors, this single ADC is multiplexed and one sensor at a time is converted, the value stored in RAM and then the next sensor converted.

There are two different kinds of sensors attached to the central microcontroller: differential and single-ended. The differential sensors are first amplified through an Op-amp network and then converted. Each differential sensor can require a different gain to maximize the sensitivity dynamic range of the sensor.

The differential sensors include the magnetometers (three axes), depth (pressure) transducer, and water velocity (measured using a two-axis strain gauge). The single-ended sensors include the accelerometers (three axes), salinity, temperature (both of the water and of the microcontroller), and battery voltage. These sensors are used to reconstruct the three dimensional trajectory (both in the position and velocity domains), as well as salinity and temperature profiles.

In order to collect enough data for useful analysis, these sensors must be periodically sampled, filtered, corrected for calibration parameters, and stored for post-processing. Depending on the rate at which we are sampling each of the sensors, the amount of data collected can become very large (currently, the prototype limits the maximum sampling rate to 20Hz for all sensors). This is, however, an arbitrary limit imposed by the software. If, after experimentation, it is found that higher data rates are required, this can be changed without modifying the hardware.

The most important aspect of the MAMMARK tag is low power consumption. In order to be of use in the field, the MAMMARK must collect and store the sensor data until the animal in question returns to a place where the tag can be recovered (provisions are in place for remote data retrieval, however, the limited bandwidth of the RF link would make this a very slow process). The main function of the software is to manage power consumption enabling long interval data collection. The tag is expected to operate unattended for up to a year on a single lithium-ion battery pack.

With only 10K bytes of RAM, the MAMMARK tag cannot store more than a few minutes of data locally. In addition any volatile storage mechanism would run the risk of data loss. Instead, the main storage is provided by flash memory (current implementation uses Secure Digital) connected to the SPI bus. This sub-system provides secure long term storage of up to 4 gigabytes (possibly larger in the future), and provides a "future proof" method for storing larger data sets as the price of storage falls due to commodity pressures. As the sensor data is sampled, it is aggregated in RAM until 512 bytes (SD block size) are collected, at which point it is written to the flash subsystem. It is estimated that this storage will be sufficient for a significant period of time even at high sensing rates (even sampling continuously at 20Hz on all sensors the MAMMARK still has over 20 days of storage capacity).

As previously stated, the key performance criteria for the device is long life and given its battery powered nature, power conservation is essential. Most sub-systems are kept in a low or powered-off state whenever possible. The power draw of each subsystem is balanced against the required time for power-up and stabilization for high quality sensor readings. The hardware includes power circuits that allow each individual sensor to be powered or de-powered as overall system requirements necessitate. In terms of the power budget, the sensor components are some of the most expensive, and thus great care is taken when sequencing the power up in order to minimize overall power consumption.

Other subsystems include external communications, a Global Positioning System (GPS) module, RF Beeper, and the previously mentioned mass storage module. Each of these modules can be individually powered. The hardware also

provides for battery recharge via an external solar cell effective only at the surface.

Locating the tag after the animal returns to the beach and possibly molts is the function of the RF beeper. The RF beeper sends out a signal that allows the location to be determined using directional antennas and will possibly include a simple encoding of GPS latitude and longitude data. Note that this will only be powered when the tag detects that it is on dry land (this is relatively easy to determine using both the depth gauge and salinity sensors). Once the tag is located, communications with the tag can occur using radio communications on one of the ISM Bands. RF communication is accomplished using an off-the-shelf RF module from ChipCon. Each tag has its own node address and multiple tags can be in communication with a single base station. Provisions also exist in the software to build a mesh wireless sensor network which allows communications with tags that are beyond the direct range of the base station.

Given the large amount of data that will need to be transferred, and the power required to transfer that data over the RF link, remote data transmission over RF is a distant option to physically recovering the tag itself. The foreseeable scenarios for RF only communication are, for instance, when the tag is attached to a large male on the beach who is too aggressive to approach (tranquilizers for the large bulls are sometimes lethal, and all efforts will be made to leave the animals unharmed). Smaller animals can be restrained using physical means that allow the tag to be removed.

IV. MAMMARK SOFTWARE

The software running on the TI MSP430 is responsible for managing the communications, sensor, storage, and power subsystems. The longer a tag is able to run the more useful it will be. The software is responsible for sequencing and minimizing the power consumption of the device. Fig. 5 shows a simplified block diagram of the MAMMARK software system.

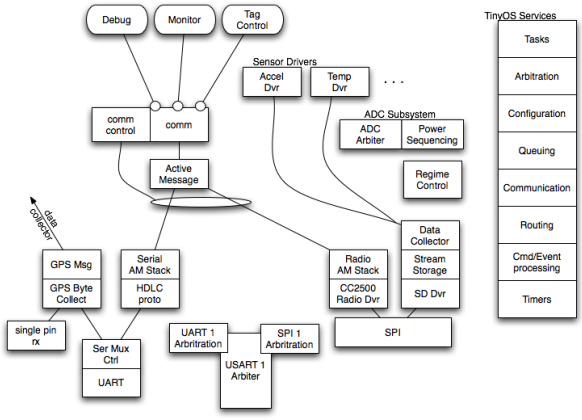

Fig. 5. TinyOS 2 based Software Block Diagram. The software to run the MAMMARK tag is divided into several subsystems implemented on top of kernel services provided by the TinyOS core.

TinyOS 2 provides a structure that makes implementing event driven software state machines relatively easy. Further, the software constructed is linked statically which makes analysis for race conditions possible. This helps minimize one of the nastier causes of embedded system failures.

A. Sensing System

The sensing system is composed of a driver for each sensor. This driver interacts with the ADC subsystem for access, power sequencing, and possible reconfiguration. For example, the accelerometer is a single chip (powered once) that provides 3 different output values (3 axes). In between each output value an appropriate settling time must be allowed before starting the sample. The driver is what instigates each change while the ADC subsystem is responsible for controlling power and settling.

When a data set is ready to be written for permanent storage the following subsystems are involved:

Driver	collects data set from sensor. Calls collector.
Collector	accepts data from sensors and other subsystems and combines the data into a form acceptable to the mass storage system (Stream Storage). 512 bytes.
StreamStorage	writes 512 byte disk blocks in sequence to a mass storage device. In this implementation, writes to a Secure Digital Card via the SD driver.
SD driver	provides the interface to the SD card via the SPI bus

Groups of sensors and their sampling periods are collected into *regimes*. One regime maybe active at a time and determines how often a sensor and which sensors are to be sampled. Changes to a regime and notification of effected sensor drivers is the responsibility of the *RegimeControl* subsystem.

B. GPS

The GPS subsystem is responsible for obtaining an accurate time/position fix from the GPS satellite constellation and writing this information into mass storage. This allows corrections to be made to position and timing information when post processing the collected sensor data.

The GPS interfaces to the system via two paths. The first path provides bidirectional communication to the GPS and is used when initilizing. It uses the same underlying hardware that other subsystems use and must be arbritrated prior to use. A receive only path is also provided by the *single rx pin*. This pin allows an interrupt driven software UART to be implemented that doesn't use any of the shared hardware.

The lowest level of the GPS subsystem is the *GPSByteCollector*. This subsystem assembles single bytes from the UART (either bytes from the h/w UART or bits from the single pin receive) and provides these bytes to the *GPSMsg* layer.

GPSMsg assembles the input byte stream into GPS messages and processes them. Position and time information is extracted from these messages and written to mass storage via the *DataCollector*.

Other parts of the system monitors the tag's surface state and generates events that informs *GPSMsg* to power on or off the GPS subsystem.

C. Communications

When a tagged animal is on the beach, we want to be able to communicate with the device to determine its status and location. Under certain conditions upload of data from the tag can also be advantageous. A group of reasonably proximate tags have the potential for forming an ad-hoc wireless sensor network. All of these reasons argue for a capable communications system.

TinyOS provides a communications stack called *Active Messaging*. Active Messaging essentially provides a reliable messaging layer that includes source, destination, and port information that allows implementation of many higher layer services.

The MamMark implementation builds the Active Messaging layer on top of both a direct serial connection and the CC2500 radio. Provisions are made so that upper layers do not need to know which type of connection is being used and tools on the base station will work with either communications link.

D. H/W interface and arbritration

Shared hardware forms the lowest level of the system. Because it is shared a mechanism must be provided to allow different subsystems access to the hardware in a coherent fashion. TinyOS 2 provides an architecture for arbritration between multiple users of a resource coupled with reconfiguration of the resource to the new owners specification.

When a subsystem wants access to a hardware resource, it must first make a request to the Arbiter controlling access to the resource. If the resource is free, the request is granted and the Arbiter will call the associated configuration routine provided by the requester. If the resource isn't available then the request will be enqueued awaiting a release of the resource. This provides a consistent mechanism for controlling access to shared resources throughout the system not just hardware.

Low level shared hardware resources include the following:

- USART 1 is shared between SPI 1 and UART 1.
- GPS, direct serial, radio, and SD all use USART 1.
- GPS communicates in a different manner than the AM communications stack and needs to arbritrate separately.
- the SD and the radio share the SPI bus.
- sensors must arbitrate for power and access to the ADC.

E. Upper Layer Functions

All upper layer functionality is at the top of the communications stack and provides services that either control the tag or ship data off the tag.

Below is a brief description of these services:

Debug	monitors and controls internal state of the tag, especially while testing and development.
Monitor	allows monitoring normal operation and observing data while it is being collected.
Tag Control	provides remote control of the tag and in particular what kind of data is being collected and simple uploading of collected data.

Other services will no doubt be added as the tag matures. In particular data offload will be provided which requires some form of wireless mesh coupled with routing mechanisms. These services will be implemented as higher layer services.

V. Sensor Models

In order to determine the expected accuracy of the MAMMARK sensor, as well as for simulation and testing work, it is important to characterize the sensor suite. A unified model is one which can be applied to variously to accelerometers, magnetometers, pressure sensors, and others. This is useful to the project due to the ability to reconnect measurement noise back through the Kalman filtering schemes to actual attitude and position through the dead reckoning filter first proposed in [8].

A general model for the sensor output follows closely the one presented in [21]:

$$\xi_m = (1 + S_f)\xi_t + b(t) + \nu_w \tag{1}$$

where ξ_m is the measured quantity at the sensor output and ξ_t is the true quantity. S_f represents a scale factor error, $b(t)$ represents the time varying bias or drift terms, and ν_w is the noise on the sensor. $b(t) + \nu_w$ is the residual measurement of the sensor with no input, and can thus be measured when the sensor is static. The sensor noise, ν_w can be assumed to be zero mean band limited white noise. This can be characterized by taking the standard deviation of the sensor's output over a short period of time with no input applied. In this work, we will ignore the scale factor error ($S_f = 0$) as this is usually calibrated out at the factory or during initial installation, and has been shown to be quite stable over long periods of time.

The total sensor bias $b(t)$ is comprised of several components, and consists of an additive error:

$$b(t) = b_0 + b_1(t) \tag{2}$$

The constant null shift, b_0, is easy to determine by computing the mean of the sensor over a long period of time when no input is applied. Note that an easy estimate of the null shift can be accomplished by measuring the output when static (as in the case of computing the bias for rate gyros while sitting still).

The time varying bias drift, $b_1(t)$, is characterized as a stochastic time sequence. Modeling the drift as band limited

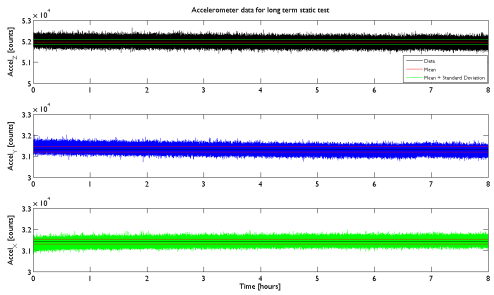

Fig. 6. Static MAMMARK Accelerometer Data

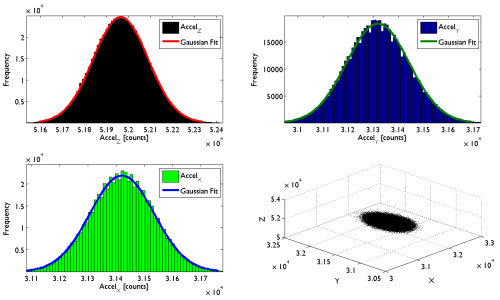

Fig. 7. Static MAMMARK Accelerometer Histograms

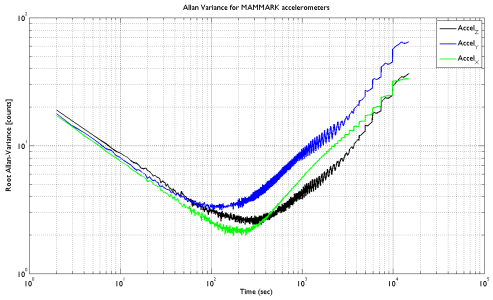

Fig. 8. Allan Variance of Static Accelerometer Data

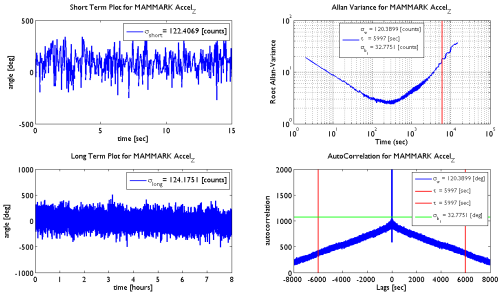

Fig. 9. Unified Model for Accelerometer Z channel

white noise would be too conservative in the short term, and too optimistic in the longer term. In order to adequately model this time sequence, yet retain a tractable model, we model the bias drift as an exponentially correlated or first order Gauss-Markov process:

$$\dot{b}_1(t) = -\frac{1}{\tau} b_1(t) + \omega_b \tag{3}$$

where τ is strictly positive and is the correlation time constant. ω_b is a Gaussian white process noise with a power spectral density given by:

$$\boldsymbol{\mathcal{E}}\left\{\omega_b\left(t_1\right)\omega_b\left(t_2\right)\right\} = \frac{2\sigma_b}{\tau} \tag{4}$$

The parameter τ defines the degree of correlation. If τ is small, then the signal is highly correlated in time, and in the limit as τ approaches infinity, the signal becomes a random constant (e.g.: Gaussian white noise). The slow time varying bias drift can be completely modeled with the parameters τ and σ_b.

Two different, but complementary, techniques will be used to extract the values of τ and σ_b: the Allan Variance and the autocorrelation function. The Allan Variance is a standard approach to characterize noise models originally developed to analyze the stability of atomic clocks [1]. Whereas the power spectral density of a signal relates the power as a function of freqency, the Allan Variance does so as a function of averaging time [13]. Using the Allan Variance, the signature of exponentially correlated noise can be revealed.

The Allan Variance plot will demonstrate a slope of $-1/2$ during the time where wide-band noise is the dominant process. Where the slope is $+1/2$, the process is dominated by correlated noise. Thus a minimum for the time constant can be extracted from the point on the Allan Variance plot where the slope changes from $-1/2$ to $+1/2$ (often referred to as the "flicker floor"). Thus a signal with *only* wide-band noise would appear to drop at a slope of $-1/2$ throughout the averaging time.

Using the autocorrelation plot, τ corresponds to the lag where the value is at $1/e$ from its zero-lag peak (approximately 37% of the peak value). Note that in order to extract the slowly varying time process, the sensor readings are decimated through averaging for a variety of window widths, with the correlation time being consistent through the averaging.

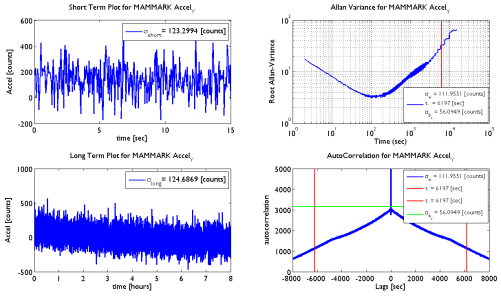

Fig. 10. Unified Model for Accelerometer Y channel

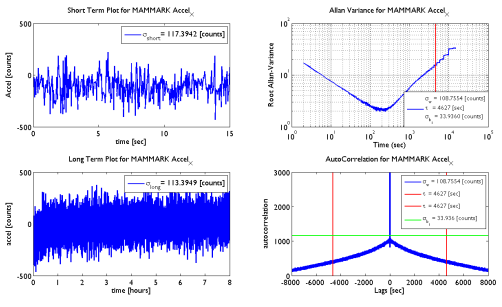

Fig. 11. Unified Model for Accelerometer X channel

Likewise, it is possible to extract the Markov process noise, σ_b by taking the square root of the peak of the autocorrelation function. Again, this should be consistent across averaging time.

Using these tools, it is possible to model the bias drift of a sensor in a unified and simple way. Note that many sensors display multiple correlation processes that have different time scales, but that these are beyond the scope of this unified model.

VI. Sensor Calibration

Measurement errors in the three axis magnetometers and accelerometers directly affect the attitude estimate.

Given that the accelerometer and magnetometer measurements make up an integral part of the trajectory reconstruction algorithms, care must be taken in calibration of these sensors to minimize the errors that are accumulated during the Dead Reckoning integration process.

In this work, we use a technique for calibrating the accelerometers and magnetometers directly from a locus of measurements using a two-step algorithm described in [11], [12], [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 given in Eq. 1:

$$a_{meas} = (1 + S_f)a_{true} + b(t) + \nu_w \tag{5}$$

where a_{meas} is the measured acceleration component, a_{true} is the true acceleration component on that axis, b is the time varying bias or offset, and ν_w 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 gravitational acceleration. 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.

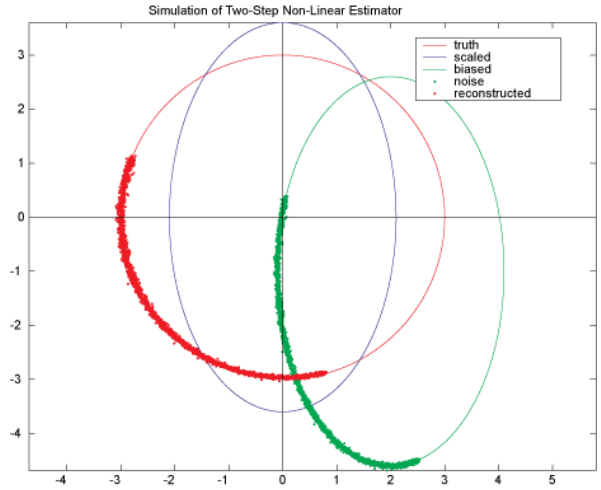

Fig. 12. Two-step Estimation Algorithm on Simulated Data

The same analysis is true for the 3D case, except that an ellipsoid is generated, rather than an ellipse. It the case of the 3D sensor, care must be take to traverse enough of the surface of the ellipsoid to have good observability of the parameters. Again, note that the only thing required for the algorithm to work is a knowledge of the true magnitude of the gravitational field, and motion of the sensor.

Note that for this analysis, we assume that the time varying portion of the bias has a long enough correlation time that the experiment can be run before the bias has shifted significantly.

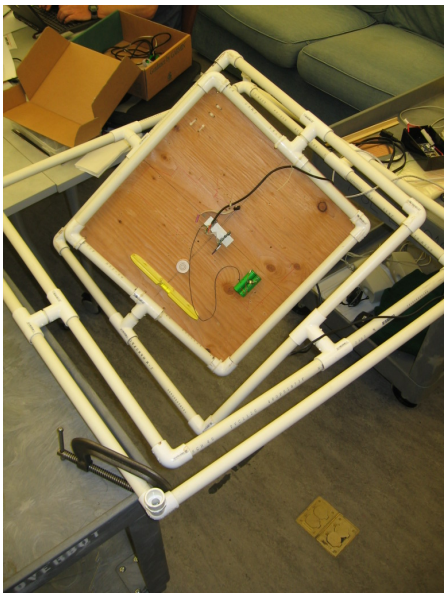

Fig. 13. Gimble for rotating MAMMARK sensor

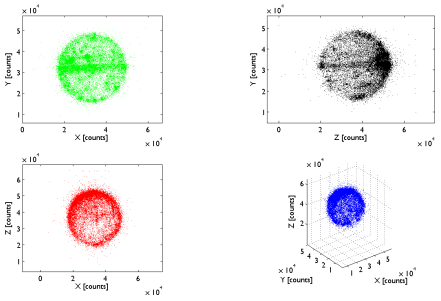

Fig. 14. Data from tumbled MAMMARK sensor

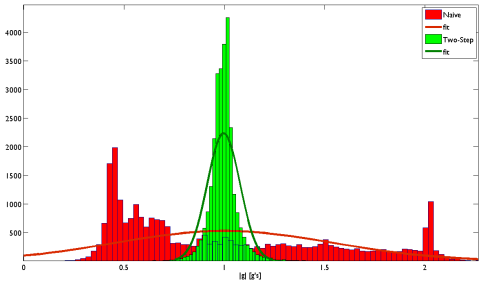

Fig. 16. Naive vs. Two-step calibration histogram

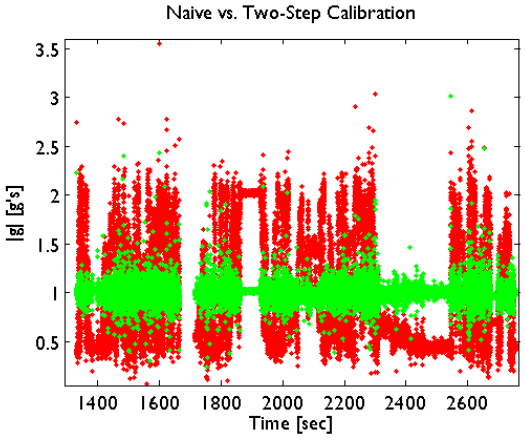

Fig. 15. Naive vs. Two-step calibration performance

More sophisticated attitude filtering can be used to estimate the bias drift from other sensor fusion type algorithms.

Fig. 12 shows the 2D version of the algorithm working on simulated data. Here, no attempt is made to model non-orthogonal axes, and a significant portion of the circle is traversed in order to ensure good parameter estimation. The red solid line is the true data, the blue solid line is the scaled (scale factor) data, and the green solid line includes the bias offsets. The green dots are the sampled points that are then used in the algorithm to reconstruct the red dots (true sampled points). We will use accelerometer data from the MAMMARK sensor to calculate the (constant) bias and scale factor errors of the sensor.

Using the gimbal pictured in Fig. 13, the MAMMARK tag was rotated and tumbled to generate the data required for calibration. As the entire process took only a few minutes, the bias drift did not have time to effect the measurements. The data is plotted to show it is roughly spherical (see Fig. 14).

Using a naive calibration (center the data on the median of the data in all three axis and scale to match the magnitude to 1 g), we get much worse performance than using the two-step algorithm. Fig. 15 shows the calibrated magnitude data for both the naive and the two-step algorithm. Histograms of the distribution of errors is shown in Fig. 16. The standard deviation of the error for the naive calibration is $0.54g$'s versus the two-step calibration of $0.08g$'s.

The two-step algorithm essentially fits a sphere of the correct radius to that data. Running the data through the extended two-step algorithm (detailed in [10]), we extract the following parameters:

$$\begin{aligned} x_0, y_0, z_0 &= 33437.94, 32094.48, 36581.56 \\ a, b, c &= 16397.69, 16547.91, 15319.98 \\ \phi, \rho, \lambda &= 1.66, -2.59, 0.37 \end{aligned}$$

Thus, ignoring the cross-coupling axes terms (which are very small), the conversion from z-axis counts to g's would be:

$$a_z = (k_z - 36581.56)/15319.98 \tag{6}$$

where k_z is the counts of the z-channel ADC, and a_z is the output of the sensor in g's. Each other channel has a similar equation. Thus we can convert the measurements from counts to g's, and further to attitude using the measurements from the magnetometers as well.

VII. Static Data Analysis

In order to determine the wide band noise of the sensors, as well as the bias drift, static data was taken with the MAMMARK sensor on the bench and recorded through the serial port. Fig. 6 shows the static data over 8 hours of 20Hz sampling. While there are some outliers, the data shows very good agreement with a simple Gaussian estimate, as shown by the histograms in Fig. 7.

In order to determine the unified model parameters, we first examine the Allan Variance of the accelerometer channels (Fig. 8). Note that the Allan Variance for each axis measurement initially decreases at a slope of $-1/2$, indicating band limited white noise. Each reaches its "flicker floor" around 100–300 seconds, and shows an absolute minimum of noise of approximately 2–4 ADC counts. The sensor looks very consistent from axis to axis, and the unified models generated will be quite similar. The characteristic $+1/2$ slope after the flicker floor is indicative of the correlated noise, which corroborates our decision to use a first order Gauss-Markov process.

Sensor	σ_ν [counts]	[milli–g's]	τ [sec]	σ_b [counts]	[milli–g's]
Accel_X	108.76	6.63	4627	33.94	2.07
Accel_Y	111.96	6.77	6197	56.09	3.39
Accel_Z	120.39	7.85	5997	32.78	2.14

TABLE I

UNIFIED MODEL FOR MAMMARK ACCELEROMETER CHANNELS

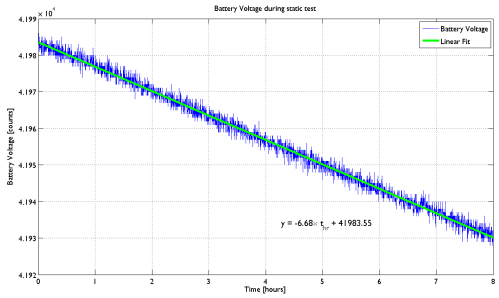

Fig. 17. Battery Draw during static test

Fig. 9 shows the short and long term variability within the static data, the Allan Variance, and the autocorrelation. Using the data, the z-channel of the accelerometer has a model parameterized by a band-limited white noise with a standard deviation of 120.4 counts on the sensor, a correlation time, τ, of 5997 seconds, and a Markov process noise with a standard deviation of 32.8 counts. Similar plots are presented for the y- and x-channels in Fig. 10 and Fig. 11, respectively. A table of the unified models for the accelerometer is presented in Table I.

While some of this variation is due to a temperature coupling (clearly visible in the X and Y channels of Fig. 6), this is less of an issue for the MAMMARK sensor than would be for an INS-type application. This is due to the operational environment that the MAMMARK will be situated in: deep ocean water. The temperature profile of the ocean column is quite constant from a depth of approximately 50 meters to 1500 meters. As such, the sensor will operate in nearly constant temperature conditions. On the other hand, the pressure will vary considerably, and our sensors have not been tested under varying pressure conditions.

Using these models, we will be able to simulate the attitude estimation and tune the filters based on our measured static data. This will allow us to extract the most information out of each sensor, and further refine the trajectory estimation.

VIII. POWER CONSUMPTION

The MAMMARK tag is powered from Lithium Ion batteries, and is anticipated to have enough battery life to last over a year of active data gathering. In order to intelligently manage the battery life, battery voltage is passed through a voltage divider, and then sampled using the ADC system. A plot of the power consumption during the static data gathering is presented in Fig. 17.

Li-Ion batteries have some unique discharge characteristics based on the internal construction of the devices. Their initial voltage is much higher than typical aqueous batteries. However, if they are discharged below a certain critical amount, they become very difficult to recharge, and eventually can be destroyed through that process. Typically, draining the cells below 3V will damage the batteries.

The initial charge on the MAMMARK battery charge is 4.38V and corresponds to 47950 counts on the ADC. The static test was performed with all sensors being sampled at a fixed 20Hz rate (which is much higher than would be used during deployment). Measurements during these tests indicate a current draw of 35mA (peak) to a more nominal usage of 0.7mA. Based on a simple linear fit to the battery data, the rate of discharge is shown to be -6.7 counts per hour under usage, which corresponds to a voltage decay of 6 mV/hour.

Note that the static tests had neither the GPS, nor the SD card operating. Both of these items consume high amounts of power on the tag. Also, there is a 1-3% self-discharge on the Li-Ion batteries that will slowly drain away regardless of use. Also, there is an assumption here that the Li-Ion battery discharges in a linear fashion, and that the power available in the battery is proportional to the measured voltage.

Based on a linear extrapolation, our early estimates are that the tag will be able to operate for 1930 hours or approximately 80 days at 20Hz sampling rate. If the power consumption scales linearly with with sample rate, than we expect double that at 10Hz, or 160 days. Thus, using our normal sampling regime, where some sensors are sampled more slowly, we expect durations of more than a year on the Li-Ion batteries.

IX. CONCLUSION

In this work, we continue to document the development of the MAMMARK marine mammal marking tag. The tag uses a very low power microcontroller combined with off-the-shelf MEMs technology sensors to reconstruct the trajectory of a diving pinniped. Attitude estimation is based on a solution to Wahba's problem using magnetometer and accelerometer measurements. Due to power management and resource constraints, both the hardware and software have been largely redesigned. The main changes in the hardware have been to move the RF block from the serial to the SPI port, and the move the GPS RX pin directly into an interruptible I/O pin on the microcontroller.

The software migration has gone to TinyOS 2, a modern community supported OS designed for wireless remote sensor applications. By using this OS, power management and resource contentions leverage the existing expertise and code base within the TinyOS community. The sensors are being sampled, and a calibration was performed using tumble and static data for the accelerometer channel.

The accelerometers channel shows a static wide-band noise of approximately 7 milli–g's, bias correlation time constants of around 6000 seconds, and bias process noise of approximately 2-3 milli–g's. Power consumption was measured via battery voltage while processing the static data, and early data

indicated a low battery drain that will translate to operations on the order of a year or more if the sample rate can be slowed down to approximately 1 Hz.

Future development will focus on getting actual data from the environment, and to make sure the tag is robust and effective.

X. Future Work

Current work has focused on updating the hardware for better performance (radio and gps hardware). Software has focused on porting the existing hardware to the TinyOS 2.x platform to provide a more robust structure as well as being able to take advantage of a wealth of existing code. The prototype hardware and software has recently come together yielding early bench results.

Future work includes:

1) Bench validation of sensors.
2) Evaluate GPS reacquire time. (critical for surfacing behavior)
3) Controlled test in closed water (submarine devices and/or human diver in closed pools).
4) Compare actual results against simulated dives to measure filter performance.
5) measure/calculate power and memory performance (lab and realistic conditions) to provide estimates for actual in-situ performance.
6) Develop base station and control mechanisms for early deployment.
7) Develop sensor network routing and data movement mechanisms.

References

[1] D. W. Allan, N. Ashby, and C. Hodge. *The Science of Timekeeping*. Hewlett Packard Application Note 1289, Sunnyvale, CA, 1997.

[2] R. D. Andrews, D. R. Jones, J. D. Williams, G. W. Oliver, P. H. Thorson, D. P. Costa, and B. J. Le Boeuf. Heart rates of northern elephant seals diving at sea and resting on the beach. *Journal of Experimental Biology*, 200:2083–2095, 1997.

[3] B. J. Le Boeuf, D. P. Costa, A. C. Huntley, and S. D. Feldkamp. Diving behavior of female northern elephant seals, mirounga angustirostris. *Canadian Journal of Zoology*, 66:446–458, 1988.

[4] B. J. Le Boeuf, D. E. Crocker, S. P. Blackwell, P. A. Morris, and P. H. Thorson. *Marine mammals: advances in behavioral and population biology*. Oxford University Press, London, UK, 1993.

[5] B. J. Le Boeuf, D. E. Crocker, D. P. Costa, S. P. Blackwell, P. M. Webb, and D. S. Houser. Foraging ecology of northern elephant seals. *Ecological Monographs*, 70(3):353–382, 2000.

[6] R. L. DeLong, B. S. Stewart, and R. D. Hill. Documenting migrations of northern elephant seals using daylength. *Marine Mammal Science*, 8:155–159, 1992.

[7] G. H. Elkaim, E. B. Decker, G. Oliver, and B. Wright. Development of a marine mammal marker (mammark) for in-situ environmental monitoring. In *Proceedings of the ION National Technical Meeting, ION-NTM 2006*, Monterey, CA, Jan 2006. ION.

[8] G. H. Elkaim, E. B. Decker, G. Oliver, and B. Wright. Marine mammal marker (mammark) dead reckoning sensor for in-situ environmental monitoring. In *Proceedings of IEEE/ION PLANS 2006*, pages 976 – 987, San Diego, CA, April 2006. IEEE/ION.

[9] G. H. Elkaim, E. B. Decker, G. W. Oliver, and B. Wright. Marine mammal marker (mammark) for in-situ environmental monitoring. *GPS World*, 17(8):30–33, 2006.

[10] G. H. Elkaim and C. C. Foster. Extension of a non-linear, two-step calibration methodology to include non-orthogonal sensor axes. *IEEE Transactions on Aerospace Electronic Systems*, 43(4), 2007.

[11] 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.

[12] 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.

[13] 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.

[14] R. D. Hill. *Elephant seals: population ecology, behavior and physiology*. University of California Press, Berkeley, CA, 1994.

[15] SIRF Inc. *Sirf Star III, GPS chipset data specification*, 2007.

[16] P. Levis, D. Gay, V. Handziski, J.-H. Hauer, B. Greenstein, M. Turon, J. Hui, K. Klues, C.Sharp, R. Szewczyk, J. Polastre, P. Buonadonna, L. Nachman, G. Tolle, D. Culler, and A. Wolisz. T2: A second generation os for embedded sensor networks. Technical Report TKN-05-007, Telecommunication Networks Group, Technische Universitat Berlin, Nov 2005.

[17] G. W. Oliver. The MAP Tag: visualizing the tracking and diving behavior of marine mammals. In *Proceedings of the 11th Biennial Conference on Biology of Marine Mammals*, Orlando, FL, 1995.

[18] G. W. Oliver, P. A. Morris, P. H. Thorson, and B. J. Le Boeuf. Homing behavior of juvenile northern elephant seals. *Marine Mammal Science*, 14(2):245–256, 1998.

[19] Texas Instruments (TI). *MSP430x161x Mixed Signal Microcontroller Data Sheet*, 2005.

[20] Texas Instruments (TI). *ChipCon 2500: Low-Cost Low-Power 2.4 GHz RF Transceiver*, 2007.

[21] L. Wenger and D. Gebre-Egziabher. System Concepts and Performance Analysis of Multi-Sensor Navigation Systems for UAV Applications. In *Proceedings of the 2nd AIAA Unmanned Unlimited Systems Conference, San Diego, CA*. AIAA, 2003.

[22] T. M. Williams, R. W. Davis, L. A. Fuiman, J. Francis, , B. J. Le Boeuf, M. Horning, J. Calambokidis, and D. A. Croll. Sink or swim:strategies for cost efficient diving in marine mammals. *Science*, 288:133–136, 2000.