DAS Departamento de Automação e Sistemas
CTC Centro Tecnológico
UFSC Universidade Federal de Santa Catarina

Development of a Prototype Attitude Determination System (PADS) for High Altitude Research Balloons

Relatório submetido à Universidade Federal de Santa Catarina

como requisito para a aprovação na disciplina

DAS 5511: Projeto de Fim de Curso

Fernando Martignone Esteves

Florianópolis, julho de 2014

Development of a Prototype Attitude Determination System for High Altitude Research Balloons

Fernando Martignone Esteves

Esta monografia foi julgada no contexto da disciplina
DAS5511: Projeto de Fim de Curso
e aprovada na sua forma final pelo
Curso de Engenharia de Controle e Automação

Prof. ***Marcelo Ricardo Stemmer***

____
Assinatura do Orientador

Acknowledgements

Firstly, I would like to thank my advisor at The Catholic University of America, Jandro L. Abot, for all the support during all the phases of this work. I would like also to thank my advisor at UFSC, Marcelo Ricardo Stemmer, for all help provided to me during all the undergraduate process.

In addition, I would like to thank my parents for all the motivation they have provided me during all of these years. Finally, my sincere gratitude to Raissa Raimundo da Silva for all the support and for being such an excellent friend and human being.

Resumo Estendido

Sistemas de determinação de atitude são um requisito para a maioria dos problemas de navegação e controle. Tradicionalmente, sistemas inerciais de orientação (AHRS) têm sido usados para prover dados de orientação com alta confiabilidade e precisão. No entanto, este tipo de solução geralmente usa sensores muito caros. Tendo em vista disso, um sistema de determinação de orientação (atitude) de baixo custo foi proposto no âmbito deste projeto usando componentes que podem ser encontrados nos grandes distribuidores de produtos electrónicos, os quais incluem um conjunto de giroscópios MEMS, clinômetros, um magnetômetro e uma câmera estrelar. Os sensores serão utilizados de forma hierárquica, o que significa que os sensores mais rápidos e menos precisos são atualizados pelos sensores mais lentos, porém mais precisos.

Este projeto é parte de um programa da NASA lançado recentemente intitulado “Undergraduate Student Instrument Project Educational Flight Opportunity” (USIP), o qual promove uma cooperação entre universidades e pesquisadores da NASA no desenvolvimento de experimentos científicos de forma a estabelecer uma mão de obra que possua conhecimentos científicos e tecnológicos no estado da arte e capacidade de gerenciar grandes projetos. A maior parte deste projeto foi desenvolvido na universidade The Catholic University of America com ajuda de parceiros em NASA-Goddard Space Flight Center.

Em um primeiro passo, as saídas de um conjunto de giroscópios foram fusionadas através de um filtro de Kalman, de forma a criar um giroscópio virtual com maior precisão do que um único giroscópio. Esta tecnologia tem provado ser muito eficaz e possui um grande potencial, tendo em vista que sensores MEMS estão experimentando uma rápida melhoria em termos de precisão, robustez, tamanho e resposta dinâmica. Em um segundo passo, um algoritmo baseado em um filtro de Kalman estendido, foi desenvolvido a fim de fusionar a orientação fornecida pelo giroscópio virtual, pelo magnetômetro e pelos clinômetros. A câmera estrelar é sensor o mais lento e fornece dados de orientação através da comparação da posição das estrelas. O sistema proposto destina-se a aplicações científicas para balões de alta altitude. Estes sistemas, tipicamente, precisam de sistemas de

apontamento na ordem de arco segundos. A carga científica foi projetada para manter todos os sensores e coletar dados em condições normais de voo. Além disso, um programa em MATLAB foi desenvolvido para processar todos os dados dos sensores e implementar os algoritmos de fusão de dados baseados no filtro de Kalman. A carga científica inclui todos os instrumentos responsáveis pela detecção de orientação, aquisição de dados e processamento, controle de temperatura e regulação de tensão para os componentes.

Abstract

Attitude determination systems are a requirement for most navigation and control problems. Traditionally, attitude and heading reference systems (AHRS) have been used to provide attitude with high reliability and accuracy. However, such solutions usually use very expensive sensors. Considering that, a low-cost attitude determination system has been proposed in the scope of this project using commercial-off-the-shelf components, which include a set of MEMS gyroscopes, clinometers, a magnetometer and a star-tracking camera. The sensors will be used in a hierarchical manner, which means that the faster and less accurate sensors are updated by the slower but more precise sensors.

This project is part of a NASA program recently released untitled as "Undergraduate Student Instrument Flight Project Educational Opportunity" (USIP), which promotes the cooperation between universities and NASA leading scientists in the development of scientific experiments in order to establish high qualified workers that have state-of-art scientific and technological knowledge and ability to manage large projects. Most of this project was developed at The Catholic University of America with the help of partners in NASA-Goddard Space Center Flight.

In a first step, the outputs from a set of gyroscopes were fused through a Kalman Filter to create a virtual gyroscope with higher accuracy than a single gyroscope. This technique has been proved to be very successful and has a great potential, considering that MEMS sensors are experiencing rapid improvements in terms of precision, robustness, size and dynamic response. In a second step, an Extended Kalman Filter algorithm was developed in order to fuse the attitude provided by the virtual gyroscope, clinometers and magnetometer. The Star Tracking Camera is the slowest sensor and provides absolute attitude data by comparing the position of the stars. The proposed system is intended to Scientific High Altitude Balloon applications, which typically require an accurate pointing system in the order of arc seconds. A payload was designed to hold all sensors and gather attitude data during flight conditions. In addition, a program in MATLAB was developed to process all the data from sensors and implement the Kalman Filter fusion algorithms. The

payload carries all the instruments responsible for the attitude sensing, data acquisition & processing, temperature control and voltage regulation.

Table of Contents

Chapter 1: **Introduction**

Attitude determination is one of the most important subsystems in spacecraft, satellite or scientific balloons missions, since it can be combined with actuators to provide rate stabilization and pointing precision for payloads. It generally uses a combination of sensors and mathematical models to provide orientation values in relation to an inertial frame. Traditionally, attitude and heading reference systems (AHRS) have been used to provide attitude with high reliability and accuracy. However, such solutions usually use very expensive sensors [1]. Therefore, in the scope of this project a novel low-cost prototype attitude determination system that uses an array of pointing sensors in a hierarchical manner, i.e., the faster and less precise sensors are updated by the slower, but more precise ones. Sensors will include a set of uncorrelated MEMS gyroscopes, two clinometers, a magnetometer, and a star tracking camera. The MEMS gyroscopes will be combined into a virtual gyroscope as demonstrated by Bayard [2] to achieve a measurement with higher accuracy that from a single gyroscope. Typical accuracies required for most of the scientific payloads in high altitude balloon missions are in the order of few arc seconds to few arc minutes, thus the system proposed should achieve accuracy on that range.

1.1: Document Structure

The document is organized in chapters in the following form: In chapter 2, a attitude dynamics discussion is presented, which is important to understand the following models and derivations. In chapter 3, a discussion of balloon systems structure and its importance in current science missions is presented. In chapter 4, a system overview is provided including its subsystems and their characteristics. In chapter 5, Kalman Filter fundamentals are presented, which give the basis to understand the following implementations. In chapter 6, the developed program denominated as “PADS Analyzer” for the sensors data analysis and its main functions are described. In chapter 7, sensors noise analyses are performed by taking advantage of the PADS Analyzer functions. In chapter 8, a virtual gyroscope concept is presented following by its implementation and results. In chapter 9, sensors attitude data fusion by using an extended Kalman Filter algorithm is discussed and the results are presented. In chapter 10, a power budget for the system is presented.

Chapter 2: **Attitude Dynamics**

The attitude of a rigid body can be defined as the imaginary rotation necessary to move the current axes frame to a reference frame. Euler angles can be used to determine the orientation of a rigid body in a 3-dimensional Euclidean Space. They represent a sequence of three elemental rotations starting from an initial reference frame and can be used to represent the spatial orientation in relation to any reference frame. There are several conventions for Euler Angles, depending on the selected rotation axes. The rotations can either occur about a fixed reference frame (extrinsic rotations) or about the rotating coordinate system axes (intrinsic rotations). For the following analysis the ZYX convention it is going to be adopted which uses intrinsic rotations. The conventions assume that a set of rotations about a fixed reference frame, where α, β and γ are the rotations about the z, y and x axes, respectively, of the initial reference frame as can be seen in Figure 1.

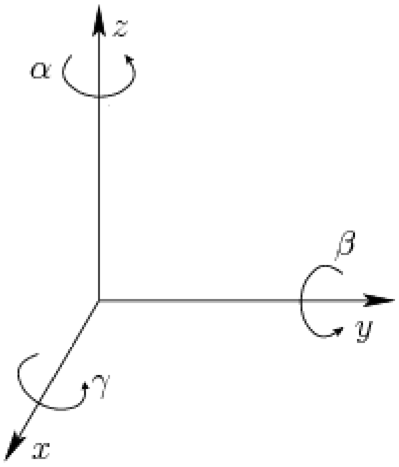


Figure 1 - ZYX Convention

The rotation matrixes for rotations around the Z, Y and X axes are respectively shown in the following equations:

$$R_z(\alpha) = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

$$R_y(\beta) = \begin{pmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{pmatrix}.$$

$$R_x(\gamma) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{pmatrix}.$$

According to the ZYX, a rotation R can be expressed as a composition of these matrixes:

$$R(\alpha,\beta,\gamma) = R_z(\alpha)\,R_y(\beta)\,R_x(\gamma)$$

The composition of matrixes leads to the following rotational matrix:

$$R(\alpha,\beta,\gamma) = \begin{pmatrix} \cos\alpha\cos\beta & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma \\ \sin\alpha\cos\beta & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma \\ -\sin\beta & \cos\beta\sin\gamma & \cos\beta\cos\gamma \end{pmatrix}$$

2.1.1: Attitude Measurements

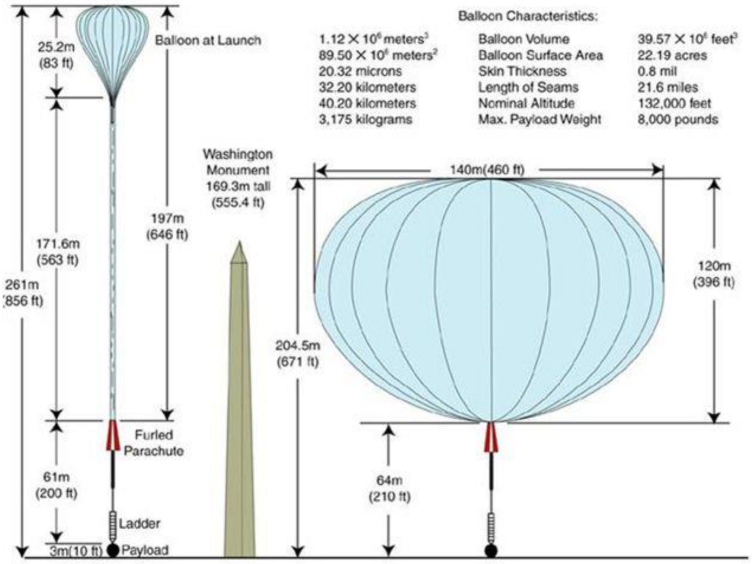
There are basically two types of attitude measurements: absolute attitude measurements and relative attitude measurements. The first class of measurements is based on the fact that by knowing the position of the system in the space, it is possible to compute the attitude in respect to an inertial frame [6]. Typically, absolute measurement sensors measure these directions in relation to a body-fixed reference frame and compare it to an inertial reference frame to determine the relative orientation in respect to the inertial frame. Relative measurements belong to the class of gyroscopic instruments.

Chapter 3: **Scientific Balloons Overview**

Scientific Balloons are very large structures, usually made by a 0.02mm thickness polyethylene film material. They can be used as a carrier of scientific instruments in a similar way as orbit satellites or Space Shuttle do. The major benefits are the low cost compared to other carriers, high capacity and small lead time. Balloons can fly in six months once they receive support for the mission [7].

3.1: Balloon System

The balloon system is basically composed by three elements: a balloon that allows the system to reach very high altitudes, a parachute for landing purposes and a gondola that carries all the science experiments. The suspended load can weight a maximum of 3600 kg and can be lifted to altitudes up to 36 km [7]. When the balloon is fully inflated it can reach dimensions up to 140m in diameter and 121m in height (enough to cover a medium sized soccer stadium).



Balloon Characteristics:

1.12×10^6 meters^3	Balloon Volume	39.57×10^6 feet^3
89.50×10^6 meters^2	Balloon Surface Area	22.19 acres
20.32 microns	Skin Thickness	0.8 mil
32.20 kilometers	Length of Seams	21.6 miles
40.20 kilometers	Nominal Altitude	132,000 feet
3,175 kilograms	Max. Payload Weight	8,000 pounds

Figure 2 - Scientific Balloon Dimensions [7]

Some of its major advantages as Science Carrier are:

- Short lead time;
- Lower cost in comparison to orbiting missions;
- Multiple launch locations around the world;

- Recoverable payloads;
- High mass payloads can fit into the carrier.

3.2: Balloon Launch and Operation

In order to launch, the balloon is inflated with helium and coupled to the gondola, where the payloads are attached. The helium expands as the balloon reaches high altitudes. All instruments are turned on through an integrated command system and start gathering data. Conventional Balloon flights last up to two days, however, they can be designed to last up to more than 100 days. At the end of the mission, the parachute opens and softly brings the payloads into the surface so science teams can recover them. NASA nominally supports 15 to 20 balloon flights annually [7]. The graph below shows the altitude versus time profile of a typical conventional balloon flight:

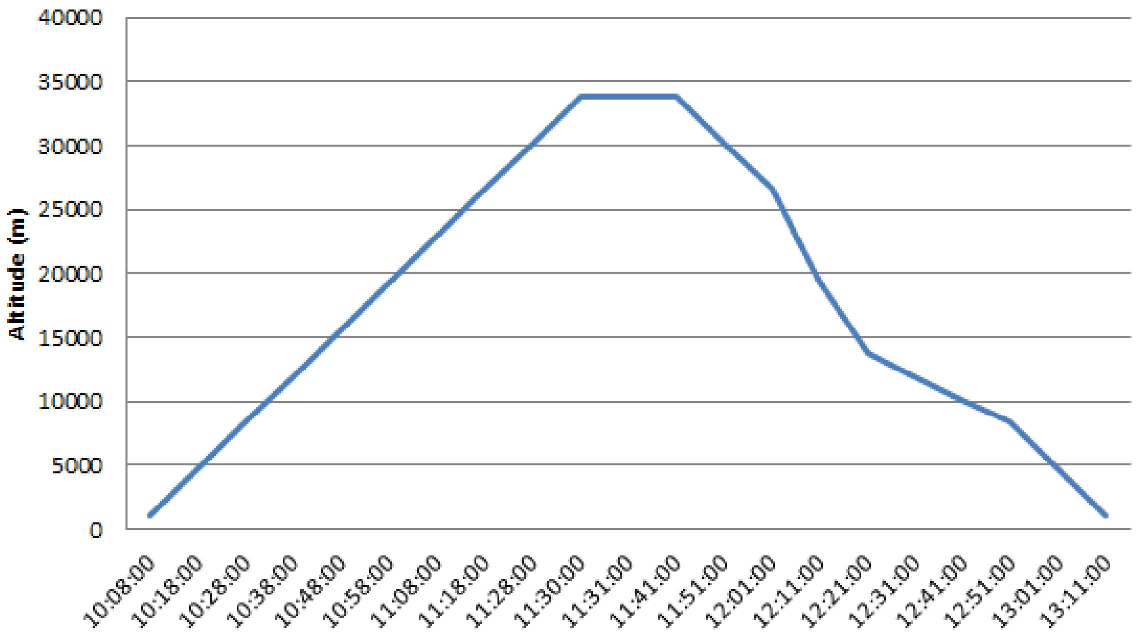


Figure 3 - Altitude versus Time Balloon Flight Profile

3.3: Flight Temperature Profile

A typical temperature profile for a high altitude balloon flight is shown in the Figure 4.

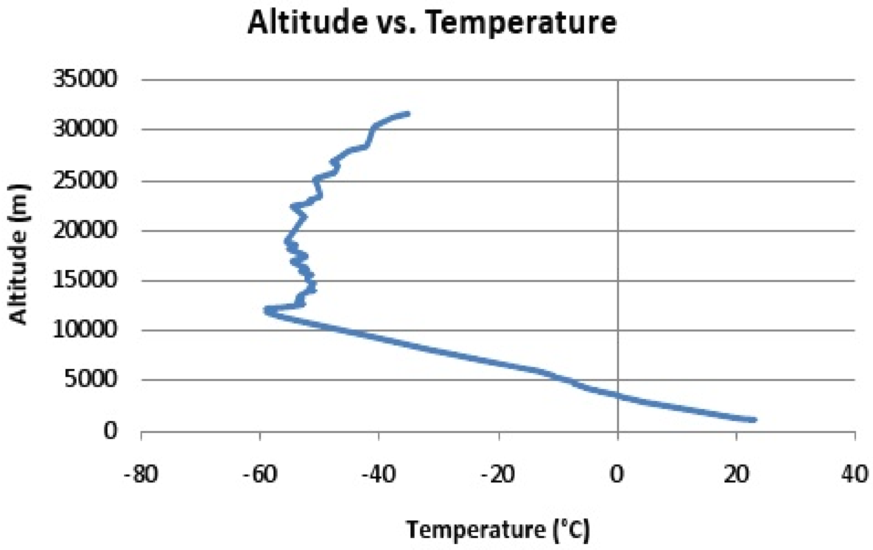


Figure 4 - Altitude versus Temperature Balloon Flight Profile

Temperatures can reach a minimum of -60 °C, which, in turn, requires the payloads to have a heating system to control the temperature and thus allow the sensors to work in normal operational conditions.

Chapter 4: **System Overview and Subsystems**

The proposed system consists of a prototype attitude measurement system for scientific high altitude balloons applications. The overall goal of the system is to collect attitude data and provide attitude estimation in the order of arc-seconds.

4.1: System Requirements

The following requirements were established taking into consideration flight conditions, costs, mass constraints and accuracy:

- Low cost system (less than 50,000 USD) using commercial off-the-shelf components;
- Orientation accuracy in the order of a few arc seconds to a few arc minutes, over the three axes X, Y and Z;
- Low mass system (less than 20 kg);
- Resist to flight temperatures of -60°C.

4.2: Subsystems Definition

The system was divided in several subsystems for design and work division purposes. The following subsystems were identified as:

1) Relative Attitude Sensing (RAS)
2) Absolute Attitude Sensing (AAS)
3) Analog Data Acquisition (ADAQ)
4) Signal Processing (SP)
5) Climate Control and Monitor (CCM)
6) CIP Interface Board (CIB)
7) Voltage Regulators Boards (VRB)
8) Pressure Vessel (PV)

9) Unpressurized Boxes (UB)

All the subsystems and their interconnections are shown in Figure 5.

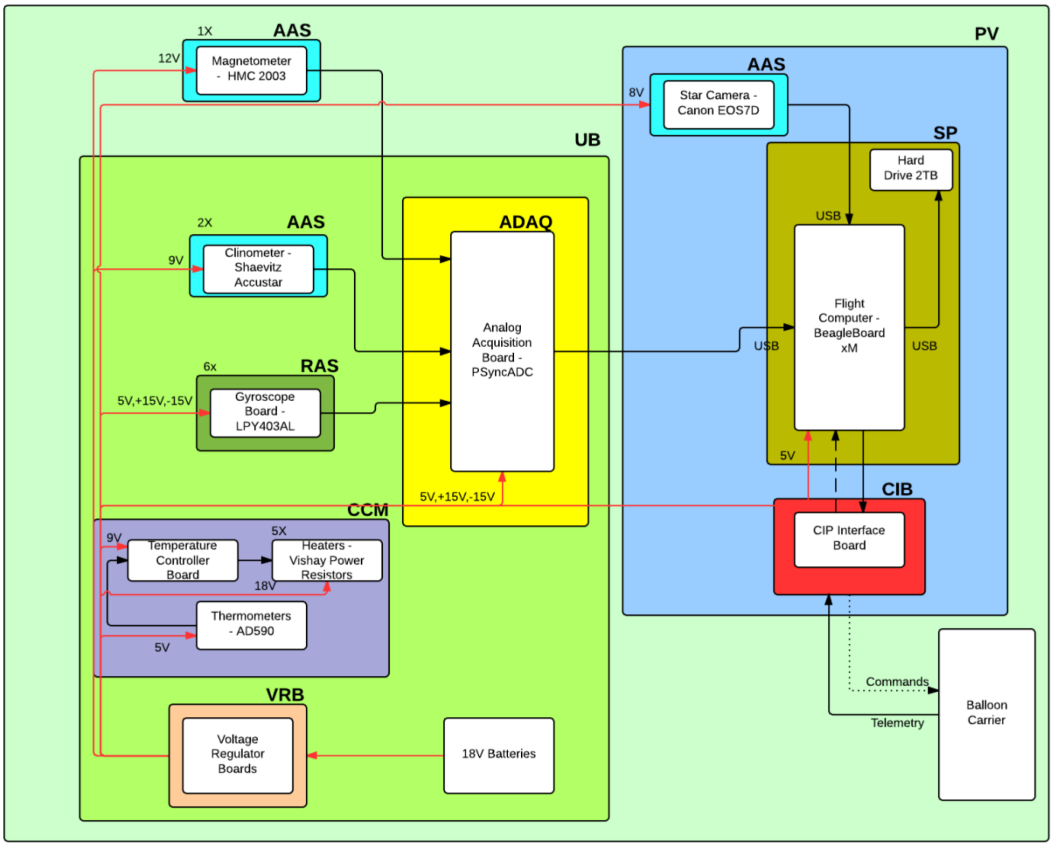


Figure 5 - Subsystems Interconnection

The subsystems from 1-6 are the ones I have directly worked on the design, assembly and test phases.

4.2.1: Relative Attitude Sensing

The RAS subsystem is located inside the UB subsystem and provides relative attitude data to the ADAQ subsystem. It is composed by the Gyroscope Boards.

4.2.1.1: Gyroscopes

The gyroscopes are the fastest sensors in the system. They provide angular rates measurements around their own body-reference axes. Values of angular displacements can be achieved by integrating the sensor output. A MEMS gyroscope can be simply modeled according to [3] as a combination of the true angular velocity $wt_i(t)$, a slowly varying random quantity usually called the bias drift $b_i(t)$ and the white noise $n_i(t)$, which corrupts the signal and becomes the angle random walk (ARW) at the gyro angle level [4]. The index i represents the axis in consideration.

$$w_i(t) = wt_i(t) + b_i(t) + n_{ai}(t)$$

$$\frac{\partial b_i(t)}{\partial t} = n_{bi}(t)$$

This model only takes into account the noises caused by the angle random walk (ARW) and the rate random walk (RRW) processes, which are going to be discussed in more details in the next sections. Other noises that can be considered in the model include the bias drift and the bias instability. However, many experiments have shown that these are the most significant noise terms in MEMS gyroscopes [4]. The following diagram represents the model described in the equations above:

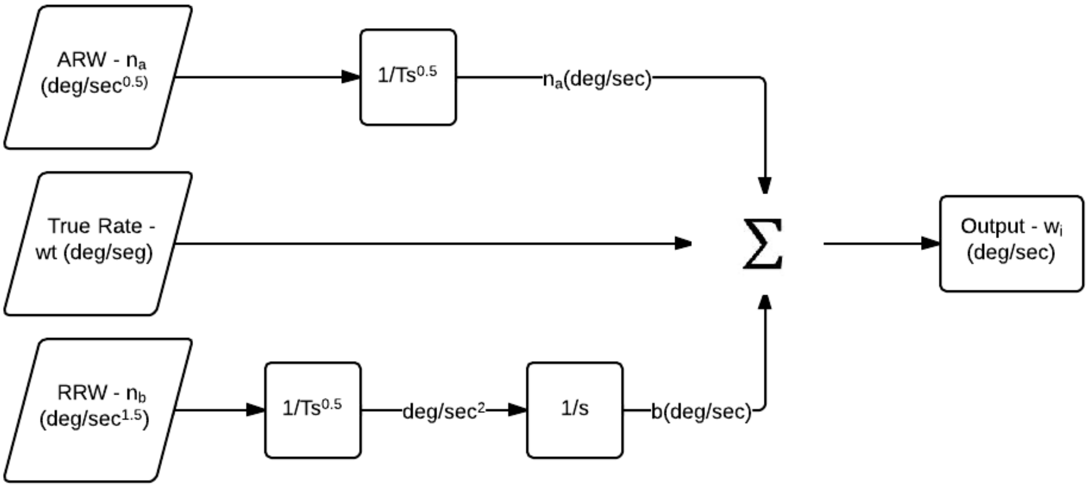


Figure 6 - Gyroscope Model

4.2.1.1.1: Angular Random Walk (ARW)

The angular random walk is the process noise produced by the white noise in the rate signal level and it is expressed in terms of deg/sec$^{0.5}$. When this component is integrated, it generates a random walk in the gyroscope angle level, which corrupts the signal over time. The standard deviation of the gyroscope measurement is proportional to the ARW value and the amount of time the signal has been integrated, as in the following relation:

$$\sigma = (ARW)\sqrt{t}$$

For instance, consider a gyroscope with an ARW of 1 deg/sec$^{0.5}$. After one second the standard deviation of the measurement will be 1 degree and in 100 seconds the standard deviation will be 10 degrees. The noise generated by integrating the white noise is usually called brown noise.

4.2.1.1.2: Rate Random Walk (RRW)

The rate random walk is the noise process produced by the white noise in the acceleration signal level of the gyroscope. It produces a random walk in the rate signal level, which is usually called bias drift.

4.2.1.1.3: Gyroscope Model Selected

The gyroscope model selected to this project was the LPY403AL. The sensor has an angular random walk of about 0.03 deg/sec$^{0.5}$ at a sample rate of 100Hz, which generates a standard deviation of 0.03º after one second of normal operation. Figure 7 shows the gyroscope model selected,

Figure 7 - LPY403AL breakout board

An amplified board was designed and assembled in order to adjust the sensor voltage range into the acquisition board range and thus increase the resolution of measurement. The design was provided by partners from NASA/GSFC. In total, six gyroscope boards were used in order to decrease the uncertainty of the measurements. As shown in Figure 8, three gyroscope boards were mounted one above each other.

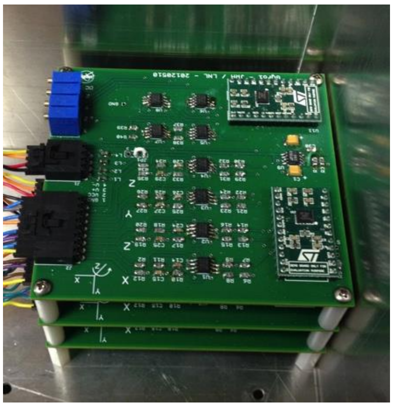


Figure 8 - Gyroscope Boards

4.2.2: Absolute Attitude Sensing (AAS)

The absolute attitude sensing provides absolute attitude data to the ADAQ and SP subsystems. It is composed by the clinometers, the magnetometer and the star tracking camera.

4.2.2.1: Clinometer

Clinometers are instruments used for measuring tilt angles. In the current system, two clinometers are used to determine the angular displacements around the X and Y axes. The clinometer model selected for the system is the Accustar Electronic Clinometer (Figure 9), which has a bandwidth of 0.5 Hz and provides a measurement with a white noise RMS of less than 0.007 degrees in a +/- 45 degrees range.

Figure 9 - Clinometer

4.2.2.2: Magnetometer

Magnetometers are instruments used for measuring magnetic fields in space. They can be used to provide a compass heading by taking advantage of the earth's magnetic field. The direction and strength of the local magnetic field are represented by Hx, Hy, Hz, which are the values of the magnetic field across each axis. The Hx and Hy information can be used to provide a compass heading in relation to the magnetic poles. The compass heading can be determined by placing the sensor in a flat area aligned to the surface and using the Hx and Hy values [8]. These values vary as the device is rotated, as seen in Figure 10.

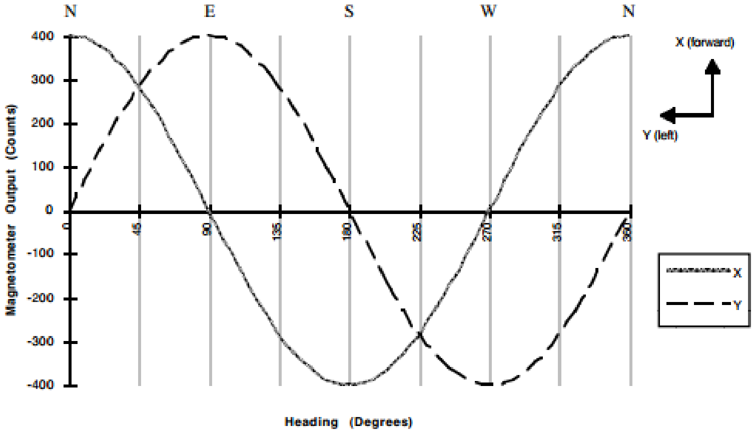


Figure 10 - Hx and Hy components in a magnetometer full rotation [8]

The compass heading is the angular displacement in relation to the local magnetic north vector and it can be obtained using the following relation:

$$\psi = arctan2(\frac{H_x}{H_y})$$

In order to obtain the true north heading, the magnetic declination angle of the current region must be subtracted. The magnetic declination is the angle between the magnetic north and the true north, and it varies depending on the position on Earth.

The magnetometer selected for the system was the model HMC2003, which provides a resolution of 40 µgauss with a nominal sensitivity of 1 Volts/ µgauss.

4.2.2.2.1: Tilt Compensation

The tilt compensation maps the current Hx and Hy measurements to the horizontal plane and provides a more accurate heading calculation by compensating the angular displacements around the X and Y axes. The following calculations are needed for the tilt compensation according to [9]

$$H_x^{'} = H_x cos(\beta) + H_y sin(\beta) sin(\phi) + H_z sin(\beta) cos(\phi)$$

$$H_y^{'} = H_y cos(\phi) + H_z sin(\phi)$$

$$\psi = arctan2(\frac{H_x^{'}}{H_y^{'}})$$

Where H_X and H_y are the magnetometer components along the X and Y axes, ϕ is the angular displacement around the X axis given by the clinometer and β is the angular displacement around the Y axis given by the clinometer.

4.2.2.3: Star Tracking Camera

The Star Tracking Camera is used to provide very accurate attitude data by comparing the position of stars. The camera selected for the system is the Canon-EOS7D. The program gphoto2 is used to communicate with the camera via flight computer and is responsible to control the camera parameters and photo commands.

Figure 11 - Star Tracking Camera

4.2.2.4: Baffle System

Baffle systems are used in Star Trackers in order to minimize the exposure of the optical system to stray light (scattered by dust particles) and prevent from Sun blinding. When the stray light is minimized, the Star Tracker is able to detect very faint stars and provide an attitude value with high precision. The major drawback is that the baffle could reduce the field of view of the camera, which requires the star tracker to detect a sufficient amount of stars in order to have enough number of stars for the attitude determination [5].

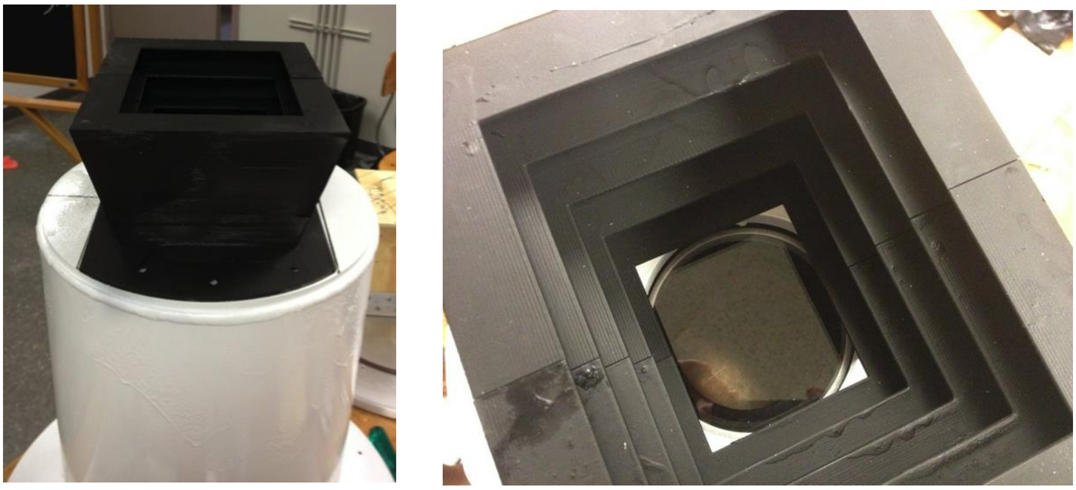


Figure 12 - External Baffle

4.2.3: Analog Data Acquisition (ADAQ)

The analog data acquisition subsystem is responsible for extracting attitude data from the RAS and AAS subsystems and send it to the Signal Processing subsystem.

4.2.3.1: Analog Acquisition Board

The analog acquisition board can read 32 differentials analog channels with 16 bits of resolution at sample rates from 0-400 Hz with a very low noise. For its range of application (+/- 10V), the measured inner noise is white, with a RMS value of 0.18mV, which represents 0.6 LSB.

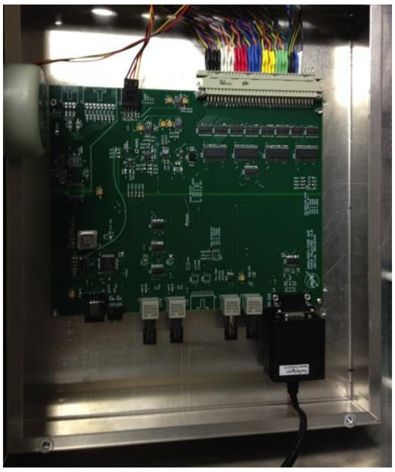


Figure 13 - Analog Acquisition Board (PSyncADC)

A program developed in Python reads the differential channels voltages, in which the sensors are coupled, and saves all the data into a file.

4.2.4: Signal Processing (SP)

The signal processing subsystem is responsible for receiving all data from the ADAQ and AAS, estimating the current attitude and storing all data in the Hard Drive.

4.2.4.1: Flight Computer

The flight computer is the element responsible for receiving all the data from sensors and camera, process them and send it to the hard drive. The flight computer selected for the application is the Beagle Board xM, which is a low-power open-source hardware single board computer. This single board computer was selected due to its processor speed, Linux compatibility, number of USB ports, low mass and short dimensions.

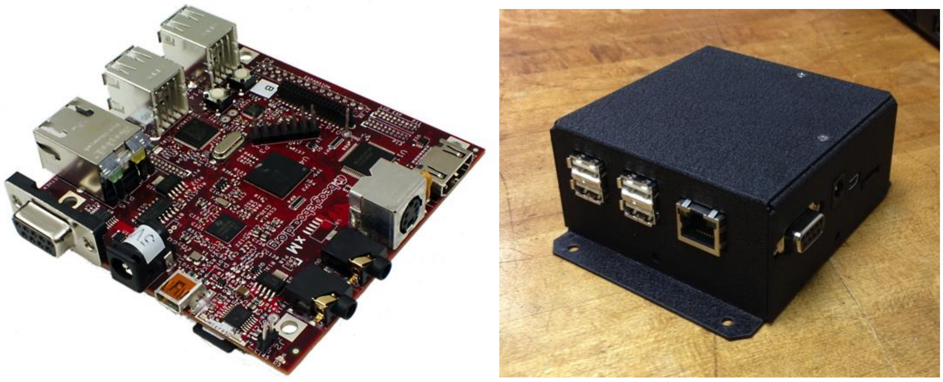


Figure 14 - Beagle Board xM

4.2.5: Climate Control and Monitor (CCM)

The climate control and monitor is the subsystem responsible of controlling the temperature and thus maintaining all system components in a normal operation mode. A simple on-off control temperature was implemented, since there are not rigid requirements for temperature. The following table shows the operational temperatures of the different components in the system:

Component	Working Temperature
Clinometer	from -30˚C to +65˚C
Gyroscope Board	from -40˚C to +85˚C
Magnetometer	from -40˚C to +85˚C
PSyncADC	from -40˚C to +85˚C
Arduino	from -40˚C to +85˚C
Star Camera	from 0˚C to +40˚C
BeagleBoard xM	from 0˚C to +85˚C
Hard Drive	from +5˚C to +40˚C

The temperature range selected for the on-off control based on the elements working temperature was chosen to be between +15˚C and +20˚C.

4.2.5.1: Temperature Sensor

The temperature sensor selected was the AD590, which is a 2-terminal sensor that produces an output current proportional to the absolute temperature and senses a wide range temperature: from -55 C to +150 C.

Figure 15 - AD590 Sensor

The AD590 produces an output current proportional to the absolute temperature. The device acts as a high impedance constant current regulator, conducting 1 µA / K for supply voltages between +4V and +30V. Thus, by using a 1000 ohm resistor, its sensitivity becomes 0.001 V / K.

The following circuit was used for measuring the temperature:

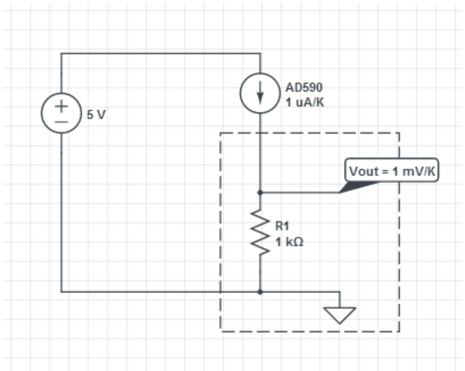


Figure 16 - AD590 Application Circuit

The temperature is read by the temperature controller, which compares it to the desired range of temperature and decides to switch the heaters on or off.

4.2.5.2: Temperature Controller

An Arduino Uno was selected to read the analog temperatures provided by the AD590 sensors and to send signals to switch on/off the heaters.

A gain amplifier was needed for the sensors signal, since the controller selected has a very low resolution and due to the fact that the Arduino can only read 10-bit analog data, which represents a resolution voltage of about 0.00488 V. This resolution is not very good considering that the voltage changes 0.001 V / K with temperature. In order to increase the resolution of the measurement an analog-to-digital converter was selected. The model selected was the ADS1115, which provides 16-bit precision at 860 samples per second over the protocol I2C.

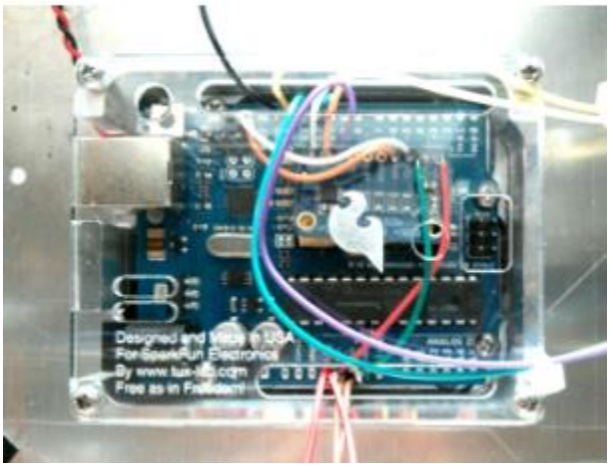


Figure 17 - Temperature Controller (Arduino Uno)

4.2.5.3: Heaters

The heaters selected for the application are wirewound, aluminum housed resistors, which can be mounted on chassis to take advantage of the heat-sink effect. The resistance value selected was 25 Ω.

Figure 18 - Wirewound Resistor

4.2.5.4: Transistor Boards

In order to switch the power supplied to the heaters, a BJT transistor was selected. The transistors base is controlled by the arduino. The transistor selected was a NPN Silicon Transistor with a maximum current of 3A and a breakdown voltage of 30 V.

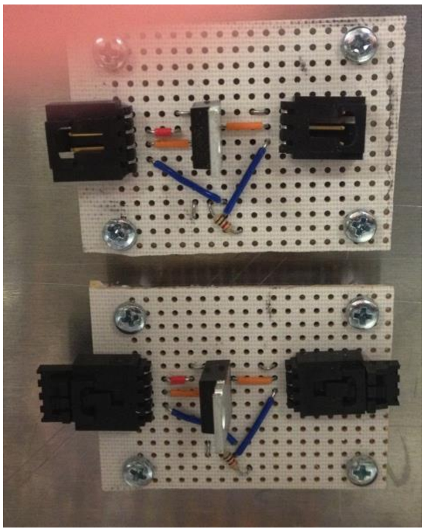


Figure 19 - Transistor Boards

4.2.6: Voltage Regulator Boards (VRB)

Since the components require different operation voltages, voltage regulator boards have been designed in order to supply the desired values. The voltages required are the following: +5V, +8V, +9V, +12V, +15V and -15V. The voltage regulator selected for the design is the LT1764A, which is a low dropout regulator with a low noise in the order of 40 µV RMS. It allows an output current of 3 amperes. The design also uses metal heat sinks on the border of the boards. Figure 20 shows the voltage regulator boards assembled:

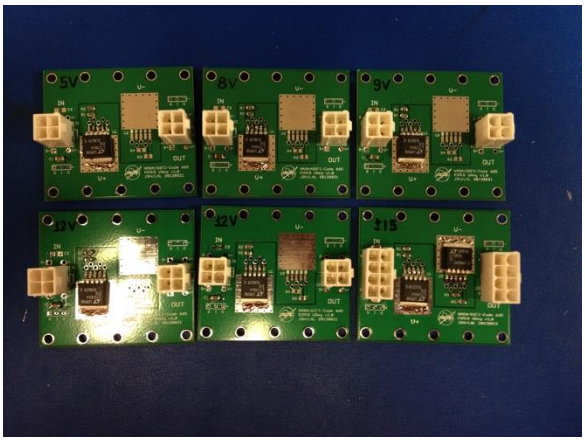


Figure 20 - Voltage Regulator Boards

4.2.7: CIP Interface Board (CIB)

The Consolidated Instrument Package (CIP) is the command and data acquisition system of the balloon carrier. It can be used to receive data from the payload for telemetry and housekeeping and send/receive commands. In our application, the CIP will be used only for commands of power on/off. An interface board has been designed to receive the command and switch the batteries on and off. A power relay with a maximum switch current of 8 amperes has been selected for this purpose. The following circuit design (Figure 21) was used for assembly the CIP Interface Board.

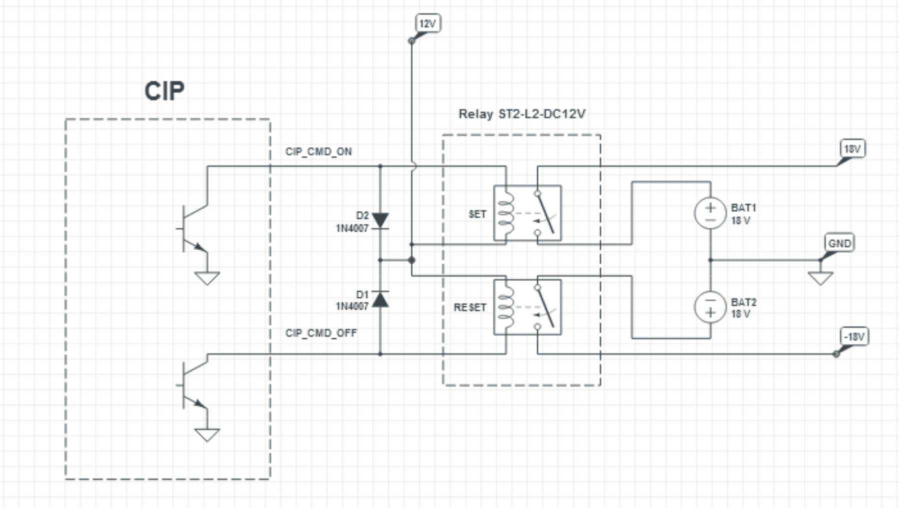


Figure 21 - CIP Interface Board Scheme

The assembled board is shown in the figure below:

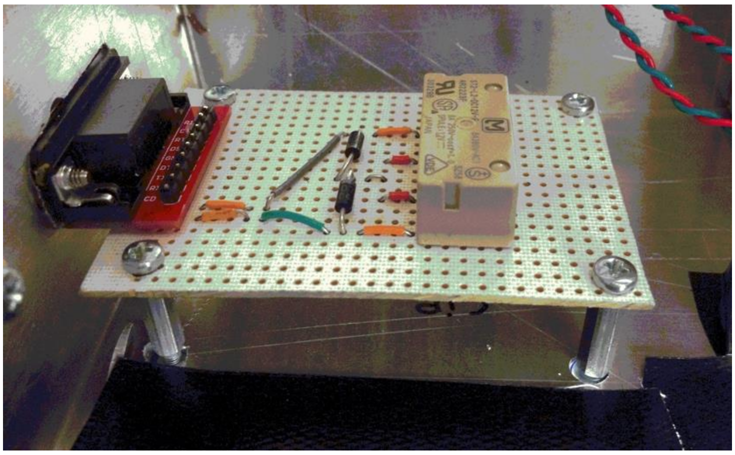


Figure 22 - CIP Board

4.2.8: Unpressurized Boxes (UB)

The unpressurized boxes subsystem is composed by the rectangular unpressurized box, the triangular box and the battery box. The components that fit inside them do not have criticality concerns about temperature.

4.2.8.1: Rectangular Unpressurized Box

The rectangular unpressurized box holds all the elements, which are not strictly required to be in the pressure vessel. The box selected is an aluminum box, which is a lightweight material, with 17 inches large, 15 inches height and 6 inches height. In Figure 23 is shown a top view of the unpressurized box with the elements inside.

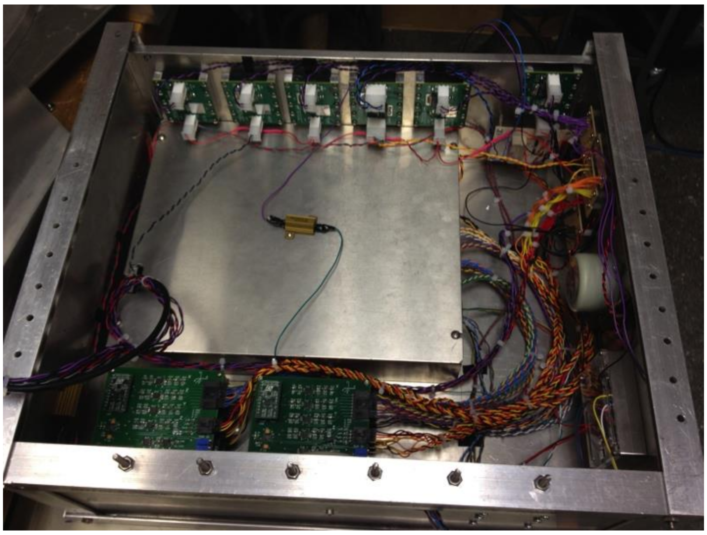


Figure 23 - Rectangular Unpressurized Box

4.2.8.2: Triangular Box

The triangular box has provides an inclination to the pressure vessel in order to remove the balloon carrier from the camera field o view and connect it to the unpressurized box.

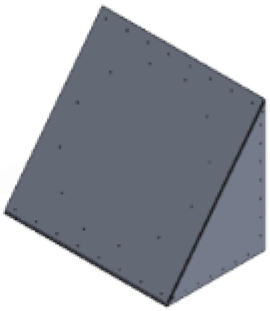


Figure 24 - Triangular box

4.2.8.3: Battery Box

The battery box holds the batteries that supply power to the elements inside the unpressurized box and the pressure vessel.

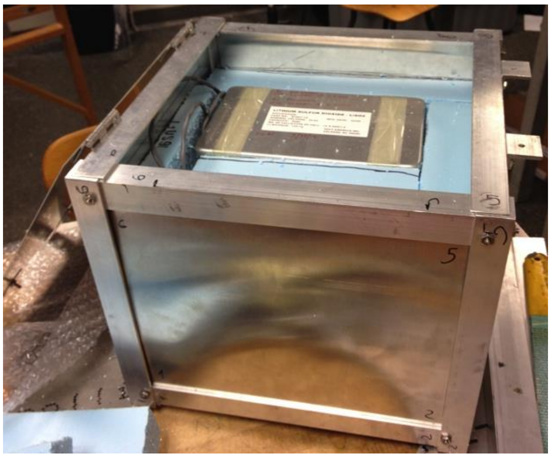


Figure 25 - Battery box

4.2.9: Pressure Vessel (PV)

The pressure vessel subsystem is the structure responsible to provide a suitable environment to the elements that are more sensitive to temperature, which include: the Flight Computer, the Star Tracking Camera and the Hard Drive. The pressure vessel consists of an aluminum cylinder attached to a bottom plate. Two hermetic connectors are mounted on the bottom surface of the pressure vessel in order to connect wires between the pressurized and the unpressurized system. A glass window and an external baffle are also attached to the lid of the cylinder.

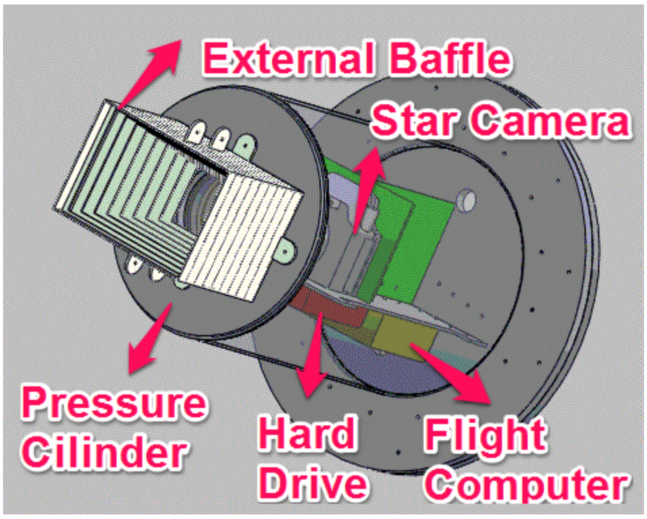


Figure 26 - Pressure Vessel Elements

A comparison between the designed system and the assembled system without the battery boxes is shown in Figure 27.

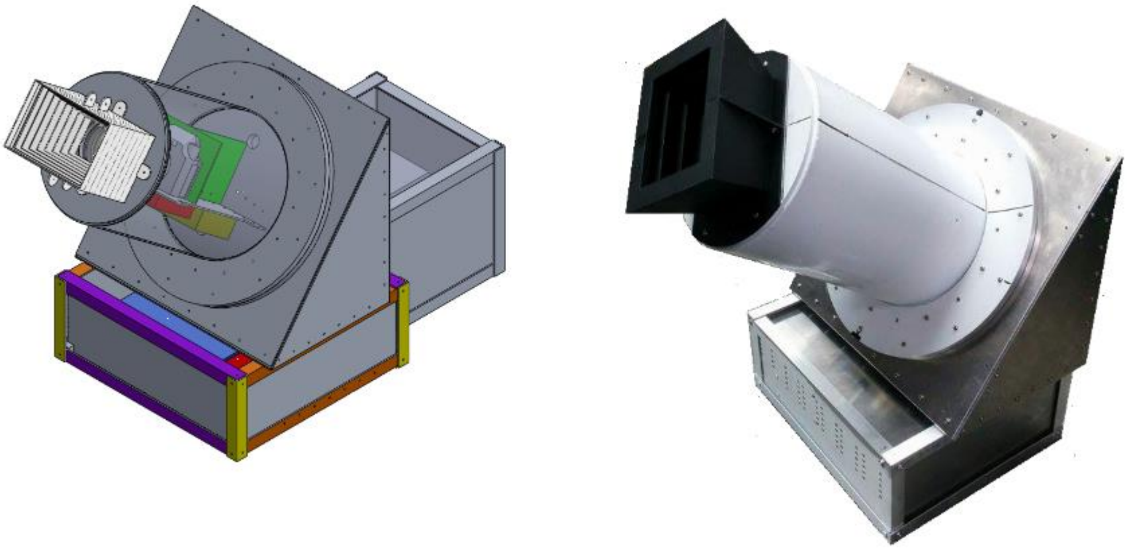


Figure 27 - Final design and real system comparison

Chapter 5: **Kalman Filter Fundamentals**

The Kalman Filter is an optimal estimator that uses a series of measurements containing noise and a prior state knowledge in a recursive manner to produce statistically optimal estimates of the current system state with a better precision than the measurement itself. This algorithm has applications in several areas such as control of vehicles, guidance and navigation and It has been successfully explored in many different applications, such as in the Apollo program for trajectories estimation, in the implementation of nuclear ballistic missile submarines and in the guidance and navigation systems of cruise missiles. It is currently used in the guidance and navigation systems of the NASA Space Shuttle and in the attitude control and navigation systems of the International Space Station [10]. The Kalman filter model assumes that the state of the system is governed by a linear stochastic difference equation:

$$x_{k+1} = Fx_k + Bu_k + w_k$$

Where, F_k is the transition model matrix that relates the state at time step k to the state at step k+1, x_k the current system state, B_k is the control gain matrix, u_k is the control vector and w_k is the zero-mean Gaussian process noise $p(w_k) \sim N(0, Q)$.

The measurement z is given by a linear combination of the state and the measurement noise.

$$z_k = Hx_k + v_k$$

Where, H_k is the observation model matrix that maps the true space into the observed space and v_k is the zero-mean Gaussian measurement noise $p(v) \sim N(0, R)$.

The initial state x_0 is assumed also to follow a Gaussian distribution:

$$p(x_0) \sim N(\hat{x_0}, P_0)$$

In the following sections of this chapter, the notation $\hat{x}_{n|m}$ will be used, which represents the estimate of x at time step n and taking into account the observations up to and including time step m. There are, basically, two estimates used in the Kalman Filter, which are typically called the *a priori* estimate represented by $x_{k+1|k}$ and the *a posteriori* estimate $x_{k+1|k+1}$. The *a priori* estimate is produced by propagating the last estimate state into the model linear stochastic equation. The *a posteriori* estimate is calculated by taking into account the current measurement and the *a priori* estimate. Thus, we can define the *a priori* and *a posteriori* estimates errors respectively as $E[x_{k+1} - \hat{x}_{k+1|k}]$ and $E[x_{k+1} - \hat{x}_{k+1|k+1}]$.

The *a priori* and *a posteriori* error covariance can be defined, as well, as:

$$P_{k+1|k} = E[(\hat{x}_{k+1|k})(\hat{x}_{k+1|k})^T]$$

$$P_{k+1|k+1} = E[(\hat{x}_{k+1|k+1})(\hat{x}_{k+1|k+1})^T]$$

The state of the filter is given by two variables:

- The *a posteriori* state represented by $x_{k|k}$*;*
- The *a posteriori* error covariance matrix $P_{k|k}$, which is a measure of the current accuracy of the state estimate.

5.1: Kalman Filter Loop

The Kalman Filter calculation is usually divided in two phases: the predict phase and the update phase. In the predict phase, the *a priori* estimate and the predicted error covariance matrix are calculated. In the update phase, the *a posteriori* estimate and the *a posteriori* error covariance matrix are calculated. These phases are discussed in the following sections.

5.1.1: Predict Phase

The predict phase generates the *a priori* estimate, since it is calculated without taking account any observations. The *a priori* estimate is given by the following relation:

$$\hat{x}_{k+1|k} = F_k\hat{x}_{k|k} + B_k u_k + w_k$$

The *a priori* error covariance is also calculated in the predict phase using the transition matrix F and the process noise covariance matrix Q.

$$P_{k+1|k} = F_k P_{k|k} F_k^T + Q$$

5.1.2: Update Phase

In the update phase, the *a posteriori* estimate is calculated by taking into account the observation at the current time step and the *a priori* estimate. In a first step, the innovation is calculated which represents the difference between the observed state z_{k+1} and the *a priori* estimate.

$$y_{k+1} = z_{k+1} - H x_{k+1|k}$$

In order to calculate the Kalman Gain, that minimizes the *a posteriori* error covariance, the innovation covariance matrix S_k is calculated as follows:

$$S_{k+1} = H P_{k+1|k} H^T + R$$

Where R is the measurements noise covariance matrix.

The Kalman gain is, then, given by the following equation:

$$K_{k+1} = P_{k+1|k} H^T S_{k+1}^{-1}$$

The *a posterior*i estimate is calculated using the *a priori* estimate and the Kalman gain to weight the innovation.

$$x_{k+1|k+1} = x_{k+1|k} + K_{k+1} y_{k+1}$$

When the measurement error covariance R_k tends to zero, which means that the measurement is trusted, the gain K weights the innovation more heavily, as can be represented by the following limit: [11]

$$\lim_{R_k \to 0} K_k = H_k^{-1}$$

On the other hand, if the *a priori* estimate error covariance P_k approaches zero, the gain K weights less the innovation, as can be represented in the following limit: [11]

$$\lim_{P_k \to 0} K_k = 0$$

The *a posteriori* error covariance is given by the following equation:

$$P_{k+1|k+1} = (I - K_{k+1}H_{k+1})P_{k+1|k}$$

5.1.3: Kalman Filter Flowchart

The following flux chart represents the whole recursive calculations of the Kalman Filter algorithm:

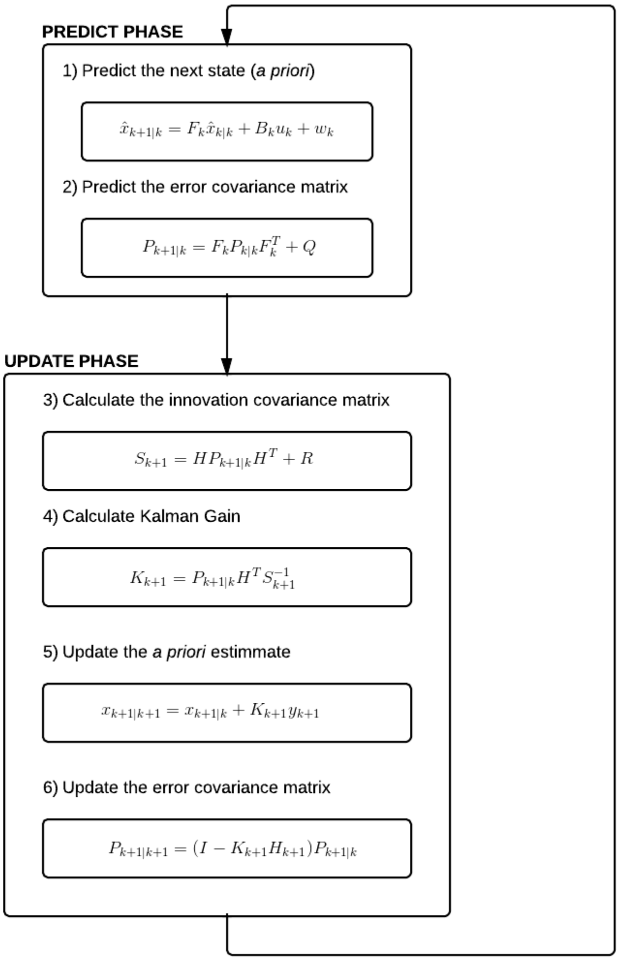

PREDICT PHASE
1) Predict the next state (*a priori*)
$\hat{x}_{k+1|k} = F_k\hat{x}_{k|k} + B_k u_k + w_k$
2) Predict the error covariance matrix
$P_{k+1|k} = F_k P_{k|k} F_k^T + Q$
UPDATE PHASE
3) Calculate the innovation covariance matrix
$S_{k+1} = HP_{k+1|k}H^T + R$
4) Calculate Kalman Gain
$K_{k+1} = P_{k+1|k}H^T S_{k+1}^{-1}$
5) Update the *a priori* estimmate
$x_{k+1|k+1} = x_{k+1|k} + K_{k+1}y_{k+1}$
6) Update the error covariance matrix
$P_{k+1|k+1} = (I - K_{k+1}H_{k+1})P_{k+1|k}$

Figure 28 - Kalman Filter Flowchart

5.2: Performance

It has been proven that the Kalman Filter is optimal for the following conditions: [13]

a) The model of the system is completely right;

b) The measurement noises can be modeled as white

c) The noises covariances are exactly determined.

5.3: Extended Kalman Filter

The classic Kalman filter is designed for estimate the state of linear stochastic systems. In order to estimate the state of non-linear systems, an extended Kalman filter needs to be implemented, which consists of linearizing the Kalman filter equations around the current estimate and apply the classic Kalman filter concepts. The model of a non-linear stochastic difference equation is then given by:

$$x_{k+1} = f(x_k, u_k) + w_k$$

With a measurement given by:

$$z_k = h(x_k) + v_k$$

5.3.1: Linearization of Process Model

For using the Extended Kalman Filter the process model is linearized using first order Taylor series around its current estimate:

$$x_{k+1|k} \approx f(\hat{x}_k, u_k) + \bigtriangledown f_x(x_k - \hat{x}_k) + w_k$$

Where $\bigtriangledown f_x$ is the Jacobian of the function f with respect to x evaluated at the current estimate $\hat{x}_{k|k}$. All the higher order terms are ignored in this equation, since it is assumed that x_k is close to $\hat{x}_{k|k}$. [12]

The linearized model can be written as:

$$x_k \approx Fx_k + U_k + w_k$$

Where $F = \bigtriangledown f_x$and $U_k = f(\hat{x}_k, u_k) - \bigtriangledown f_x \hat{x}_k$

5.3.2: Linearization of the observation model

Same linearization needs to be applied to the observation model using first order Taylor series around the current estimate:

$$z_k = h(\hat{x}_k) + \bigtriangledown h(x_k - \hat{x}_k) + v_k$$

Where $\bigtriangledown h$ is the Jacobian of function h evaluated at the current estimate $\hat{x}_k$. The linearized observation model above can be written as:

$$Z_k \approx Hx_k + W_k + v_k$$

Where the matrixes H and W_k are defined as:

$$H = \bigtriangledown h$$

$$W_k = h(\hat{x}_k) - \bigtriangledown h\ \hat{x}_k$$

5.3.3: Extended Kalman Filter (EKF) Loop

The phases in the Extended Kalman Filter are the same than the classic Kalman Filter. The predict phases and update phases are discussed below:

5.3.3.1: EKF Predict Phase

In order to obtain the *a priori* estimate, the non-linear model is used with the values of the last state estimate and the control inputs.

$$x_{k+1|k} = f(x_{k|k}, u_k)$$

The covariance state error matrix can be calculated by:

$$P_{k+1|k} = \mathbf{F_k} P_{k|k} \mathbf{F_k^T} + Q$$

Where $\mathbf{F_K}$ is the Jacobian of function f evaluated at time step k.

5.3.3.2: EKF Update Phase

The calculations for the extended Kalman filter are quite similar to the ones presented in the classic version.

The innovation covariance matrix S_k is given by:

$$S_{k+1} = \mathbf{H} P_{k+1} \mathbf{H^T} + R$$

Where **H** is the Jacobian of function h. The Kalman gain is given by:

$$K_{k+1} = P_{k+1|k} \mathbf{H^T} S_{k+1}^{-1}$$

The following equations are used for updating the state and state error covariance matrix.

$$x_{k+1|k+1} = x_{k+1|k} + K_{k+1}(z_{k+1} - h(x_{k+1|k}))$$

$$P_{k+1|k+1} = (I - K_{k+1}H_{k+1})P_{k+1|k}$$

5.3.4: Extended Kalman Filter Loop

The following flowchart shows all the recursive calculations needed for the extended Kalman Filter:

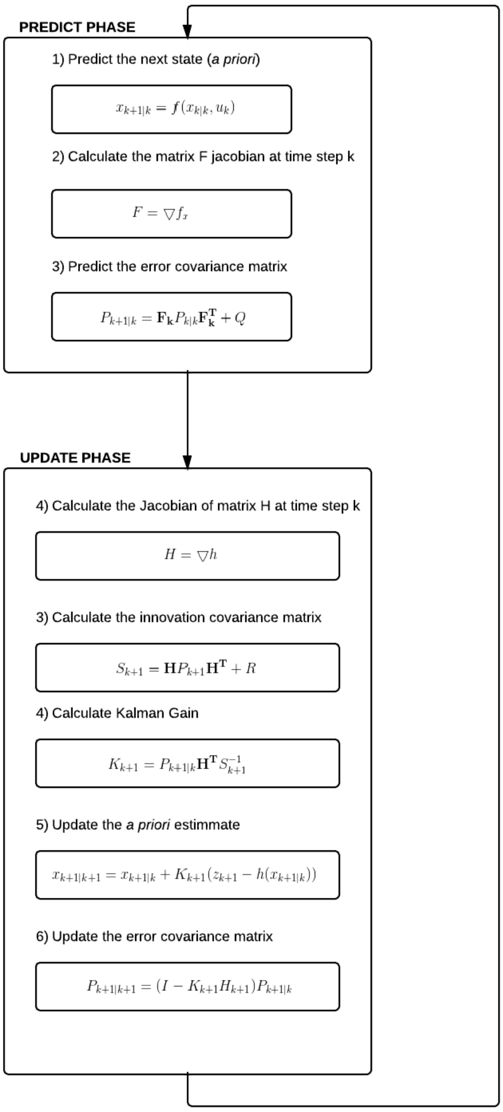
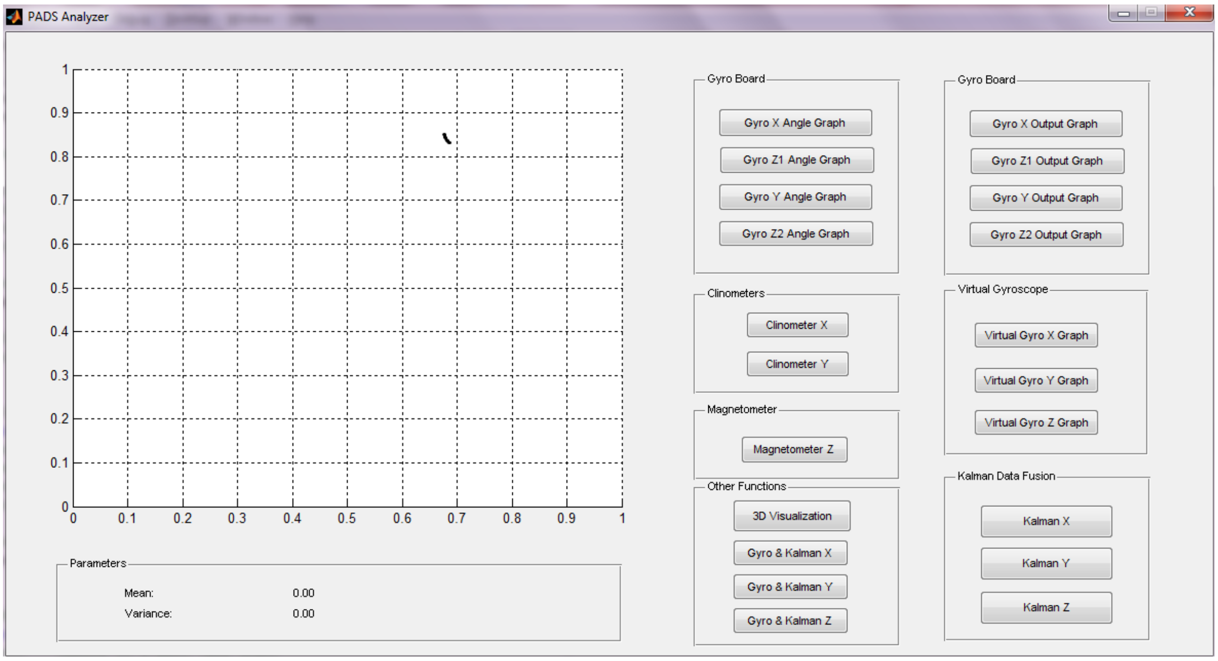

PREDICT PHASE
1) Predict the next state (*a priori*)
$x_{k+1|k} = f(x_{k|k}, u_k)$
2) Calculate the matrix F jacobian at time step k
$F = \bigtriangledown f_x$
3) Predict the error covariance matrix
$P_{k+1|k} = \mathbf{F_k} P_{k|k} \mathbf{F_k^T} + Q$
UPDATE PHASE
4) Calculate the Jacobian of matrix H at time step k
$H = \bigtriangledown h$
3) Calculate the innovation covariance matrix
$S_{k+1} = \mathbf{H} P_{k+1} \mathbf{H^T} + R$
4) Calculate Kalman Gain
$K_{k+1} = P_{k+1|k} \mathbf{H^T} S_{k+1}^{-1}$
5) Update the *a priori* estimmate
$x_{k+1|k+1} = x_{k+1|k} + K_{k+1}(z_{k+1} - h(x_{k+1|k}))$
6) Update the error covariance matrix
$P_{k+1|k+1} = (I - K_{k+1}H_{k+1})P_{k+1|k}$

Figure 29 - Extended Kalman Filter Flowchart

Chapter 6: **PADS Analyzer**

In order to view and analyze the data from sensors, a program in MATLAB with a GUI interface, denominated “PADS Analyzer” was developed. Several functions were implemented on it. Each of these functions is going to be discussed below. The program interface is shown in Figure 30:



Mean: 0.00

Variance: 0.00

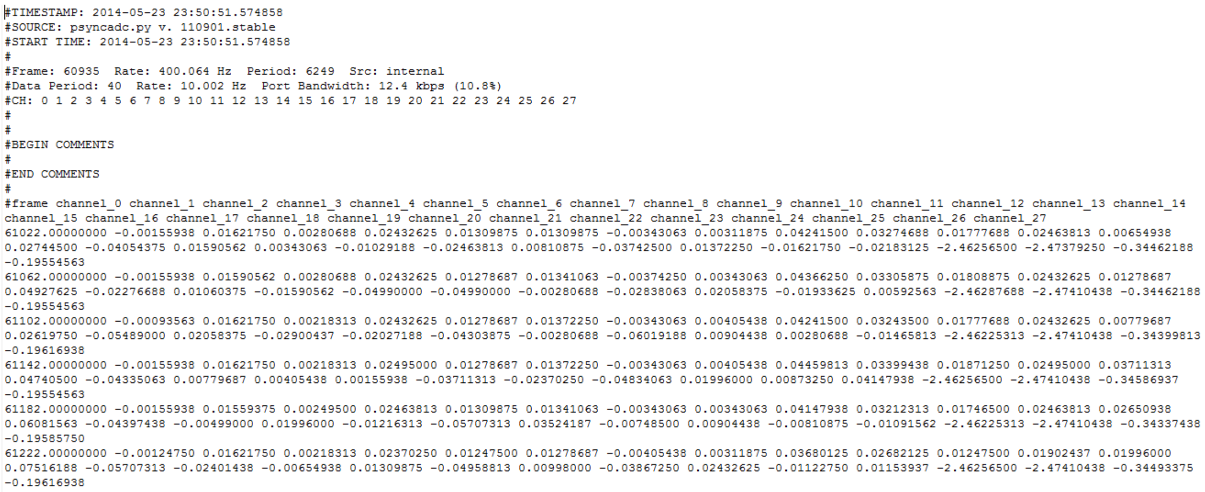
Figure 30 - PADS Analyzer

All functions can be accessed by buttons that appear in the right side of the interface screen. In the left side of the screen, there is an axes that can be used by the functions for plotting purposes and a display, which shows basic parameters of the signal being processed.

6.1: PADS Analyzer Data Flow

The sensor data is read by the Analog Acquisition Board, which runs a program in Python that saves all the data into a text file. The text file is organized in a way in which each row represents a new measurement and each column represents a channel that is coupled to a specific sensor.

A sample of this file is showed below:



```
#TIMESTAMP: 2014-05-23 23:50:51.574858
#SOURCE: psyncadc.py v. 110901.stable
#START TIME: 2014-05-23 23:50:51.574858
#
#Frame: 60935  Rate: 400.064 Hz  Period: 6249  Src: internal
#Data Period: 40  Rate: 10.002 Hz  Port Bandwidth: 12.4 kbps (10.8%)
#CH: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
#
#
#BEGIN COMMENTS
#
#END COMMENTS
#
#frame channel_0 channel_1 channel_2 channel_3 channel_4 channel_5 channel_6 channel_7 channel_8 channel_9 channel_10 channel_11 channel_12 channel_13 channel_14
channel_15 channel_16 channel_17 channel_18 channel_19 channel_20 channel_21 channel_22 channel_23 channel_24 channel_25 channel_26 channel_27
61022.00000000 -0.00155938 0.01621750 0.00280688 0.02432625 0.01309875 0.01309875 -0.00343063 0.00311875 0.04241500 0.03274688 0.01777688 0.02463813 0.00654938
0.02744500 -0.04054375 0.01590562 0.00343063 -0.01029188 -0.02463813 0.00810875 -0.03742500 0.01372250 -0.01621750 -0.02183125 -2.46256500 -2.47379250 -0.34462188
-0.19554563
61062.00000000 -0.00155938 0.01590562 0.00280688 0.02432625 0.01278687 0.01341063 -0.00374250 0.00343063 0.04366250 0.03305875 0.01808875 0.02432625 0.01278687
0.04927625 -0.02276688 0.01060375 -0.01590562 -0.04990000 -0.04990000 -0.00280688 -0.02838063 0.02058375 -0.01933625 0.00592563 -2.46287688 -2.47410438 -0.34462188
-0.19554563
61102.00000000 -0.00093563 0.01621750 0.00218313 0.02432625 0.01278687 0.01372250 -0.00343063 0.00405438 0.04241500 0.03243500 0.01777688 0.02432625 0.00779687
0.02619750 -0.05489000 0.02058375 -0.02900437 -0.02027188 -0.04303875 -0.00280688 -0.06019188 0.00904438 0.00280688 -0.01465813 -2.46225313 -2.47410438 -0.34399813
-0.19616938
61142.00000000 -0.00155938 0.01621750 0.00218313 0.02495000 0.01278687 0.01372250 -0.00343063 0.00405438 0.04459813 0.03399438 0.01871250 0.02495000 0.03711313
0.04740500 -0.04335063 0.00779687 0.00405438 0.00155938 -0.03711313 -0.02370250 -0.04834063 0.01996000 0.00873250 0.04147938 -2.46256500 -2.47410438 -0.34586937
-0.19554563
61182.00000000 -0.00155938 0.01559375 0.00249500 0.02463813 0.01309875 0.01341063 -0.00343063 0.00343063 0.04147938 0.03212313 0.01746500 0.02463813 0.02650938
0.06081563 -0.04397438 -0.00499000 0.01996000 -0.01216313 -0.05707313 0.03524187 -0.00748500 0.00904438 -0.00810875 -0.01091562 -2.46225313 -2.47410438 -0.34337438
-0.19585750
61222.00000000 -0.00124750 0.01621750 0.00218313 0.02370250 0.01247500 0.01278687 -0.00405438 0.00311875 0.03680125 0.02682125 0.01247500 0.01902437 0.01996000
0.07516188 -0.05707313 -0.02401438 -0.00654938 0.01309875 -0.04958813 0.00998000 -0.03867250 0.02432625 -0.01122750 0.01153937 -2.46256500 -2.47410438 -0.34493375
-0.19616938
```

Figure 31 - Data text file sample

This file is continuously updated by the Analog Acquisition Board with a sample time determined by the user. The PADS Analyzer reads this data, process it , saves related parameters and data into a file and allows the user to view the data through graphs in real time. A simple data flowchart of the integrated system is shown in Figure 32:

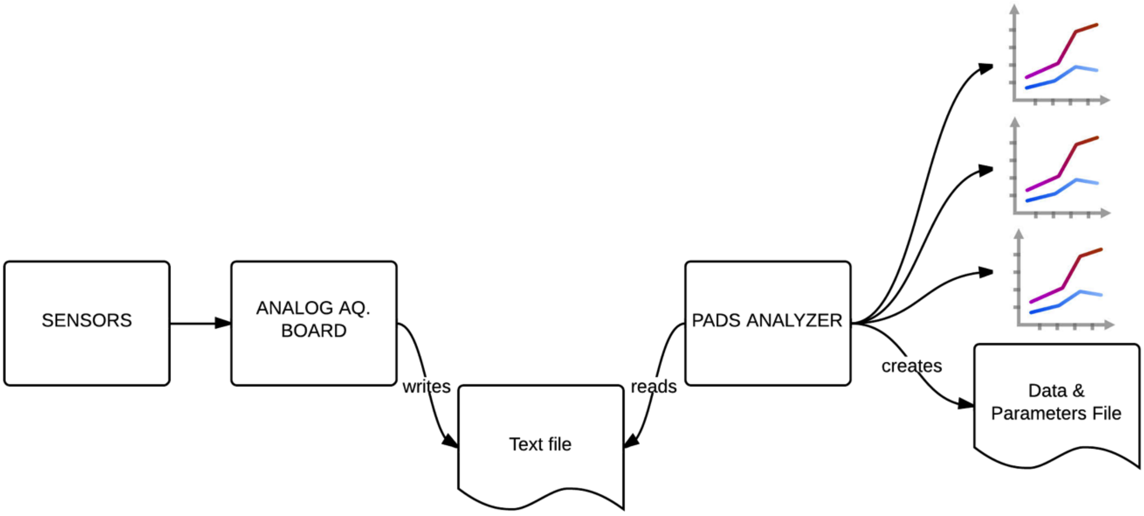
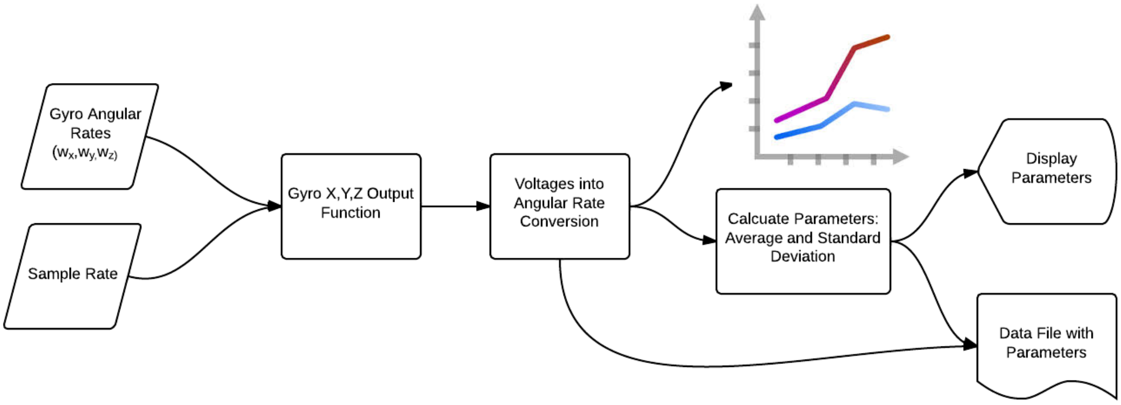


Figure 32 - PADS Analyzer Flowchart

6.2: Gyroscope X,Y,Z Output Functions

The Gyroscope X,Y,Z Output functions allow the user to view the non-integrated gyroscope data. It basically converts the voltage readings into values of angular velocity in degrees per second and plots the data in real time in the main axes. Related parameters such as average and standard deviation are calculated and displayed in the main screen. All sensors data and parameters are saved into a file.

A flow chart for the function is shown in Figure 33:



Gyro Angular Rates (w_x,w_y,w_z)
Sample Rate
Gyro X,Y,Z Output Function
Voltages into Angular Rate Conversion
Calcuate Parameters: Average and Standard Deviation
Display Parameters
Data File with Parameters

Figure 33 - Gyroscope X,Y,Z Output Functions Flowchart

6.3: Gyroscope X,Y,Z Angle Output Functions

The Gyroscope X,Y,Z Angle Output functions allows the user to view the integrated gyroscope data. It subtracts the biases from the gyroscope outputs and integrates the values in order to provide a value of angular displacement. The gyroscope biases are estimated by calculating the average values of the stationary sensor output. It also calculates related parameters and saves them along with the sensors data in a text file.

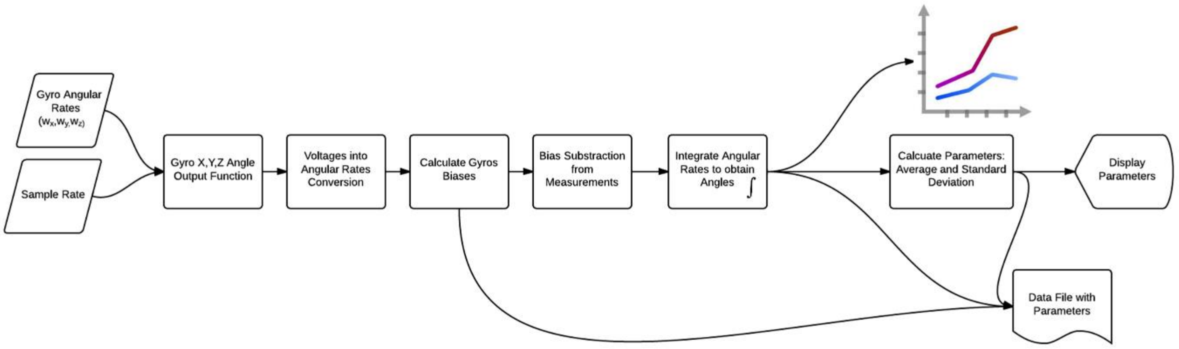

Gyro Angular Rates (wx,wy,wz)
Sample Rate
Gyro X,Y,Z Angle Output Function
Voltages into Angular Rates Conversion
Calculate Gyros Biases
Bias Substraction from Measurements
Integrate Angular Rates to obtain Angles ∫
Calcuate Parameters: Average and Standard Deviation
Display Parameters
Data File with Parameters

Figure 34 - Gyroscope X, Y, Z Angle Output Functions Flowchart

6.4: Clinometer X, Y Functions

The Clinometer X, Y functions allow the user to view the clinometers data and related parameters. The function receives the voltage provided by the clinometer and converts it to an angle. It calculates basic parameters and saves all the data and parameters into a text file. A flowchart of the function is shown in Figure 35.

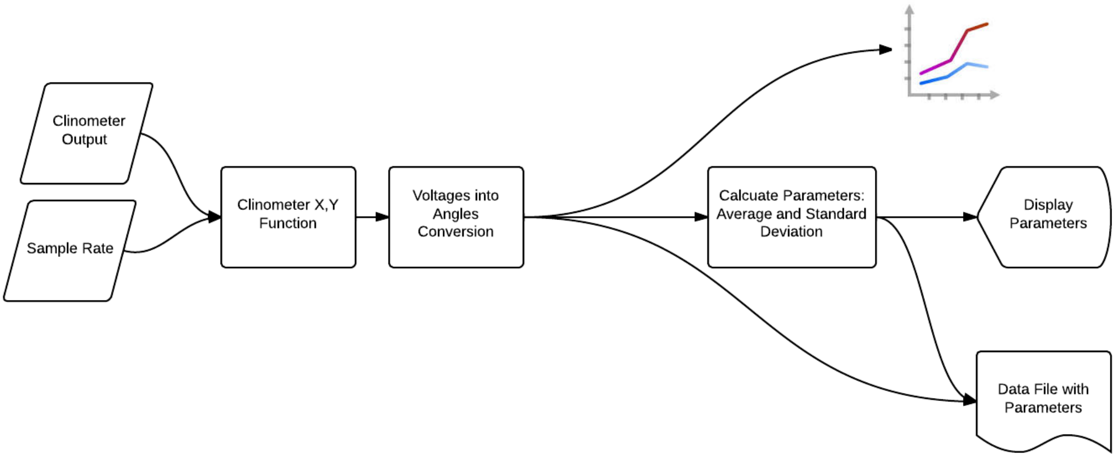

Clinometer Output
Sample Rate
Clinometer X,Y Function
Voltages into Angles Conversion
Calcuate Parameters: Average and Standard Deviation
Display Parameters
Data File with Parameters

Figure 35 - Clinometer X, Y Functions Flowchart

6.5: Magnetometer Z Function

The Magnetometer Z function receives the magnetometer X and Y readings and calculates a compass heading that provides a value of the Z displacement regarding to the local magnetic flux reference. Tilt compensation is also performed using the X and Y displacement values from the clinometers in order increase the accuracy of the measurement.

A flowchart of the Magnetometer Z function is shown in Figure 36.

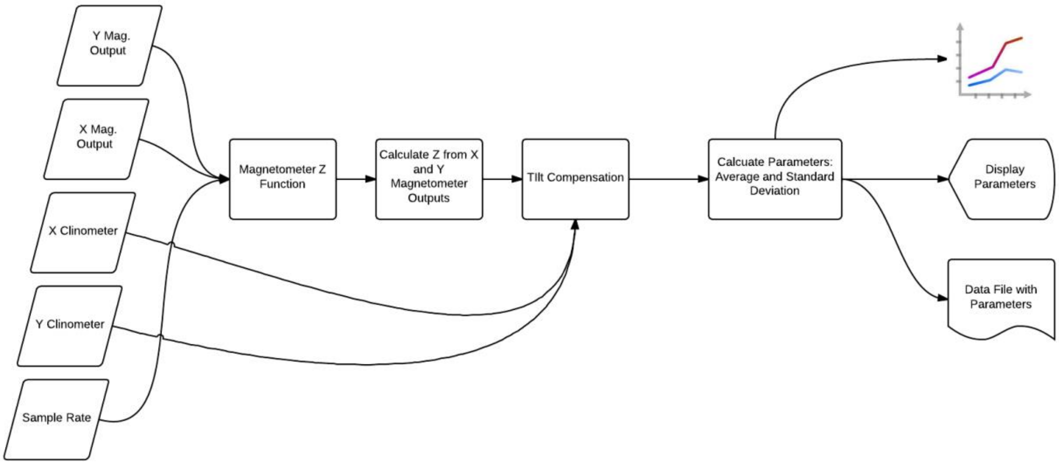

Y Mag. Output
X Mag. Output
X Clinometer
Y Clinometer
Sample Rate
Magnetometer Z Function
Calculate Z from X and Y Magnetometer Outputs
Tilt Compensation
Calcuate Parameters: Average and Standard Deviation
Display Parameters
Data File with Parameters

Figure 36 - Magnetometer Z Function Flowchart

6.6: Virtual Gyro X, Y, Z Functions

The Virtual Gyro X, Y, Z functions estimate the true angular rate by using an Kalman Filter algorithm, which combines the different gyroscopes outputs into a single value, with more accuracy than the one provided by a single gyroscope. A flowchart for the Virtual Gyro X, Y, a Z function is shown in Figure 37.

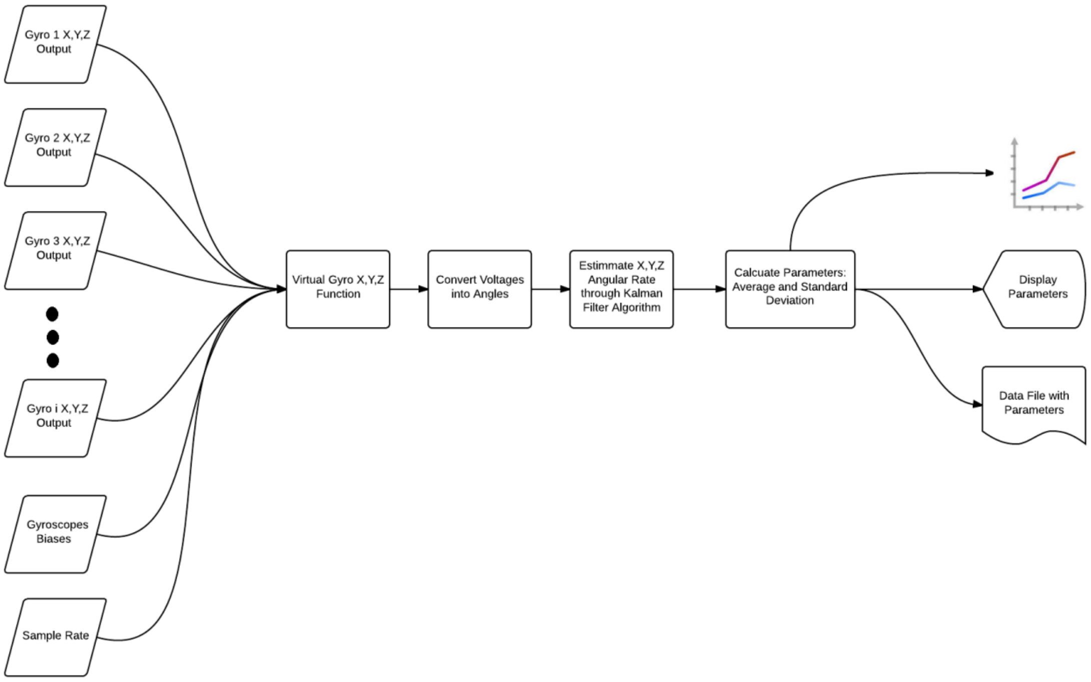

Gyro 1 X,Y,Z Output
Gyro 2 X,Y,Z Output
Gyro 3 X,Y,Z Output
Gyro i X,Y,Z Output
Gyroscopes Biases
Sample Rate
Virtual Gyro X,Y,Z Function
Convert Voltages into Angles
Estimmate X,Y,Z Angular Rate through Kalman Filter Algorithm
Calcuate Parameters: Average and Standard Deviation
Display Parameters
Data File with Parameters

Figure 37 - Virtual Gyro X, Y, Z Flowchart

6.7: Kalman X, Y, Z Functions

The Kalman X, Y, Z functions use an Extended Kalman Filter algorithm to fuse the attitude provided by the virtual gyroscope and the attitude provided by the clinometers and magnetometer. The virtual gyroscope estimate is used to predict the next attitude states until a new clinometer X, Y or magnetometer Z value is available.

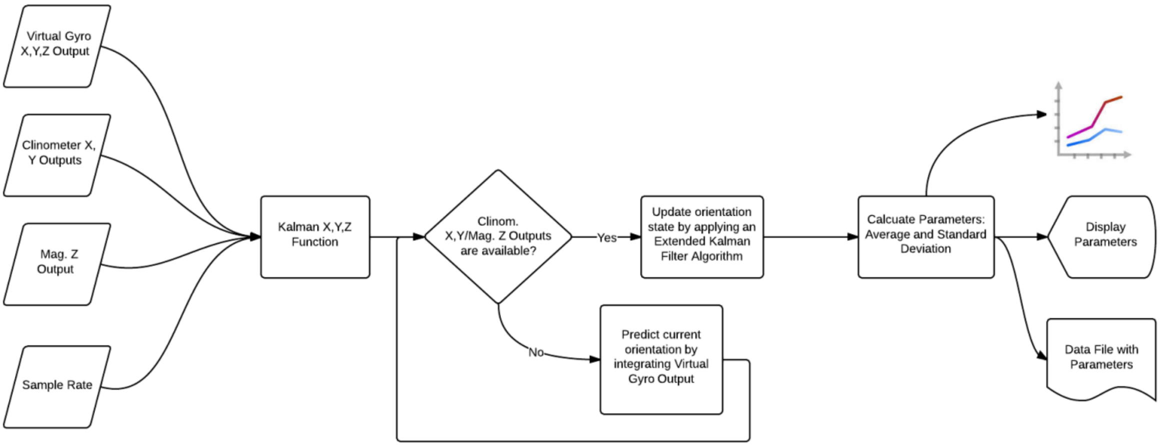

Virtual Gyro X,Y,Z Output
Clinometer X, Y Outputs
Mag. Z Output
Sample Rate
Kalman X,Y,Z Function
Clinom. X,Y/Mag. Z Outputs are available?
Yes
No
Update orientation state by applying an Extended Kalman Filter Algorithm
Predict current orientation by integrating Virtual Gyro Output
Calcuate Parameters: Average and Standard Deviation
Display Parameters
Data File with Parameters

Figure 38 - Kalman X, Y, Z Functions Flowchart

6.8: 3D Visualization Function

A 3D Visualization function was created in order to visualize and check the attitude models and measurements by sensors. The 3D Visualization function applies the attitude estimated by the Kalman X, Y, Z to a rotating block, so the user can see in real time the rotation motion applied to the sensors. The 3D Visualization Function flow chart is shown in the Figure 39:

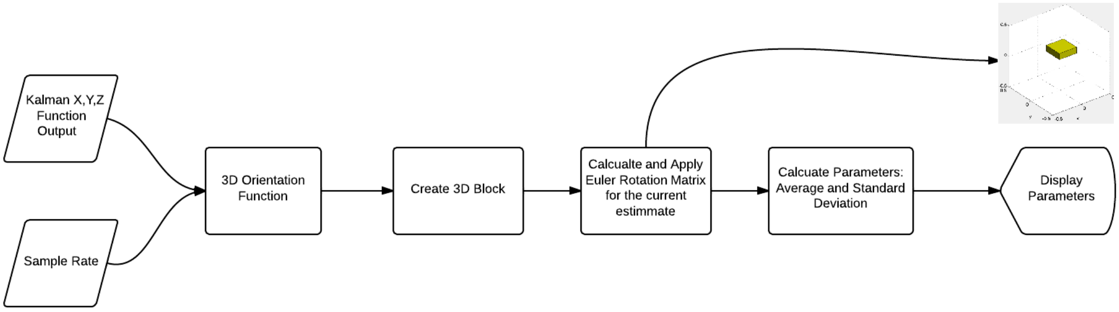

Kalman X,Y,Z Function Output
Sample Rate
3D Orientation Function
Create 3D Block
Calcualte and Apply Euler Rotation Matrix for the current estimmate
Calcuate Parameters: Average and Standard Deviation
Display Parameters

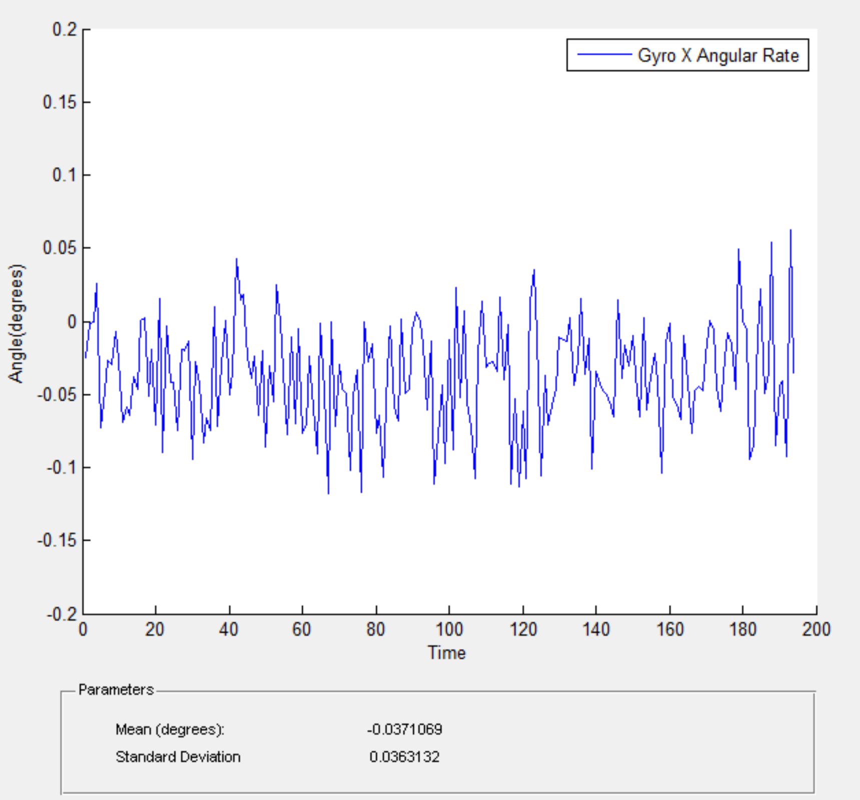
Figure 39 - 3D Visualization Function Flowchart

Chapter 7: **Sensors Noise Analysis**

In this chapter the stationary noise of each sensor is going to be analyzed. This noise analysis is important in order to check if the sensors are working as specified and to obtain important parameters for using in the Kalman Filter fusion algorithms.

7.1: Gyroscope Noise Analysis

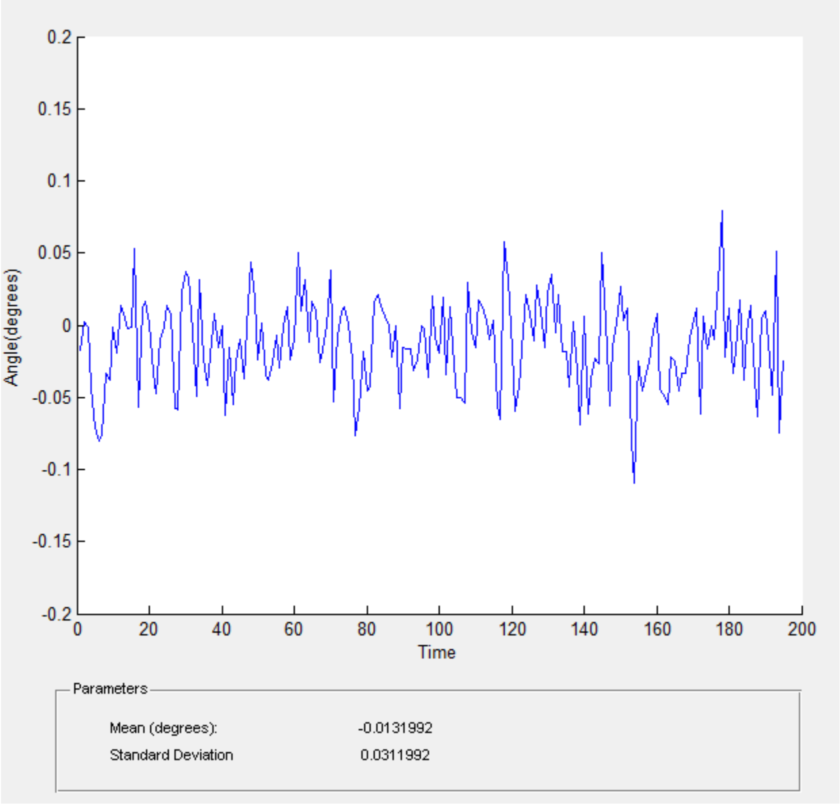
In order to analyze the noise from the gyroscopes the Gyroscope X, Y, Z Graph functions and the Gyroscope X, Y, Z Angle functions of the PADS Analyzer were used. The non-integrated outputs in the X, Y, Z axes are shown below for the gyroscopes in a stationary state.



Mean (degrees): -0.0371069

Standard Deviation 0.0363132

Figure 40 - Gyroscope X Axis Sample Angular Rate

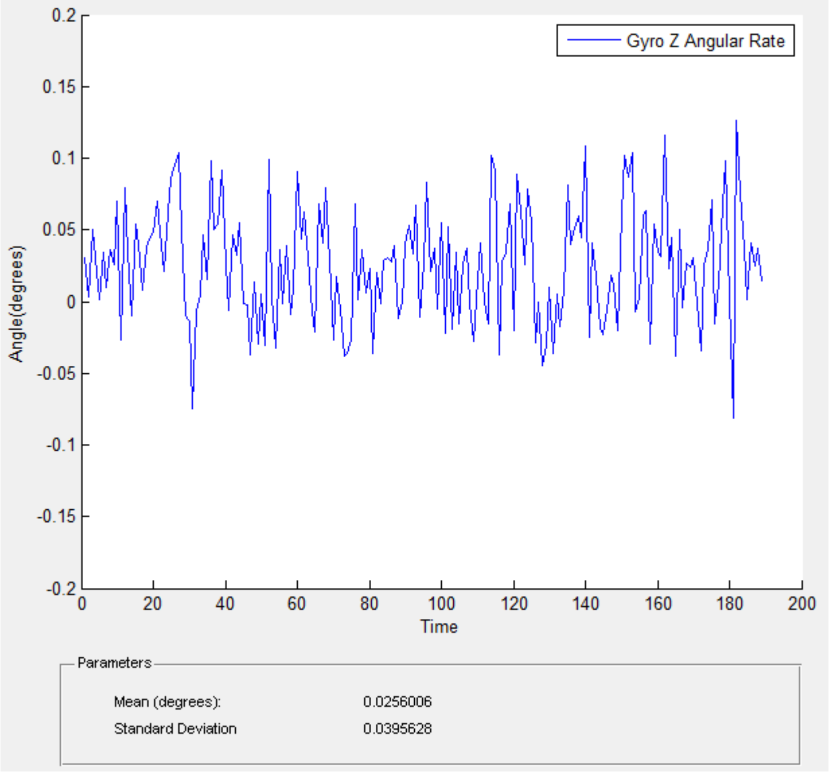


Parameters

Mean (degrees): -0.0131992

Standard Deviation 0.0311992

Figure 41 - Gyroscope Y Axis Sample Angular Rate



Parameters

Mean (degrees): 0.0256006

Standard Deviation 0.0395628

Figure 42 - Gyroscope Z Axis Sample Angular Rate

The values found for the RMS (standard deviation) and biases are shown in the following table:

Output	*RMS*	*Bias (degrees/sec)*
Gyroscope X	*0.03631*	*-0.03711*
Gyroscope Y	*0.03120*	*-0.01320*
Gyrosocpe Z	*0.03956*	*0.02560*

The outputs shown in Figures 43, 44 and 45 are the result of the white noise integration, which is usually called the angle random walk (ARW). The angle random walk can be calculated from the RMS parameters by using the following formula:

$$\sigma_\theta(t) = \sigma \cdot \sqrt{\delta t \cdot t}$$

$$\mathrm{ARW} = \sigma_\theta(1)$$

This quantity represents the standard deviation of the angle indirect measurement at time equal to one second, as discussed in section 4.2.1.1.

The angle random walk parameters for the three axes are showed in the following table:

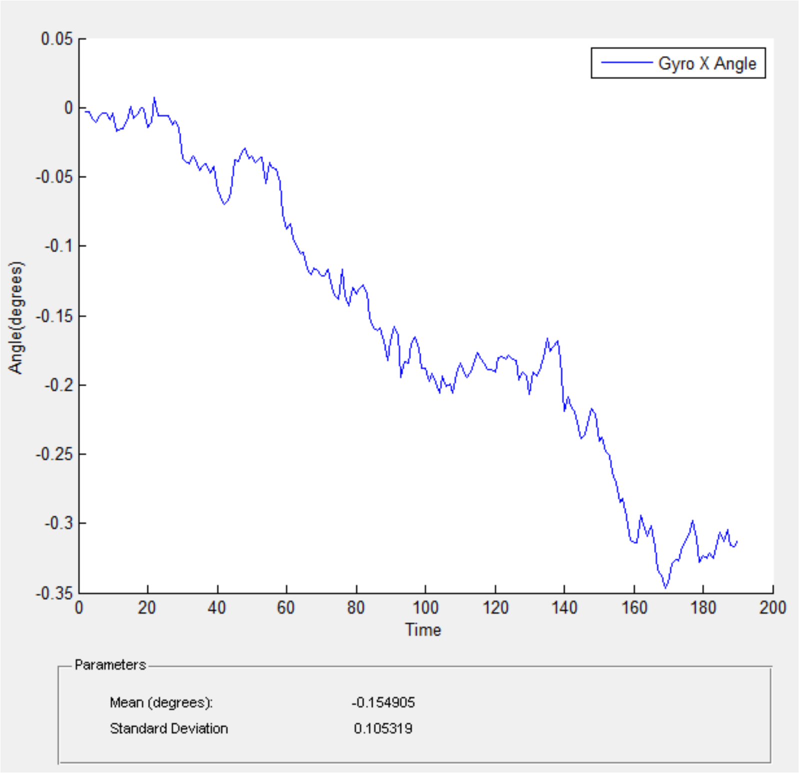
Output	Angle Random Walk (deg/sec$^{0.5}$)
Gyroscope X	*0.03631*
Gyroscope Y	*0.03120*
Gyroscope Z	*0.03956*

The rate random walk parameters were obtained by observing the bias change over time. An one hour test was performed to get data samples, that were clustered and used for the bias calculation. Then, a best-fit linear estimation in MATLAB was used to calculate the RRW parameters, which are shown in the following table:

Output	Rate Random Walk (dps /sec$^{0.5}$)
Gyroscope X	4.14 x 10^{-5}
Gyroscope Y	4.05 x 10^{-5}
Gyroscope Z	3.99 x 10^{-5}

The most relevant gyroscope noise parameters are caused by the ARW process.

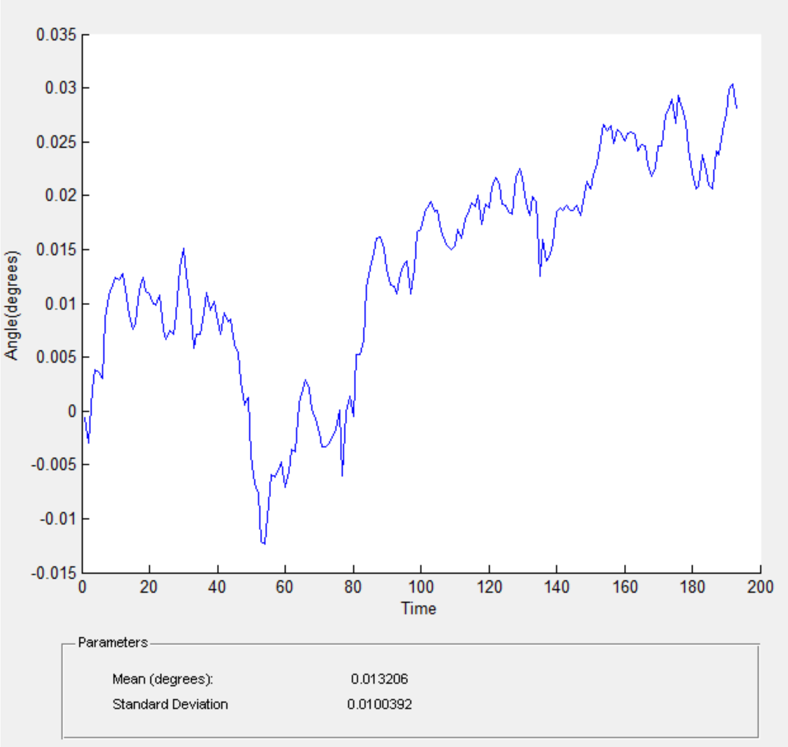
In order to visualize the integrated output of the gyros, the Gyroscope X,Y,Z Angle functions are used. For the gyroscope in a stationary state, the angular displacement in the X, Y, Z axes are shown in Figure 43, 44 and 45.



Mean (degrees): -0.154905

Standard Deviation 0.105319

Figure 43 - Angle Random Walk Gyroscope X Output

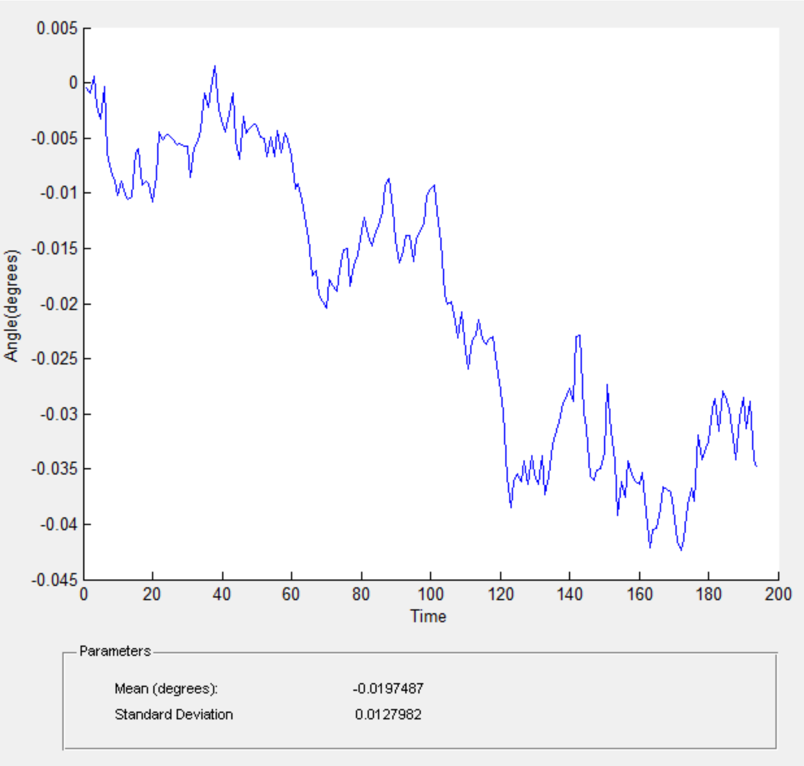


Parameters

Mean (degrees): 0.013206

Standard Deviation 0.0100392

Figure 44 - Angle Random Walk Gyroscope Y Output



Parameters

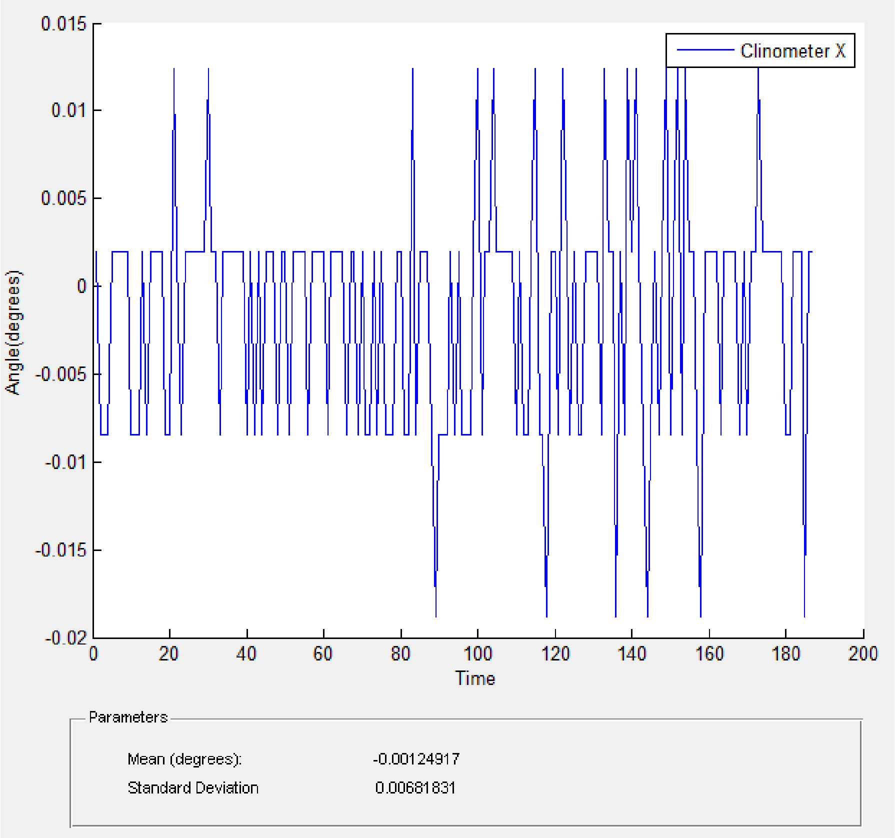
Mean (degrees): -0.0197487

Standard Deviation 0.0127982

Figure 45 - Angle Random Walk Gyroscope Z Output

7.2: Clinometer Noise Analysis

For the clinometer noise analysis, the Clinometer X,Y functions were used. The following graphs were obtained with the clinometer in a stationary state for the X and Y axes, respectively.

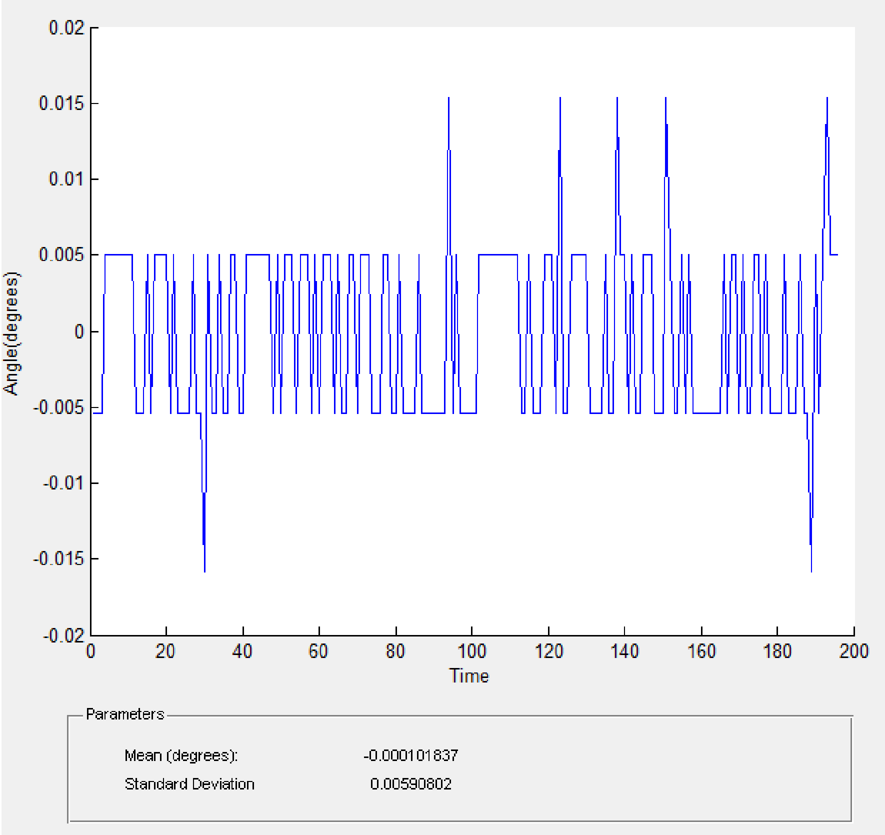


Parameters

Mean (degrees): -0.00124917

Standard Deviation 0.00681831

Figure 46 - Clinometer X Output



Mean (degrees): -0.000101837

Standard Deviation 0.00590802

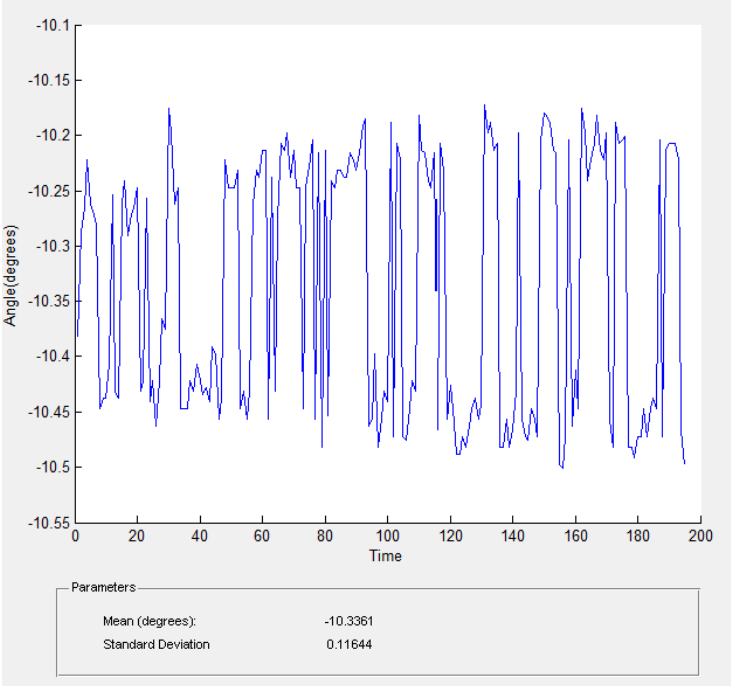
Figure 47 - Clinometer Y Output

The RMS parameters for the X and Y clinometers are given in the table below:

Output	RMS
Clinometer X	0.006818
Clinometer Y	0.005908

7.3: Magnetometer Noise Analysis

For the magnetometer noise analysis, the Magnetometer Z from the PADS Analyzer function was used. In order to calculate a heading compass from the values of the magnetic field in the X and Y direction is used. The graph shown in Figure 48 was obtained for the magnetometer in stationary state.



Parameters

Mean (degrees): -10.3361

Standard Deviation 0.11644

Figure 48 - Magnetometer Z Output

The RMS parameter for the Magnetometer Z function is shown in the table below:

Output	RMS
Magnetometer Z	0.1164

The high noise and standard deviation obtained were, in part, due to the not complete magnetic-Isolated place used during the tests.

Chapter 8: **Virtual Gyroscope**

In order to improve the measurement system accuracy, several MEMS gyroscopes were combined to create a virtual gyroscope with a higher resolution. This approach was originally proposed by Bayard in [2]. The virtual gyroscope technology fuses several MEMS gyroscopes and uses signal processing to improve the accuracy of the sensors. This technology takes advantage of the fact that MEMS sensors can be easily fabricated in a large number, in a single wafer with a relative low cost. Some simulations results have shown that by using four single gyroscopes, with a correlation factor of -0.333, the gyros drift could be divided by about 139 times [14]. Improving the accuracy by improving the mechanical sensing design has demonstrated not to be very easy and efficient, so new methods have been proposed in the last years. Creating virtual sensors by combining multiple sensing units has demonstrated to be a very efficient technique and it has been employed in many different engineering areas [15]. In 1992 Weis and Allan created a smart clock, by combining three inexpensive wrist watches with an error of 40 seconds a month, and achieved an error of 1 second a month [14].

8.1: Correlation between MEMS sensors

Bayard in [2] has shown that combining independent sensors (i.e. with no correlation between each other) can improve the accuracy by 1/sqrt(N), where N is the number of sensors. If there is correlation between sensors, this accuracy can be improved even further. Correlation could be created by fabricating the sensors in a single silicon chip with a same readout. In this project, since commercial-off-the-shelf components are being used, the selected MEMS gyroscopes are assumed to be uncorrelated.

8.2: Virtual Gyro Kalman Filter

A Kalman filter algorithm was used to optimally combine the measurements provided by the single gyroscopes into a measurement with higher precision. The state of the system was treated as the true angular rate w and the drift values b_i and the state model equations were established as follows:

$$\dot{x} = Fx(t) + Bu(t) + w(t)$$

$$z(t) = Hx(t) + v(t)$$

Where w(t) = [n_{b1} n_{b2} n_{b3} n_w] and v(t) = [n_{a1} n_{a2} n_{a3}] are the process noise and the measurement noise with covariances given by the matrixes Q and R, respectively:

$$p(w_k) \sim N(0, Q)$$

$$p(v) \sim N(0, R)$$

For the following analysis, a virtual gyroscope composed by three gyroscopes will be considered. More gyroscopes can be added to the virtual gyroscope by expanding the equations that will be presented in this chapter.

The state of the system X(t) is composed by the drift rate bias of each gyroscope and by the true angular rate w.

$$X = \begin{bmatrix} b1 \\ b2 \\ b3 \\ w \end{bmatrix}$$

The virtual gyroscope transition matrix can be then modeled, according to [15], by:

$$F = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The process noise is given by the matrix Q:

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

In this model, the bias drifts derivatives and variances are assumed to be negligible and the true angular rate derivative variance was assumed v_w=1.

The observation model is given by:

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

The measurement noises are assumed to be independent and with a variance of about 0.0009, given by the noise analysis performed in chapter 7. The measurement noise matrix R is, then, given by:

$$R = \begin{bmatrix} 0.0009 & 0 & 0 \\ 0 & 0.0009 & 0 \\ 0 & 0 & 0.0009 \end{bmatrix}$$

The linear space model described above can be discretized according to [16] by the following equations:

$$x_{k+1} = F_d x_k + B_d u_k + w_k$$

$$y_k = H x_k + v_k$$

The discretized equations, assuming a first order approximation for a small time step T are shown below.

$$F_d = e^{FT} \approx I + FT$$

$$B_d = F^{-1}(F_d - I)B = TB$$

$$Q_d = Q; R_d = R; C_d = C; D_d = D; R_d = R; Q_d = Q$$

The final approximate solution, then, is given by:

$$x_{k+1} = (I + FT)x_k + TBu_k$$

$$y_k = Hx_k + v_k$$

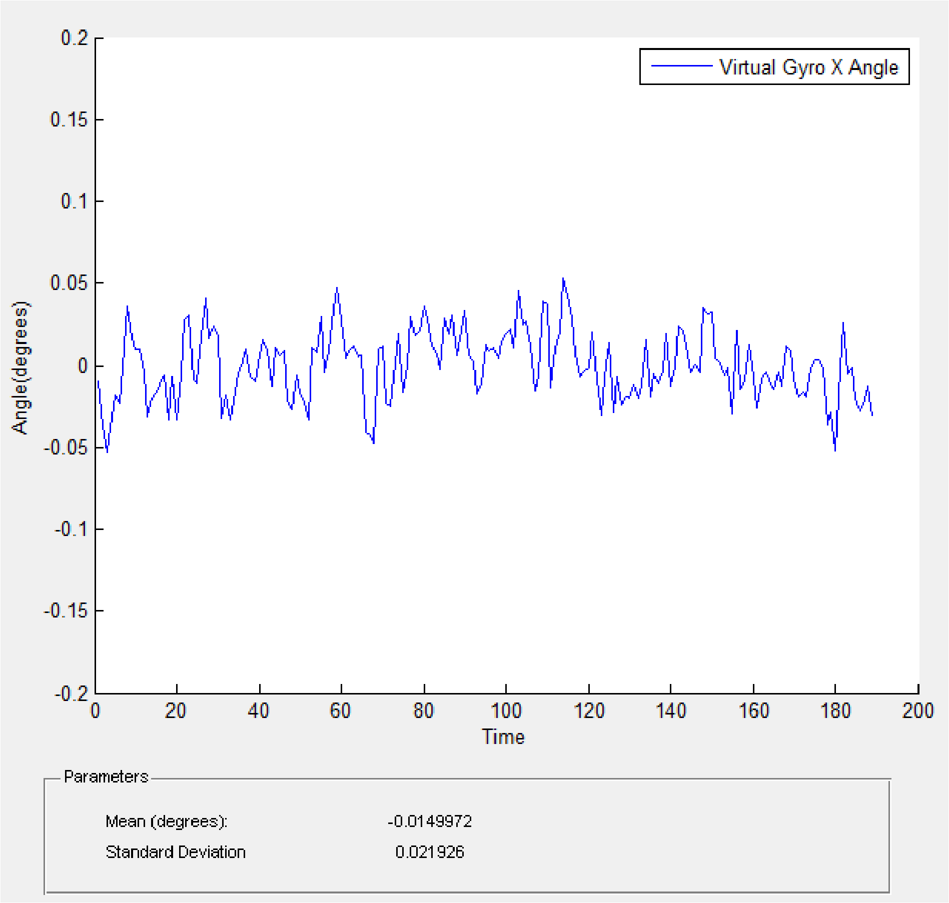
The following matrixes are given by applying the discretization model to the gyroscope state model:

$$F_d = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$B_d = T_s \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

8.3: Virtual Gyro Implementation Results

The virtual gyroscope was implemented by the Virtual Gyroscope function of the PADS Analyzer. The graph below is a comparison of the output provided with a single gyroscope and the one provided by the virtual gyroscope.



Parameters

Mean (degrees): -0.0149972

Standard Deviation 0.021926

Figure 49 - Virtual Gyro X Output

In the table below there is a comparison between the RMS values of the outputs provided by single gyroscopes and the one provided by the virtual gyroscope.

Gyroscope Output	Gyroscope 1 RMS	Gyroscope 2 RMS	Gyroscope 3 RMS	Virtual Gyroscope RMS
X	0.03631	0.03613	0.03685	0.02193
Y	0.03120	0.03157	0.03111	0.02233
Z	0.03956	0.03987	0.03992	0.02691

The following reduction factors were achieved by using three gyroscopes into the virtual gyroscope concept:

Gyroscope Output	Reduction Factor
X	1.6612
Y	1.4014
Z	1.4784

Chapter 9: **Extended Kalman Filter Data Fusion Algorithm Implementation**

In order to estimate the current attitude of the system an extended Kalman filter algorithm was implemented. The algorithm combines the data from the different sensors and provides an estimate for the current attitude.

9.1: Predict Phase Implementation

The gyroscopes sensors are used to predict the state of the measurement until a new measurement from the clinometers or the magnetometer is available. The angular rates provided by the gyroscopes are relative to their own body reference frame, thus a conversion is needed to provide attitude estimation in relation to an inertial frame of reference. For this convention, Euler angles are used to this conversion. The Euler rotation angles use the ZYX convention. Then, the rotation matrix can be expressed as:

$$R = \begin{bmatrix} c\theta c\psi & -c\phi s\psi + s\theta c\psi s\phi & s\phi s\psi + c\phi s\theta c\psi \\ c\theta s\psi & c\phi c\psi + s\theta s\phi s\psi & -c\psi s\phi + c\phi s\theta s\psi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix}$$

Where c(.) and s(.) denote cosin and sin functions, respectively.

The Euler angular rates can be then expressed, according to [17], as:

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & tan(\theta)sin(\phi)) & tan(\theta)cos(\phi)) \\ 0 & cos(\phi) & -sin(\phi) \\ 0 & \frac{sin(\phi)}{cos(\theta)} & \frac{cos(\phi)}{cos(\theta)} \end{pmatrix} \begin{pmatrix} w1 - b1 \\ w2 - b2 \\ w3 - b3 \end{pmatrix}$$

Where w1, w2 and w3 are the angular velocities in relation to the gyroscope body frame reference. The biases drift are negligible, therefore, are assumed to be zero.

$$\dot{b}1 = 0 \; \dot{b}2 = 0 \; \dot{b}3 = 0$$

A simple linear integration, then, was used to predict the state until a new measurement is available.

$$\phi = \phi + \dot{\phi}dt \ ; \ \theta = \theta + \dot{\theta}dt \ ; \psi = \psi + \dot{\psi}dt$$

Since the gyroscopes are significantly faster than the other sensors in the system, the state is propagated several times until an update occurs.

In order obtain a linear model for the extended Kalman filter algorithm, the model is linearized around its current estimate.

$$\dot{\phi} = \overline{\dot{\phi}} + \frac{\partial \dot{\phi}}{\partial \phi}(\phi - \overline{\phi})$$

The higher terms are ignored, since it is assumed that ϕ value is close to the linearization point [12]. Expanding this equation to all the angles and considering the partial derivatives will lead to the following matrix:

$$F = I + MT_s$$

Where T_s is the sampling time and M is defined as:

$$M = \frac{\partial(\dot{\phi}, \dot{\theta}, \dot{\psi})}{\partial(\phi, \theta, \psi)}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = M \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}$$

The matrix M is given, by applying the partial derivatives, by:

$$M = \begin{bmatrix} w2\, tan\theta\, cos\phi - w3\, tan\theta\, sin\phi & w2\frac{cos\phi}{cos^2\theta} + w2\frac{cos\phi}{sin^2\theta} & 0 \\ -w2\, sin\phi - w3\, cos\phi & 0 & 0 \\ w2\,\frac{cos(\phi)}{cos(\theta)} - w3\,\frac{sin(\phi)}{cos(\theta)} & w2\,\frac{sin(\phi)sin(\theta)}{cos^2(\theta)} + w3\,\frac{cos(\phi)sin(\theta)}{cos^2(\theta)} & 0 \end{bmatrix}$$

The observation matrix is defined as:

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The predicted error covariance can be calculated by:

$$P_{k+1|k} = F_k P_{k|k} F_k^T + Q$$

Where the process noise matrix Q_k was defined as:

$$Q = \begin{bmatrix} \sigma_\phi^2 & 0 & 0 \\ 0 & \sigma_\theta^2 & 0 \\ 0 & 0 & \sigma_\psi^2 \end{bmatrix}$$

Where σ_i is the noise associated to the prediction process leaded by the gyroscopes. This variance can be estimated by using the gyroscope angle random walk value which is about 0.03 deg/sec$^{0.5}$. Considering a time between samples of about 0.01 seconds, the variance is estimated to have a value of 0.003 degrees. Thus the matrix Q is defined as:

$$Q = \begin{bmatrix} 0.003 & 0 & 0 \\ 0 & 0.003 & 0 \\ 0 & 0 & 0.003 \end{bmatrix}$$

9.2: Update Phase Implementation

In the update phase, the values of the angles provided by the clinometer and magnetometer are combined with the *a priori* estimate provided by the gyroscopes, in order to estimate the current attitude. The sensors noise matrix R is defined as:

$$R = \begin{bmatrix} \sigma_\phi^2 & 0 & 0 \\ 0 & \sigma_\theta^2 & 0 \\ 0 & 0 & \sigma_\psi^2 \end{bmatrix}$$

Where σ_ϕ and σ_θ are the standard deviation from the clinometers and σ_ψ is the standard deviation value of the magnetometer. These values are estimated, according to the analysis made in section 7.1, to be:

$\sigma_\theta = 0.000036$; $\sigma_\phi = 0.000036$ and $\sigma_\psi = 0.01$.

The innovation covariance matrix and the Kalman gain are respectively, calculated by:

$$S_{k+1} = H P_{k+1|k} H^T + R$$

$$K_{k+1} = P_{k+1|k} H^T S_{k+1}^{-1}$$

The *a posteriori* covariance matrix and the *a posteriori* state are, respectively, given by:

$$P_{k+1|k+1} = (I - K_{k+1} H_{k+1}) P_{k+1|k}$$

$$x_{k+1|k+1} = x_{k+1|k} + K_{k+1} y_{k+1}$$

9.3: Extended Kalman Filter Results

In Figure 50, a single output from the Kalman X,Y,Z function is shown. The orientation state is predicted by the gyroscopes until a new measurement is available. In the figure below Kalman filter updates occurs every 10 time units.

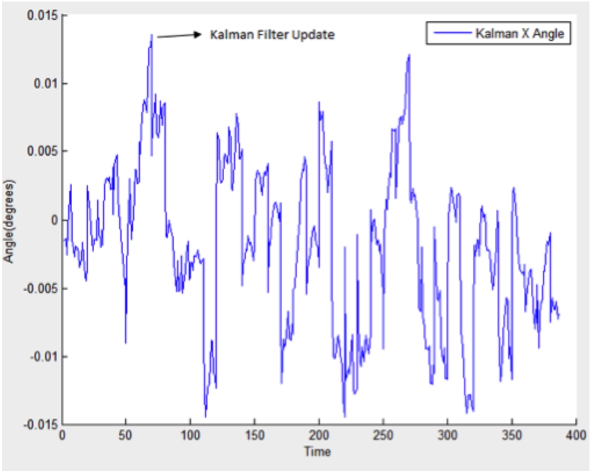


Figure 50 - Extended Kalman Filter result over the X axis

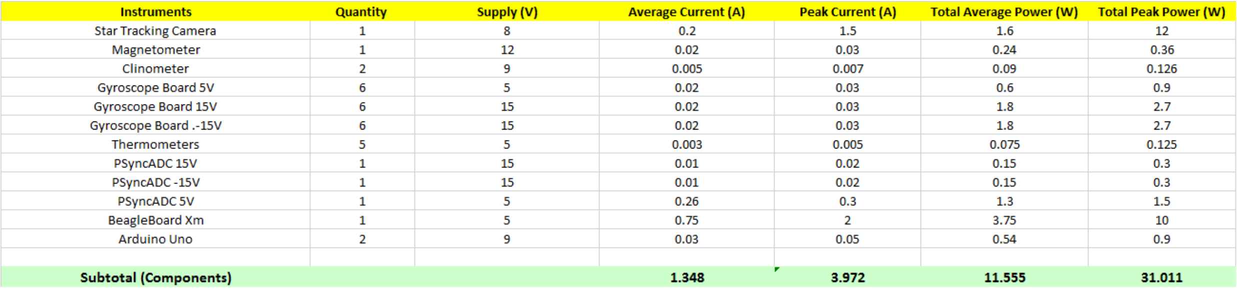
A standard deviation of about 0.005 degrees was achieved by combining a set of 3 uncorrelated MEMS gyroscopes and clinometers over the X and Y axes.

Chapter 10: **Power Budget**

In this chapter, the power budget of the system is going to be discussed. The following power estimations were based in calculations, observations and in components specifications. The budget was divided into three parts: Instruments, Voltage Regulators and Heaters. Each of these parts is going to be discussed separately:

10.1: Instruments

The instruments include all the system sensors, the acquisition board and the controllers. The estimated power consumption for these elements is shown in the table below:



Instruments	Quantity	Supply (V)	Average Current (A)	Peak Current (A)	Total Average Power (W)	Total Peak Power (W)
Star Tracking Camera	1	8	0.2	1.5	1.6	12
Magnetometer	1	12	0.02	0.03	0.24	0.36
Clinometer	2	9	0.005	0.007	0.09	0.126
Gyroscope Board 5V	6	5	0.02	0.03	0.6	0.9
Gyroscope Board 15V	6	15	0.02	0.03	1.8	2.7
Gyroscope Board .-15V	6	15	0.02	0.03	1.8	2.7
Thermometers	5	5	0.003	0.005	0.075	0.125
PSyncADC 15V	1	15	0.01	0.02	0.15	0.3
PSyncADC -15V	1	15	0.01	0.02	0.15	0.3
PSyncADC 5V	1	5	0.26	0.3	1.3	1.5
BeagleBoard Xm	1	5	0.75	2	3.75	10
Arduino Uno	2	9	0.03	0.05	0.54	0.9
Subtotal (Components)			**1.348**	**3.972**	**11.555**	**31.011**

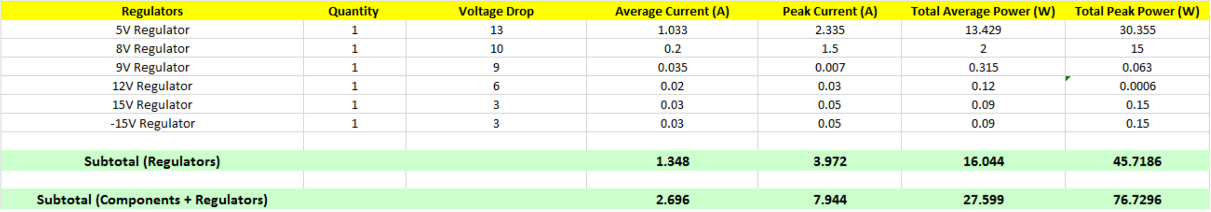
The flight computer (Beagle Board xM) is the element that requires most of the power followed by the Analog Acquisition Board (PsyncADC) and the Star Tracking Camera. The flight computer has a peak of 2 amperes when booting and the camera demands a peak current of 1.5 amperes when taking pictures. The instruments total average power is estimated to be 11.55 W, with a peak of 31.01 W.

10.2: Voltage Regulators

Voltage regulators have a good significance on the system power budget since they dissipate a big amount of power. The power dissipated by each regulator was estimated by taking into account the voltage drop and the current passing through the regulator, by applying the following relation:

$$P_{dissipated} = V_{drop} I$$

The estimated power dissipation for the voltage regulators is showed in the table below.

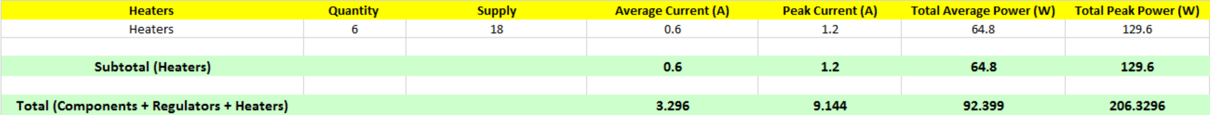


Regulators	Quantity	Voltage Drop	Average Current (A)	Peak Current (A)	Total Average Power (W)	Total Peak Power (W)
5V Regulator	1	13	1.033	2.335	13.429	30.355
8V Regulator	1	10	0.2	1.5	2	15
9V Regulator	1	9	0.035	0.007	0.315	0.063
12V Regulator	1	6	0.02	0.03	0.12	0.0006
15V Regulator	1	3	0.03	0.05	0.09	0.15
-15V Regulator	1	3	0.03	0.05	0.09	0.15
Subtotal (Regulators)			**1.348**	**3.972**	**16.044**	**45.7186**
Subtotal (Components + Regulators)			**2.696**	**7.944**	**27.599**	**76.7296**

The average power dissipated by the voltage regulators is estimated to be 16.044 W with a peak power of 45.72 W.

10.3: Heaters

Heaters are some of the most power demanding components. They help on stabilizing the temperature during harsh flight conditions in order to maintain the sensors in normal operation modes. The power required for them was based in thermal calculations, which are not in the scope of this project. The table below shows the power required by the heating system.



Heaters	Quantity	Supply	Average Current (A)	Peak Current (A)	Total Average Power (W)	Total Peak Power (W)
Heaters	6	18	0.6	1.2	64.8	129.6
Subtotal (Heaters)			**0.6**	**1.2**	**64.8**	**129.6**
Total (Components + Regulators + Heaters)			**3.296**	**9.144**	**92.399**	**206.3296**

The total power estimated for the system has an average of 92.40 W and a peak power of 206.33 W.

Chapter 11: **Conclusions**

In this project, a relative low-cost attitude determination system for high altitude scientific balloons has been developed by taking advantage of different sensors capabilities and performances in a hierarchical manner. Using multiple inexpensive MEMS gyroscopes has proven to increase significantly the precision of the system. Furthermore, an attitude precision in the order of arc seconds was achieved by combining sensors through an Extended Kalman Filter Algorithm. In the future, high-correlated MEMS gyroscopes can be used along with a study of their covariance, which has the potential to increase dramatically the precision of the system.

Bibliography:

[1] D. Li, R. J. Landry, and O. Lavoie, "Low-cost MEMS Sensor-based Attitude Determination System by Integration of Magnetometers and GPS: A Real-data Test and Performance Evaluation," in 2008 IEEE/ION Position, Location and Navigation Symposium, 2008, pp. 342-350.

[2] Bayard, D.S.; Ploen, S.R, “High accuracy inertial sensors from inexpensive components”, US Patent. US20030187623A1, 2003.

[3] Luttwak, A. (2011), “Human motion tracking and orientation estimation using inertial sensors and RSSI measurements”, M.Sc. Thesis. The Hebrew University of Jerusalem: Israel.

[4] H. L. Chang, L. Xue, W. Qin, G. M. Yuan, and W. Z. Yuan, “An integrated MEMS gyroscope array with higher accuracy output,” in Proc. Sensors, 2008, vol. 8, pp. 2886–2899.

[5] Rotteveel, J. ; Le Mair, A. ; American Institute of Aeronautics and Astronautics, “Multi-Aperture Miniaturized Star Sensors, Modular Building Blocks for Small Satellite AOCS Systems”, Conference; 23rd, Small Satellites, 2009

[6] Hall, C. , “Chapter 4: Attitude Determination”, Course Notes, 2003

[7] National Aeronautics and Space Administration (NASA), “Scientific Balloons”, Available at: http://sites.wff.nasa.gov/code820.

[8] Honeywell, “3-Axis Magnetic Sensor Hybrid HMC2003”, Honeywell. "3-Axis Magnetic Sensor Hybrid HMC2003." Honeywell, n.d. Web. 3 Aug. 2012.

[9] Ozyagcilar, T. (2012), “Implementing a tilt-compensated eCompass using accelerometer and magnetometer sensors”. Freescale semiconductor, AN 4248, Rev. 3.

[10] M. Grewal and A. Andrews, “Applications of Kalman filtering in aerospace 1960 to the present,” IEEE Control Syst. Mag., vol. 30, no. 3,pp. 69–78, Jun. 2010

[11] Gary Bishop and Greg Welch, "An Introduction to the Kalman Filter". Universityy of Norh Carolina SIGGRAPH 2001 course notes. ACM Inc., North Carolina, 2001.

[12] Huang, Shoudong, “Understanding the Extended Kalman Filter”, ARC Centre of Excellence for Autonomous Systems (CAS), University of Technology Sydney, April. 2010.

[13] Simon, D. “Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches”, Wiley-Interscience, 2006

[14] Honglong Chang, Liang Xue, Chengyu Jiang, Michael Kraft, Weizheng Yuan, “Combining Numerous Uncorrelated MEMS Gyroscopes for Accuracy Improvement Based on an Optimal Kalman Filter”. IEEE T. Instrumentation and Measurement 61(11): 3084-3093 (2012)

[15] Chang H., Xue L., Qin W., Yuan G., Yuan W, “An integrated MEMS gyroscope array with higher accuracy output”, Sensors. 2008;8:2886–2899.

[16] Brown, R.G., Hwang, P.Y.C., “Introduction to random signals and applied Kalman filtering”, 3. ed. New York: John Wiley & Sons, 1985

[17] Roni, Yadlin, “Attitude Determination and Bias Estimation Using Kalman Filtering”, United States Air Force Academy, 2010.