Master's Thesis - Projet de Fin d'Études

Control System of the Balloon Experimental Twin Telescope for Infrared Interferometry

Author:
Marc Casalprim Torres
UPC-ETSETB
ISAE-SUPAERO

Supervisor:
Dr. Stephen Rinehart
NASA Goddard Space Flight Center

September 2017

Contents

Glossary **vii**

Introduction **1**

1 BETTII **3**

1.1 Design . . . 3

1.1.1 Interferometry . . . 4

1.2 Sensors . . . 5

1.2.1 Star cameras . . . 5

1.2.2 Gyroscopes . . . 7

1.3 Control system . . . 7

1.3.1 Elevation actuators . . . 8

1.3.2 Azimuth actuators . . . 8

1.3.3 Attitude estimation . . . 10

1.4 Commands and Telemetry system . . . 15

2 Flight Campaign **17**

2.1 Data analysis tools development . . . 18

2.2 Pre-flight tests . . . 20

2.2.1 Inertia value measurement . . . 21

2.3 Flight . . . 22

3 Flight Data **25**

3.1 Gyroscopes . . . 25

3.2 Momentum dump . . . 26

3.3 Attitude estimator . . . 28

3.3.1 Covariance matrix 28

3.3.2 Post-flight estimation 30

Conclusions and future work **35**

A Equatorial coordinate system **37**

B Flight plots **39**

C Delayed star camera solution **45**

D Kalman filter Python code **47**

Bibliography **51**

List of Figures

1.1 CAD design of BETTII 4

1.2 Dewar, the cryogenic instrument. 4

1.3 Interferogram of a sum of cosine waves at different frequencies. 5

1.4 Left star camera of BETTII. 6

1.5 Example of an image taken by a star camera, with the solution found by the Astrometry.net software printed on. 6

1.6 The three SRS-2000 gyroscopes mounted in an orthogonal assembly. 7

1.7 Compensated controlled moment gyroscopes. 8

1.8 Galil Motion control boards. 9

1.9 Momentum Dump mechanism. 9

1.10 Coordinate systems relevant to the pointing control system. 10

1.11 Ford and cRIO computers. 15

1.12 Main GUI of the Aurora software. 16

2.1 BETTII balloon launch. 18

2.2 A KST session showing plots of the BETTII control system. 19

2.3 Real time signal processing of the detectors. Developed for the scientific team of BETTII. The data shown in the figures is taken with bad conditions and is not useful. 20

2.4 Indoors and outdoors hanging tests of BETTII 21

2.5 Inertia value measurement 22

2.6 Measured azimuth velocity. Left, information from the telemetry received during flight. Right, information extracted from the disks after the flight. 23

3.1 Power spectral density of the three gyroscopes during flight 25

3.2 Signal from the gyroscopes. 26

3.3 Commands of the momentum dump during the braking of the payload. 27

3.4 Measured azimuth velocity during the braking. 27

3.5 Estimated attitude. 28

3.6 Evolution of the first three diagonal elements of the matrices $\boldsymbol{P}$ and $\boldsymbol{P_{rot}}$ . . . 29

3.7 Evolution of the uncertainties in the attitude of the telescope. 30

3.8 Estimated attitude post-flight. 31

3.9 Estimated biases post-flight. 31

3.10 Estimated attitude post-flight. 32

3.11 Post-flight attitude uncertainties. 33

A.1 BETTII reference frames on the equatorial coordinate system. 37

B.1 Measured velocities from the three gyroscopes, in arcesc/s. 39

B.2 Detail of the gyroscopes at the same time window of the estimator figures 3.5 and B.4 40

B.3 Detail of the gyroscopes at the moment of launch. 40

B.4 Estimated gyroscope biases, in arcsec/s. 41

B.5 Innovations of the estimator. 41

B.6 Measured altitude from the GPS. 42

B.7 Azimuth and Pitch measured by the magnetometer HMR3300. 42

B.8 Measured currents. 43

B.9 Unwrapped azimuth measured by the magnetometer. 43

B.10 Temperatures during flight. 44

List of Tables

1.1 Quaternions involved in the telescope attitude calculation. Siderostats' elevation angle of 60 degrees 12

1.2 Quaternions involved in the telescope attitude. Siderostats' elevation angle of 60 degrees. Using the equatorial representation, in degrees. 12

3.1 Estimated noise uncertainties of the three gyroscopes during the flight, in arcsec/s. 26

Glossary

ARW	*Angular Random Walk*
BETTII	*Balloon Experimental Twin Telescope for Infrared Interferometry*
CAD	*Computer-Aided Design*
CCMG	*Compensated Controlled Moment Gyroscopes*
CIP	*Consolidated Instrument Package*, communications device provided by CSBF
CPU	*Central Processing Unit*
CRIO	*Compact Reconfigurable I/O*
CSBF	*Columbia Scientific Balloon Facility*
DC	*Direct Current*
DEC	*Declination*
FIR	*Far-infrared*, 0.7-2.5 μm
FPGA	*Field Programmable Gate Array*
GPS	*Global Positioning System*
GSFC	*Goddard Space Flight Center*
JWST	*James Webb space Telescope*
NASA	*National Aeronautics and Space Administration*
NIR	*Near-infrared*, 30-200 μm
OPD	*Optical Path Difference*
PID	*Proportional-Integral-Derivative*, common controller architecture
PSD	*Power Spectral Density*
RA	*Right Ascension*
TCP	*Transmission Control Protocol*

Introduction

In this thesis, we will study the control system of an astronomical balloon-borne telescope called BETTII: Balloon Experimental Twin Telescope for Infrared Interferometry. The BETTII project is a collaboration between NASA, the University of Maryland, Johns Hopkins University, Cardiff University, and University College London, with assistance from the Far-Infrared Telescope Experiment team in Japan. Developed at NASA's Goddard Spaceflight Center (GSFC), this instrument is implementing the "Double-Fourier" interferometry, a new observation technique that could lead to future space telescopes with high angular resolution in the far-infrared spectrum.

At infrared wavelength, observations from the Earth's surface are limited by the low atmospheric transmission. Observations at this region of the spectrum have a high scientific value, but the resolution of space-based telescopes is limited by the cost and complexity of building and flying progressively larger aperture telescopes. High altitude platforms are a good compromise between ground and space observatories. They can feature larger optics than space-based observatories at a reduced cost, but they are also less sensitive, due to the surrounding thermal emission from the atmosphere and their components.

BETTII is a high-altitude observatory in a balloon platform that, instead of using a single-aperture like a conventional telescope, uses interferometry between 30 and 110 μm. It has a cryogenic payload and it will attempt to study clustered star formation in nearby star clusters.

This document is the result of a 5 months experience at NASA-GSFC and CSBF between April and September of 2017 and is centered around the data collected from BETTII's first flight. Due to the nature of a flight campaign, several different tasks have been conducted during this stay including hardware work, electrical debugging and software developing. The structure will be as follows: first, there is a brief description of BETTII and its different subsystems. Secondly, the events during the launch campaign are described, giving some examples of the problems that arose during and before the flight. Then, some useful information is processed and shown from the flight data. Finally, some propositions of future work and conclusions are presented.

Chapter 1

The Balloon Experimental Twin Telescope for Infrared Interferometry

The Balloon Experimental Twin Telescope for Infrared Interferometry (BETTII) will provide access to the cosmos, similar to that of an orbiting observatory. Flying on a high altitude balloon at 40 km (130 kft), BETTII will explore a region of the electromagnetic spectrum, the far-infrared (FIR), using a technique called interferometry.

BETTII is an 8-meter interferometer which operates at wavelengths of 30-110 μm on a high-altitude balloon. This long baseline will allow unprecedented angular resolution (~ 0.5") in this band, and the high atmospheric transmission at balloon altitudes will allow the unique double-Fourier instrument on BETTII to obtain a high spectral resolution, $R \equiv \frac{\lambda}{\Delta\lambda} \sim 100$. The combination of these capabilities will provide spatially resolved spectroscopy on astrophysically important sources, exploring the physical processes that lurk below the resolution limits of current FIR facilities.

This project is a first step toward the coming era of space-based infrared interferometry, enabling a technology that will transform astronomy and astrophysics. Data acquired with BETTII will be complimentary to observations with space observatories such as Herschel and the James Webb Space Telescope, exploring the FIR wavelength range with unprecedented high angular resolution. These data will be a powerful tool for understanding star formation in clusters, and with future flights, for understanding active galactic nuclei and the late stages of stellar evolution.

1.1 Design

The whole payload consists of a long 8 meters trust, built of 2 meters carbon fiber tubes, sitting on a gondola made of aluminum bars. The goal is to receive and combine two beams of light reflected from two spatially separated mirrors. These siderostat mirrors, controlled in elevation by two motors, are located on the two extremes of the trust. The reflected beams are, lately, compressed in a 20:1 ratio by two specially built telescopes. After the telescope, the light passes through a set of mirrors, whose goals are to adjust the phase difference between the beams and center the targets on the detectors. The science sensors are located inside the cryogenic instrument (also referred in this document as Dewar), using Helium and Nitrogen in order to cool down the detectors at less than 4 Kelvin. Thus, it achieves good enough signal conditions in the far-infrared spectrum.

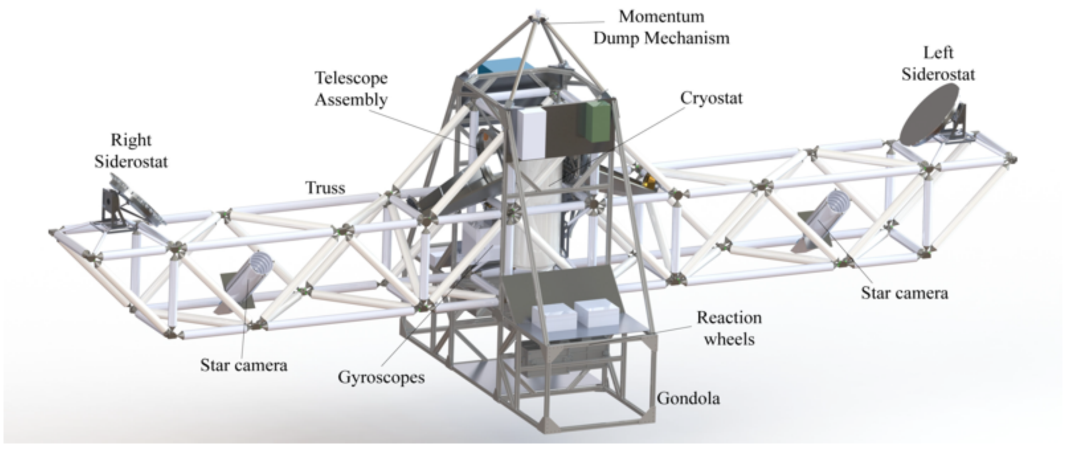


Figure 1.1: CAD design of BETTII

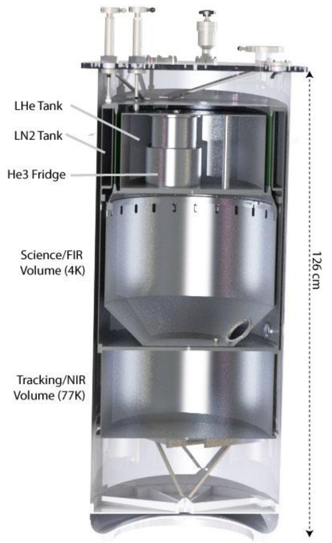


Figure 1.2: Dewar, the cryogenic instrument.

1.1.1 Interferometry

Because light behaves like a wave, the intensity of the two light beams combined depends on the brightness of the source and the relative phase of each beam. Changes in this relative phase create a modulation of that intensity. This modulation is called an interferogram, which describes the measured intensity variation as a function of the phase difference between the two beams. In this work, however, instead of expressing the phase difference in radians we will express it in terms of physical distance (optical path difference, OPD) to relate more easily to opto-mechanical considerations and avoid the dependence on the wavelength.

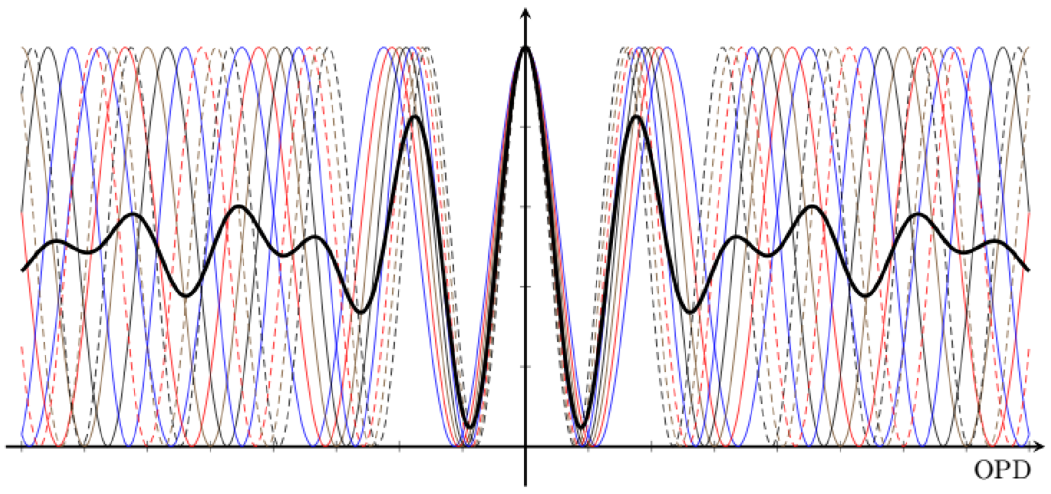


Figure 1.3: Interferogram of a sum of cosine waves at different frequencies.

In this project, there are two delay lines, a warm delay line and a cold delay line. They are a set of moving mirrors that introduce a path difference between the two beams of light. The warm delay line is outside the cryogenic chamber; its main goal is to center the interferogram and correct any optical delays caused, for example, by a residual misalignment of the telescope axis with respect to the source. The cold delay line has a ten times higher frequency control loop and is located inside the Dewar. It will generate a scan of optical path differences in order to obtain the desired interferogram. By scanning the OPD, we obtain a modulation of each pixel on the detector. We get the spectrum of the source performing a Fourier transform of the interferogram but we can also obtain the spatial information of the sources looking at their amplitudes and phases. That is why it is called a double Fourier interferometry, because we obtain information on both the spectral and the spatial characteristic of the source.

The two siderostat mirrors have a diameter of 50 cm and are separated by 8m. Theoretically, the resulting angular resolution is about 0.5" for the 30-55 μm band and ~1" for the 55-110 μm band. This resolution is much better than the existing telescopes that operate also in the far-infrared, which are limited by the mirror size. For reference, the James Webb Space Telescope will achieve the same resolution but for shorter wavelengths around the 25 μm band.

1.2 Sensors

1.2.1 Star cameras

BETTII uses two star cameras located on both arms and pointing to 45 degrees of elevation. They measure the payload orientation. The star camera software, provided by the Cardiff University, solves from a given picture the inertial orientation and also gives the uncertainties of this three measurements. The lenses used provide low chromatic aberration, a wide field of view ($\sim 10°$) and a collecting area of 90 mm^2. However, they do not have an auto-focus feature, that is why a specially designed and remotely controlled auto-focus mechanism has been implemented. This auto-focus system will allow us to correct any unfocusing due to, for example, a contraction of the lens caused by the low temperatures. Later on, during the flight campaign, an alternative camera solution finder software called Astrometry.net was implemented in the system. Astrometry.net is an open-source and free to use software. From the ground, we can chose which software to use, as it was observed that they behave differently depending on the conditions. Other parameters such as the exposure time or the field of view are also commandable from the ground.

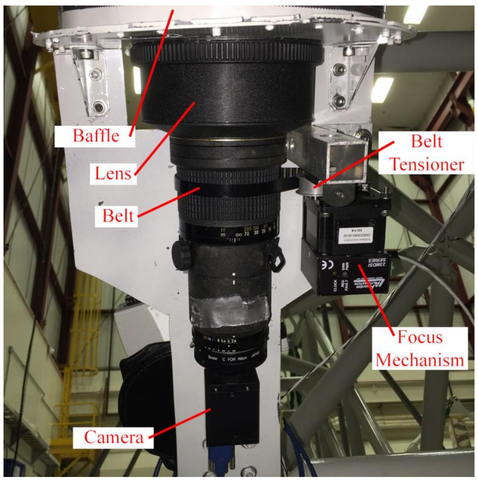
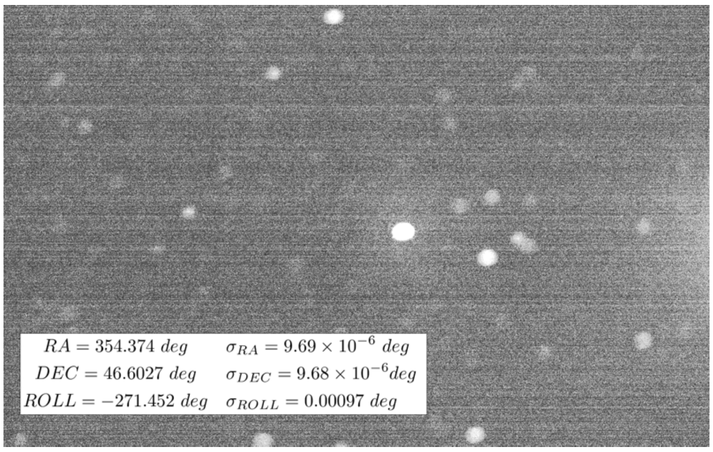


Figure 1.4: Left star camera of BETTII.



$RA = 354.374\ deg$ $\sigma_{RA} = 9.69 \times 10^{-6}\ deg$
$DEC = 46.6027\ deg$ $\sigma_{DEC} = 9.68 \times 10^{-6} deg$
$ROLL = -271.452\ deg$ $\sigma_{ROLL} = 0.00097\ deg$

Figure 1.5: Example of an image taken by a star camera, with the solution found by the Astrometry.net software printed on.

1.2.2 Gyroscopes

The gyroscopes used to measure the angular velocity of BETTII are three SRS-2000 Sagnac effect gyroscopes from Optolink. These devices offer a high precision and have very low angular random walk and biases. The gyrsocopes' bandwidth is 50 Hz but they can be read at a rate of 2000 Hz. Each gyroscope has a thermometer and a Peltier plaque that controls the temperature, since their performance depends on the temperature stability. The three gyroscopes were mounted in an orthogonal configuration, shown in figure 1.6. The

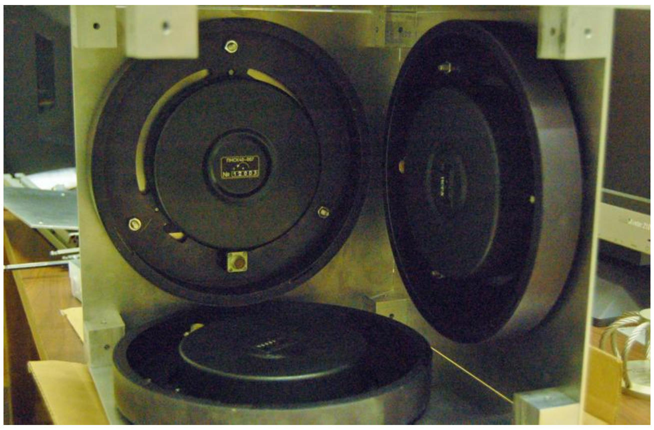


Figure 1.6: The three SRS-2000 gyroscopes mounted in an orthogonal assembly.

angular random walk (ARW) indicates the effect of integrating the noisy measure of the velocity. The specification from the manufacturer is $ARW = 5 \times 10^{-4} deg/\sqrt{h}$. This means that if we integrate the gyroscopes for 1 hour, the 1σ uncertainty on our position would be 5×10^{-4} $deg \sim 1.8''$. This effect varies as the square root of the integration time, for an integration of 1 second it would be $0.03''$, only 60 times less. There is also a bias drift, which the manufacturer defines as 0.005 deg/h at a fixed temperature.

1.3 Control system

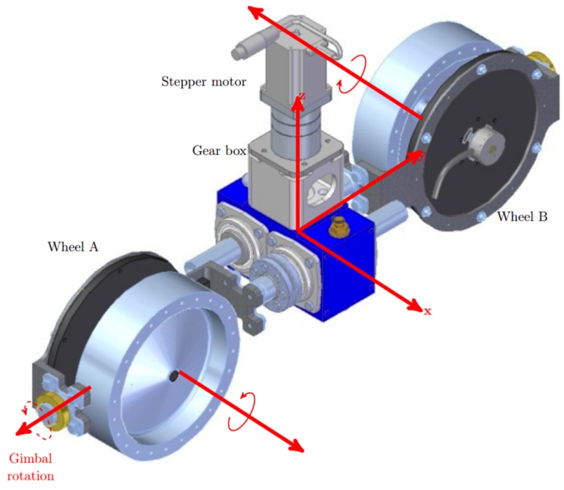
The goal of the control system is to use the information measured by the star camera and the gyroscopes sensors to maintain the telescope pointing at a desired attitude. The control loop consists of an attitude estimator that will be compared to the desired coordinates of a target star to compute the desired azimuth and elevation. Then, these azimuth and elevation are fed to two PID controllers. This system will be key to track any target star and will use three different types of actuators, one to control the elevation and two to control the azimuth. The control system runs in a 100Hz loop implemented in LabView for the on-board computer called Boop, described briefly in the section 1.4.

1.3.1 Elevation actuators

The elevation of BETTII's telescope is determined entirely by the angle of the two siderostat mirrors. This angle is changed by two industrial DC brushless rotators manufactured by Griffin Motion and they are controlled with a commercial controller that provides a high angular precision.

1.3.2 Azimuth actuators

In order to control the payload's azimuth, two actuators are used: a CCMG and a momentum dump mechanism. The Compensated Controlled Moment Gyroscopes (CCMG) is a system with two counter-rotating reaction wheels mounted on a gimbal. The wheels are controlled by a commercial controller from Galil Motion that operates a brushless DC motor together with a 13 bit encoder that measures the wheels' rotation. In addition, there is a stepper motor that moves the shaft's angle, also controlled by a Galil Motion controller. This latter device that controls the stepper motor operates an additional motor, the one from the momentum dump mechanism that we will see later.



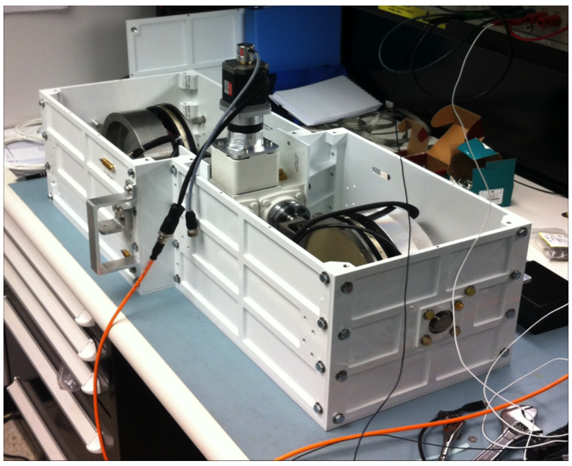


Figure 1.7: Compensated controlled moment gyroscopes.

During power-up, the wheels start accelerating until they reach a constant speed of 3000 rpm in about 10 minutes. This spinning velocity will be maintained during all the flight duration. At 3000 rpm, the CCMG wheels have a stored momentum of $M_{CCMG} = 20.8 Nms$. Depending on the angle θ between the horizontal axis and the rotation axis of the wheels, the momentum along the z axis will be a projection of the total stored momentum:

$$M_{CCMG,z} = 20.8\ sin\theta \tag{1.1}$$

When the rotation axes of the wheels are aligned with z ($\theta = 90°$), we have the maximum momentum. The torque τ_{CCMG} is the variation of the momentum with time:

$$\tau_{CCMG} = \frac{d}{dt} M_{CCMG,z} = 20.8\ \dot{\theta}\ \cos\theta \quad [Nm] \tag{1.2}$$

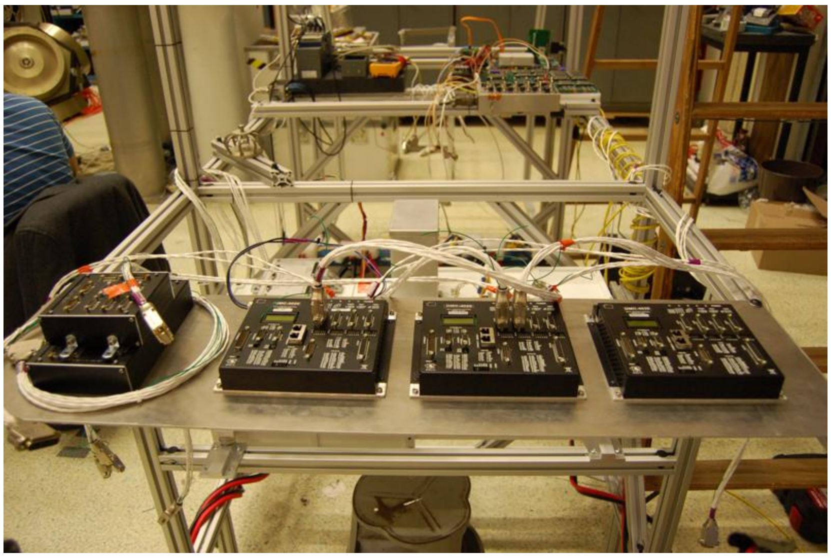


Figure 1.8: Galil Motion control boards.

The variation on the rotation axis angle is translated to a torque. We will always work with small angles, because at angles around 90 degrees the effect on the torque is practically nonexistent. This is the main function of the momentum dump, that uses the attachment to the balloon like a momentum storage. The principle behind the mechanism is to rotate a motor connected, via a double-bearing assembly, to the pin that supports all the payload structure.

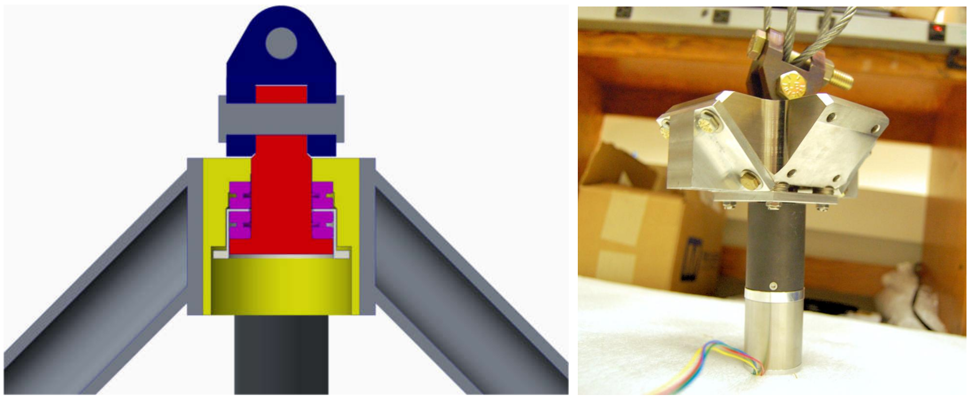


Figure 1.9: Momentum Dump mechanism.

When the payload is hanging, the friction transfers the rotation of the motor to the balloon train. Then, when the train is sufficiently twisted, a momentum unloading starts in the opposite direction. All this process occurs slowly because of the low friction of the bearings. The control loop of this part has been tuned to maximize the speed. To achieve a given momentum on the z axis, the momentum generated by the CCMG is calculated and added to the payload inertia:

$$M_z = M_{CCMG,z} \; + \; J_z \, \omega_z \tag{1.3}$$

where the inertia J_z is estimated with a procedure explained in the section 2.2.1 where the CCMG gimbal is moved at a constant rate.

1.3.3 Attitude estimation

The attitude of BETTII is estimated using the information from the star cameras and the gyroscopes. In case that the star cameras fail and we can not get an absolute measurement of the position, the system is designed to use alternatively the magnetometer and the GPS. Once we get the target star at the NIR detector, we use the tip/tilt mirrors position to estimate more accurately the attitude of the telescope. The information from the sensors is fused on an extended Kalman filter that will estimate the attitude and the gyroscopes' biases.

1.3.3.1 Reference frames

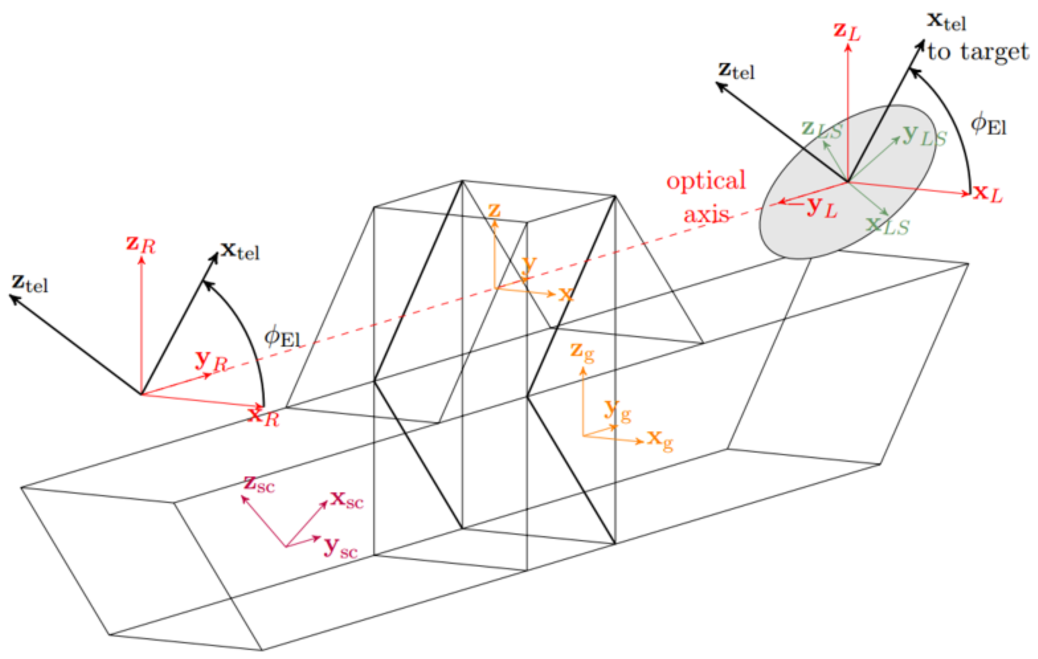

$\mathbf{z}_L$
$\mathbf{x}_{\mathrm{tel}}$
to target
$\mathbf{z}_{\mathrm{tel}}$
ϕ_{El}
$\mathbf{z}_{LS}$
$\mathbf{y}_{LS}$
optical axis
$\mathbf{y}_L$
$\mathbf{x}_L$
$\mathbf{x}_{LS}$
$\mathbf{z}$
$\mathbf{y}$
$\mathbf{x}$
$\mathbf{z}_R$
$\mathbf{x}_{\mathrm{tel}}$
$\mathbf{z}_{\mathrm{tel}}$
ϕ_{El}
$\mathbf{y}_R$
$\mathbf{x}_R$
$\mathbf{z}_{\mathrm{g}}$
$\mathbf{y}_{\mathrm{g}}$
$\mathbf{x}_{\mathrm{g}}$
$\mathbf{z}_{\mathrm{sc}}$
$\mathbf{x}_{\mathrm{sc}}$
$\mathbf{y}_{\mathrm{sc}}$

Figure 1.10: Coordinate systems relevant to the pointing control system. The subscripts sc, g, tel, L, R and LS indicate the star camera, gyroscope, telescope, left, right and left siderostat reference frames, respectively. The gyroscope reference frame is nominally aligned with the gondola reference frame, which has no subscript. Credit: Maxime Rizzo [Riz16]

It is important to set the key coordinate systems of BETTII before going into the details of the attitude estimation. We consider that the gondola reference frame G is the one defined by the gyroscopes (axes with 'g' subscript). The star cameras are nominally aligned with the gyroscope reference frame but with a rotation about y_g of -45 degrees. With the aid of laser trackers, a more precise rotation between the two reference frames had been found. The final goal of the control system is to point the detectors to a target star. The detectors will see the light reflected from the siderostats. The telescope reference frame T is defined as the result from the rotation of the gondola reference frame with the angle of the two siderostats. In this manner, the x_{tel} axis will represent the line of sight of our detectors.

Stars can be located using the equatorial coordinate system, in terms of right ascension (RA) and declination (DEC) coordinates. This celestial coordinate system is useful because the stars are practically fixed in this frame, and they do not change their coordinates during the observation time. That does not occur if we use Azimuth and Elevation coordinates referenced to the Local Horizon plane. The star cameras will also provide solutions in this equatorial coordinate system and that is the reason why we will use it as our inertial reference frame I to describe the attitude of the gondola. It is important to determine the inertial attitude in order to calculate the desired local azimuth ϕ_{Az} and elevation ϕ_{El} of the gondola. Ideally, starting with only the target's location in RA and DEC, the system will be able to control the entire gondola to within ±15" of the target. Once we achieve this goal, an additional fine pointing control using small mirrors will be done to center each beam to ±1.5" at the science detector.

1.3.3.2 Quaternions

To represent the attitudes and rotations of the different reference frames we use quaternions. Briefly, quaternions extend the complex numbers and are very useful to describe three-dimensional rotations. They have a real part q_r and three imaginary parts q_i,q_j,q_k. If the values of the quaternion fulfill

$$\boldsymbol{q} = \begin{bmatrix} q_i \\ q_j \\ q_k \\ q_r \end{bmatrix} = \begin{bmatrix} k_x \ \sin\theta/2 \\ k_y \ \sin\theta/2 \\ k_z \ \sin\theta/2 \\ \cos\theta/2 \end{bmatrix} \tag{1.4}$$

the quaternion $\boldsymbol{q}$ is called quaternion of rotation. A quaternion of rotation has unit norm, θ describes the angle of rotation around the axis defined by the vector $\boldsymbol{k} = [k_x, k_y, k_z]^T$.

The notation ${}^{G}_{I}\mathbf{q}$ represents a quaternion describing the rotation between the inertial reference frame I and the gondola reference frame G or, equivalently, the coordinates of the inertial attitude of the gondola in the equatorial coordinate system. In this manner, the quaternion ${}^{S}_{G}\mathbf{q}$ should describe the rotation of -45 degrees applied to the gondola reference frame to become the right star camera reference frame S. The rotations are applied using the quaternion multiplication. In this work we do not use the Hamilton multiplication but the "natural order" multiplication, in accordance with the JPL proposed Standard Conventions [Bre99].

As an example, if we want to calculate where the telescope is pointing to – described by the quaternion ${}^{T}_{I}\mathbf{q}$ – using only the measured attitude from the star camera software, the typical chain of rotations is as follows:

$${}^{T}_{I}\mathbf{q}^{meas} = {}^{T}_{G}\mathbf{q} \otimes {}^{G}_{I}\mathbf{q}^{meas} = {}^{T}_{G}\mathbf{q} \otimes {}^{G}_{S}\mathbf{q} \otimes {}^{S}_{I}\mathbf{q}^{meas} \tag{1.5}$$

The rotation ${}^{T}_{G}\mathbf{q}$ is a pure rotation about the y_g axis of the elevation angle applied to the siderostats. The rotations ${}^{T}_{G}\mathbf{q}$ and ${}^{G}_{S}\mathbf{q}$ are considered to be very precise, obtained by optical alignments and accurate angle measurements. In the tables 1.1 and 1.2 we can see the evolution of these rotations for a measurement of the star camera of $RA, DEC, ROLL = 100, 20, 30$ degrees. The approach used for the estimation of the telescope attitude is almost the same as shown in the previous example but obtaining a better estimation of the ${}^{G}_{I}\mathbf{q}$ quaternion, instead of using directly the solutions provided by the star cameras.

	${}^{T}_{I}\mathbf{q}$	${}^{T}_{G}\mathbf{q}$	${}^{G}_{S}\mathbf{q}$	${}^{S}_{I}\mathbf{q}$
qi	0.384	0.000	−0.006	0.292
q_j	0.006	−0.500	0.382	0.087
q_k	0.712	0.000	−0.006	0.758
q_r	0.587	0.866	0.924	0.577

Table 1.1: Quaternions involved in the telescope attitude calculation. Siderostats' elevation angle of 60 degrees

	${}^{T}_{I}\mathbf{q}$	${}^{T}_{G}\mathbf{q}$	${}^{G}_{S}\mathbf{q}$	${}^{S}_{I}\mathbf{q}$
RA	90.984	0.000	−1.216	100.000
DEC	32.696	60.000	−44.960	20.000
$ROLL$	33.142	0.000	−1.189	30.000

Table 1.2: Quaternions involved in the telescope attitude. Siderostats' elevation angle of 60 degrees. Using the equatorial representation, in degrees.

1.3.3.3 State equations

The goal is to estimate an attitude quaternion ${}^{G}_{I}\mathbf{q}$ describing the inertial attitude of the gondola. In addition, we want to estimate the biases of the three gyroscopes $\mathbf{b}$(t) to have better velocities measurements and improve our predictions. A Kalman filter will be used to obtain the estimate of these values. The seven-dimensional state vector of our system is:

$$\mathbf{x}(t) = \begin{bmatrix} {}^{G}_{I}\mathbf{q}(t) \\ \mathbf{b}(t) \end{bmatrix} \tag{1.6}$$

The evolution of the state vector is defined by:

$${}^{G}_{I}\dot{\mathbf{q}}(t) = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}(t))\,{}^{G}_{I}\mathbf{q}(t) \tag{1.7}$$

$$\dot{\mathbf{b}}(t) = \mathbf{n_b}(t) \tag{1.8}$$

where

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix} = \begin{bmatrix} \lfloor \hat{\boldsymbol{\omega}} \times \rfloor & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & 0 \end{bmatrix} \tag{1.9}$$

with $\boldsymbol{\omega} = \boldsymbol{\omega^{meas}} - \boldsymbol{b} - \boldsymbol{n_g}$. Where $\mathbf{n_g}$ is the noise measured by the three gyroscopes and $\mathbf{n_b}$ is the bias instability. These two noises are considered white and Gaussian with variances σ_g^2 and σ_b^2 respectively. However, the noise values are unknown so we use an estimate of the state vector $\hat{\mathbf{x}} = \left[{}^G_I\hat{\mathbf{q}}(t), \hat{\mathbf{b}}(t)\right]^T$ that follows the same dynamic equations but with the noises equal to zero. When dealing with propagation of quaternions, the Kalman filter faces some numerical complications. To avoid this problem, a multiplicative Kalman filter is implemented by using small angle approximation of the quaternions [TR05] [Riz16]. This modified Kalman filter will have a new six-dimensional state vector called error vector $\tilde{\boldsymbol{x}} = \left[\delta\boldsymbol{\theta}, \Delta\mathbf{b}\right]^T$, where $\Delta\mathbf{b} = \mathbf{b} - \hat{\mathbf{b}}$ and $\delta\boldsymbol{\theta}$ are the three angular errors between the true and estimated attitude quaternions taken directly from the difference quaternion ${}^G_I\delta\mathbf{q} = {}^G_I\mathbf{q} \otimes {}^G_I\hat{\mathbf{q}}^{-1} \approx \left[\frac{1}{2}\delta\boldsymbol{\theta}, 1\right]^T$. With these new variables we can write our gyroscope model as:

$$\boldsymbol{\omega}^{meas} = \boldsymbol{\omega} + \boldsymbol{b} + \boldsymbol{n_g} \tag{1.10}$$

$$\hat{\boldsymbol{\omega}} = \boldsymbol{\omega}^{meas} - \hat{\boldsymbol{b}} \tag{1.11}$$

$$\boldsymbol{\omega}^{true} = \hat{\boldsymbol{\omega}} - \mathbf{n_g} - \Delta\mathbf{b} \tag{1.12}$$

The resulting continuous state equations from this new error state vector are [TR05]:

$$\dot{\delta\boldsymbol{\theta}} = -\hat{\boldsymbol{\omega}} \times \delta\boldsymbol{\theta} - \Delta\mathbf{b} - \mathbf{n_g} \tag{1.13}$$

$$\dot{\Delta\mathbf{b}} = \dot{\mathbf{b}} - \dot{\hat{\mathbf{b}}} = \mathbf{n_b} \tag{1.14}$$

Combining these results we may write the error state equation as

$$\dot{\tilde{\mathbf{x}}} = \begin{bmatrix} \dot{\delta\boldsymbol{\theta}} \\ \dot{\Delta\mathbf{b}} \end{bmatrix} = \mathbf{F} \begin{bmatrix} \delta\boldsymbol{\theta} \\ \Delta\mathbf{b} \end{bmatrix} + \mathbf{G} \begin{bmatrix} \mathbf{n_g} \\ \mathbf{n_b} \end{bmatrix} \tag{1.15}$$

with

$$\mathbf{F} = \begin{bmatrix} -\lfloor\hat{\boldsymbol{\omega}}_\times\rfloor & -\boldsymbol{I}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix} \tag{1.16}$$

and

$$\mathbf{G} = \begin{bmatrix} -\boldsymbol{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \boldsymbol{I}_{3\times3} \end{bmatrix} \tag{1.17}$$

where $\lfloor\hat{\boldsymbol{\omega}}_\times\rfloor$ is the skew-symmetric matrix made out of the elements of $\hat{\boldsymbol{\omega}}$, $\lfloor\hat{\boldsymbol{\omega}}_\times\rfloor\delta\boldsymbol{\theta} = \hat{\boldsymbol{\omega}} \times \delta\boldsymbol{\theta}$. To obtain a discrete transition matrix $\boldsymbol{\Phi}_k$ of the process, we can integrate the state equation between the times t_{k-1} and $t_k = t_{k-1} + \Delta t$:

$$\boldsymbol{\Phi}_k = \boldsymbol{\Phi}(t_k, t_{k+1}) = \boldsymbol{e}^{\boldsymbol{F}\Delta t} = \boldsymbol{I}_{6\times6} + \boldsymbol{F}\Delta t + \frac{1}{2!}\boldsymbol{F}^2\Delta t^2 + ... = \begin{bmatrix} \boldsymbol{\Theta}_k & \boldsymbol{\Psi}_k \\ \mathbf{0}_{3\times3} & \boldsymbol{I}_{3\times3} \end{bmatrix} \tag{1.18}$$

with $\boldsymbol{\Theta}_k \sim \boldsymbol{I}_{3\times3} - \Delta t\lfloor\hat{\boldsymbol{\omega}}_\times\rfloor + \frac{\Delta t^2}{2}\lfloor\hat{\boldsymbol{\omega}}_\times\rfloor^2$ and $\Psi_k \sim -\boldsymbol{I}_{3\times3}\Delta t + \frac{\Delta t^2}{2}\lfloor\hat{\boldsymbol{\omega}}_\times\rfloor$. The full details of these calculations are given in [TR05]. In the final implementation of the Kalman filter, we use the first order terms of the $\boldsymbol{\Theta}_k$ and $\boldsymbol{\Psi}_k$ expressions.

1.3.3.4 Kalman filter

Prediction

In summary, assuming we receive a measurement ω_k^{meas} and we have an estimate of the quaternion $\hat{\boldsymbol{q}}_{k-1|k-1}$ and the bias $\hat{\boldsymbol{b}}_{k-1|k-1}$ at time step $k-1$, the propagation equations for our estimations are:

$$\hat{\boldsymbol{b}}_{k|k-1} = \hat{\boldsymbol{b}}_{k-1|k-1} \tag{1.19}$$

$$\hat{\boldsymbol{\omega}}_{k|k-1} = \boldsymbol{\omega}_k^{meas} - \hat{\boldsymbol{b}}_{k|k-1} \tag{1.20}$$

$$\hat{\boldsymbol{q}}_{k|k-1} = \exp\left(\frac{1}{2}\boldsymbol{\Omega}(\hat{\boldsymbol{\omega}}_{k|k-1})\Delta t\right)\hat{\boldsymbol{q}}_{k-1|k-1} \tag{1.21}$$

The propagation of the covariance matrix $\boldsymbol{P}_{k-1|k-1}$ of the error vector depends on the covariance matrix of the process noise $\boldsymbol{Q_d}$ and the state transition matrix $\boldsymbol{\Phi}_k$. The matrix $\boldsymbol{\Phi}_k$ is computed using $\hat{\boldsymbol{\omega}}_{k|k-1}$ in the expressions of $\boldsymbol{\Theta_k}$ and $\boldsymbol{\Psi_k}$. The noise covariance matrix $\boldsymbol{Q_d}$ is considered constant and has the following theoretical expression [TR05]:

$$\boldsymbol{Q_d} \sim \begin{bmatrix} \sigma_g^2 \Delta t \boldsymbol{I}_{3\times3} & -\sigma_b^2 \frac{\Delta t^2}{2} \boldsymbol{I}_{3\times3} \\ -\sigma_b^2 \frac{\Delta t^2}{2} \boldsymbol{I}_{3\times3} & \sigma_b^2 \Delta t \boldsymbol{I}_{3\times3} \end{bmatrix} \tag{1.22}$$

where $\sigma_g = 1.5 \times 10^{-7}\ rad/\sqrt{s}$ represents the uncertainty related to the ARW noise and $\sigma_b = 1.8 \times 10^{-6}\ rad\ s^{-3/2}$ the bias instability, obtained from the gyroscopes manufacturer. In practice, the best results are not achieved with the previous values. An empiric tuning of σ_g and σ_b is performed once the whole control system is functional. According to the Kalman filter equations, the matrix $\boldsymbol{P}_{k|k}$ propagation is computed:

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{\Phi}_k \boldsymbol{P}_{k-1|k-1} \boldsymbol{\Phi}_k^T + \boldsymbol{Q}_d \tag{1.23}$$

Update

The update phase of the Kalman filter will use the difference between our prediction $\hat{\boldsymbol{q}}_{k|k-1}$ and the new star camera measurement $\boldsymbol{q}_k^{meas}$. This difference is called innovation, $\boldsymbol{z}_k$, and it will be equal to the angular part $\delta\boldsymbol{\theta}_k^{meas}$ of the difference quaternion $\delta\boldsymbol{q}^{meas} = \boldsymbol{q}_k^{meas} \otimes \hat{\boldsymbol{q}}_{k|k-1}^{-1}$. The measurement model will be as follows:

$$\boldsymbol{z}_k = \delta\boldsymbol{\theta}_k^{meas} \approx \boldsymbol{H}\tilde{\boldsymbol{x}}_k + \boldsymbol{n}_k^{meas} \tag{1.24}$$

where $\boldsymbol{H} = \begin{bmatrix} \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} \end{bmatrix}$ and $\boldsymbol{n}_k^{meas}$ is the star camera measurement noise, characterized by a covariance matrix $\boldsymbol{R}_k$. Then, the covariance matrix of the innovation is $\boldsymbol{S}_k = \boldsymbol{H}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^T + \boldsymbol{R}_k$ and the Kalman gain results $\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1}\boldsymbol{H}^T\boldsymbol{S}_k^{-1}$. Once we calculate the previous matrices, we can start the update:

$$\tilde{\boldsymbol{x}}_{k|k} = \boldsymbol{K}_k \boldsymbol{z}_k = \begin{bmatrix} \delta\boldsymbol{\theta} \\ \Delta\boldsymbol{b} \end{bmatrix} = \begin{bmatrix} 2\boldsymbol{v} \\ \Delta\boldsymbol{b} \end{bmatrix} \tag{1.25}$$

$$\delta\boldsymbol{q} = \begin{cases} \begin{bmatrix} \boldsymbol{v} \\ \sqrt{1 - \boldsymbol{v}^T\boldsymbol{v}} \end{bmatrix}, & \text{if } \boldsymbol{v}^T\boldsymbol{v} \leq 1 \\ \frac{1}{\sqrt{1+\boldsymbol{v}^T\boldsymbol{v}}} \begin{bmatrix} \boldsymbol{v} \\ 1 \end{bmatrix}, & \text{otherwise} \end{cases} \tag{1.26}$$

$$\hat{\boldsymbol{q}}_{k|k} = \delta\boldsymbol{q} \otimes \hat{\boldsymbol{q}}_{k|k-1} \tag{1.27}$$

$$\hat{\boldsymbol{b}}_{k|k} = \hat{\boldsymbol{b}}_{k|k-1} + \Delta\boldsymbol{b} \tag{1.28}$$

$$\hat{\boldsymbol{\omega}}_{k|k} = \hat{\boldsymbol{\omega}}_k^{meas} - \hat{\boldsymbol{b}}_{k|k} \tag{1.29}$$

$$\boldsymbol{P}_{k|k} = (\boldsymbol{I}_{3\times3} - \boldsymbol{K}_k\boldsymbol{H})\boldsymbol{P}_{k|k-1}(\boldsymbol{I}_{3\times3} - \boldsymbol{K}_k\boldsymbol{H})^T + \boldsymbol{K}_k\boldsymbol{R}_k\boldsymbol{K}_k^T \tag{1.30}$$

After finishing this steps, the loop restarts propagating the estimated states until a new measurement is received and the update is performed again. However, there exists a delay of a few seconds between the trigger of the star cameras and the reception of the solution. To solve this issue, the $\boldsymbol{q}_k^{meas}$ measurement is propagated until the current time step k, using propagation matrices calculated since the instant $k-N$ where the image was taken. The details of these issue are explained in the appendix C. In case the star cameras are not working, a less accurate absolute position measurement is available using a magnetometer and GPS data.

1.4 Commands and Telemetry system

In this section, we will discuss briefly the different communications and software used to get the telemetry data from BETTII. The payload has an on-board computer operating a Linux kernel, called Ford, running different processes such as the star camera solutions finder or the operation of the science detectors. It was specially built to withstand the harsh conditions during flight. Another device, a cRIO 9038 from National Instruments that we call Boop, has a CPU running a Real Time software that computes the Kalman filter and a FPGA for the applications that require accurate timings like the control loops.

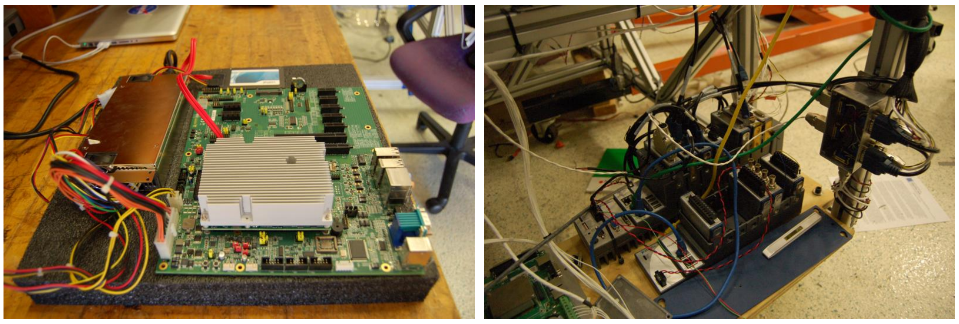
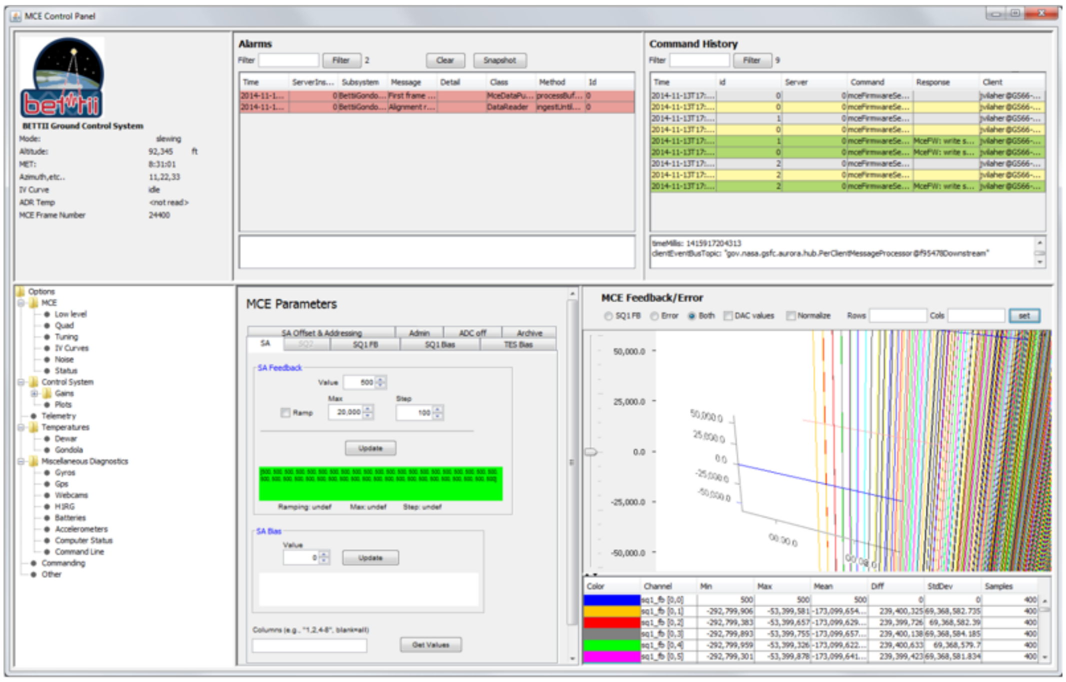


Figure 1.11: Ford and cRIO computers.

All the telemetry data from Boop, such as the covariance matrices and the estimated state of the Kalman filter, is sent via a TCP communication to Ford, that parses and compresses the data before sending it through the downlink [Vil14]. At ground, a common PC running a specially developed Java software called Aurora can read all the coming data and store it in an archive. Aurora operates all the different systems in BETTII, so it is also used to send different commands and change the different subsystems behaviour, modifying, for instance, parameters of the blob finding algorithm or the different PID gains.

During flight, the downlink and uplink communications are managed by an on-board device called CIP that is provided by the CSBF balloon facility. The CIP is used routinely in all the CSF balloon programs and has mature electronic equipments for telemetry, command and tracking. On ground, a transceiver is connected to a PC running Aurora, that will process all the messages and send the commands.

When the payload is not in flight configuration, the CIP is substituted by a wireless router. The use of a router for the communication provides more features such as remote desktop that eases the testing and debugging of the systems.



BETTII Ground Control System

Mode:	slewing
Altitude:	92,345 ft
MET:	8:31:01
Azimuth,etc..	11,22,33
IV Curve	idle
ADR Temp	<not read>
MCE Frame Number	24400

Alarms

Filter 2

Time	ServerIns...	Subsystem	Message	Detail	Class	Method	Id
2014-11-1...	0	BettiGondo...	First frame ...		MceDataPu...	processBuf...	0
2014-11-1...	0	BettiGondo...	Alignment r...		DataReader	ingestUntil...	0

Command History

Filter 9

Time	id	Server	Command	Response	Client
2014-11-13T17:...	0	0	mceFirmwareSe...		jvlaher@GS66-...
2014-11-13T17:...	0	0	mceFirmwareSe...		jvlaher@GS66-...
2014-11-13T17:...	1	0	mceFirmwareSe...		jvlaher@GS66-...
2014-11-13T17:...	0	0	mceFirmwareSe...		jvlaher@GS66-...
2014-11-13T17:...	1	0	mceFirmwareSe...	MceFW: write s...	jvlaher@GS66-...
2014-11-13T17:...	0	0	mceFirmwareSe...	MceFW: write s...	jvlaher@GS66-...
2014-11-13T17:...	2	0	mceFirmwareSe...		jvlaher@GS66-...
2014-11-13T17:...	2	0	mceFirmwareSe...		jvlaher@GS66-...
2014-11-13T17:...	2	0	mceFirmwareSe...	MceFW: write s...	jvlaher@GS66-...

```
timeMills: 1415917204313
clientEventBusTopic: "gov.nasa.gsfc.aurora.hub.PerClientMessageProcessor@f95478Downstream"
```

SA Feedback: Value 500; Ramp; Max 20,000; Step 100

Ramping: undef Max: undef Step: undef

SA Bias: Value 0

Columns (e.g., "1,2,4-8", blank=all)

Color	Channel	Min	Max	Mean	Diff	StdDev	Samples
🟦	sq1_fb [0,0]	500	500	500	0	0	400
🟧	sq1_fb [0,1]	-292,799,906	-53,399,581	-173,099,654...	239,400,325	69,368,582.735	400
🟥	sq1_fb [0,2]	-292,799,383	-53,399,657	-173,099,629...	239,399,726	69,368,582.39	400
⬛	sq1_fb [0,3]	-292,799,893	-53,399,755	-173,099,657...	239,400,138	69,368,584.185	400
🟩	sq1_fb [0,4]	-292,799,959	-53,399,326	-173,099,622...	239,400,633	69,368,579.7	400
🟪	sq1_fb [0,5]	-292,799,301	-53,399,878	-173,099,641...	239,399,423	69,368,581.834	400

Figure 1.12: Main GUI of the Aurora software.

Chapter 2

Flight Campaign

NASA and other agencies organize balloon launch campaigns year-round across multiple continents. The Columbia Scientific Balloon Facility (CSBF) at Palestine, Texas, was the location for BETTII's spring flight campaign. For a typical launch, the scientific payload is attached on the bottom of a 100m long train that includes a parachute and a ladder. The top of the ladder is attached to the bottom of the large helium-filled balloon. High-altitude balloons fly between 30 and 40 km, above the 99% of the atmosphere, which make them particularly suited for studying the universe at infrared, far-infrared and sub-millimeter wavelengths.

At these altitudes, temperatures are around 240K (about $-33°C$), while the air pressure is 0.5% of the sea level pressure (about 5 mbar). There are also high altitude winds, large laminar flows that move the balloon and payload as one. These disturbances excite pendulum motions which are typically of the order of a few arcminutes and have periods of a few to many tens of seconds.

The payload's temperature distribution is influenced by the air temperature, the sunlight and the infrared radiation of the Earth, which can result in complex temperature gradients across the instrument. For this reason, several thermometers were installed on the structure of BETTII to study the temperature distribution. The flight during this campaign is at night, where a better temperature uniformity is expected.

Cosmic rays can also affect balloon experiments, damaging the electronics. However, this is more taken into account for long-duration flights around Antarctica, where payloads are exposed to the cosmic rays environment for many weeks.

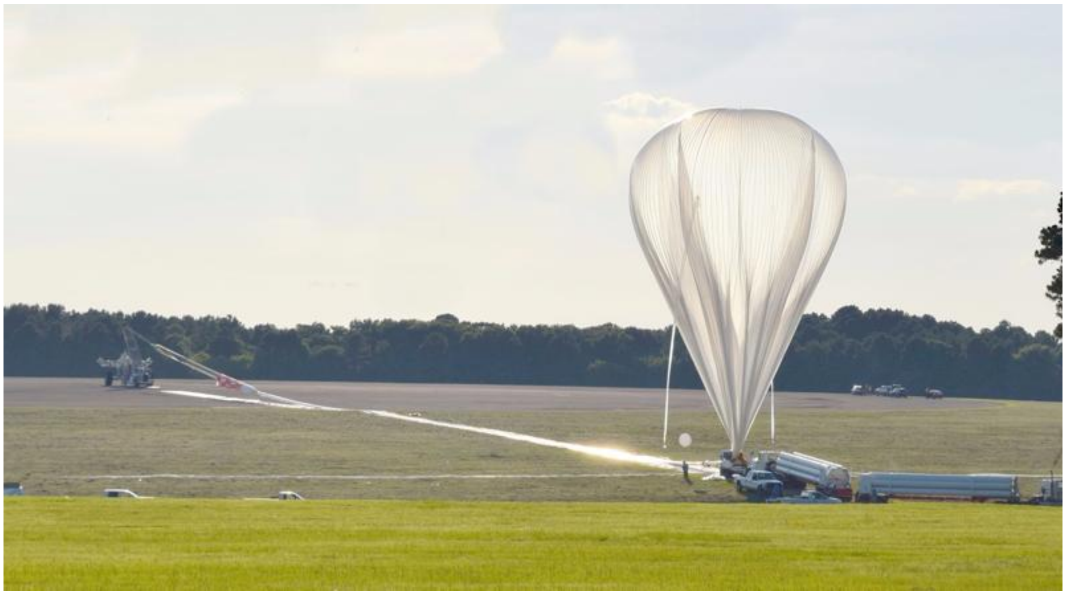


Figure 2.1: BETTII balloon launch. On the left, the payload is captured by the launch vehicle "Big Bill", until the balloon is inflated and released. The parachute assembly can be seen in orange, connecting the bottom of the balloon with BETTII.

2.1 Data analysis tools development

All the information gathered from the payload is sent through the telemetry down-link and stored in an archive created by the Aurora software. Usually, to plot all this information, an open-source tool called KST plot is used. KST reads the several binary files representing each field of data and draws them in X-Y or Fourier transform plots. It has a very intuitive user interface allowing different styles and saving the configured sessions for a later use.

However, KST is not intended to generate new data from the existing information and that is a very useful feature that the BETTII team wanted. This need is why one of the main tasks during this launch campaign was to develop software capable of processing all types of data and generate more useful information for debugging and analysis[1]. The language used was Python because it is very common in astronomy and it was understandable by all the team. The main libraries used for the development are:

- NumPy, a fundamental Python library that provides numerical arrays and functions
- SciPy, a scientific Python library, which supplements and slightly overlaps NumPy. NumPy and SciPy historically shared their code base but were later separated.
- matplotlib, a plotting library based on NumPy
- pandas, a data analysis and manipulation library.

[1]GitHub repository: `https://github.com/androidside/pythonFlightDataProcessing`

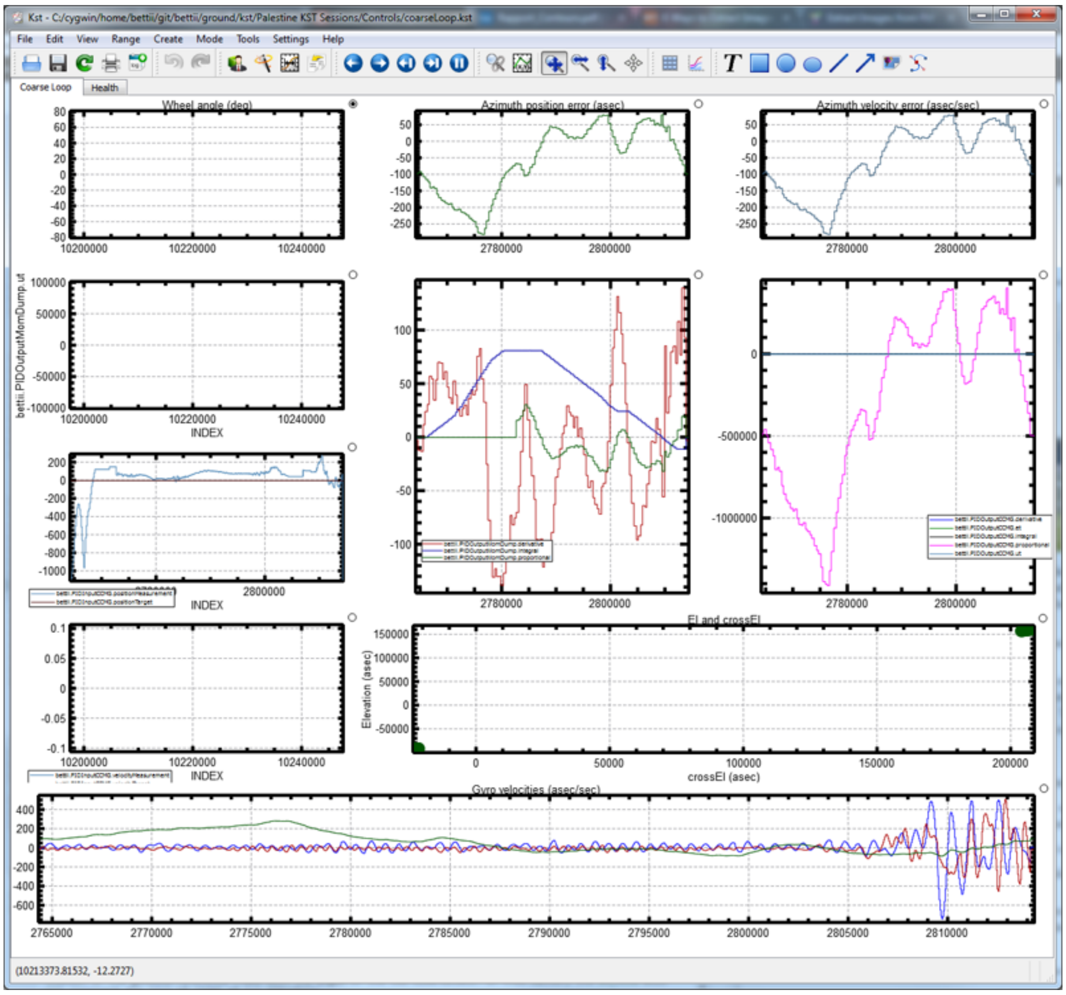

Kst - C:/cygwin/home/bettii/git/bettii/ground/kst/Palestine KST Sessions/Controls/coarseLoop.kst
File Edit View Range Create Mode Tools Settings Help
Coarse Loop
Health
Wheel angle (deg)
Azimuth position error (asec)
Azimuth velocity error (asec/sec)
bettii.PIDOutputMomDump.ut
INDEX
El and crossEl
Elevation (asec)
crossEl (asec)
Gyro velocities (asec/sec)
(10213373.81532, -12.2727)

Figure 2.2: A KST session showing plots of the BETTII control system.

With this software, which adapts to Aurora's archiving system and deals with specific issues such as parsing errors, many interesting plots can be obtained that will be shown later on this document. A good example of the adequacy of this tools over KST are the figures that were designed for the science team to show in real time the information from the four FIR detectors during flight (figure 2.3). Unfortunately, the detectors were not cold enough during flight in order to provide useful information.

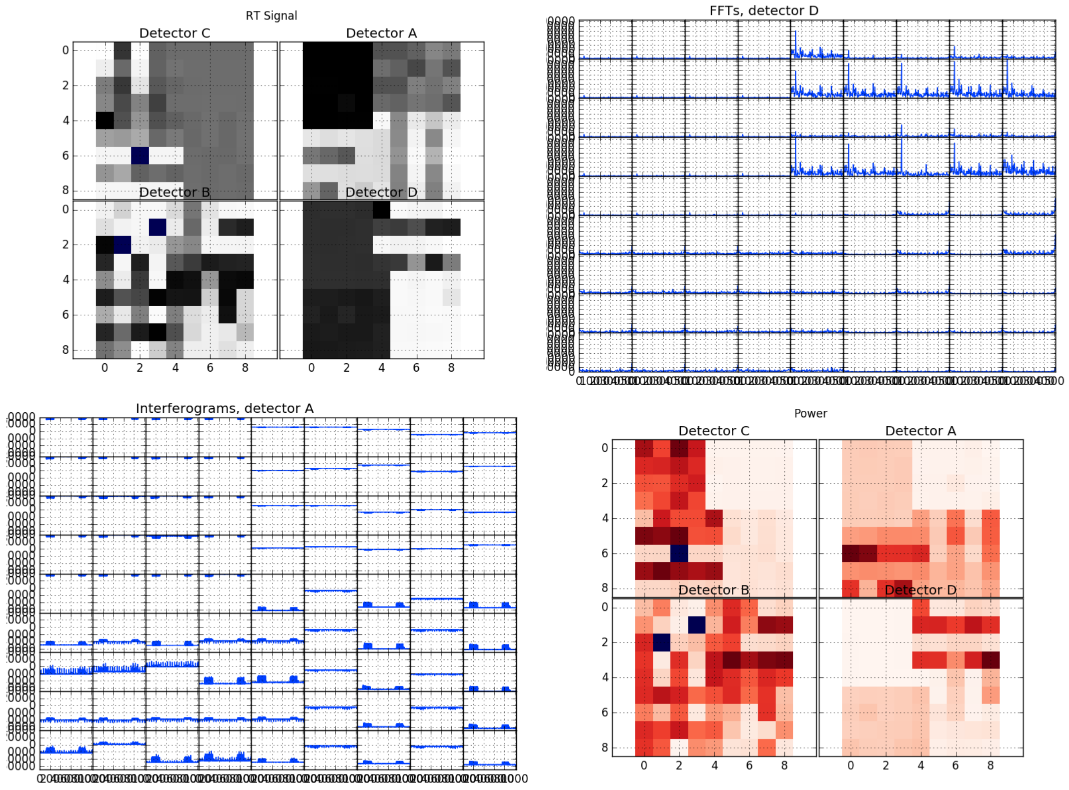


Figure 2.3: Real time signal processing of the detectors. Developed for the scientific team of BETTII. The data shown in the figures is taken with bad conditions and is not useful.

2.2 Pre-flight tests

Before launch, a lot of work is involved, not only transporting the payload to the launch facilities but also tuning the different subsystems of BETTII and running several tests. The work during a flight campaign is very special, trying to fix and improve every unexpected event before the day of the launch. One of the main problems during this campaign was the Dewar, as it had a cold leak and we couldn't refrigerate enough the detectors to obtain relevant scientific data. In addition, one wheel of the CCMG started to fail, which was a system that had never failed before. Luckily, all these problems were solved by the team in order to be ready on June 2, the nominal launch day.

In the weeks before the launch, at CSBF, different tests of the control system were done. During the day, if the other members of the team allowed us, we performed hanging tests to determine the PID gains of the CCMG and momentum dump control loops for different scenarios. During the night we could obtain star camera solutions, that allowed us to test the attitude estimation and tune its different parameters.

Once all the subsystems were ready and the Dewar was again assembled to the telescope, after solving the cold leak, we could perform more complete tests hanging BETTII from a crane outdoors at night. The results from that test were not very successful. The control system is designed to withstand the perturbations at high altitudes. The winds of that night were too strong to counteract and point to a bright star. All the following hanging tests were performed indoors. There, without large perturbations and the total mass loaded, we could extract more data such as the inertia J_z in azimuth.

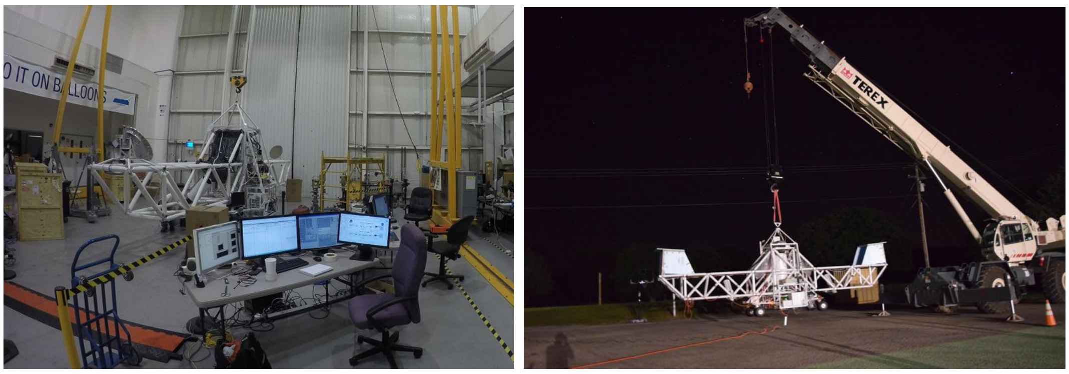


Figure 2.4: Indoors and outdoors hanging tests of BETTII

2.2.1 Inertia value measurement

The value of the inertia J_z commented in the section 1.3.2 is important to command properly the momentum dump. The moment of inertia depends on the mass distribution of BETTII and it can change every time we add or remove a component. For this reason, it has to be calculated with a simple procedure shortly before the launch. This procedure consists on applying a constant torque τ_{CCMG} in order to measure the azimuth acceleration $\dot{\omega}$ while hanging from a crane. A constant torque is commanded by rotating the CCMG wheels' gimbal at a constant rate. Following the equations 1.3 and 1.2 and according to the conservation of momentum, we can express:

$$J_z \dot{\omega}_z = \tau_{CCMG} = 20.8 \; \dot{\theta} \; \cos\theta \tag{2.1}$$

The results of this procedure, realized three days before the launch, are shown in the figure 2.5. From the data obtained, an average torque τ_{CCMG} of 0.184 Nms and acceleration $\dot{\omega}$ of 11×10^{-5} rad/s^2 were measured. Thus, the resulting estimated inertia value is $J_z = 1070.4$ kg m^2. The same value was obtained on the previous flight campaign at Fort Summer on 2016. This confirms the expected result, because no design changes were applied that affected the weight distribution nor the inertia of the payload.

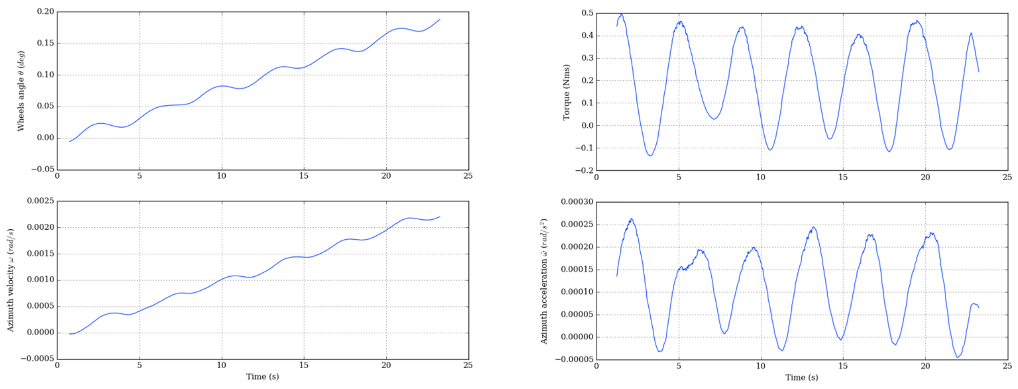


Figure 2.5: Measured azimuth velocity, commanded wheels' angle, torque and azimuth acceleration during the test performed to estimate the J_z value.

2.3 Flight

The launch, initially planed for the 2nd of June, was postponed to the 8th of June due to the bad weather conditions during that week. The extra time was useful to perform additional flight rehearsals where the roles of every member of the team were clarified. I was the responsible to monitor different data during the ascension, such as the temperatures of the 90 thermometers distributed along the structure or the currents consumed on every voltage line. Once at float, when the conditions are more stable, we could start to operate the control system.

The preparations started at 12:15PM, when the flight batteries were connected, Ford was turned on and the filling of the Helium tank of the Dewar had started. After the final checks, at 15:00, the CSBF staff attached BETTII to the launch vehicle "Big Bill" and it was moved to the launchpad. At 18:20 the balloon filling started. The launch occurred at 19:11. Some problems happened during the ascension. The altitude provided by the GPS stopped working when we crossed 12000 m of altitude (figure B.6). It seems that the change on the dynamic platform model of the MAX-M8Q GPS module was not applied, as the default mode has a 12 km limit on altitude [uBl17].

We couldn't accelerate the CCMG wheels during the flight. The possible cause can be a lack of communication between the Galil controllers and Ford. Temperatures dropped drastically, below $-50°C$. The several heaters installed were not enough to maintain the temperatures above $-10°C$ at the devices of interest such as the star cameras. Probably due to the harsh conditions, the auto-focus mechanism couldn't work properly and it was difficult to obtain star camera solutions.

However, despite all the problems that we faced, usual on a first flight like this one, we achieved great goals like braking the payload using only the momentum dump mechanism. To achieve that, we had to change the PID gains and see the response in real time. This is a demonstration of the good performance and the importance of the telemetry system and real time data representation. In addition, we could tune the star cameras software parameters to finally obtain solutions and analyze the estimator behaviour.

Towards the end of the flight, at 3:00AM, we started to lose a lot of telemetry packets and we had big gaps of time without any information. We tried to connect to a different ground station located at Fort Summer, NM, that was closer to BETTII at that time, but the problem remained. At 7:00AM the descend started. Before the descend, all the systems were turned off for security reasons.

Unfortunately, when the parachute deployed, it detached from the payload and BETTII plummeted 40km in free fall to earth. The consequences of the impact were severe, with the complete destruction of all the systems. Luckily, the SSDs where the telemetry data was stored survived and the information could be recovered, including the moments where the transmission failed. At figure 2.6 we can appreciate how the recovery of this data improved a lot the information received through the CIP, having a higher rate of data points and solving the several telemetry gaps, especially towards the end of the flight.

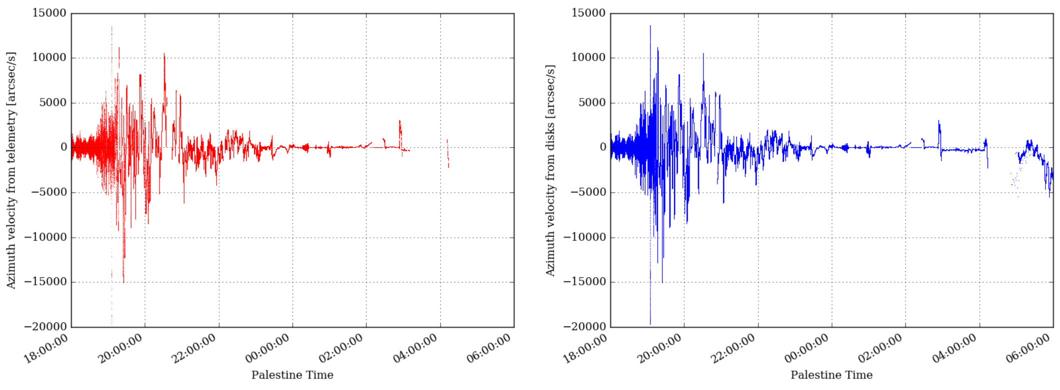


Figure 2.6: Measured azimuth velocity. Left, information from the telemetry received during flight. Right, information extracted from the disks after the flight.

Chapter 3

Flight Data

In this section, we will show some information extracted from the recovered disks. In the appendix B there are more figures showing data during the entire flight, such as the altitude, the temperatures, the gyroscopes readings, the current consumption or the magnetometer measurements. We can observe on these figures how there are several gaps due to the Boop and Ford reboots, particularly towards the end of the flight. These reboots were performed in order to solve the telemetry problems.

3.1 Gyroscopes

A first analysis we can perform is to study the power spectral density (PSD) of the three gyroscopes. In this manner we can extract frequential information about the type of perturbations we found during the flight. If we focus on the moments where we didn't actively control the attitude and after braking the payload movements caused by the ascension, we observe that dominant frequencies are very low. This phenomenon is caused by the pendulum motions resulting from the attachment to the balloon.

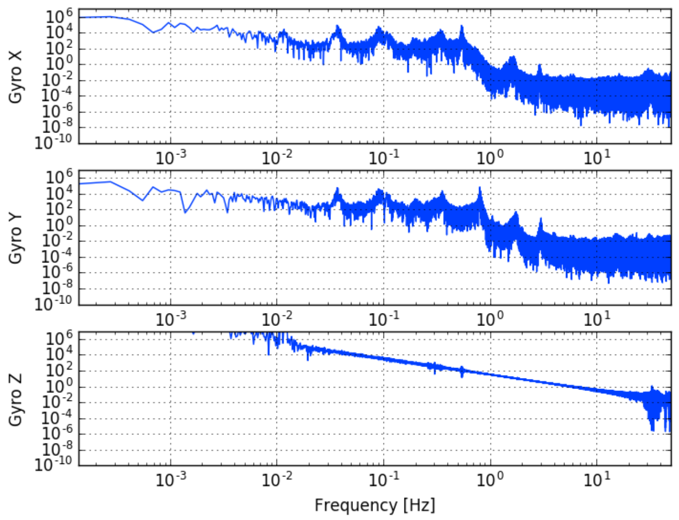
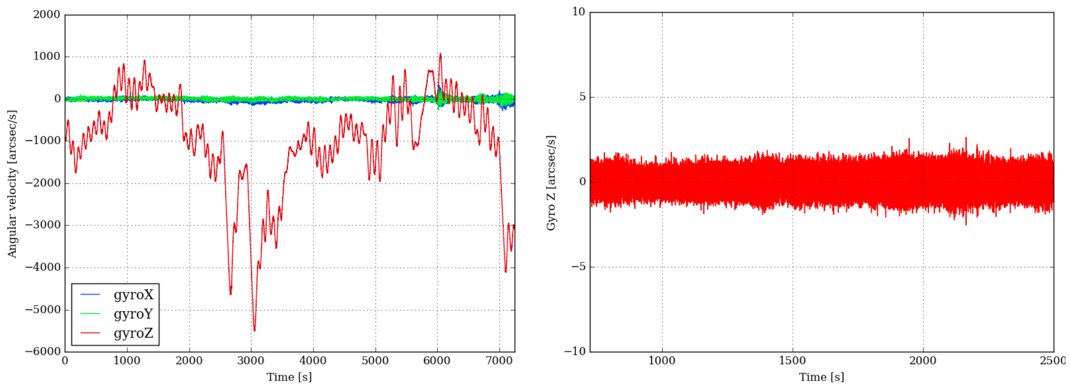


Figure 3.1: Power spectral density of the three gyroscopes during flight

σ_{g_x}	0.37
σ_{g_y}	0.42
σ_{g_z}	0.43

Table 3.1: Estimated noise uncertainties of the three gyroscopes during the flight, in arcsec/s.

If we filter these low frequencies, below 3 Hz, we can obtain an estimate of the gyroscopes noise, shown in the table 3.1. The uncertainty values, of the order of 0.4 arcsec/s, are two times higher than the expected theoretical values specified by the manufacturer. This increment is caused by a problem in the close-loop algorithms inside the electronics of the gyroscopes. The problem was known by the manufacturer, and the solution was to inject a random phase perturbation in the closed loop. This had the effect of increasing, by a factor of 4, the noise variance of the gyroscopes.



(a) Measured velocities by the three gyroscopes.

(b) Azimuth gyroscope. High-pass filtered at 3 Hz.

Figure 3.2: Signal from the gyroscopes.

3.2 Momentum dump

During the flight, the CCMG controller lost connection with Ford, and consequently we tried to brake the payload using only the momentum dump. After tuning the PID gains, we could limit the oscillations of the azimuth velocity below 200 arcsec/s.

These results confirm the excellent performance of the momentum dump mechanism, a component specially designed for BETTII. The fact of being able to maintain stability and brake the resulting oscillation from the balloon ascension allowed us to obtain enough good star camera images. In addition, we could obtain images of a star at the NIR detectors during a short period of time.

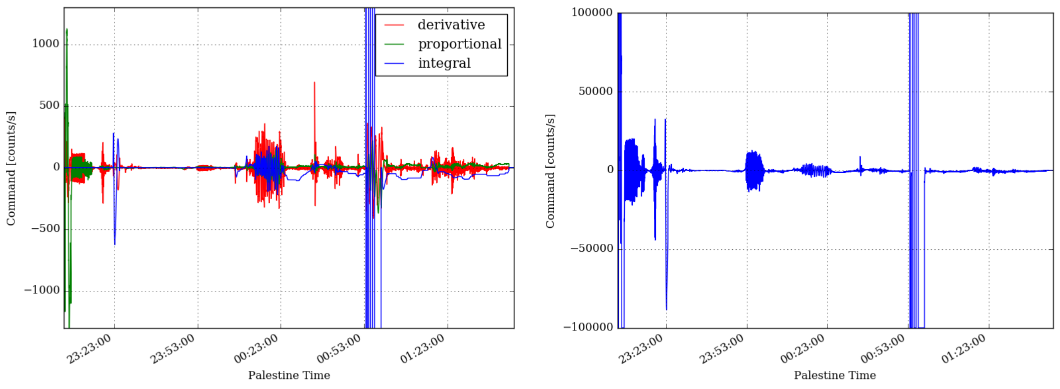


Figure 3.3: Commands of the momentum dump during the braking of the payload. To generate the command $u(t)$ (right), a scaling gain is applied after the sum of the three PID contributions (left). There exists a software limit of 100000 counts per second, that is equal to 4.7 rpm.

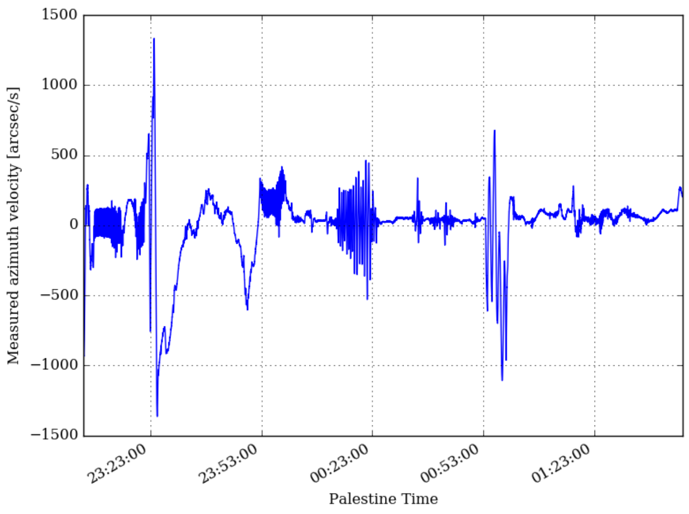


Figure 3.4: Measured azimuth velocity during the braking.

3.3 Attitude estimator

Regarding the attitude estimator, we obtained enough information to evaluate its behavior. Despite the auto-focus mechanism failure, we were able to get good star camera images and the star camera solution finder software could measure an absolute attitude of BETTII. These star camera solutions were critical for the correct behaviour of the estimator and allowed us to know where the telescope was pointing at every time. In this section we will show some results from the attitude estimator, archived from 02:09AM to 4:13AM, where we obtained a total of 50 star camera solutions.

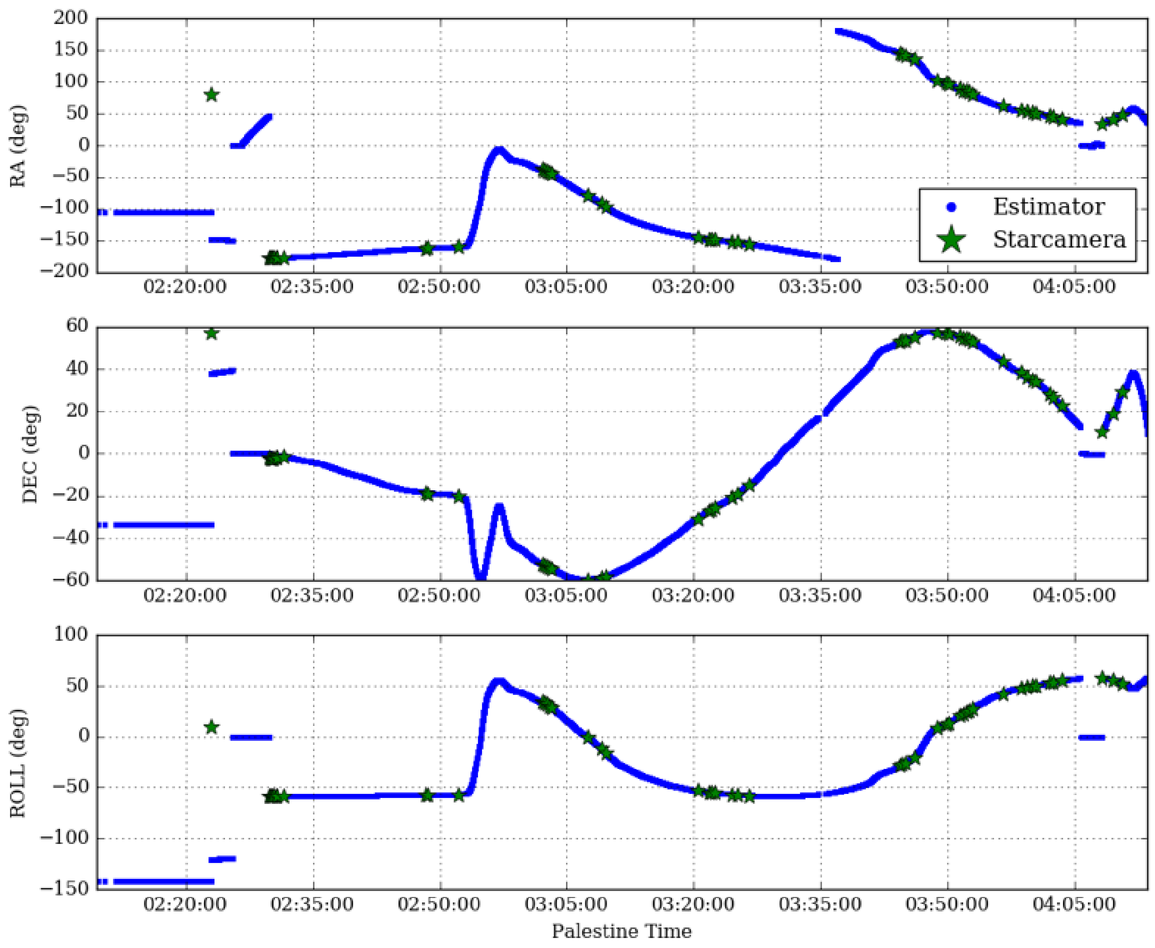
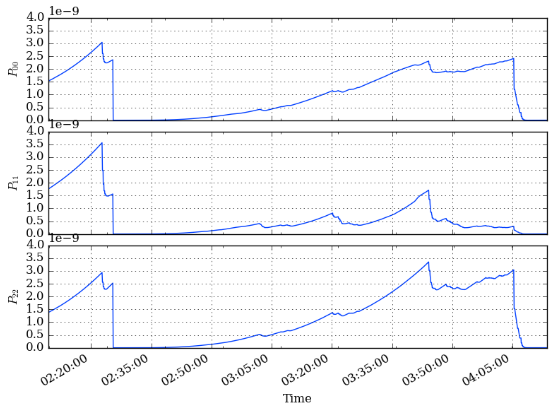


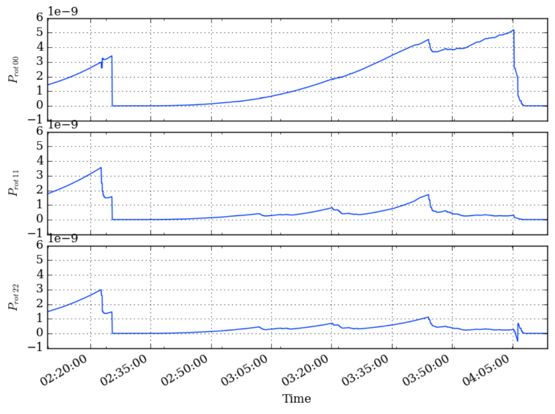
Figure 3.5: Evolution of the estimated attitude and the star camera measurements. There are two times where the estimator was intentionally reset to zero during the flight.

3.3.1 Covariance matrix

The covariance matrix $\boldsymbol{P}$, describes the uncertainty we have in the state estimation. Neglecting the crossed elements and focusing only on the diagonal, we can obtain a simple visualization of the attitude uncertainty. The first three values of the diagonal correspond to the state $\boldsymbol{\delta\theta}$. The uncertainty of this state, due to the small-angle approximation, can be interpreted directly as the uncertainty in attitude on the gyroscopes reference frame G. In the figure 3.6a we can see how there is a certain correlation between the first three values of the matrix $\boldsymbol{P}$ diagonal.



(a) $\boldsymbol{P}$



(b) $\boldsymbol{P_{rot}}$

Figure 3.6: Evolution of the first three diagonal elements of the matrices $\boldsymbol{P}$ and $\boldsymbol{P_{rot}}$

The main cause of this correlation is the way the star camera solutions are propagated. A solution provided by the star camera software has an uncertainty two orders of magnitude higher in ROLL than in RA and DEC. In the figure 1.5 we can see an example of a star camera solution, where the roll uncertainty is higher. Then, this solution is rotated 45 degrees in order to be operated on the gyroscopes reference frame. What we practically see is the roll uncertainty projected to the two other components.

In order to obtain a better representation of the attitude uncertainty of the telescope, we can undo the rotation ${}^{G}_{S}\boldsymbol{q}$ that we apply to the the star camera solutions, to the upper left 3×3 block $\boldsymbol{P}_{3\times3}$ of the matrix $\boldsymbol{P}$:

$$\boldsymbol{P}_{rot} = {}^{S}_{G}\boldsymbol{M}\ \boldsymbol{P}_{3\times3}\ {}^{S}_{G}\boldsymbol{M}^{T} \tag{3.1}$$

where ${}^{S}_{G}\boldsymbol{M}$ corresponds to the rotation matrix associated with the quaternion ${}^{S}_{G}\boldsymbol{q}$, a rotation of approximately -45 degrees around the y_g axis.

$${}^{S}_{G}\boldsymbol{M} = (2q_r^2 - 1)\boldsymbol{I}_{3\times3} - 2q_r\lfloor\hat{\boldsymbol{v}}\times\rfloor + 2\boldsymbol{v}\boldsymbol{v}^{T} \tag{3.2}$$

with ${}^{S}_{G}\boldsymbol{q} = [\boldsymbol{v}\ q_r]^{T}$. In the figure 3.6b we can see the result of this rotation. With P_{rot}, the uncertainties are more uncorrelated, obtaining lower values for the RA and DEC uncertainties, the most important coordinates for the telescope. This result demonstrates the importance of having the star cameras pointing to where the telescope will see. This is one of the main reasons why the star cameras are located at an elevation angle of 45 degrees.

An approximation of the uncertainties in the telescope's attitude is shown in the figure 3.7, relating the square root of the first element of the diagonal of $\boldsymbol{P}_{rot}$ to the $ROLL$, the second to the DEC and the third to the RA coordinates. When we were receiving star camera solutions frequently, the uncertainties arrived below 4 arcseconds for both RA and DEC coordinates.

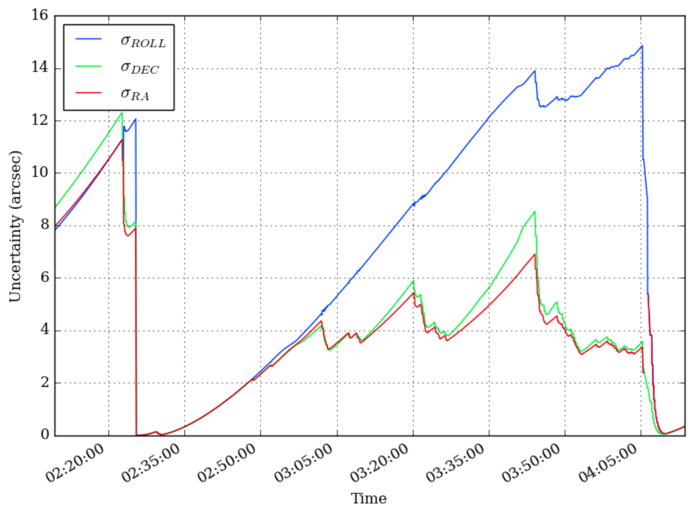


Figure 3.7: Evolution of the uncertainties in the attitude of the telescope.

3.3.2 Post-flight estimation

In the figure 3.5 we can observe how there are instants where the estimated attitude is not valid, due to resets of the estimator. A good exercise is to implement the Kalman filter on a common PC with Python that will use the measurements stored in the disks in order to have a good estimate of the attitude of BETTII, even at the instants were the estimator was not working. To do that, we will use the same equations implemented on Boop and described in the section 1.3.3.4.

The result of this implementation is shown at figure 3.8, where the attitudes obtained are very similar to the ones on the figure 3.5 but filling the gaps where the estimator was reset to zero. One of the advantages of estimating the attitude after the flight was the possibility to estimate the telescope's attitude even before we started receiving star camera solutions. However, this task became impossible because there were several instants where the whole system was rebooted and we lost information of the gyroscopes during large periods of time, as we can observe on the first instants of the figure B.2.

Regarding the biases, the gyroscopes are very precise and have small and stable biases according to the manufacturer. Consequently, the estimated biases are very low, of the order of tenths of arcseconds per second (figures B.4 and 3.9). The error induced by this low order of magnitude is very similar to the possible misalignments between the gyroscopes or the residual errors on the reference frame measurements. We can apply a matrix to the gyroscopes measurement $\boldsymbol{\omega}_k^{meas}$ that will correct any angular error between reference frames or even the orthogonalization errors and scale factor errors. To estimate this matrix, two Kalman filters

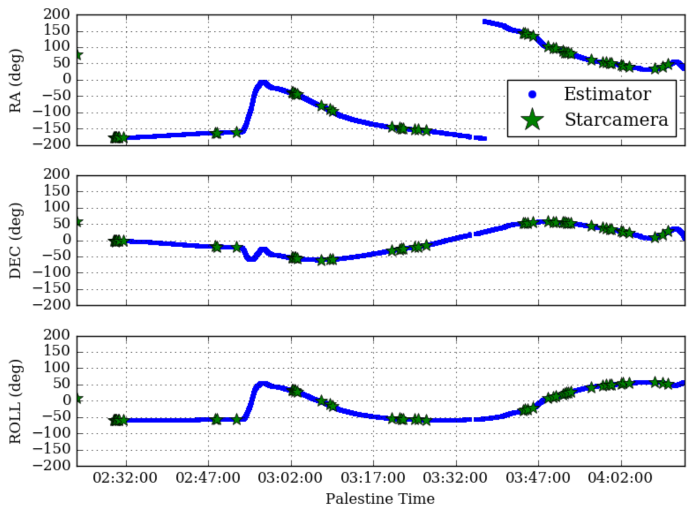


Figure 3.8: Evolution of BETTII's attitude, estimated after the flight. The estimator was implemented in Python, using the same gains as in flight, the case of the figure 3.5.

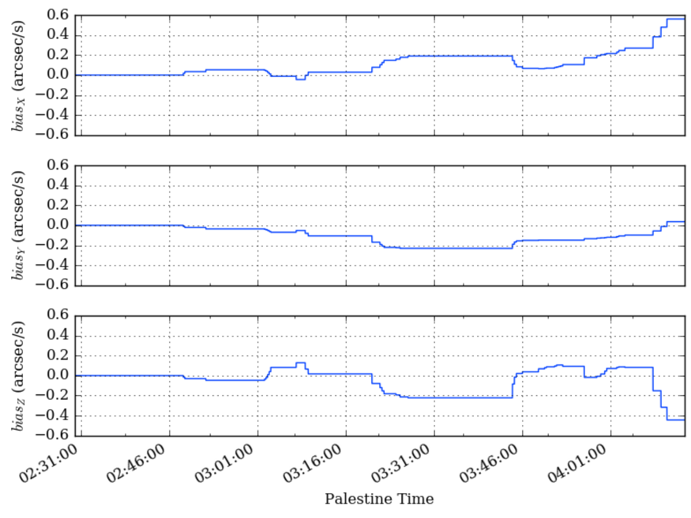


Figure 3.9: Evolution of the estimated gyroscopes biases. The estimator was implemented in Python, using the same gains as in flight.

have been proposed [Riz16] and implemented in Python.

Here, however, we propose an additional Kalman filter that will only estimate the attitude and not the biases. That will be equal to the presented filter but assuming the biases to be zero (equivalently, forcing the σ_b parameter of the Q_d matrix to be zero), reducing the dimensions of the state vector from six to only three elements. Apart from the high computational advantage that this supposes, this filter will not assume any error to the biases and it will avoid bad propagations caused by a wrong estimation of the biases.

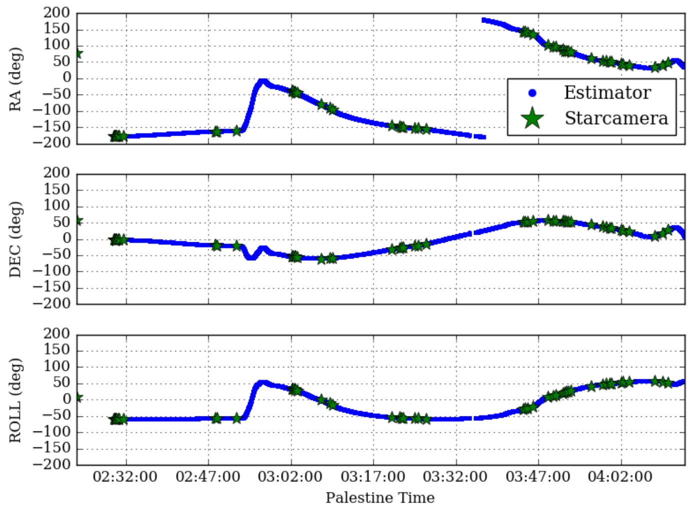


Figure 3.10: Evolution of BETTII's attitude, estimated after the flight. The estimator is a reduced Kalman filter that uses only the state $\delta\boldsymbol{\theta}$.

In the figure 3.11 we can see a comparison of the estimated telescope attitude uncertainty, calculated from the covariance matrix $\boldsymbol{P}$ in the manner explained in the previous section. We can see how the uncertainties from both filters are below the ones calculated during flight. In addition, the proposed reduced Kalman filter estimates an uncertainty lower than 1.8" when the star cameras were frequent.

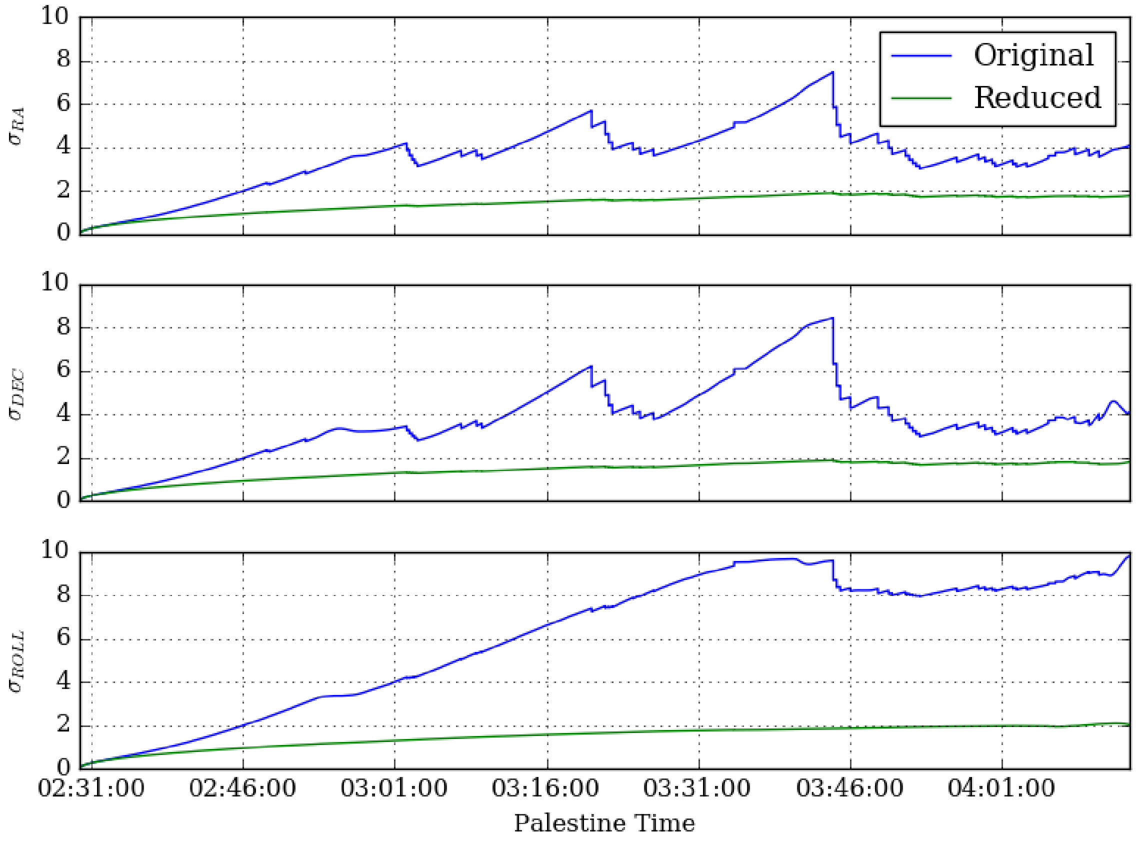


Figure 3.11: Comparison of the telescope attitude uncertainties obtained from the two different Kalman filters implemented on ground, after the flight

Conclusions and future work

This project was intended to analyze the data obtained during the first flight of BETTII. An aerospace project such as BETTII includes many subsystems that require experts in different disciplines. Since my specialty in this last year of master in Toulouse was in automation and estimation, I focused on the control system. However, given my training as a telecommunications engineer in Barcelona, I could also collaborate on other aspects such as the design and debug of electronic circuits and software development in C ++, Python, MatLab, LabView and Java for the control and telemetry system.

The main conclusion that can be drawn from this experience is that carrying out a complex project such as BETTII successfully is not easy. We had a small and highly experienced NASA team, who worked together, anticipating and taking care of all the possible problems in detail but, unfortunately, a small failure at the time of descent ended up destroying a payload that took years in the making. This incident was a tough shock to the team but the initial pessimism gave way to the redesign and improvement of BETTII. Some of these improvements focus primarily on correcting the problems seen during the flight.

In order to avoid the future cold leaks seen during the campaign, the team is currently working on a new Dewar design, where there will be no such marked emphasis on lightness but on a better robustness. In addition, a better design of the tests before the flight could have avoided the problems of the GPS and the controllers of the wheels of the CCMG.

Regarding the control system, two main improvements are proposed. The gyroscopes used to date, the SRS-2000, offer very small biases. The error induced by these biases was much smaller than that related to the ARW. The consequence of this is that the Kalman filter considered very stable and small biases. Reducing the state vector in just $\delta\boldsymbol{\theta}$ gave similar results but with the computational benefits of reducing the order of the filter in half. There were not enough resources to ensure excellent orthogonalization of the gyroscopes, and this was a reason for discussions since it was not possible to know whether the sources of the errors were from the estimator, from bad orthogonalizations or from misalignments of the gyroscopes with respect to the star cameras. In order to find answers to these questions, other estimators were developed that attempted to find possible residual errors.

In order to avoid that kind of problems, what is proposed is to use gyroscopes that come already orthogonalized from the factory. The proposed gyroscopes are from the same manufacturer, the TRS-500 of Optolink, with some characteristics worse than those used up to such as higher biases. With this gyroscopes, the original Kalman filter with bias estimation will have more sense. Another improvement that would reduce the errors caused by the chain of rotations involved in the attitude calculation of the telescope is to mechanically mount and align the star cameras and gyroscopes. In this way the reference axes of the star cameras and gyroscopes are nominally the same.

Most of the mechanical, optical and electronic components, as well as the software infrastructure, were specially designed and built from scratch for BETTII. It was not expected, then, a complete success in BETTII's first flight from the scientific point of view. Despite that, engineering wise, the flight was a success and it almost completed all the set goals for the flight. We can state that if BETTII receives enough funds to move forward, it will demonstrate with success the suitability of the double Fourier interferometry in space. This would lead to a new generation of space-based telescopes, capable of increasing considerably the obtained resolutions until now in the infrared region.

Personally, I highly value this experience, where I had the opportunity to work with high-level professionals that are doing cutting-edge research in a project that has been in development for over 6 years and I was able to see, first-hand, how its first launch materialized. Not only have I acquired new technical knowledge, but also I have seen the general view of how things are planned and organized and how a big and complex mission like this is carried out.

Appendix A

Equatorial coordinate system

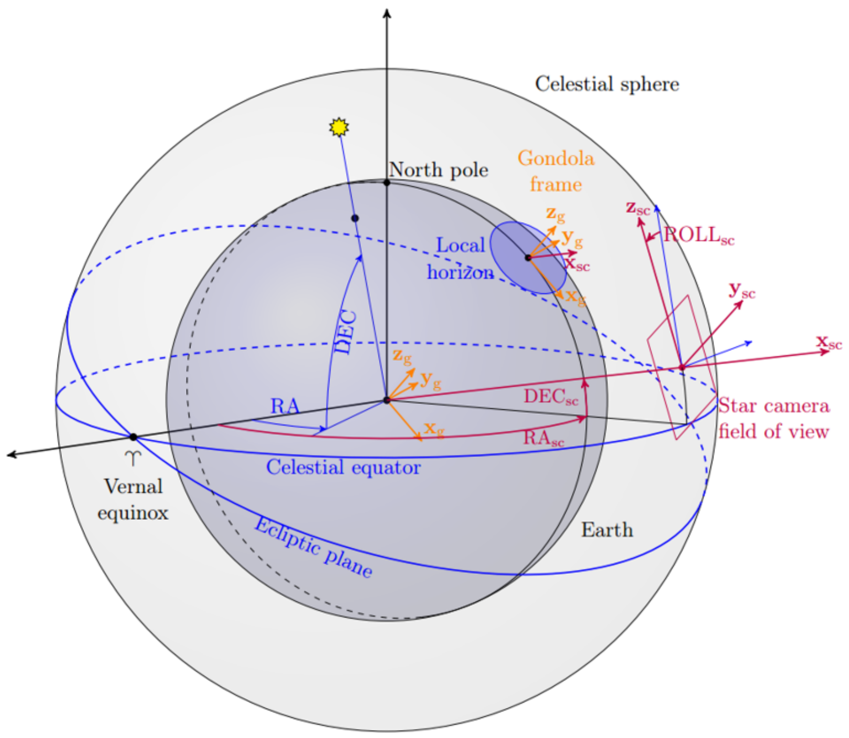


Figure A.1: BETTII reference frames on the equatorial coordinate system. Credit: Maxime Rizzo [Riz16]

In the figure A.1, the celestial sphere is shown as the outermost circle. A location of a star (yellow) in the sky is represented by a pair of spherical coordinates, the right ascension (RA) and the declination (DEC). The zero for both RA and DEC coordinates is the vernal equinox, that corresponds to one node of the intersection between the celestial equator and the ecliptic plane.

Appendix B

Flight plots

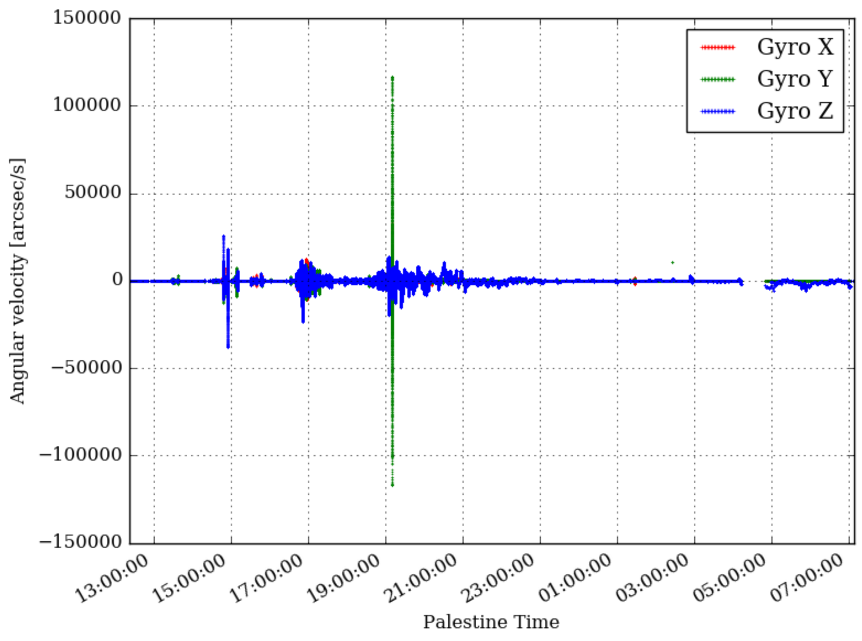


Figure B.1: Measured velocities from the three gyroscopes, in arcesc/s.

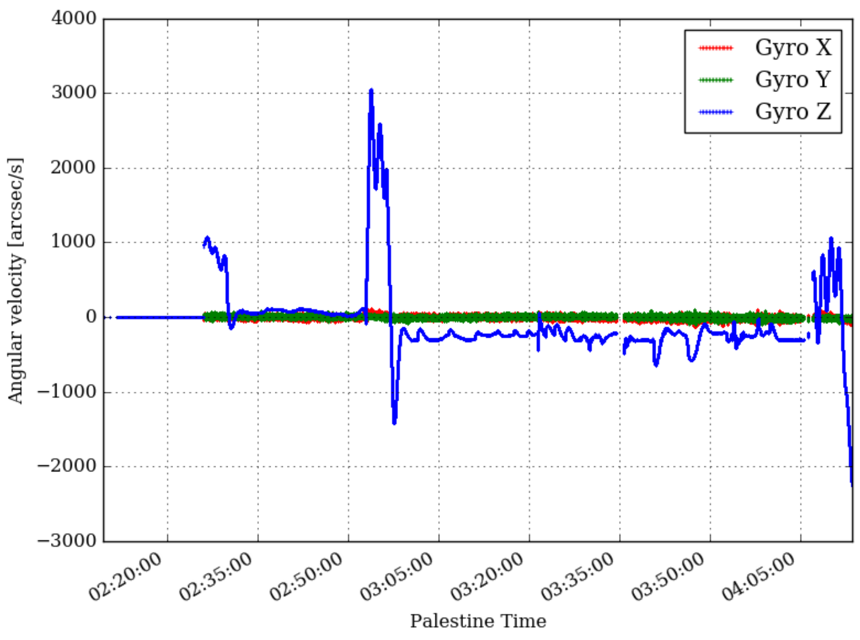


Figure B.2: Detail of the gyroscopes at the same time window of the estimator figures 3.5 and B.4

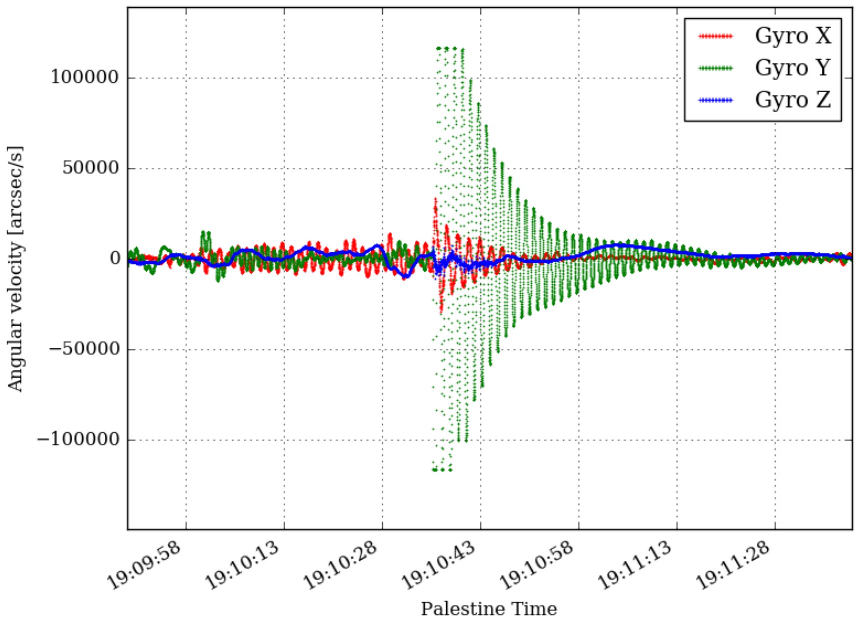


Figure B.3: Detail of the gyroscopes at the moment of launch.

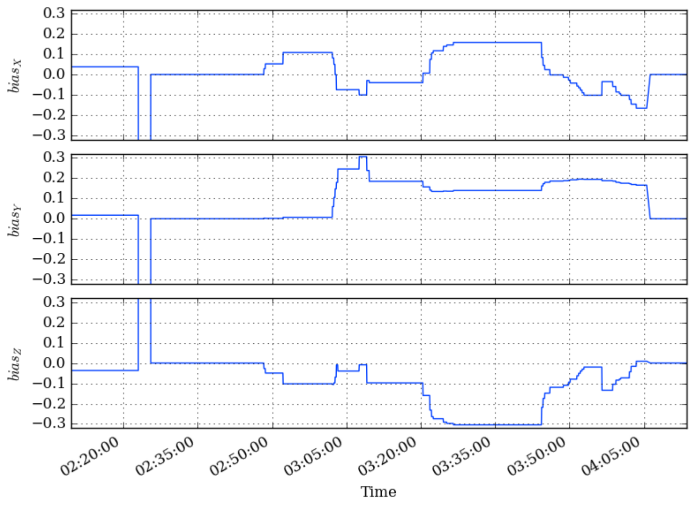


Figure B.4: Estimated gyroscope biases, in arcsec/s.

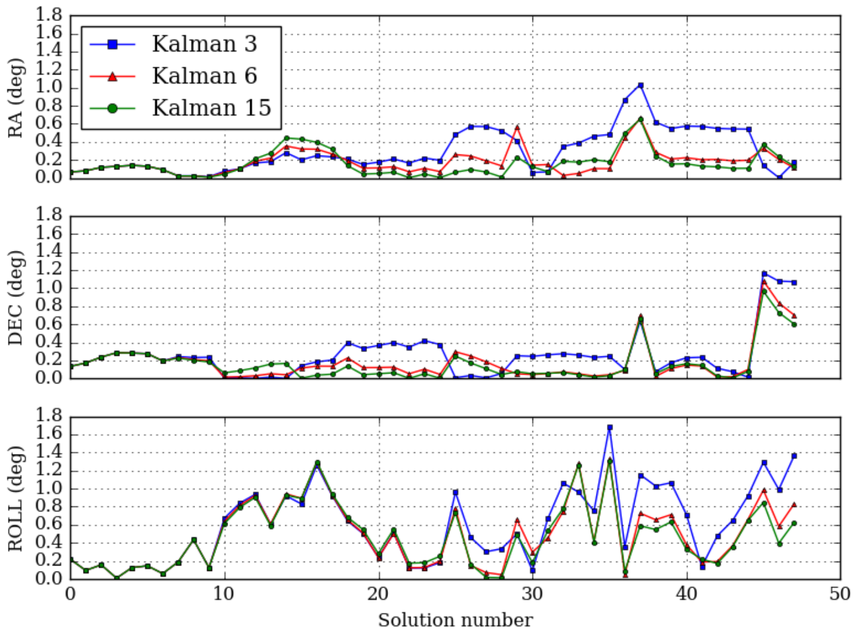


Figure B.5: Absolute error between the star camera measurement and the estimated attitude, in the star camera reference frame. The estimations are performed with a Python code on ground. Three different Kalman filters were used, with the same gains. The fifteen states Kalman filter [Riz16] estimates, in addition to the 6 states presented in this thesis, the nine elements of the matrix applied to the gyroscopes measurements.

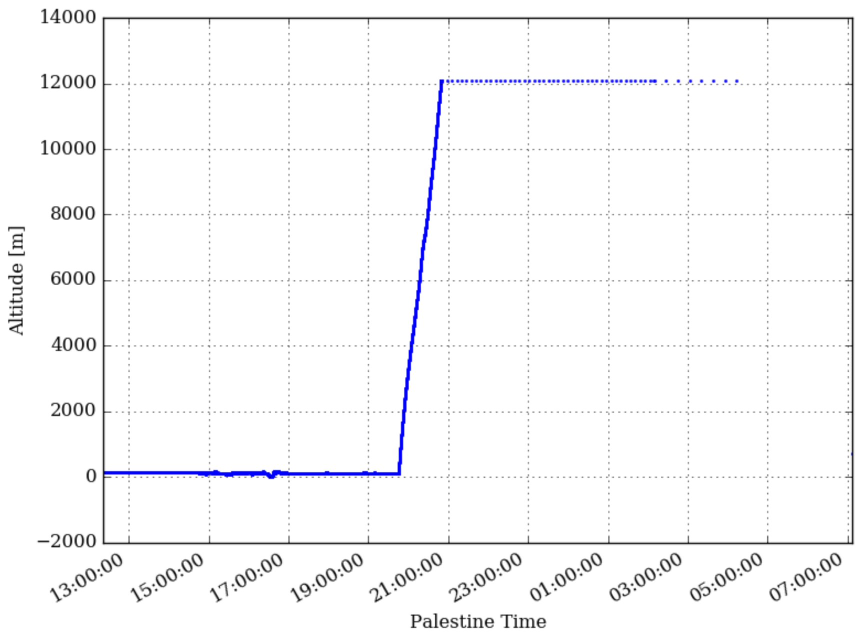


Figure B.6: Measured altitude from the GPS [uBl17]. The readouts failed above 12 km, issue explained briefly in section 2.3

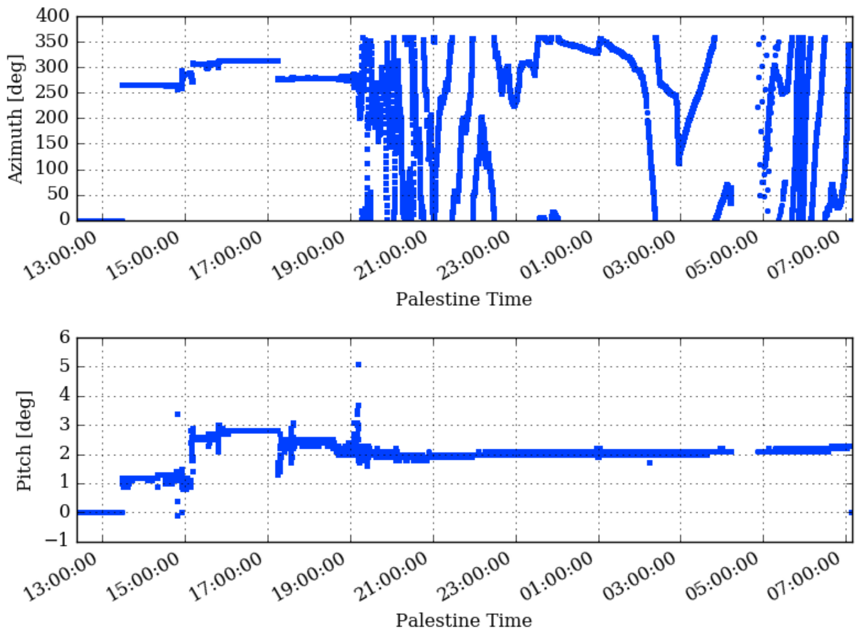


Figure B.7: Azimuth and Pitch measured by the magnetometer HM3500. The pitch is measured from a built-in accelerometer.

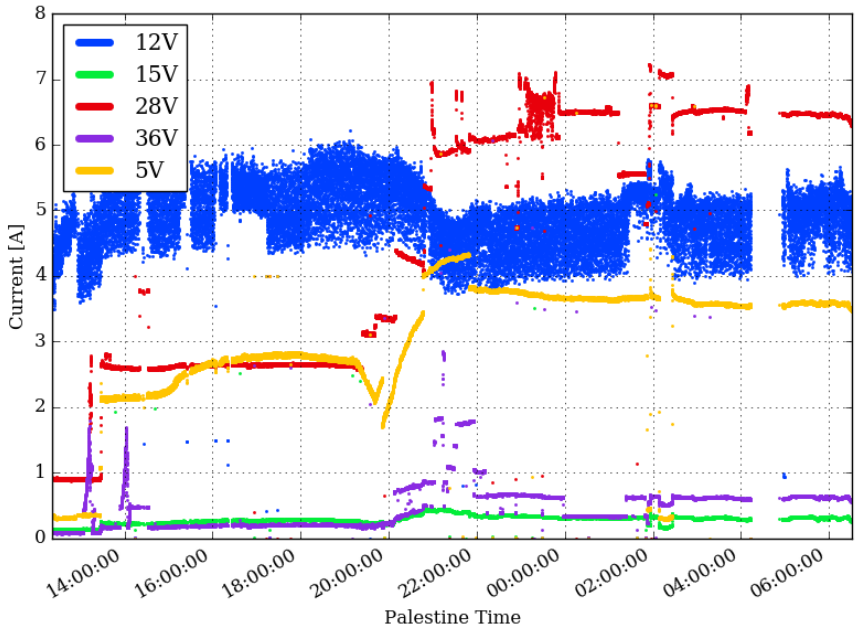


Figure B.8: Currents measured by the current sensors on the main power board. The implementation of this feature was one of my tasks during this internship. Sudden changes in the consumption at 36V and 28V lines are caused by turning on and off the heaters. The high, oscillating consumption in the 12V power line is caused by Ford and Boop, the on-board computers.

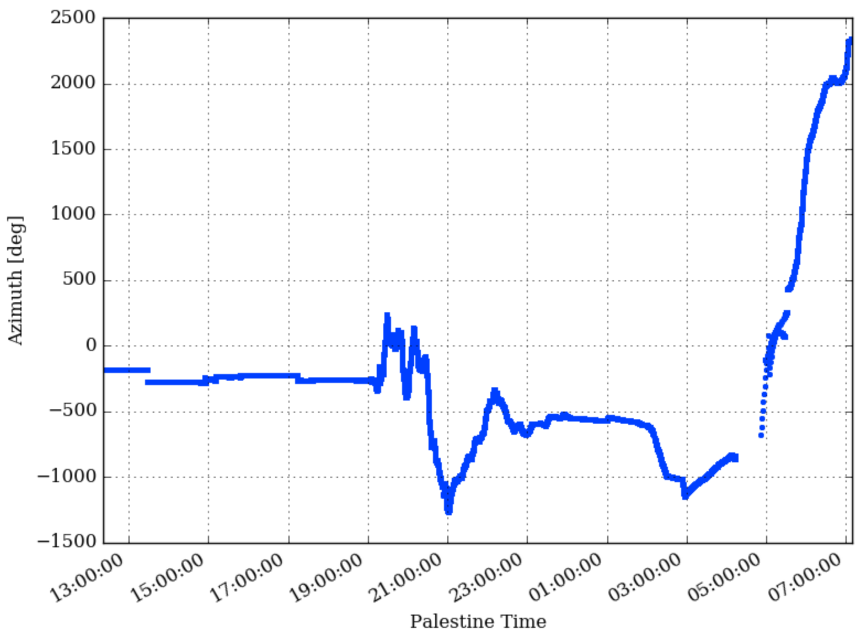


Figure B.9: Unwrapped azimuth measured by the magnetometer during the flight. It is the same azimuth data shown in figure B.7 but with the azimuth information unwrapped.

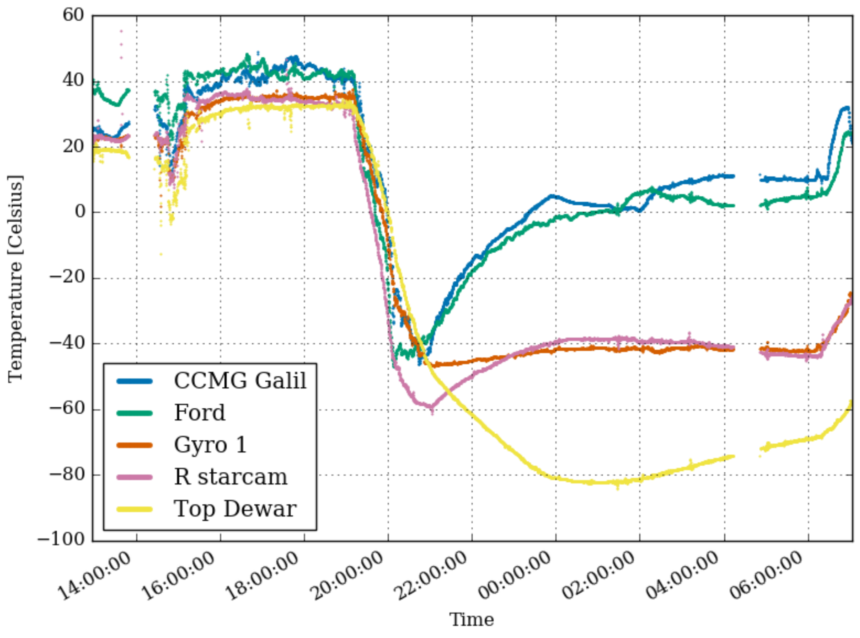


Figure B.10: Temperatures of BETTII during flight, measured from five different thermometers.

Appendix C

Delayed star camera solution

The star camera solution finder software takes several loop cycles, N in general, to calculate and send the attitude measurement $\boldsymbol{q}^{meas}$ and its uncertainties $\boldsymbol{R}$. Consequently, there exists a delay of a few seconds between the trigger of the star cameras and the reception of the solution.

More in detail, at an instant k the estimator will receive a measurement $\boldsymbol{q}_{k-N}^{meas}$ that corresponds to an image taken at the time step $k-N$. To calculate the actual measurement $\boldsymbol{q}_k^{meas}$ at the instant k, we will need to propagate $\boldsymbol{q}_{k-N}^{meas}$ and its covariance matrix $\boldsymbol{R}_{k-N}$. These propagations can be found following the equations 1.21 and 1.21:

$$\boldsymbol{q}_k^{meas} = \prod_{i=k-N}^{k} \exp\left(\frac{1}{2}\boldsymbol{\Omega}(\hat{\boldsymbol{\omega}}_i)\Delta t\right)\boldsymbol{q}_{k-N}^{meas} = \boldsymbol{C}_k \ \boldsymbol{q}_{k-N}^{meas} \tag{C.1}$$

$$\boldsymbol{R}_k = \boldsymbol{A}_k \boldsymbol{R}_{k-N} \boldsymbol{A}_k^T + \boldsymbol{B}_k \tag{C.2}$$

where the matrices $\boldsymbol{A}_k$ and $\boldsymbol{B}_k$ are defined recursively as

$$\boldsymbol{A}_k = \boldsymbol{\Phi}_k \boldsymbol{A}_{k-1} = \prod_{i=k-N}^{k} \boldsymbol{\Phi}_i \tag{C.3}$$

$$\boldsymbol{B}_k = \boldsymbol{\Phi}_k \boldsymbol{B}_{k-1} \boldsymbol{\Phi}_k^T + \boldsymbol{Q}_d \tag{C.4}$$

with $\boldsymbol{A}_{k-N} = \boldsymbol{I}_{6\times6}$ and $\boldsymbol{B}_{k-N} = \boldsymbol{0}_{6\times6}$.

When we trigger the star camera we will start to keep track of the matrices $\boldsymbol{A}_k$, $\boldsymbol{B}_k$ and $\boldsymbol{C}_k$, and then we will apply them once the estimator receives the star camera measurement.

Appendix D

Kalman filter Python code

The following code implements the 6 states Kalman filter used for the results shown in the post-flight estimation section 3.3.2. This code is part of a project developed for this thesis and available in the following GitHub repository: `https://github.com/androidside/pythonFlightDataProcessing`

```
class Estimator6(Estimator):
    '''
    Class implementing the original 6 state Kalman filter.
    '''
    EST_FILENAME=Estimator.EST_FILENAME+"6"
    def estimate(self,P0=0*np.eye(6),b0=np.zeros(3),q0=None,Qd=np.eye(6),SCg
   =5,ts=None,te=None,progress=False):
        gyros=self.gyros.loc[ts:te] #cropping time of the gyros pd.Dataframe()
   , ts: start time, te: end time
        if progress: #print progress of the estimator
            start_time = timer()
            printOn=0
            print "Initializing..."
        if q0 is None: #initializing the estimated attitude quaternion
            fig=gyros.index[0]
            fis=self.sc.index[0]
            if fig>=fis: #if gyros start later than the first SC solution
                isc=next((i for i,ind in enumerate(self.sc.index) if ind>fig))
    #first index of the SC where gyros are available
                q0=self.sc.qI2G.iloc[isc-1] #nearest past starcamera solution
   to the start of the gyros assigned to the quaternion
                sc=self.sc.iloc[isc:]
            else: #if gyros start before than the first SC solution
                q0=self.sc.qI2G.loc[fis]
                ig=next((i for i,ind in enumerate(gyros.index) if ind>fis)) #
   first index of the Gyros where SC solutions are available
                gyros=gyros.loc[ig:]
                sc=self.sc.iloc[1:]
        else: sc=self.sc.loc[ts:te]
        nextSCindex=0
        i0=gyros.index[0]
        q_prop=q0

        bias=b0
        L=len(gyros.index)
        props=[q_prop]*L
        i_prop=[i0]*L
        biases=[bias]*L
        Ps=[P0]*L

        C=np.matrix([[0.999999, 0.000309,-0.002822],[0.001062,
    0.999995,-0.0031],[0.00129,0.002992,0.999991]]) #gyros alignment matrix
        P=P0
        H=np.eye(3,6)
        dt=0.01

        if progress: print "Starting estimation:"
        for j in range(L-1):
            #Prediction
            dt=(gyros.index[j+1]-gyros.index[j])/400.
            w=(gyros.iloc[j,:3].as_matrix())*(1/3600.*np.pi/180)-bias #
    arcsec2rad conversion - bias
            w=np.asarray(C.dot(w).T).T[0]
            wx=w[0];wy=w[1];wz=w[2]
            Ow=np.matrix([[0,wz,-wy,wx],[-wz,0,wx,wy],[wy,-wx,0,wz],[-wx,-wy,-
    wz,0]]) #Omega(omega)
            A=expm(0.5*Ow*dt)
            q_prop=Quat(A.dot(q_prop.q))
            Theta=np.eye(3)-dt*vec2skew(w)
            Psi=-dt*np.eye(3)
            Phi=np.concatenate((np.concatenate((Theta,np.zeros((3,3)))),np.
    concatenate((Psi,np.eye(3)))),axis=1)
            P=Phi.dot(P.dot(Phi.T))+Qd

            ind=gyros.index[j]
            if nextSCindex < len(sc.index) and ind >= sc.index[nextSCindex]: #
    Update!
                qmeas=sc.qI2G.iloc[nextSCindex]
                dq=qmeas*q_prop.inv()
                z=dq.q[:3]
                Mrot=np.matrix
    ([[0.693865,0,0.720106],[0,1,0],[-0.720106,0,0.693865]]) #matrix used in
    LabView to rotate R to the Gyros ref. frame (~46 deg)
                R=(SCg*np.diag(sc.iloc[nextSCindex][['ra_err','dec_err','
    roll_err']].tolist()))**2 #measured uncertainties multiplied by the gain
    SCg (=5 by default)
                R=Mrot*R*Mrot.T
                S=np.matrix(H.dot(P.dot(H.T))+R)
                K=P.dot((H.T).dot(S.I))
                x=np.asarray(K.dot(z))[0]
                dq=x[:3]
                db=x[3:]
                n=dq.dot(dq)
                if n<=1:
                    qd=Quat([dq[0],dq[1],dq[2],np.sqrt(1-n)])
                else:
                    qd=Quat(np.array([dq[0],dq[1],dq[2],1])/np.sqrt(1+n))
                q_prop=qd*q_prop

                bias=bias+db
                J=(np.eye(6)-K.dot(H))
                P=J.dot(P.dot(J.T))+K.dot(R.dot(K.T))
                nextSCindex=nextSCindex+1
```

```
                if progress: print "UPDATE! (%s/%s)" % ((nextSCindex+1),len(sc
    .index))
            props[j+1]=q_prop
            biases[j+1]=bias
            i_prop[j+1]=ind
            Ps[j+1]=P

            percentage=100.0*j/L
            if progress and percentage>=printOn:
                et=timer()-start_time
                print 'Estimating %0.1f%%' % percentage
                print "Elapsed time: %0.2f seconds." %et
                if percentage>0: print "Remaining time: %0.2f minutes." % (et/
    percentage*(100-percentage)/60)
                print "Data duration: %0.2f minutes" % (L/2400.)
                printOn=printOn+0.1

        d={'qest': props,
           'biases':biases,
           'biasX':[bias[0] for bias in biases],
           'biasY':[bias[1] for bias in biases],
           'biasZ':[bias[2] for bias in biases],
           'RA':[q.ra for q in props],
           'DEC':[q.dec for q in props],
           'ROLL':[q.roll for q in props],
           'P':Ps
           }
        self.est=pd.DataFrame(d,index=i_prop)
```

Bibliography

[Bre99] W. G. Breckenridge. *Quaternions - Proposed Standard Conventions.* Tech. rep. INTEROFFICE MEMORANDUM IOM 343-79-1199. JPL, 1999 (cit. on p. 11).

[Riz16] Maxime Rizzo. "BETTII: A pathfinder for high angular resolution observations of star-forming regions in the far-infrared." PhD thesis. University of Maryland, 2016 (cit. on pp. 10, 13, 32, 37, 41).

[TR05] Nikolas Trawny and Stergios I Roumeliotis. "Indirect Kalman filter for 3D attitude estimation." In: *University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep* 2 (2005) (cit. on pp. 13, 14).

[uBl17] uBlox. *u-blox M8 - Receiver Description.* `https://www.u-blox.com/sites/default/files/products/documents/u-blox8-M8_ReceiverDescrProtSpec_(UBX-13003221)_Public.pdf`. 2017 (cit. on pp. 22, 42).

[Vil14] Jordi Vila. "Telemetry System of the Ballon Experimental Twin Telescope for Infrared Interferometry." MA thesis. ISAE-SUPAERO, 2014 (cit. on p. 15).

Abstract — The Balloon Experimental Twin Telescope for Infrared Interferometry (BETTII) is a far infrared telescope on a balloon-based platform that will validate the interferometry technique for future space telescopes. It is one of the most complex balloon projects carried out at NASA-GSFC and it requires a lot of different subsystems. The work during this thesis is centered around the coarse control system of the telescope's attitude and the analysis of the data collected during its first flight.

Keywords: balloon, interferometry, telescope, infrared, bettii, astronomy, control system

NASA - Goddard Space Flight Center
Greenbelt, MD