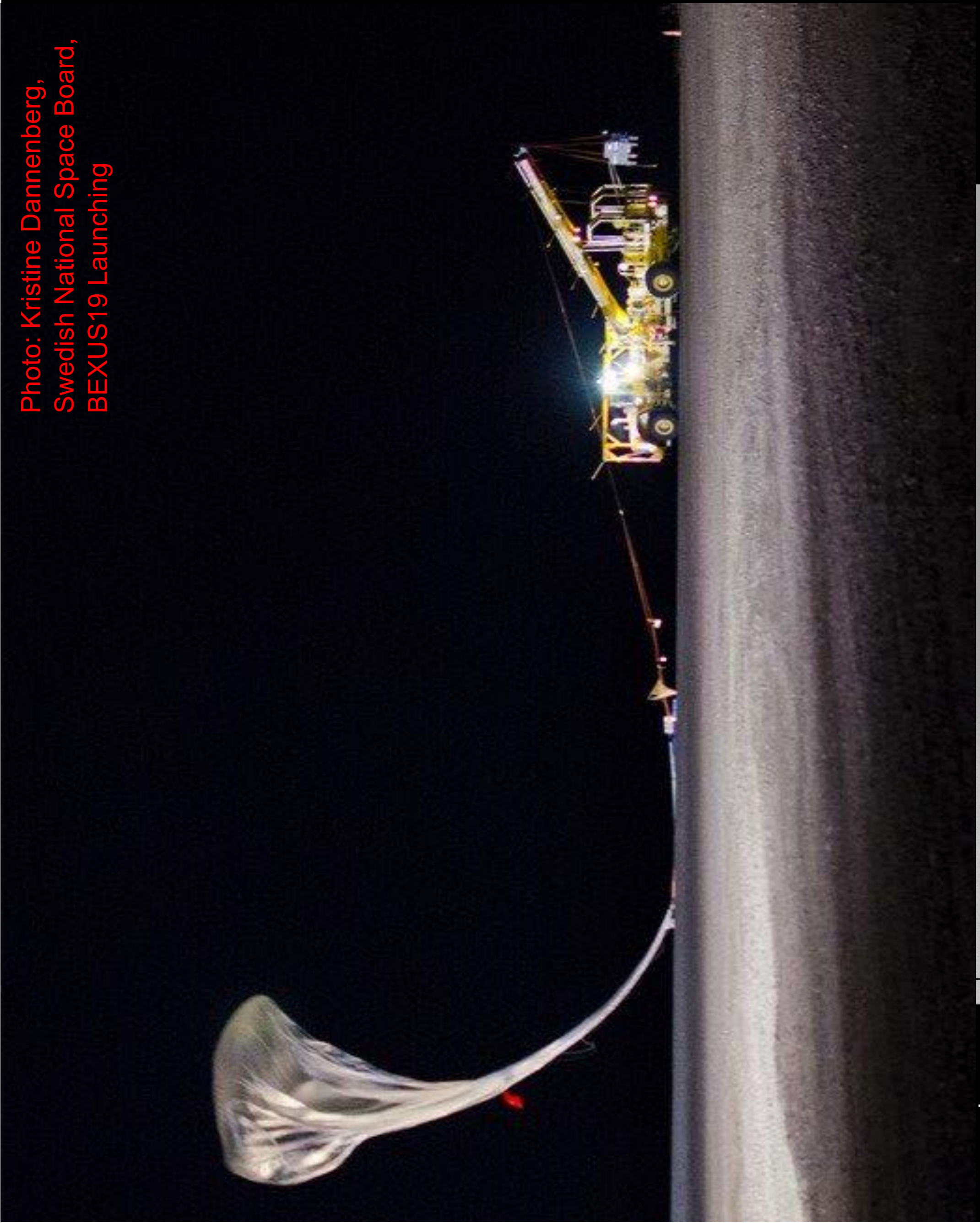


Photo: Kristine Dannenberg,
Swedish National Space Board,
BEXUS19 Launching

PROYECTO
FIN DE
CARRERA

Star Tracker for BEXUS19 Mission

Manuel Milla
Peinado

INGENIERÍA
TELECOMUNICACIÓN

ESTUDIOS DE INGENIERÍA
DE TELECOMUNICACIÓN

PROYECTO FIN DE CARRERA

Star Tracker for BEXUS19 Mission

Manuel Milla Peinado

Curso 2014/2015
Tutor: Andrés María Roldán Aranda

Universidad de Granada

ESTUDIOS DE INGENIERÍA
DE TELECOMUNICACIÓN

PROYECTO FIN DE CARRERA

"Star Tracker for BEXUS19 Mission"

CURSO: 2014/2015

Manuel Milla Peinado

ugr Universidad de Granada

ESTUDIOS DE INGENIERÍA DE TELECOMUNICACIÓN

"Star Tracker for BEXUS19 Mission"

REALIZADO POR:

Manuel Milla Peinado

DIRIGIDO POR:

Andrés María Roldán Aranda

DEPARTAMENTO:

Electrónica y Tecnología de los Computadores

El tribunal constituido para la evaluación del proyecto PFC titulado:

"Sistema de caracterización de dispositivos magnetorresistivos"

Realizado por el alumno: Ignacio Rodríguez López

Y dirigido por el tutor: Andrés María Roldán Aranda

Ha resuelto asignarle la calificación de:

□ SOBRESALIENTE (9 - 10 puntos)

□ NOTABLE (7 – 8.9 puntos)

□ APROBADO (5 – 6.9 puntos)

□ SUSPENSO

Con la nota[1]: □ puntos.

El Presidente:

El Secretario:

El Vocal:

Granada, a _____ de ______________ de 2013

[1] Solamente con un decimal.

D. Andrés María Roldán Aranda, Profesor del departamento de Electrónica y Tecnología de los Computadores de la Universidad de Granada, como director del Proyecto Fin de Carrera de D. Manuel Milla Peinado,

Informa:

que el presente trabajo, titulado:

"Star Tracker for BEXUS19 Mission"

ha sido realizado y redactado por el mencionado alumno bajo nuestra dirección, y con esta fecha autorizo a su presentación.

Granada, a de Diciembre de 2014

Fdo.

Los abajo firmantes autorizan a que la presente copia de Proyecto Fin de Carrera se ubique en la Biblioteca del Centro y/o departamento para ser libremente consultada por las personas que lo deseen.

Granada, a de Abril de 2013

Fdo.

Star Tracker for BEXUS19 Mission

Manuel Milla Peinado

PALABRAS CLAVE:

Star Tracker, attitude,CCD, camera, FOV, Hipparcos Catalog, ICRS, algorithm, BEXUS, MATLAB, Raspberry Pi, Orion, pattern,Capricorn, baffle, stellar database, HIP, identify, flight, analysis, ESA ,SNSB ,DLR.

RESUMEN:

El objetivo de este trabajo es el diseño de un star tracker para su uso en un globo estratosférico, este trabajo ha sido realizado dentro del programa REXUS/BEXUS, el autor del mismo fue miembro del equipo GranaSAT que participó en este programa durante la campaña BEXUS18/19.

KEYWORDS:

El objetivo de este trabajo es el diseño de un star tracker para su uso en un globo estratosférico, este trabajo ha sido realizado dentro del programa REXUS/BEXUS, el autor del mismo fue miembro del equipo GranaSAT que participó en este programa durante la campaña BEXUS18/19.

ABSTRACT:

The aim of this project is the implementation of its use in a stratospheric ballon, this project was realised under the REXUS/BEXUS programme, the author was member of the GranaSAT team who took part in the programme during the BEXUS18/19 campaign.

Dedicado a

Mis padres, Manuel y Sacramento. Sin ellos nada de esto hubiera sido posible.

Agradecimientos:

En primer lugar, quisiera expresar todo mi agradecimiento a mi familia. Gracias a mis padres por el apoyo constante y la confianza depositada, por haber estado siempre a mi lado y por todo el esfuerzo que han dedicado a formarme. Gracias por comprenderme en todo momento y sacar lo mejor de mí en los momentos de flaqueza. Personalmente, gracias a mi padre por inculcarme siempre valores de esfuerzo y superación. A mi madre, gracias por conocerme como ninguna otra persona y saber sacarme siempre una sonrisa. Sin vosotros no habría llegado hasta aquí.

Darle las gracias también a mi hermano Juan Manuel, pues con él he compartido innumerables horas de trabajo y diversión durante todo este tiempo. Tú más que nadie has sabido sobrellevar mi carácter, gracias por tu cercanía y por todas las risas que compartimos.

No quisiera olvidarme del resto de mi familia. Gracias a todos ellos por su afecto, especialmente a mis abuelos por haberme tenido siempre presente.

Mostrar también mis agradecimientos al Departamento de Electrónica y Tecnología de los Computadores de la Universidad de Granada, y muy especialmente a mi tutor, Andrés María Roldán Aranda. Gracias por tu constante dedicación, atención y supervisión durante todo este tiempo. Por tu trato cercano y amable, y por todo el conocimiento transmitido.

Todos son igualmente partícipes de este trabajo.

INDEX

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

ϵ Relative error in the angle between two stars.

k k-vector.

ABS Acrylonitrile Butadiene Styrene.

ADS Attitude Determination System.

CCD Coupled Charged Devide.

CMOS Complementary Metal Oxide Semiconductor.

CPU Central Processing Unit.

DEC Declination, δ.

DLR German Aerospace Center.

ECI Earth-Centered Inertial.

ECTD Electronics and Computer Technology Department.

ESA European Space Agency.

ESTEC Space Research and Technology Centre.

FOV Field Of View.

GC Generated Catalog.

GPIO General Purpose Input/Output.

HIP Hipparcos Identification Number.

HP Hipparcos Catalog.

I2C Inter-Integrated Circuit.

IAU International Astronomy Union.

ICRS International Celestial Reference System.

LED Light-Emitting Diode.

LIS Lost In Space.

MORABA MObile ROcket BAsis.

QE Quantum Efficiency.

RA Right Ascension, α.

RAM Random Access Memory.

ROI Region Of Interest.

SAIL Space Avionics and Instrumentation Laboratory.

SED Student Experiment Document.

SNSB Swedish National Space Board.

SSC eSrange Space Center.

TCP/IP Transmission Control Protocol / Internet Protocol.

UGR University of Granada.

USB Universal Serial Bus.

UTIAS SFL University of Toronto Institute for Aerospace Studies Space Flight Laboratory .

WBS Work Breakdown Structure.

ZARM Center of Applied Space Technology and Microgravity.

CHAPTER 1 INTRODUCTION

The following master thesis was carried out as a partial fulfillment of the 3+2 Telecommunication Engineering degree. The objective of this work was the designing, implementation and testing of a Star Tracker for its use in a stratospheric balloon under the REXUS/BEXUS programme, defined in its webpage as follows [37]:

> *The REXUS/BEXUS programme is realised under a bilateral Agency Agreement between the German Aerospace Center (DLR) and the Swedish National Space Board (SNSB). The Swedish share of the payload has been made available to students from other European countries through a collaboration with the European Space Agency (ESA).*
>
> *EuroLaunch, a cooperation between the Esrange Space Center (SSC) and the Mobile Rocket Base (MORABA) of DLR, is responsible for the campaign management and operations of the launch vehicles. Experts from DLR, SSC, ZARM and ESA provide technical and logistic support to the student teams throughout the project.*

The author of this document was part of the GranaSAT team, a group of eleven students from the University of Granada (UGR) (Spain) selected for the BEXUS18/19 campaign during the selection workshop at the European Space Research and Technology Centre (ESTEC) (December 3^{rd}-5^{th} Nordwijk, Holland).

For a period of one year, the team designed an Attitude Determination System (ADS) based on three different methods, namely, a Horizon Sensor, a Magnetometer ADS and a Star Tracker. The October, 8^{th}, 2014 the experiment was launched in Kiruna (Sweden) from the Esrange Space Center(SSC) facilities and tested during two and a half hours of flight. It was proven to be a complete success and laid the bases for the development of the GranaSAT nano-satellite.

1.1 Chapter Description

- **Chapter 2: System Analysis:** in this chapter, we will analyse how a Star Tracker can be used to determine the attitude of a spacecraft. We start outlining the GranaSAT experiment, its purposes and the subsystem involved.

  In section 2.1, we first introduce how the attitude of a spacecraft can be determined through stellar observations. Some commercial star trackers its performance is described in 2.2. A brief survey of identifications algorithms is carried out in section 2.2.1.

  Finally, in section 2.3; we finish this chapter showing the specific requirements for our Star Tracker.

- **Chapter 3: Hardware Implementation** in this chapter, we will describe the hardware chosen for the GranaSAT Star Tracker subsystem. Three main components had to be chosen, these are a camera (section 3.1), a baffle (section 3.2) and a computer (section 3.3)

- **Chapter 4: Software Implementation** in this chapter, the algorithms programmed will be described. We can split the software development in two stages, one concerning the data which the Star Tracker needs to work properly (section 4.1) and other which implements the software running during the flight (section 4.2).

- **Chapter 5: Verification and Testing** in this chapter, we will explain the test procedures followed before the flight. Later, an entire section 5.6 is dedicated to discuss the results obtained during the flight. An additional section 5.6.1.1 is dedicated to study these situations which made the Star Tracker algorithms fail.

- **Chapter 6: Conclusions and Future lines**: the conclusions derived from the participation of the author in the REXUS/BEXUS programme are discussed in this section. We also describe briefly the possible improvements and future work related with this project.

CHAPTER 2 ANALYSIS

The GranaSAT experiment was mainly aimed to test in real flight conditions three ADS (Star Sensor, Horizon Sensor and Magnetometer) and evaluate their possible implementation in a future nano-satellite designed by students of the University of Granada. Each ADS was designed by a different student. Figure 2.1 shows the Work Breakdown Structure(WBS), which is the task assignment to each student.

Both Star Tracker and the Horizon Sensor subsystems shared the same CCD camera (DMK 41BU02)[23] for the image acquisition. Magnetic field measurements were taken by the LSM303[47] magnetometer. The box which contains these components was called *upper box*. Apart from the mentioned camera and magnetometer sensor, the *upper box* contained I2C temperature sensors and the power electronics required for its operation. Figure 2.2 shows a sketch of this box and its iteration with the *inner box*, which will be described below.

The *inner box* (figure 2.3), mainly contained the experiment computer and the power electronics required for its operation. The power for BEXUS experiments is provided by a pack of 8 series LSH20[40] batteries. The maximum current consumption is limited to 1A. The data obtained during the flight was real-time processed by a Raspberry Pi Model B[39] and sent to the Ground Station.

The communication from the ground station to the student experiments is provided by the E-link[13] (Figure 2.4). The E-Link consists on the Ground Unit equipment and an Airborne Unit (figure 2.5). Technical information of both units is shown in table 2.1.

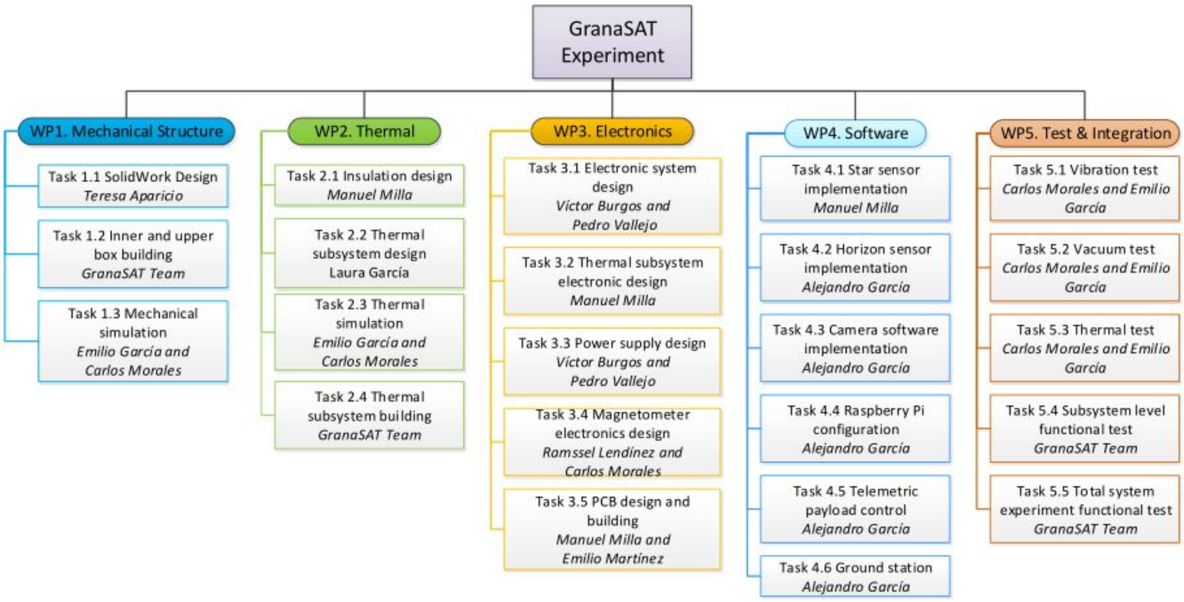

GranaSAT Experiment
WP1. Mechanical Structure
Task 1.1 SolidWork Design Teresa Aparicio
Task 1.2 Inner and upper box building GranaSAT Team
Task 1.3 Mechanical simulation Emilio García and Carlos Morales
WP2. Thermal
Task 2.1 Insulation design Manuel Milla
Task 2.2 Thermal subsystem design Laura García
Task 2.3 Thermal simulation Emilio García and Carlos Morales
Task 2.4 Thermal subsystem building GranaSAT Team
WP3. Electronics
Task 3.1 Electronic system design Víctor Burgos and Pedro Vallejo
Task 3.2 Thermal subsystem electronic design Manuel Milla
Task 3.3 Power supply design Víctor Burgos and Pedro Vallejo
Task 3.4 Magnetometer electronics design Ramssel Lendínez and Carlos Morales
Task 3.5 PCB design and building Manuel Milla and Emilio Martínez
WP4. Software
Task 4.1 Star sensor implementation Manuel Milla
Task 4.2 Horizon sensor implementation Alejandro Garcia
Task 4.3 Camera software implementation Alejandro Garcia
Task 4.4 Raspberry Pi configuration Alejandro Garcia
Task 4.5 Telemetric payload control Alejandro Garcia
Task 4.6 Ground station Alejandro Garcia
WP5. Test & Integration
Task 5.1 Vibration test Carlos Morales and Emilio García
Task 5.2 Vacuum test Carlos Morales and Emilio García
Task 5.3 Thermal test Carlos Morales and Emilio García
Task 5.4 Subsystem level functional test GranaSAT Team
Task 5.5 Total system experiment functional test GranaSAT Team

Figure 2.1 – *GranaSAT WBS*

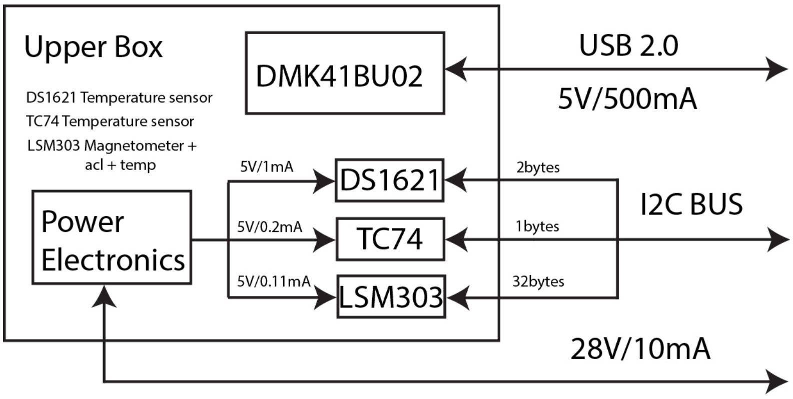


Figure 2.2 – *Upper Box Sketch*

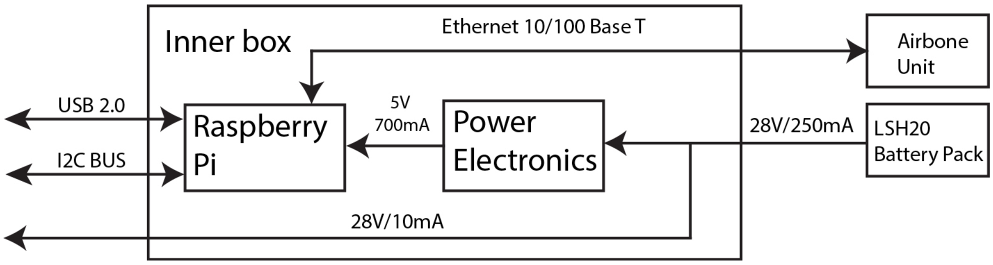


Figure 2.3 – *Inner Box Sketch*

The E-Link allows the students to communicate with their experiment through TCP/IP protocols using an Ethernet 10/100 Base-T interface connected to the Airborne Unit. At least one interface is provided for each experiment. The radio interface is transparent to the BEXUS experiments and provides a channel with 2Mbps of bandwidth which has to be shared for all the experiments in the Gondola. Thanks to the E-Link features it was possible to implement a server-client architecture to control the experiment remotely.

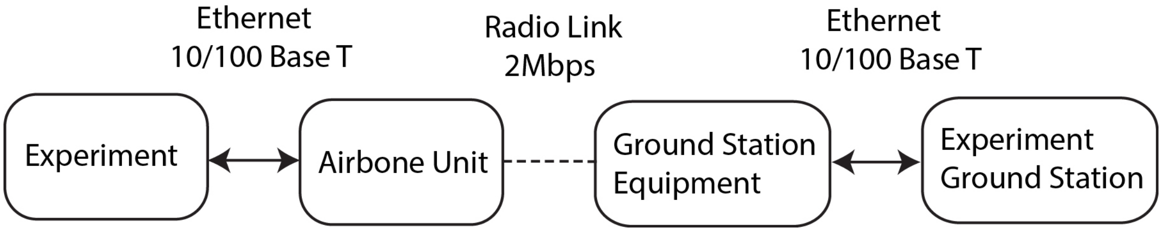


Figure 2.4 – *BEXUS Communication*

The flight configuration of the experiment installed in the BEXUS Gondola is shown in figure 2.6. For a complete description of the GranaSAT experiment (not only the Star Tracker part) see the Student Experiment Documentation (SED)[48], which can be obtained mailing to *grupo.granasat@gmail.com*.

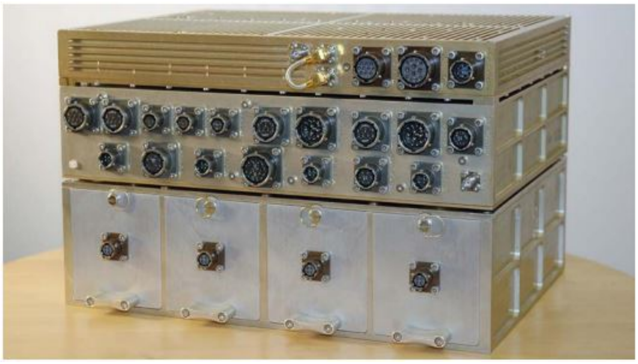


Figure 2.5 – *Airborne Unit*

2.1 The Attitude Estimation Problem

The spatial orientation of a spacecraft is called *attitude*. The attitude can be determined with respect to a reference frame through observations in the spacecraft frame (figure 2.7). In 1902, the W. Right brothers successfully designed and tested the first 3-axis controllable airplane [15].

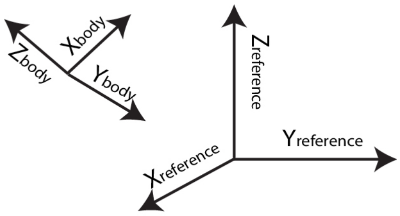


Figure 2.7 – *Reference Frames*

	Airborne Unit	**Ground Unit**
Antenna	Vertical Polarised ommi	1.8 meter parabolic dish
Operating frequency	S-band	S-band
Max Output Power	Peak 10 watt	Peak 10 Watt
Modulation	DSSS	DSSS
Channel bandwidth	Nominal ± 11MHz	Nominal ± 11MHz
Maximun range at LOS	500 km at 30 km altitude (TBD)	500 km at 30 km altitude (TBD)
Data bandwidth	2 Mbps duplex nominal	2 Mbps duplex nominal
User interfaces	2 Ethernet 10/100 Base 3 asynchronous duplex RS-232/422 channels	Ethernet 10/100 Base-T 2 asynchronous RS-232/422 channels 1 synchronous channel up to 1 Mbps
Power supply	20 to 38 volt DC	-
Operation time	Nominal > 11 hours	-
Weight	Nominal 20 Kg with batteries	-

Table 2.1 – *E-Link Units*

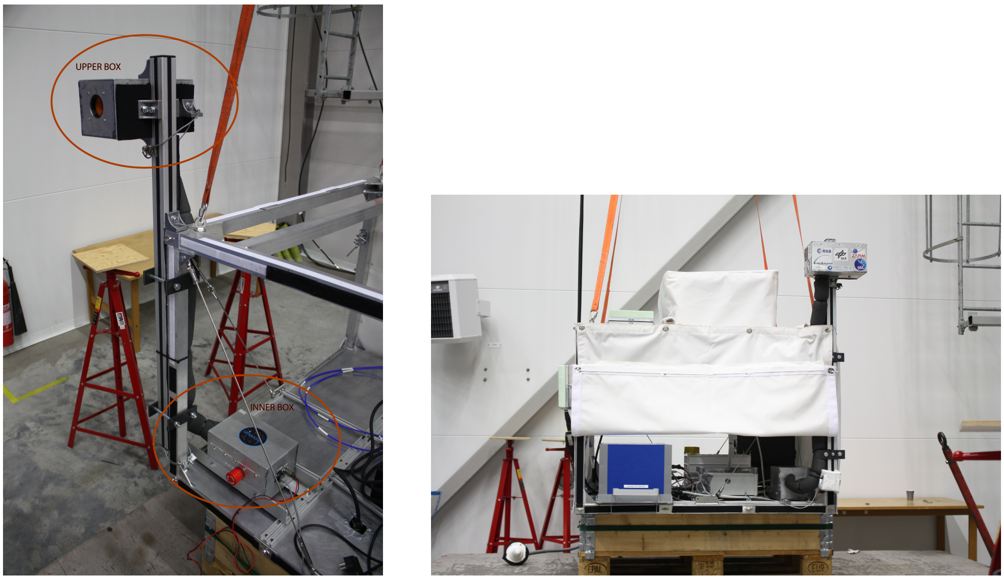
The (ICRS) is the current celestial reference system since 1st January, 1998. The IAU made this decision in August of 1997 in its 23rd General Assembly [14]. The origin of the celestial coordinates in the ICRS is the barycenter of the solar system. As it appears in [21] (figure 2.8a), the pole at J2000.0[1] is shifted from the ICRS pole by 17.3 ± 0.2 mas[2]. The Right Ascension (RA) origin is offset from the mean equinox at J2000.0 by 78 ± 10 mas. Figure 2.8b shows the graphical explanation of the RA and Declination (DEC), the equatorial coordinates at J2000 epoch. The origin of the J2000 equatorial coordinates is the center of the Earth.

The celestial coordinates of stars and proper motions in the Hipparcos catalog are expressed in the ICRS for the J1991.25 epoch [12]. Using the proper motion, parallax and orbital elements, the coordinates for a star in a certain epoch can be moved to any arbitrary epoch. This will be useful later for the development of a stellar database in section 4.1.1

The earliest solution to the problem of estimating the rotation matrix between two coordinate systems (i.e the body frame and the reference frame) was first introduced by Harold in 1964 [18]. Later in 1965, Grace Wahba, a PhD student in a summer job in IBM,

[1] In astronomy, epoch JXXXX.YY is an instant in time arbitrarily selected as a point of reference. For example, J2000 means 12h on January 1

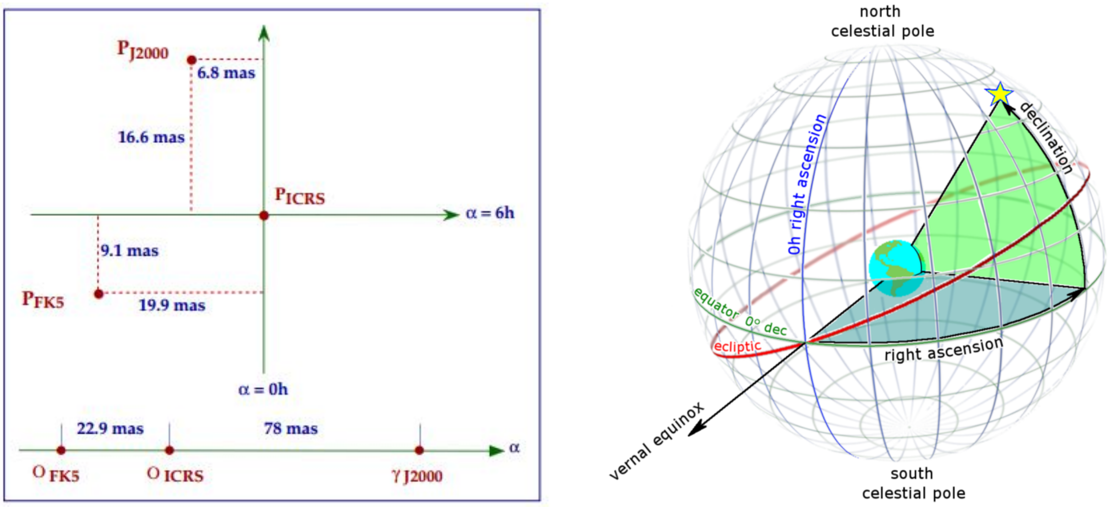
[2] milliarcsecond



(a) *GranaSAT Experiment Flight Configuration.*

(b) *Gondola*

Figure 2.6 – *GranaSAT Experiment*



(a) *Location of the poles and the origin of right ascensions of the FK5 and that of the celestial pole and equinox at J2000.0 with respect to the ICRS [21].*

(b) *Equatorial coordinates.*

Figure 2.8 – *ICRS and Equatorial Coordinates*

formulated the generalization of the attitude problem [56].

Formally, the problem was to find the rotation matrix M which brings the best least squares coincidence of a set of v_n^* points in the spacecraft frame to their coordinates v_n in a known reference frame, Equation 2.1.1.

Several algorithms have been proposed to solve the Wahba's problem, in sections 2.1.1,2.1.2 and 2.1.3 we present 3 of the most used ones.

$$\sum_{j=1}^{n} \left\| v_j^* - M v_j \right\|^2 \tag{2.1.1}$$

2.1.1 TRIAD Method

This method was first proposed by Black in 1964[18] before the advent of the Wahba's problem. It can be seen as a particular deterministic solution using just the information of a pair of vectors (i.e the vector which points towards the sun and the Earth magnetic field vector).

If we know the coordinates of two vectors in the spacecraft reference frame $\vec{r}$ and $\vec{b}$ their measured values in a well-known inertial reference frame. It exists a matrix A that:

$$\vec{b_i} = A\vec{r_i} \quad \text{where } i\text{=1,2} \tag{2.1.2}$$

The TRIAD method estimates the matrix A as follows:

$$\hat{S} = \frac{\vec{r}_1}{\|\vec{r}_1\|} \tag{2.1.3}$$

$$\hat{s} = \frac{\vec{b}_1}{\left\|\vec{b}_1\right\|} \tag{2.1.4}$$

$$\hat{M} = \frac{\vec{r}_1 \times \vec{r}_2}{\|\vec{r}_1 \times \vec{r}_2\|} \tag{2.1.5}$$

$$\hat{m} = \frac{\vec{m}_1 \times \vec{m}_2}{\|\vec{m}_1 \times \vec{m}_2\|} \tag{2.1.6}$$

The attitude matrix is finally:

$$\hat{A} = \left[\hat{S} \vdots \hat{M} \vdots \hat{S} \times \hat{M}\right] \left[\hat{s} \vdots \hat{m} \vdots \hat{s} \times \hat{m}\right]^T \tag{2.1.7}$$

2.1.2 QUEST

In contrast to the TRIAD method which uses only information of two vectors to compute the attitude, the QUEST method is a statistical procedure which uses information from all measured vectors to solve the Wahba's problem (i.e in a Star Tracker we would have a measured vector from each identified star). This method is described in[43].

The QUEST algorithm approximates the q-method, which is described in [17]. QUEST is less computationally expensive than the q-method due to its approximation of the largest eigenvalue problem derived from the q-method.

The quaternion attitude is computed finding the Rodriguez parameters in the following Equation 2.1.8. It involves to solve multiple linear equations. Gaussian elimination method is usually applied.

$$\left[(\lambda_{opt} + \sigma)\mathbf{1} - \mathbf{S}\right]\mathbf{p} = \mathbf{Z} \tag{2.1.8}$$

where:

$$\mathbf{B} = \sum_{k=1}^{N} w_k(\mathbf{v}_{kb}\mathbf{v}_{ki}^T) \tag{2.1.9}$$

$$\lambda_{opt} = \sum w_k \tag{2.1.10}$$

$$\sigma = tr(\mathbf{B}) \tag{2.1.11}$$

$$\mathbf{B} = \mathbf{B} + \mathbf{B}_T \tag{2.1.12}$$

$$\mathbf{p} = \text{Rodriuez parameters} \tag{2.1.13}$$

$$\mathbf{Z} = [B_{23} - B_{32}B_{31} - B_{13}B_{12} - B21]^T \tag{2.1.14}$$

Once the Rodriguez parameters $\mathbf{p}$ have been found, the attitude quaternion can be calculated using Equation 2.1.15:

$$\overline{\mathbf{q}} = \frac{1}{\sqrt{1 + \mathbf{p}^T\mathbf{p}}}\begin{bmatrix}\mathbf{p}\\ 1\end{bmatrix} \tag{2.1.15}$$

$$\overline{\mathbf{q}} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} + q_4 \tag{2.1.16}$$

2.1.3 SVD Method

This method was proposed by Markley[28]. The method gives a solution for the Wahba's problem using the Singuluar Value Decomposition of a matrix. It proceeds as follows:

- Suppose that it is available a set of n vectors (notated as $\mathbf{b}$) measured in the spacecraft frame.
- It is available their values $\mathbf{r}$ in a inertial reference frame.
- The matrix B is constructed as it appears in Equation 4.2.13:

$$B = \sum_{i=1}^{n} \mathbf{b}_i \mathbf{r}_i^T \tag{2.1.17}$$

- Applying the Singular Value Decomposition to the matrix B we can obtain:

$$B = USV^T \tag{2.1.18}$$

- The current attitude matrix is finally found:

$$A_c = U_+ V_+^T \tag{2.1.19}$$

where:

$$U_+ = U \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & det(U) \end{bmatrix} \tag{2.1.20}$$

$$V_+ = V \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & det(V) \end{bmatrix} \tag{2.1.21}$$

2.2 Star Tracker

A Star Tracker is a device able to compute the attitude of a spacecraft through stellar images. As seen in section 2.1 to achieve this, it is necessary to measure a set of vectors in the spacecraft reference frame. As show in Figure 2.9, every vector can be obtained using the 2D coordinates in the image frame(this is their x, y pixels) and the focal length of the Star Tracker optics. This procedure will be fully explained in chapter 4.

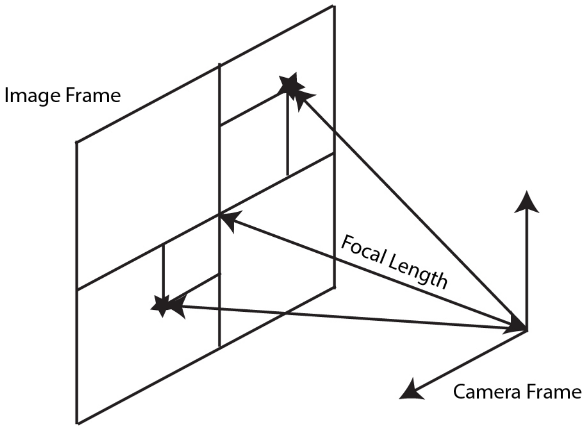


Figure 2.9 – *Obtaining the Measured Vectors*

As it appears in [25], a Star Tracker has the following main elements, which are sketched in Figure 2.10:

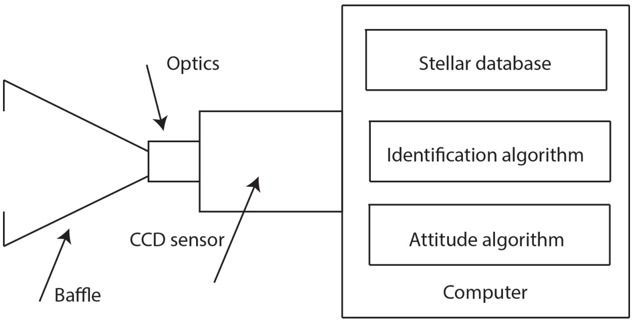


Figure 2.10 – *Star Tracker Sketch*

- **Baffle:** This element is used in order to avoid stray lights.
- **Camera Optics:** Element that contains the lens and determines (in conjunction with the camera sensor) the Field of View (FOV) of the Star tracker.
- **Camera Sensor:** Optoelectronic sensor responsible for the image acquisition.
- **Computer:** Element responsible for the image processing and attitude determination operations. In these tasks the following elements are involved:
  - Stellar Database: Completion of stars and their locations in a certain reference frame.
  - Identification Algorithm: Set of operations that allows the system to identify the stars present in an image.
  - Attitude Determination Algorithm: Set of operations that allows the system to compute the attitude through the identification of the stars.

Nowadays, star trackers (Figure 2.11) are widely used in space missions [3, 53, 44, 49, 42, 41]. The following table 2.2 shows some of its important parameters.

	Update Rate (Hz)	Power Consumption (W)	FOV (degrees)	Accuracy (arcsec)
CT-633 **(Ball Aerospace)**	5	8-9	17.5 x 17.5	35
ST-5000 **(University of Wisconsin-Madison)**	10	12	5x7	0.5
Hydra-M **(SODERN)**	16	7	-	11
HE-5AS **(TERMA)**	4	7	22	1.5
A-STR **(Office Galileo)**	10	8.5W @ 20°C 10.5W @ 50°C	16.4° x 16.4°	100
SHEFEX **(DLR)**	10	-	14.87 × 11.15	360

Table 2.2 – *Commercial Star Trackers Comparative*

Figure 2.11 – *Commercial Star Tracker*
Courtesy of Ball Aerospace & Technology Corp.

2.2.1 Identification Algorithms

The Identification Algorithm is the core of the Star Tracker functionality as it is impossible to compute the attitude before the correct identification of the stars in the image. In this section, three well-known identification algorithms will be described[32].

2.2.1.1 Matching Group Algorithm

This algorithm was first proposed by Bezooijen in [55]. It seeks to find the longest chain of observed stars that matches a group of guide stars. It maximizes the rate of success by exploiting the whole information content in the star pattern.

The process to obtain the star pattern is shown in Figure 2.12.

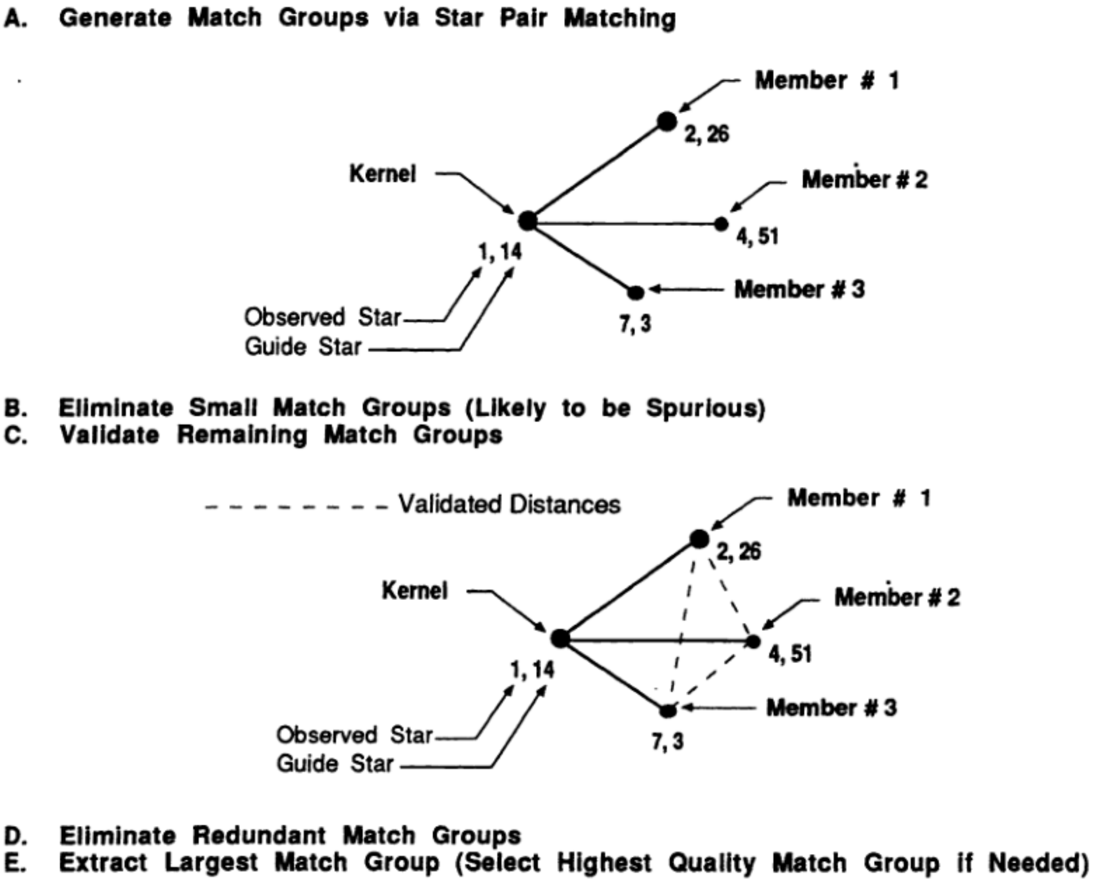

A. Generate Match Groups via Star Pair Matching
Member # 1
2, 26
Kernel
Member # 2
4, 51
1, 14
Member # 3
Observed Star
Guide Star
7, 3
B. Eliminate Small Match Groups (Likely to be Spurious)
C. Validate Remaining Match Groups
Validated Distances
Member # 1
2, 26
Kernel
Member # 2
4, 51
1, 14
Member # 3
Observed Star
Guide Star
7, 3
D. Eliminate Redundant Match Groups
E. Extract Largest Match Group (Select Highest Quality Match Group if Needed)

Figure 2.12 – *Matching Group Algorithm. Figure Extracted from [55]*

1. In step A, we have formed that group because we have found that stars number 2, 4 and 7 are pairs of the *kernel* star.

2. In step B, we discard the small groups that have appeared.

3. In step C, we validate the angular distances of the $non - kernel$ stars.

4. In step D, redundant groups are discarded.

5. In step E, we finally get the largest group of stars.

2.2.1.2 Voting Algorithm

The Voting algorithm is defined in [24] and aims to find the star pattern regarding only in the information of the angular distance between star pairs. The algorithm is very well defined by the authors and its pseudocode for the star identification routine is reproduced below:

Algorithm 1 Voting Method LIS Routine

```
detect $n$ possible stars $S_i$, $1 \leq i \leq n$ in the image
for all possible stars do
    estimate position $P$
    estimate localization uncertainty $e$
end for
for $i = 1$ to $i = n$ do
    for $j = i + 1$ to $j = n - 1$ do
        compute the distance $d_{ij} = |P_i - P_j|$
        if $d_{ij} < D$ then
            compute the distance uncertainty region $R_{ij} = [d_{ij} - e_{ij}, d_{ij} + e_{ij}|$
            locate $k$ entries in $T$ such $T(k).d \in R_{ij}$
            for all entries $T(k)$ do
                append to voting list $V_i$ and $V_j$ of possible stars $S_i$ and $S_j$ respectively
                the two catalogue stars $T(k).ID1$ and $T(k).ID2$
            end for
        end if
    end for
end for
for all possible stars $S_i$ do
    assign $St(i)$ to the catalogue star which got the maximal number of votes
end for
for $i = 1$ to $i = n$ do
    for $j = i + 1$ to $j = n$ do
        if $\left|S_{St(i)} - S_{St(j)}\right| \in R_{ij}$ then
            add vote for the match $\left(S_i, S_{St(i)}\right)$ and for the match $\left(S_j, S_{St(j)}\right)$
        end if
    end for
end for
all possible stars whose votes are clustered together are assumed correct.
estimate the attitude using a least squared quaternion based method.
```

2.2.1.3 The Grid Algorithm

This method was introduced by Padgett and Kreutz-Delgado[33]. Instead of looking for the star pattern checking the angular distances between stars as in the Matching Group

Algorithm or in the Voting Method, the key of this algorithm is building the star pattern for each star reference star included in the Stellar Database.

The pattern associated to each reference star is obtained as follows, Figure 2.13.

1. Select a reference star r for which the pattern is going to be obtained.
2. Reallocate the star r and all the stars that fall in a certain pr radius in a manner that r is in the North Pole (Which is the image center).
3. Define a radius br. Orient a grid of $g \times g$ elements in that manner that the equator of the grid pass through the center of the star r and its closest neighbor. The closest neighbor is considered the star which is never inside the radios br and its distance from it is minimum.
4. At this point we can obtain a g^2 length vector known as pat_i. Each position $i * g + j$ in the vector either has 0 value if there is no star in the $cell(i, j)$ or 1 value if there is a star in $cell(i, j)$.

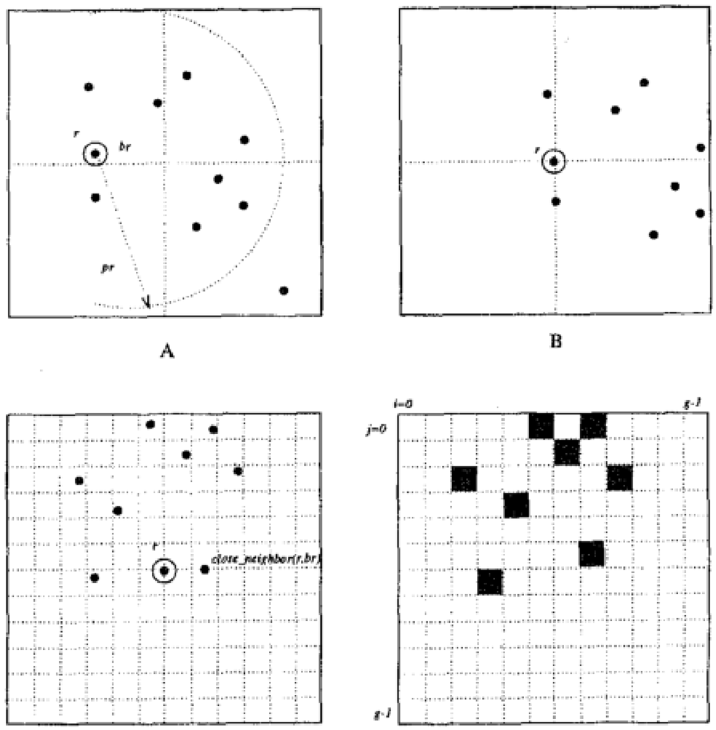


Figure 2.13 – *Pattern Building in Grid Algorithm. Extracted from [33]*

In this algorithm, once an image has been taken, the previous procedure is applied to each possible star. Then, one pattern pat_j is obtained from each star. Given a set of patterns pat_i previously obtained from the reference stars, we will identify the stars in the image as the stars that maximises the following equation Eq 2.2.1:

$$match(pat_j, pat_i) = \sum_{k=0}^{g^2-1} \left(pat_j\left[k\right] \& pat_i\left[k\right]\right) \tag{2.2.1}$$

2.3 Requirements

In this section, we will present the Star Tracker requirements for the GranaSAT experiment. These requirements will be deduced from sections 2.1 and 2.2. We have decided to split the requirements in two separate parts. In section 2.3.1, we will present the hardware requirements for the Star Tracker subsystem while in section 2.3.2 the specific requirements for the software will be presented.

2.3.1 Hardware Requirements

Regarding to the information in 2.2, the following elements are required:

- **Baffle**: the baffle shall be designed not to interfere with the FOV and physically implemented with the Electronics and Computer Technology Department (ECTD) facilities.
- **Optics**: The optics shall ensure an appropriate FOV without perspective distortion.
- **CCD camera**:
- **Computer**: The computer chosen shall fulfill the following specifications.
  - At least one USB 2.0 interface must be available.
  - At least one Ethernet port must be available.
  - It has to support I2C communication.
  - Standard C code support.

As the budget for the GranaSAT experiment was very limited, an additional requirement is added:

- The cost of the Star Tracker components cannot be higher than 1000€. The assignment to each component is broken down below:
  - **Baffle**: 100€
  - **Camera**: 600€
  - **Computer**: 100€
  - **Optics**: 200€

2.3.2 Software Requirements

Again, from Sections 2.1 and 2.2 we can define the following software requirements:

- A **stellar database** has to be built.
- An **identification algorithm** has to be implemented.
- An **attitude computation method** has to be implemented.
- As the Star Tracker is not the only subsystem in the GranaSAT experiment, the use of the computer resources must not be monopolised.

CHAPTER 3

HARDWARE IMPLEMENTATION

In this chapter, the relevant hardware components for the Star Tracker subsystem are presented. The complete experiment hardware description is available in the SED.

We start introducing a brief survey of three possible commercial cameras. In our case, the camera used was not the best option for our purposes; however, the analysis of the possible camera options is done in order to provide useful information about an appropriate camera (sensor) for its use in a future student-designed nano-satellite.

A study of the FOV of our Star Tracker is carried out (section 3.1.5). This parameter would be obtained from the camera sensor and the optics, section 3.1.5.

The chapter continues with the description of the baffle, section 3.2. This element was designed by Teresa L. Aparicio. We would like to thank her for her permission to show her baffle design in this document.

Finally, in section 3.3, we compare three of the most used single-board computers, which were considered mainly because of its low-power consumption, computationally performance and size, which is comparable to a credit card.

In table 3.1; we show a summary of the final costs for each component:

Component	**Price (€)**
Camera (DMK 41BU02)	510
Computer (Raspberry Pi B)	40
Optics (FL-HC1214-2M)	150
Baffle	40
Total	740

Table 3.1 – *Componets Price*

3.1 Camera

A set of USB 2.0 compatible industrial cameras were studied for its use in the experiment. This kind of cameras usually has CCD or CMOS sensors which have been tested before in star trackers [11].Even though the camera finally used was available from the beginning, an analysis of the commercial cameras on the market was done.

3.1.1 DMK 41BU02

The DMK 41BU02 (figure 3.1) is an industrial camera from The Imaging Source company [23]. It uses as a sensor the CCD ICX205AL[46] from Sony. The technical specifications of the camera are shown in table 3.2

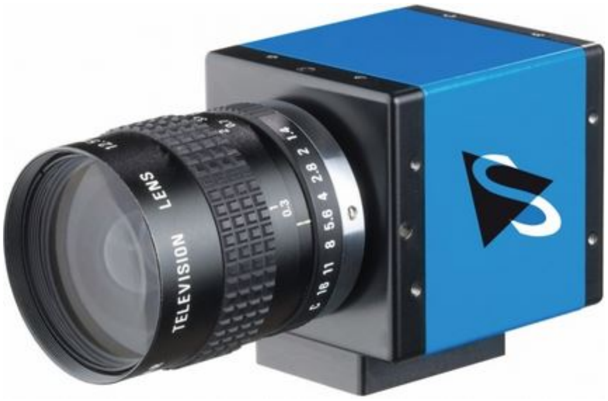


Figure 3.1 – *DMK 41BU02*

Sensor	Sony ICX205AL
Sensor size	1/2”
Resolution (px)	1024 x 960
Pixel size (μm)	4.65
Maximum frame rate (fps)	15
Dimensions (mm)	50.6 x 50.6 x 56
Weight (g)	265

Table 3.2 – *DMK 41BU02 Specifications [23]*

3.1.2 DMK 21BU618

The DMK 21BU618 [22] is an industrial camera from The Imaging Source company which shares the mechanical design with the DMK 41B02; however, it uses as a sensor the CCD ICX618ALA [45] from Sony. The technical specifications of the camera are shown in table 3.3.

Sensor	Sony ICX618ALA
Sensor size	1/4”
Resolution (px)	640 x 480
Pixel size (um)	5.6
Maximum frame rate (fps)	60
Dimensions (mm)	50.6 x 50.6 x 56
Weight (g)	265

Table 3.3 – *DMK 21BU618 Specifications [23]*

3.1.3 Demokit Aptina MT9P031

As it appears in [11] the Aptina MT9P031[2] sensor was used in S3S nanosatellite star tracker. It was developed by a collaboration of Sinclair Interplanetary, UTIAS SFL and Ryerson SAIL.

A demokit of the MT90P031 sensor (figure 3.2) can be found in the market. This sensor can be used for specific applications. The following table 3.4 shows the specifications of the sensor.

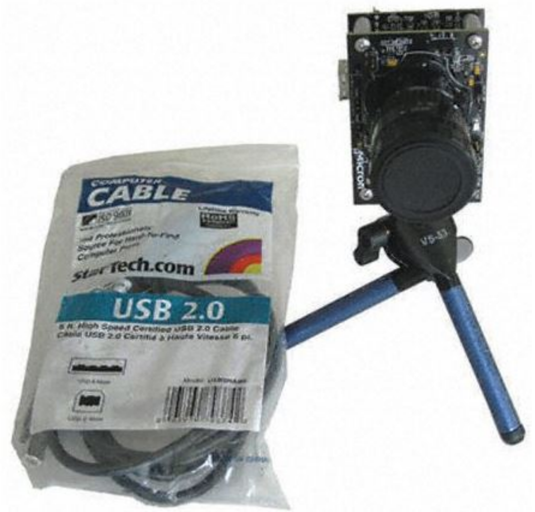


Figure 3.2 – *Demokit for the MT9P031 sensor*

Sensor	Aptina MT90P031
Sensor size	1/2.5”
Resolution (px)	2592 x 1944
Pixel size (um)	2.2
Maximum frame rate (fps)	15
Dimensions (mm)	82.7 mm × 48.3 mm × 35.0 mm

Table 3.4 – *Aptina MT90P031 Specifications [2]*

3.1.4 Camera Selection

The following table 3.5 shows the maximum Quantum Efficiency (QE) at a certain wavelength for the three camera's sensor proposed,[1, 35, 20]. At the same time, in the third column the price of each camera which uses these sensors is shown[27, 50].

Sensor	QE (%)	Price (€)
Aptina	62%@490nm	1000
ICX205AL	42%@490nm	510
ICX618ALA	68%@600nm	410

Table 3.5 – *Sensors Comparative*

Because of its price, which is greater than the budget assignment defined in section 2.3.1, the Aptina demokit could be discarded from the beginning. Regarding to its features, the most convenient camera for our purposes would be the DMK 21BU618. Even though its resolution its lower than the DMK 41BU02, its near infrared peak QE makes this camera appropriate for low-light applications. In spite of this, the DMK 41BU02 was available in the ECTD because it was used in other projects. Due to the endless budget problems which arose during the GranaSAT experiment implementation, we decided not to buy a new camera and carry out the experiment with the resources available at that moment.

3.1.5 Field Of View

The FOV defines the angle of vision of the camera, i.e., the part of the world that will appears in the image (Figure 3.3). It is computed through the effective focal length of the optics f, and the size of the image sensor[30]. If the optics are focused towards the infinity, which is our case, the FOV can be obtained using equations (3.1.1)-(3.1.2):

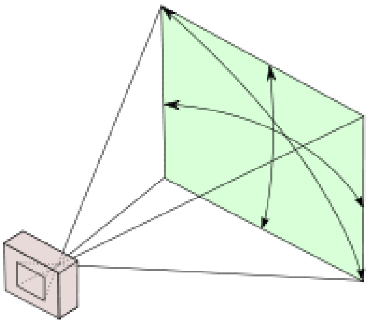
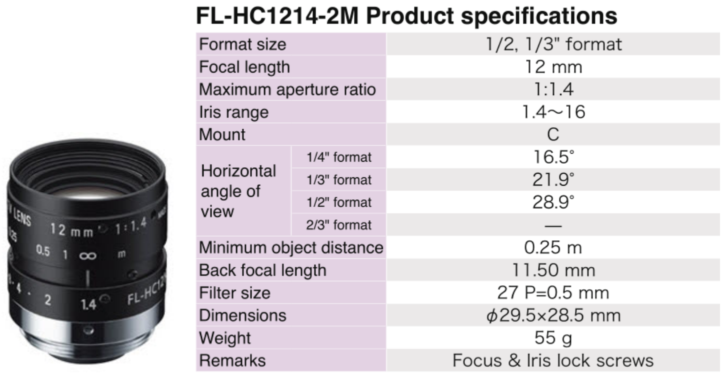


Figure 3.3 – *Angle of View*

$$\alpha_h = 2atan\left(\frac{pixels_{horizontal}pixel_{size}}{2f}\right) \tag{3.1.1}$$

$$\alpha_v = 2atan\left(\frac{pixels_{vertical}pixel_{size}}{2f}\right) \tag{3.1.2}$$

As the camera DMK 41BU02 uses C mount lens, the FL-HC1214-2M from Ricoh [38] was acquired, the following figure 3.4 shows its specifications:



FL-HC1214-2M Product specifications

Format size		1/2, 1/3" format
Focal length		12 mm
Maximum aperture ratio		1:1.4
Iris range		1.4∼16
Mount		C
Horizontal angle of view	1/4" format	16.5°
	1/3" format	21.9°
	1/2" format	28.9°
	2/3" format	—
Minimum object distance		0.25 m
Back focal length		11.50 mm
Filter size		27 P=0.5 mm
Dimensions		ϕ29.5×28.5 mm
Weight		55 g
Remarks		Focus & Iris lock screws

(a) *FL-HC1214-2M* **(b)** *FL-HC1214-2M Specs*

Figure 3.4 – *Ricoh FL-1214-2M Lens*

Having f=12mm (figure 3.4b) and an image resolution of 1280x960 (Table 3.2), the FOV of our Star Tracker can be obtained knowing the pixel size, which is 4.65 μm. Then the FOV is 28.90° × 21.88° that coincides with the value shown in figure 3.4 for 1/2" sensors.

3.2 Baffle

The baffle used in the GranaSAT experimet was designed by Teresa L. Aparicio who was in charge of the mechanical design of the experiment. It was manufactured on ABS plastic with a 3D printer from Imprusa[16]. To determine its length, the following formula 3.2.1 were used, f' is the focal length of the optics, as it appears in figure 3.4, $f' = 12 \pm 5\%$. A 3D model of the baffle is shown in figure 3.5.

$$\frac{\theta_{PE}}{f'} = \frac{1}{1.4} \tag{3.2.1}$$

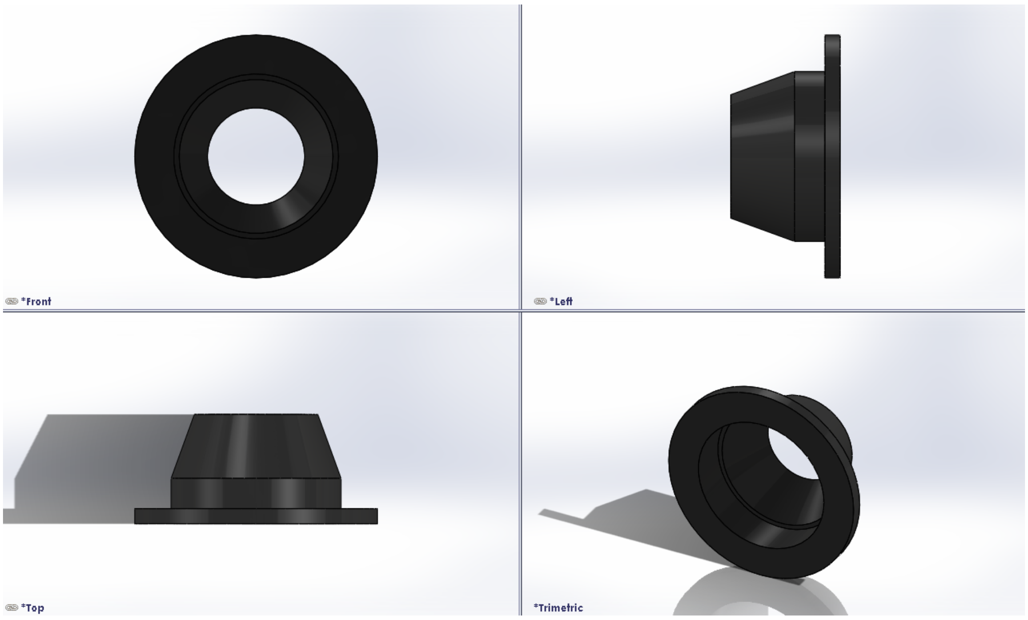


Figure 3.5 – *Baffle designed by Teresa L. Aparicio*

3.3 Computer

The experiment's computer was the most important component for the GranaSAT experiment. As it is shown in figure 3.6 it had to handle:

- The communication with the camera (USB 2.0 port required).
- The communication with the I2C sensors (I2C bus capability).
- Status LEDs (GPIO Required)
- The star tracker and horizon sensor routines (CPU + RAM).
- The communication with the experiment ground station (Ethernet port required).

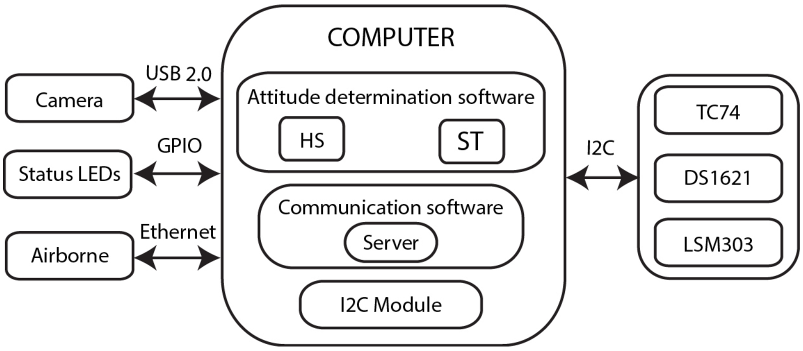


Figure 3.6 – *Computer Requirements*

In sections 3.3.1, 3.3.2 and 3.3.3 we present the different options considered. All of them are low-cost credit card sized Linux compatible computers.

3.3.1 BeagleBoard Black

The BeagleBone Black (figure 3.7) is a single board computer developed by the BeagleBoard.org foundation. This foundation is a US-based501(c) non-profif corporation aiming to provide education in and promotion of the design and use of open-source software and hardware in embedded computing. BeagleBoard.org provides a forum for the owners and developers of open-source software and hardware to exchange ideas, knowledge and experience [5].

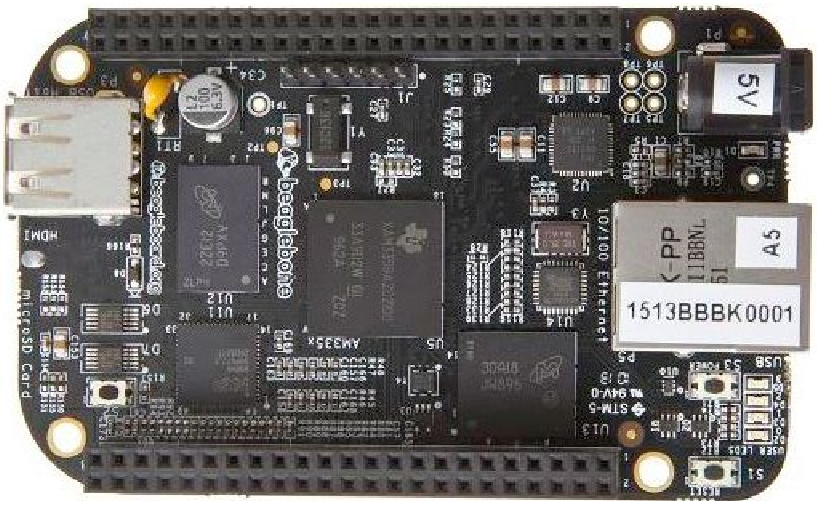


Figure 3.7 – *BeagleBoard Black*

It is compatible with Ubuntu and Android SOs. Its hardware specifications for the C revision are shown in table 3.6[7].

Processor	Sitara AM3358BZCZ100 1GHz, 2000 MIPS
Graphics Engine	SGX530 3D, 20M Polygons/S
SDRAM Memory	512MB DDR3L 800 MHz
Onboard Flash	4GB, 8bit Embedded MMC
PMIC	TPS65217C PMIC regulator. One additional LDO
Debug Support	Optional Onboard 20-pin CTI JTAG, Serial Header
Power Source	miniUSB/USB/DC Jack/5VDC via external header
PCB	3.4” x 2.1” 6 layers
Indicators	1-Power,2-Ethernet, 4-User Controllable LEDs
USB 2.0	1 port Type A socked 500 mA
Serial Port	UART0 access via 6 pin 3.3V TTL.
Ethernet	10/100, RJ45
SD/MMC Connector	microSD, 3.3 V
Video Out	16b HDMI 1280x1024 (MAX) @24Hz
Audio	Via HDMI Interface, Stereo
Expansion Connectors	Power 5V,3.3V,I2C,GPIO (69max)
Weight	1.4 oz (39.69 grams)

Table 3.6 – *BlackBone Black Specifications*

3.3.2 Raspberry Pi Model B

The Raspberry Pi is developed by the Raspberry Pi foundation, an education charity organization based in the UK. Its objectives are the promotion computer science subjects [36].

Raspberry Pi computers are available in different configurations depends on its performance. All of them works with linux distributions. For our experiment we studied the use of the Raspberry Pi model B, figure 3.8.

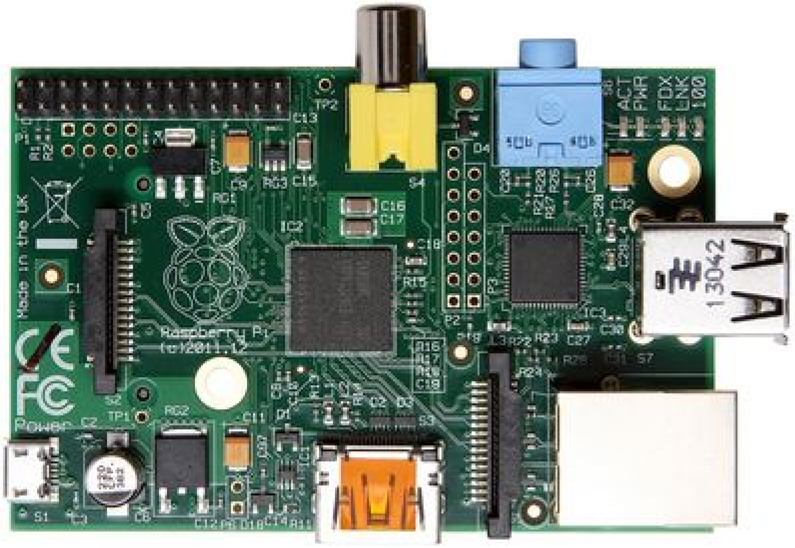


Figure 3.8 – *Raspberry Pi Model B*

The following table 3.7 shows the technical specifications for the RPI model B[10].

Chip	Broadcom BCM SoC full HD multimedia application processor
CPU	700 MHz low power ARM1176JZ-F Applications Processor
GPU	Dual Core VideoCore IC Multimedia Co-Processor
Memory	512 MB SDRAM
Ethernet	onboard 10/100 Ethernet RJ45 jack
USB 2.0	Dual USB connector
Video Output	HDMI (rev 1.3 & 1.4) Composite RCA (PAL and NTSC)
Audio Output	3.5 mm jack, HDMI
Onboard Storage	SD, MMC, SDIO card slot
Operating System	Linux
Dimensions	85.0 mm x 56.0 mm x 17 mm

Table 3.7 – *RPI model B Specifications*

3.3.3 OMAP4430 PandaBoard

PandaBoard (figure reffig:pandaboard) is an open development platform based on the Texas Instrument OMAPXXXX system of a chip (SoC)[34]. Its technical specifications are shown in table 3.8.

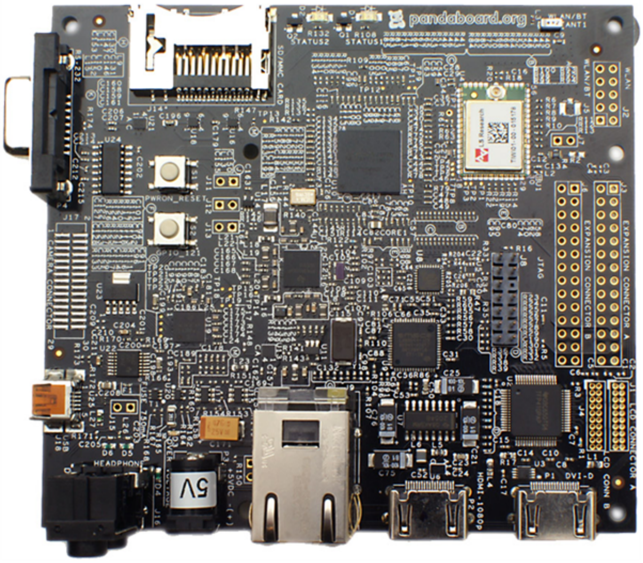


Figure 3.9 – *OMAP4430 PandaBoard*

Processor	OMAP4430
POP Memory	Micron 8Gb LPDDR2 (EDB8064B1PD-8D-F)
PMIC	TI (TWL6030)
Debug Support	14-pin JTAG ,GPIO, LEDS UART via DB-9 connector
PCB	4.5”x4.0” (114.3 x 101 mm) 8 layers
Indicators	3 LEDs
USB	4 USB ports up to 500mA on ecah port
Audio Connectors	3.5 mm, L+R out
SD/MMC Connector	6 in 1 SD/MMC/SDIO
Video	DVI- D or HDMI
Power Conector	micro USB

Table 3.8 – *OMAP4430 PandaBoard Specifications*

3.3.4 Computer Selection

The decision to choose the computer for the experiment was mainly made due to the budget constraint and the immediate availability of it. The following table 3.9 shows the price on the market from three distributors [27, 9, 52] for the computers analyzed in sections 3.3.1, 3.3.2 and 3.3.3. All prices shown do not include shipments costs.

As we can extract from table 3.9 and the technical specifications for each computer, the most powerful one (PandaBoard) is at the same time the most expensive. Therefore, it is unaffordable due to the budget assigned to the computer, which was 100€.

	RS Components	**DigiKey**	**Tigal**
BeagleBoard Black	44.89 €	43.98 €	49 €
Raspberry Pi model B	31.14 €	-	-
PandaBoard OMAP4430	-	177.56 €	177,00 €

Table 3.9 – *Computer Price Comparative*

The two available options were the BeableBoard or the Raspberri Pi; their performance is similar and they both fit the requirements for the computer defined in section 2.3.1. Finally, we decided to use for our experiment the Raspberry Pi because it existed stock of some of this boards in the ECTD.

CHAPTER

4

SOFTWARE IMPLEMENTATION

The following chapter describes the software developed concerning only the Star Tracker subsystem. It outside of the scope of this chapter to show the code written, however, it will provide a familiarized reader with all the necessary steps to reproduce the program in any programming language.

During the development of the GranSAT Star Tracker, it was necessary to build a Generated Catalog (GC) and a k-vector [31] (k). This data is obtained only once before the flight, it was called *pre-launch data* and it will be detailed in 4.1. The GC is the stellar database that appears in figure 2.10. It contains relevant information of the selected *guide stars* from the Hipparcos Catalog [12] (HP). The process to obtain these stars will be explained in 4.1.1. Once the GC has been obtained, the vector k is generated through the angular distance d information of the stars in the GC.

Both The GC and k are generated and stored in text files using MATALB[51] scripts as shown in figure 4.1.

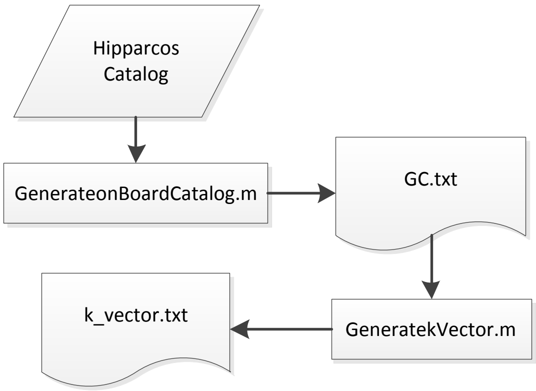


Figure 4.1 – *Pre-Launch Data Generation*

During the flight, the Star Tracker software runs always a set of C routines in the same order, these are called the Lost In Space (LIS) routine, because it seeks to find the stars appearing in the image and the attitude of the spacecraft without any previous information. Figure 4.2 shows the flow diagram of the LIS routine which will be detailed in 4.2. As introduction, they are outlined below:

- Image Acquisition: the camera captures an image for its processing.
- Centroiding: this function gets the (x,y) coordinates in the image frame of each possible star.
- Getting the unitary vectors: this function transform the 2D coordinates from the previous step into a 3D coordinates in the camera body frame.
- Finding the star pattern: this routine identifies the stars appearing in the image.
- Computing the attitude: this routine is responsible for the attitude computation.

4.1 Pre-Launch Data

4.1.1 Building the Generated Catalog

The Hipparcos and Tycho catalogs were the principal outcome of the ESA mission Hipparcos. From November 1989 to March 1993 the Hipparcos satellite collected high quality of astrometric and photometric data[12]. In this thesis, we use the HP information to build the GC as well as to obtain the coordinates of the *guide stars* in the ECI reference frame.

We can use the VizieR web application[8] to download the whole HP. Every star in the HP has 67 fields, but only the following are useful for our purposes:

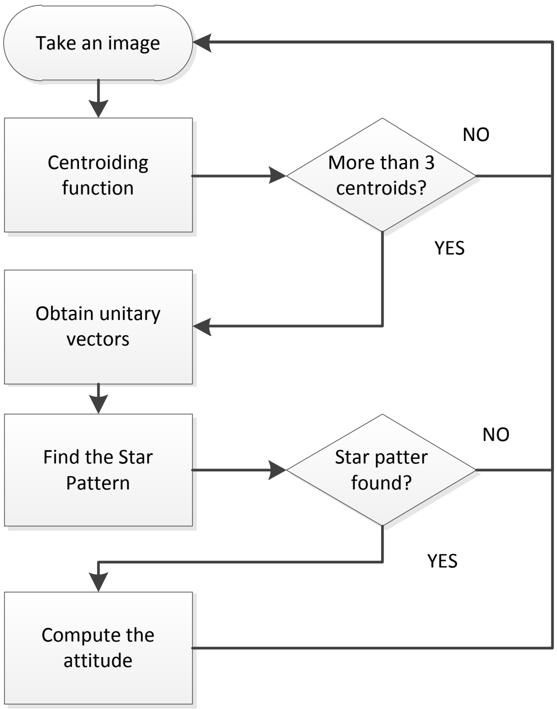


Figure 4.2 – *Star Tracker Flow*

- **Hipparcos Catalogue Identifier** (HIP,Field H1) : number between 1 and 118218 that uniquely identifies a star in the Hipparcos Catalog.

- **V magnitude** (Field H5) : the V magnitude of the star.

- **Equatorial Coordinates** (Fields H8,H9): the Right Ascension (α) and Declination (δ) for the star in the ICRS for the J1991.25 epoch.

- **Trigonometric Parallax,** π (Field H11) : the expected trigonometric parallax π, expressed in milliarcsec.

- **Proper Motions** (Fields H12,H13) : the proper motion components for the equatorial coordinates in the ICRS J1991.25 expressed in milliarces per Julian year (mas/yr)

As the data in the HP is given for the J1991.25 epoch, we have to convert this data to the current epoch, which was the predicted date for the flight (October 8^{th}, 2014). This operation can be done using NOVASv3.1[4] software, which is available in [54].

The procedure to obtain the GC (Algorithm 2) is the same as appears in [24] and its pseudo code is shown below:

Algorithm 2 GC procedure

```
for $i = 1 : guideStars.Number - 1$ do
    $[x_i, y_i, z_i] = convertRADEC(guideStars(i).RA, guideStars(i).DEC)$
    for $j = i + 1 : guideStars.Number$ do
        $[x_j, y_j, z_j] = convertRADEC(guideStars(j).RA, guideStars(j).DEC)$
        $d_{ij} = acos(x_i x_j + y_i y_j + z_i z_j)$
        if $d_{ij} <$ FOV then
            $add2GC(guideStar(i).HIP, guideStar(i).HIP, d_{ij})$
        end if
    end for
end for
$Short(GC)$                    ▷ Short in ascending order according with $d_{ij}$
```

After this process, the GC can be seen as a shorted database of pairs of stars whose angular distance is smaller than a certain value, FOV. As shown in table 4.1, the first two columns are the HIP for each star of the doublet. The third column is the angular distance in degrees between these two stars.

HIP	HIP	d (º)
...	...	...
82363	83153	6.0364
57399	58001	6.0414
20885	21402	6.0518
...	...	...
30122	31685	13.5857
85696	90422	13.5858
85693	86032	13.5859
...	...	

Table 4.1 – *Example of GC Structure*

The size of the GC depends on the number of *guide stars* chosen. Therefore, it is impossible to include all of the 118218 stars characterized in the HP as *guide stars*. Usually, apart from discarding binary stars, the decision to include or not a star as a *guide star* is made considering its V magnitude (i.e the apparent magnitude of a star, $Vmag$)[57]. The first column of the table 4.2 shows the cutoff $Vmag$ selected while in the second column appears the number of GC entries is. The FOV was computed in section 3.1.5. As we can see, the size in number of entries in the GC grows nearly exponentially.

The $Vmag$ selected depends mainly on the camera sensor. The greater the sensibility of the sensor is, the less bright the stars will appear in the image with less exposure time.

In order to find an appropriate $Vmag$ for our GC, it is useful to see what happens during a real night-sky observation. Figure 4.3 shows an image taken with the DMK 41BU02 with the parameters shown in Table 4.3. For convenience, the image appears in negative color.

$Vmag$	GC Nº of entries
2	169
3	1763
4	14422
5	137328

Table 4.2 – *GC Nº of Entries*

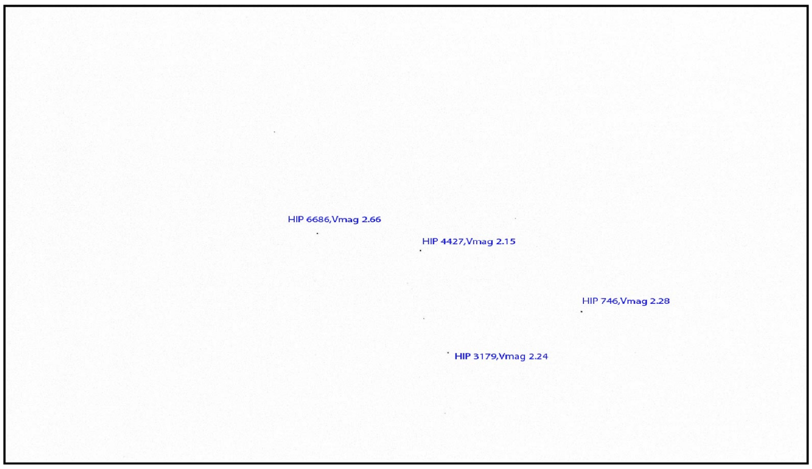


Figure 4.3 – *Experimental Image 1*

Parameter	Value
Exposure time	0.2 s
Gain	Maximum (1023)
Gamma	100
Brightness	50

Table 4.3 – *Parameters of the DMK41BU02 for Image 4.3*

The previous image (Figure 4.3) was taken in the countryside with no moon in the sky. As it can be observed only stars with Visual Magnitude lesser than 3 could be observed. Changing the exposure time from 0.2 s to 1 s (Figure 4.4), we can observe that stars with Visual Magnitude around 5 appear in the image.

After an analysis of sky covering (this is, $Vmag$ vs FOV) which will be detailed in chapter 5 the cutoff $Vmag$ =4.5 was selected. The number of *guide stars* with $Vmag < 4.5$ in the HP is 909.

Once we have obtained an appropriate number of *guide stars*, we can build the GC following the steps described at the beginning of this section (Algorithm 2). We obtained a GC with 47477 entries. Finally, the following Figure 4.5 shows a representation of the *guide stars* in the Celestial Sphere for their coordinates in the ICRS at J1991.25.

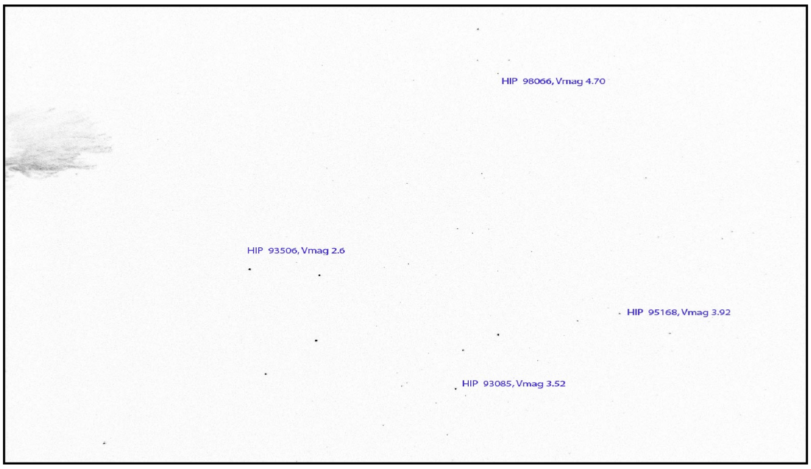


Figure 4.4 – *Experimental Image 2*

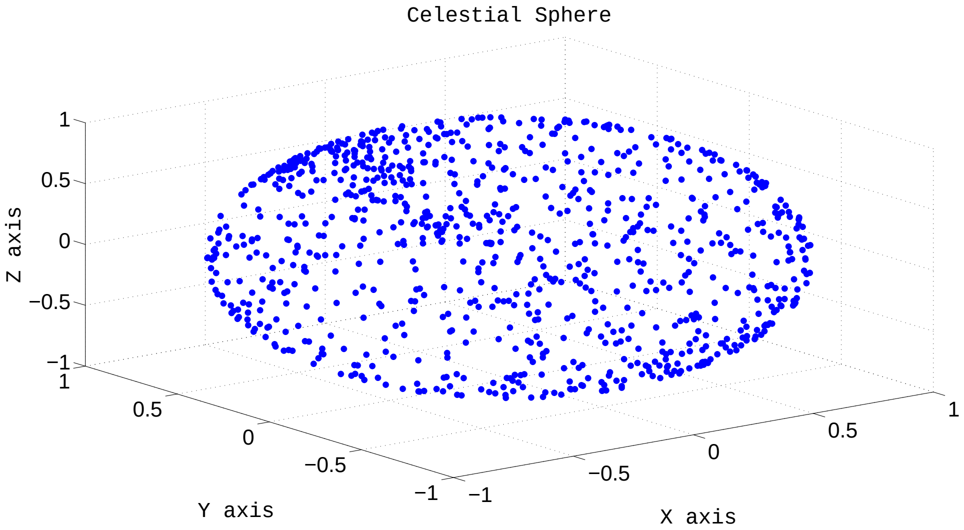


Figure 4.5 – *Celestial Sphere For the Guide Stars*

4.1.2 Generating the k-vector, k

The k-vector searching technique[31] is used to increase the efficiency in speed terms when searching information in the GC. The author of this technique claims that it is from 10 to 50 times more efficient than the well-known binary searching technique.

This technique requires the generation of a n-long vector (k) where n is as equal as entries in the previous obtained GC. In our case, we will obtain a n=43477 k-vector. The following Figure 4.6 shows the angular distance in degrees versus the pair index in our GC. As we can see, the greater the index is, the greater the angular distance between any two *guide stars* will be too.

The vector k will be created from the shorted GC as shown in algorithm 3:

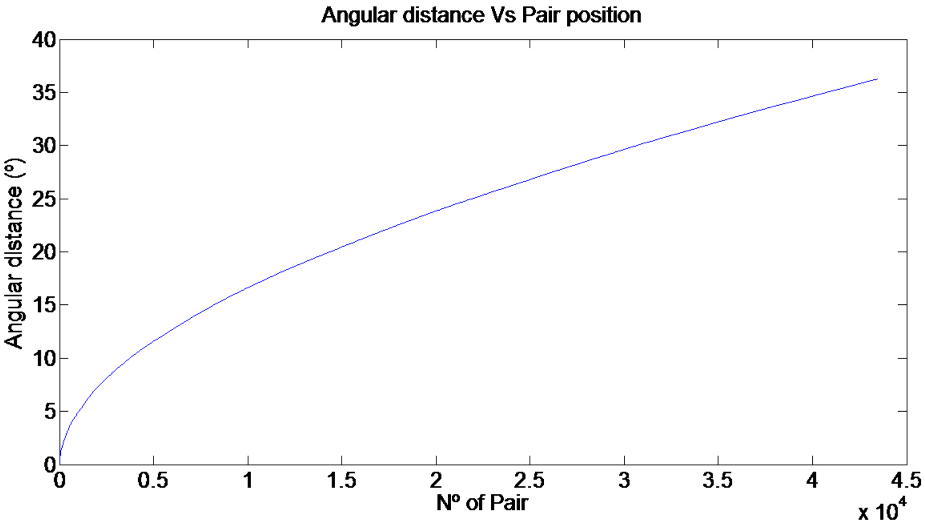


Figure 4.6 – *Anguar Distance vs Index in GC*

Algorithm 3 k Vector Generation

```
m = (y_max − y_min)/(n−1)        ▷ y_max, y_min are the maximun and d_ij in the GC
q = y_min − m
for i ≤ n do
    z(i) = im + q
end for
k(1) = 0
k(n) = n
for i = 2 : i ≤ n − 1 do
    number = find(GC.d_ij < z(i))
    k(i) = number
end for
```

As it appears in algorithm 3, every position $k(i)$ of the k vector represents the number of GC entries below the $z(i)$ value. Once k has been generated, the following expressions 4.1.1-4.1.2 are used for finding in the GC all entries whose angular distance is inside the range $[a, b]$

$$j_a = \left\lfloor \frac{a-q}{m} \right\rfloor \qquad \text{Where } \lfloor\rfloor \text{ symbolizes the } \textit{floor} \text{ operation} \tag{4.1.1}$$

$$j_b = \left\lceil \frac{b-q}{m} \right\rceil \qquad \text{Where } \lceil\rceil \text{ symbolizes the } \textit{ceil} \text{ operation} \tag{4.1.2}$$

Then, the values $k_{start} = k(j_a) + 1$ and $k_{end} = k(j_b)$ defines the index of the range $[a, b]$ in the GC. This is graphically explained in Figure 4.7.

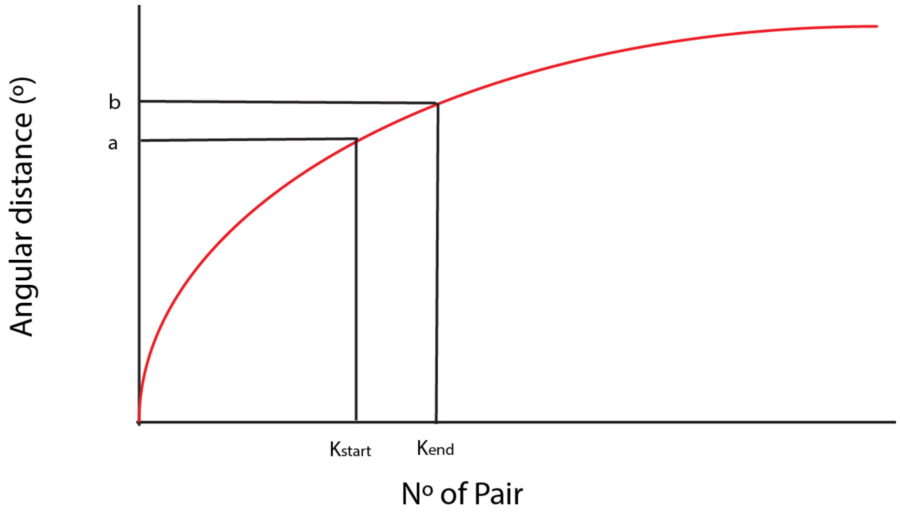


Figure 4.7 – *k-vector Range Searching Technique*

The k-vector range searching technique requires a relatively linear segments to work. Some authors often apply the k-vector technique dividing the GC in small parts[58]. This is done to avoid losing entries which should be include into a certain search due to the non linearity of the GC . This means generating a different vector k for each part. Figure 4.8 is an example of this.

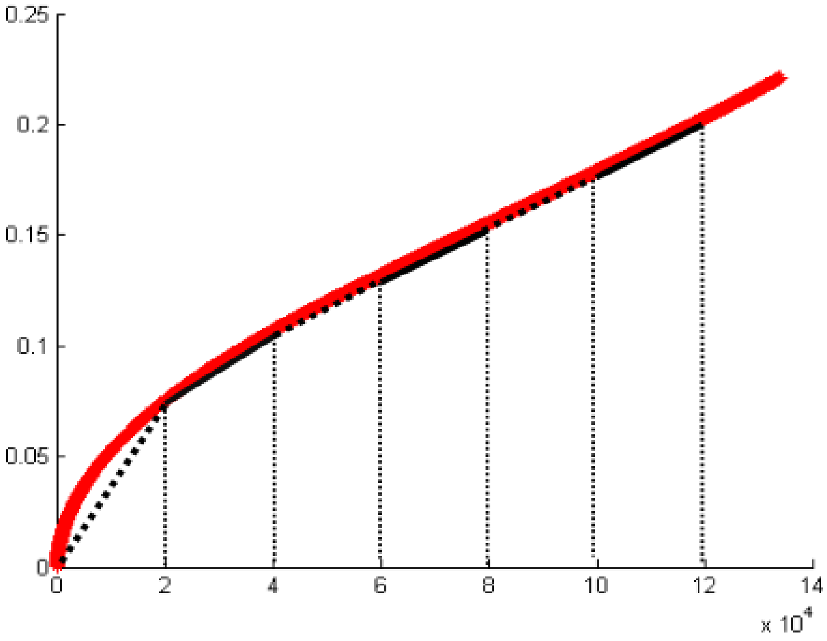


Figure 4.8 – *GC Divided in Parts. Image extracted from [58]*

Instead of dividing the GC into different parts, we propose the following empiric method 4 that has demonstrated to work properly:

Algorithm 4 k Vector Search Routine

```
$j_a = \left\lfloor \frac{a-q}{m} \right\rfloor$
$j_b = \left\lceil \frac{b-q}{m} \right\rceil$
$k_{start} = k(j_a) + 1$
$k_{end} = k(j_b)$
while $GC(k_{start}).d_{ij} > a$ && $GC(k_{start} - 1).d_{ij} > a$ && $k_{start} - 1 > 1$ do
    $k_{start} - -$
end while
while $GC(k_{end}).d_{ij} < b$ && $GC(k_{end} + 1).d_{ij} < b$ && $k_{end} + 1 < n$ do
    $k_{end} + +$
end while
```

4.2 Lost In Space routine

4.2.1 Centroiding

Centroiding is the first sub-algorithm applied to an image after receiving it. The objective is to find the coordinates (x_{cm}, y_{cm}) in the image frame of each possible star. The algorithm used has been adapted from the one appearing in [26].

Considering the image as a matrix with n rows and m columns, the algorithm 5 seeks to find each star's coordinates (x_{cm}, y_{cm}).

Algorithm 5 Centroiding Routine

```
for $i = 1 : n$ do
    for $j = 1 : m$ do
        if $image(i, j) \geq Threshold$ then
            $iBorder = computeIborder(ROI, n, m)$
            $DN = computeBrightness(ROI, n, m)$
            $x_{cm} = computeXcm(ROI, n, m, DN)$
            $y_{cm} = computeYcm(ROI, n, m, DN)$
            $centroid = [x_{cm}, y_{cm}, DN]$
            $append(listOfCentroids, centroid)$
        end if
    end for
end for
$centroids = clusterCentroids(listCentroids)$
$centroids = shortCentroids(centroids)$
```

The parameters used are:

- **Threshold**: as we are using grey color images, it is an integer between 0 and 255. It is used to discriminate between a noise pixels and star pixels.

- **Region Of Interest (ROI)**: odd integer value that defines a square region.

Auxiliary functions and their functionality is explained below:

1. $computeIBorder()$

   After detecting a pixel p which value is greater than the $threshold$, this function computes the average intensity of the pixels in the edge of the square region defined by ROI and centered in p. In Figure 4.9 we have dashed in red the edge pixel of a ROI=5. The reader must note that as we are using grey scale images, the closer the pixel is to pure white, the greater its value is.

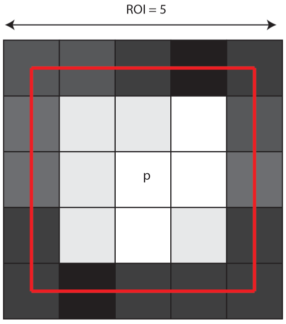
   

   Figure 4.9 – *Region of Interest*

   Finally, the average intensity $iBorder$ is computed using Equation 4.2.1:

   $$iBorder = \frac{\sum Pixels \in ROI_{edge}}{4(ROI-1)} \tag{4.2.1}$$

2. $computeBrightness()$

   This function computes the brightness DN using equation 4.2.2:

   $$DN = \sum_{x=(n-ROI/2)+1}^{(n+ROI/2)-1} \sum_{y=(m-ROI/2)+1}^{(m+ROI/2)-1} (image(x.y) - iBorder) \tag{4.2.2}$$

3. Computing x_{cm} and y_{cm}

   Both coordinates are calculated through the $computeXcm()$ and $computeYcm()$ functions which implement Equations 4.2.3 and 4.2.4 respectively:

   $$x_{cm} = \sum_{x=(n-ROI/2)+1}^{(n+ROI/2)-1} \sum_{y=(m-ROI/2)+1}^{(m+ROI/2)-1} (x\frac{image(x.y)}{DN}) \tag{4.2.3}$$

   $$y_{cm} = \sum_{x=(n-ROI/2)+1}^{(n+ROI/2)-1} \sum_{y=(m-ROI/2)+1}^{(m+ROI/2)-1} (y\frac{image(x.y)}{DN}) \tag{4.2.4}$$

4. clusterCentroids()

   Quite often, a star will have more than one pixel with a greater value than the defined threshold. In this case, we need to cluster these centroids to be able to correspond each star with a unique centroid. In our case, we average the position of each centroid that belongs to the same star, eq 4.2.5-4.2.6:

$$x_{cm} = \frac{\sum_{x_i \in Star} x_i DN_i}{\sum_{x_i \in Star} DN_i} \tag{4.2.5}$$

$$y_{cm} = \frac{\sum_{y_i \in Star} y_i DN_i}{\sum_{y_i \in Star} DN_i} \tag{4.2.6}$$

5. shortCentroids(): This function shorts the clustered centroids in descending order according with is brightness DN.

4.2.2 Getting the Unitary Vectors

Once the centroids have been obtained, we can derive from them the unitary vector ($\vec{u}$) associated with each one. This will be done assuming firstly that the camera optics can be simplified as a *pinhole system*. As it is shown in figure 4.10, in a *pinhole system* every ray that comes from the infinity pass through a the *pinhole*. The distance between this point and the image plane is F (optics focal length), x_0, y_0 are the image principal points, this is, the image center[26]. In our particular case, F, x_0 and y_0 will be obtained in chapter 5.

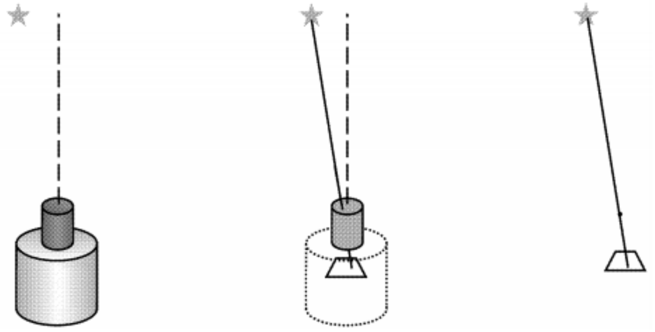


Figure 4.10 – *Pinhole System.*
Image extracted from [26]

Each centroid with coordinates x, y has an associated unitary vector $\vec{u} = \vec{i} + \vec{j} + \vec{k}$ that is computed using Equations 4.2.7, 4.2.8 and 4.2.9:

$$\mathbf{i} = cos(atan2(x - x_0, y - y_0))cos\left(\frac{\pi}{2} - atan\left(\sqrt{\left(\frac{x - x_0}{F}\right)^2 + \left(\frac{y - y_0}{F}\right)^2}\right)\right) \tag{4.2.7}$$

$$\mathbf{j} = sin(atan2(x - x_0, y - y_0))cos\left(\frac{\pi}{2} - atan\left(\sqrt{\left(\frac{x - x_0}{F}\right)^2 + \left(\frac{y - y_0}{F}\right)^2}\right)\right) \tag{4.2.8}$$

$$\mathbf{k} = sin\left(\frac{\pi}{2} - atan\left(\sqrt{\left(\frac{x - x_0}{F}\right)^2 + \left(\frac{y - y_0}{F}\right)^2}\right)\right) \tag{4.2.9}$$

4.2.3 Finding The Star Pattern

The algorithm developed for identifying the stars in the image is a variation of the Matching Group algorithm described in Section 2.2.1.1. After the previous steps 4.2.1 and 4.2.2, a list of unitary vectors, each one corresponding with a centroid, is available. With this information, we will find the HIP number of each star in the image. In order to make this section more understandable, capital letters represent an unitary vector associated with a certain star.

The algorithm will be explained step by step:

- **Step 1**. Assume that we have a list of $\mathbf{N} \geq 3$ unitary vectors. Get the first three ones $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$.

- **Step 2**. Select the vector $\mathbf{A}$ as $Kernel$ and compute its angular distance with the vectors $\mathbf{B}$ and $\mathbf{C}$.

$$d_{AB} = acos\left(\mathbf{A} \cdot \mathbf{B}\right) \tag{4.2.10}$$

$$d_{AC} = acos\left(\mathbf{A} \cdot \mathbf{C}\right) \tag{4.2.11}$$

- **Step 3**. Use the k vector range searching technique to find in the GC the pairs of stars whose angular distance d_{XY} is inside the range $[d_{XY} - d_{XY}\epsilon, d_{XY} + d_{XY}\epsilon]$. This is done for both d_{AB} and d_{AC}. The ϵ value represents the relative error obtained when measuring angles, it depends on the camera calibration and its value will be obtained in Chapter 5.

  The GC entries inside the range $[d_{AB} - d_{AB}\epsilon, d_{AB} + d_{AB}\epsilon]$ and $[d_{AC} - d_{AC}\epsilon, d_{AC} + d_{AC}\epsilon]$ are stored in separates structures S_1, S_2.

- **Step 4**. For each star present in both S_1 and S_2 check if the other two star have an angular distance inside the range $[d_{BC} - d_{BC}\epsilon, d_{BC} + d_{BC}\epsilon]$, Figure 4.11

$$d_{BC} = acos\left(\mathbf{B} \cdot \mathbf{C}\right) \tag{4.2.12}$$

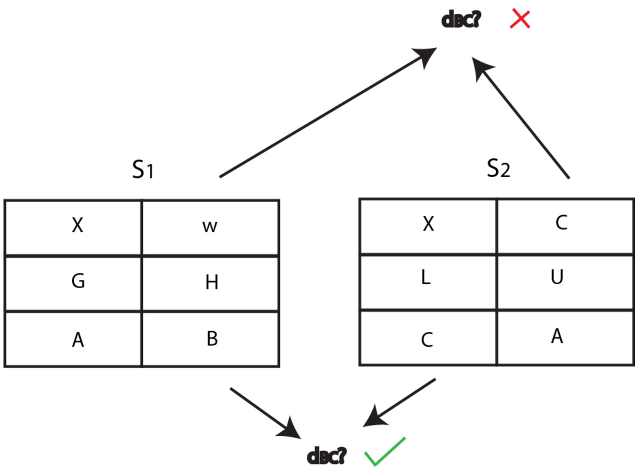


Figure 4.11 – *Checking the B-C Stars*

If at least one triplet of stars $\mathbf{A}, \mathbf{B}, \mathbf{C}$ is found, store the triplet and proceed with step **5**, otherwise, discard the image and process a new one.

- **Step 5**. Repeat steps from steps **2** to **4**, but selecting **B** as the *kernel*. If at least one triplet $\mathbf{B}, \mathbf{A}, \mathbf{C}$ is found, store the triplet and proceed with step **6**, otherwise, discard the image and process a new one.

- **Step 6**. Repeat steps from steps **2** to **4**, but selecting **C** as the *kernel*. If at least one triplet $\mathbf{C}, \mathbf{A}, \mathbf{B}$ is found, store the triplet and proceed with step **7**, otherwise, discard the image and process a new one.

- **Step 7**. At this point, it is possible to have the same three stars repeated three times with the only difference in its order, if that happens, chose these stars as the possible first three stars. Note that more than one possible triplet can be found due to the ϵ value.

- **Step 8**. If $\mathbf{N} > 3$ proceed with step **9**, otherwise, if only one triplet is found, return this triplet as the stars appearing in the image. If more than one triplet is found and $\mathbf{N} = 3$, the star field was not uniquely identified, in this case, discard the image and process a new one.

- **Step 9**. For each one of the remaining unitary vectors: **if** $\exists \mathbf{D} \in ([d_{AD} - d_{AD}\epsilon, d_{AD} + d_{AD}\epsilon] \bigcap [d_{BD} - d_{BD}\epsilon, d_{BD} + d_{BD}\epsilon] \bigcap [d_{CD} - d_{CD}\epsilon, d_{CD} + d_{CD}\epsilon])$ add **D** to the list of identified stars. Otherwise, discard **D** and proceed to **E**.

  Note that if the list of identified stars grows, the number of ranges to check grows as well. Suppose that **D** has been found in the three ranges expressed above, now the star with unitary vector **E** has to be present in all ranges: $[d_{AE} - d_{AE}\epsilon, d_{AE} + d_{AE}\epsilon]$, $[d_{BE} - d_{BE}\epsilon, d_{BE} + d_{BE}\epsilon]$,$[d_{CE} - d_{CE}\epsilon, d_{CE} + d_{CE}\epsilon]$,$[d_{DE} - d_{DE}\epsilon, d_{DE} + d_{DE}\epsilon]$. This makes the algorithm more robust against false stars with the more stars are added.

It is in this step when the possible multi-triplets found in step **7** are discarded since no new stars are added or its number is lower than the number of identified stars for the true match.

4.2.4 Getting the attitude

In section 4.2.3, we have identified each vector with a star. The direction cosine matrix (attitude) will be obtained using the SVD method described in section 2.1.3. The following example describes this process:

Assume that we have identified four stars **A**, **B**, **C** and their unitary vectors in the camera frame are the following:

$$\vec{a_b} = \begin{bmatrix} -0.0795 \\ 0.1993 \\ 0.9767 \end{bmatrix} \quad \vec{b_b} = \begin{bmatrix} -0.0606 \\ 0.1857 \\ 0.9807 \end{bmatrix} \quad \vec{c_b} = \begin{bmatrix} -0.1006 \\ 0.2105 \\ 0.9724 \end{bmatrix}$$

Their values in the ECI reference frame are:

$$\vec{a_r} = \begin{bmatrix} 0.1036 \\ 0.9944 \\ 0.0210 \end{bmatrix} \quad \vec{b_r} = \begin{bmatrix} 0.0838 \\ 0.9959 \\ 0.0339 \end{bmatrix} \quad \vec{c_r} = \begin{bmatrix} 0.1218 \\ 0.9925 \\ 0.0052 \end{bmatrix}$$

The matrix **B** is computed using equation 4.2.13

$$B = \sum_{i=1}^{n} \mathbf{b}_i \mathbf{r}_i^T = \begin{bmatrix} -0.0256 & 0.0619 & 0.3018 \\ -0.2392 & 0.5921 & 2.9131 \\ -0.0042 & 0.0116 & 0.0588 \end{bmatrix} \tag{4.2.13}$$

The matrix **B** is decomposed in the matrix **U**, **S**, **V** using the Single Value Decomposition:

$$USV^T = B$$

$$U = \begin{bmatrix} -0.1031 & 0.7963 & 0.5961 \\ -0.9945 & -0.0704 & -0.0779 \\ -0.0200 & -0.6008 & 0.7991 \end{bmatrix}$$

$$S = \begin{bmatrix} 2.9989 & 0 & 0 \\ 0 & 0.0011 & 0 \\ 0 & 0 & 0.0000 \end{bmatrix}$$

$$V = \begin{bmatrix} 0.0802 & -0.8364 & 0.5421 \\ -0.1985 & 0.5196 & 0.8310 \\ -0.9768 & -0.1743 & -0.1244 \end{bmatrix}$$

The direction cosine matrix A is finally obtained:

$$A = U_+ V_+^T = \begin{bmatrix} -0.3512 & 0.9296 & -0.1122 \\ -0.0631 & 0.0961 & 0.9934 \\ 0.9342 & 0.3559 & 0.0249 \end{bmatrix}$$

Now we can obtain the coordinates in the ECI reference system of the vectors observed in the camera body frame through the matrix A:

$$\vec{a_r'} = \begin{bmatrix} 0.1036 \\ 0.9944 \\ 0.0210 \end{bmatrix} = \begin{bmatrix} -0.3512 & 0.9296 & -0.1122 \\ -0.0631 & 0.0961 & 0.9934 \\ 0.9342 & 0.3559 & 0.0249 \end{bmatrix} \begin{bmatrix} -0.0795 \\ 0.1993 \\ 0.9767 \end{bmatrix} \tag{4.2.14}$$

$$\vec{b_r'} = \begin{bmatrix} 0.0838 \\ 0.9959 \\ 0.0339 \end{bmatrix} = \begin{bmatrix} -0.3512 & 0.9296 & -0.1122 \\ -0.0631 & 0.0961 & 0.9934 \\ 0.9342 & 0.3559 & 0.0249 \end{bmatrix} \begin{bmatrix} -0.0606 \\ 0.1857 \\ 0.9807 \end{bmatrix} \tag{4.2.15}$$

$$\vec{c_r'} = \begin{bmatrix} 0.1219 \\ 0.9925 \\ 0.0052 \end{bmatrix} = \begin{bmatrix} -0.3512 & 0.9296 & -0.1122 \\ -0.0631 & 0.0961 & 0.9934 \\ 0.9342 & 0.3559 & 0.0249 \end{bmatrix} \begin{bmatrix} -0.1006 \\ 0.2105 \\ 0.9724 \end{bmatrix} \tag{4.2.16}$$

CHAPTER

5

VERIFICATION AND TESTING

In this chapter, we will explain the test procedures followed to validate the design of the GranaSAT Star Tracker. We can split this chapter in two sections. In the first one, which corresponds with sections from 5.1 to 5.5 we present the actions designed before the flight to test our system.

Because of the impossibility of going to the countryside enough times to test the Star Tracker, we present a method used to build artificial images which allow us to test it properly in laboratory, this is presented in section 5.1.

These artificial images allow us to realise a sky covering analysis in section 5.2, testing the Centroiding algorithm in section 5.3, obtaining the ϵ paramenter in section 5.4, and finally find the attitude accuracy in section 5.5.

In the last section of this chapter (section 5.6), we present to the reader the results obtained during the BEXUS19 flight, lastly, in section 5.6.1.1 we perform a failure analysis in order to study the cases which makes our design to fail.

5.1 Artificial Star Fields

In order to test the different Star Tracker related algorithms developed, a set of artificial test images were built following the next steps. This process is described in [29].

- First of all, a random direction cosine matrix, A, is obtained from Eq 5.1.1:

$$A = \begin{bmatrix} cos\theta_1 & sin\theta_1 & 0 \\ -sin\theta_1 & cos\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos\theta_2 & 0 & -sin\theta_2 \\ 0 & 1 & 0 \\ sin\theta_2 & 0 & cos\theta_2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\theta_3 & sin\theta_3 \\ 0 & -sin\theta_3 & cos\theta_3 \end{bmatrix} \tag{5.1.1}$$

Where θ_1, θ_2 and θ_3 are the yaw, pitch and roll angles. Figure 5.1:

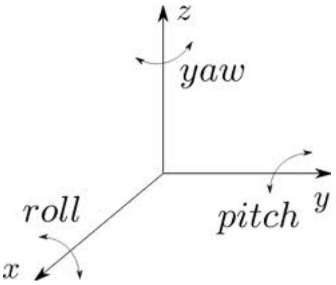


Figure 5.1 – *Yaw, pitch and roll*

- Once the direction cosine matrix has been generated, the *boresight* direction, which defines the celestial region pointed is computed using Eq 5.1.2:

$$\vec{u}_{boresight} = A^T \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \tag{5.1.2}$$

- Every star present in the image has to fulfill Eq 5.1.3:

$$cos^{-1}\left(\vec{u}_{star} \cdot \vec{u}_{boresight}\right) < FOV \tag{5.1.3}$$

where $\vec{u}_{star}$ can be obtained from the RA and DEC coordinates in the HP.

- The coordinates in the image frame can be determined using the Thales' theorem, Eq 5.1.4

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{f}{pp}\frac{1}{z}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 640 \\ 480 \end{bmatrix} \tag{5.1.4}$$

Where f is the focal length, pp is the pixel size and x, y, z are the cartesian coordinates of the star. Once the (u, v) coordinates has been obtained, they are stored in conjunction with the HIP number of the star in a text file.

- The brightness for each star is computed with Eq 5.1.5

$$star_{brightness} = t_{exp}\frac{\pi}{4}d^2\tau\lambda 10^{-0.4m}G \tag{5.1.5}$$

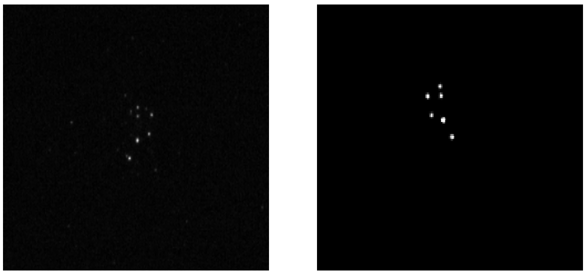
Where t_{exp} is the exposure time in seconds, d is the lens aperture diameter in cm, τ is the transmittance, λ is the spectral range in μm, m is the Visual Magnitude of the star and G is the gain parameter.

- Once the *brightness* has been obtained for each star, it is spread over the surrounding pixels for an image with resolution $n \times m$ as explained in algorithm 6:

Algorithm 6 Spreading Brightness

```
Define a number of $np$ points
for $i = 1$ to $i = np$ do
    $u_i = u_{star} + \sigma * randn()$        ▷ $randn()$ randomly generated number with Gaussian distribution with zero mean and standard deviation one
    $v_i = v_{star} + \sigma * randn()$
    if $u_i \leq n$ && $v_i \leq m$ then
        $image(u_i, v_i) + +$
    end if
end for
for $i = 1$ to $i = n$ do
    for $j = 1$ to $j = m$ do
        $image(i, j) = \frac{image(i,j)}{Saturation_{limit}}$
        if $image(i, j) \geq 1$ then
            $image(i, j) = 255$
        else
            $image(i, j) = image(i, j) \times 255$
        end if
    end for
end for
```

Figure 5.2a shows an image of the Pleiades constellation taken with the DMK 41BU02 camera. In order to show the performance of the algorithm used for creating artificial images, figure 5.2b shows an artificial image of the Pleiades constellation. Table 5.1 shows the parameters used by the algorithm to build this image.



(a) *Pleiades Constellation* **(b)** *Artificial Pleiades Constellation*

Figure 5.2 – *Image Comparison*

Exposure Time (s)	50
Aperture Diameter (cm)	1.4
Transmittance	1
Spectral Range (μm)	0.55
Gain	100000

Table 5.1 – *Parameters Used to Create Image in Figure 5.2b*

5.2 Sky Covering

In this section, we will analise the probability of finding enough stars in the camera FOV to allow the Star Tracker algorithm to work properly. As explained in section 4.2.3, the algorithm implemented needs at least three stars registered as *guide stars* in the image to be able to find the star pattern.

Using the method to build artificial images described in section 5.1, one thousand of images were build. In all of them, the *boresight* direction was randomly obtained to ensure the most of real situations as possible. The following histogram (figure 5.3) shows the probability of finding at least one *guide star* in an image which Visual Magnitude is inside the intervals defined in table 5.2:

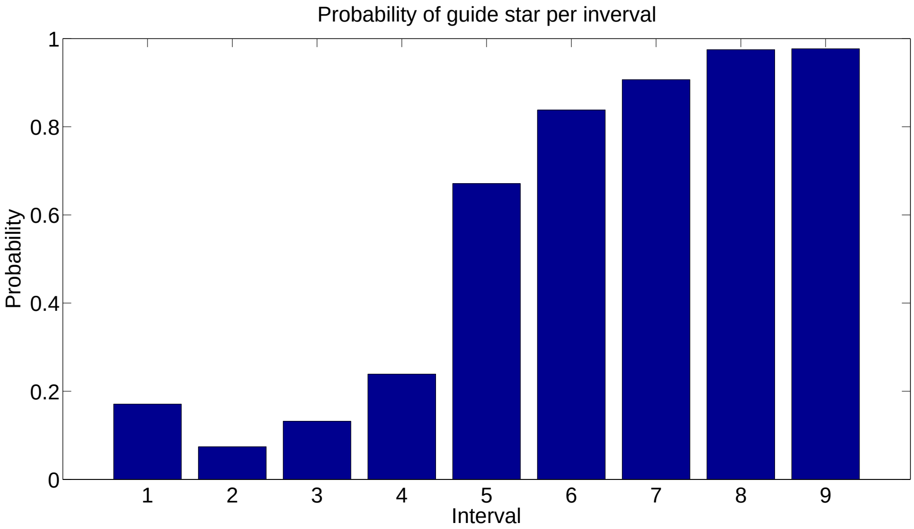


Figure 5.3 – *Probability of guide stars per interval*

Interval	Visual Magnitude Range
1	Vmag $\leq$ 0.5
2	0 $\geq$ Vmag $<$ 1
3	1 $\geq$ Vmag $<$ 1.5
4	1.5 $\geq$ Vmag $<$ 2
5	2 $\geq$ Vmag $<$ 2.5
6	2.5 $\geq$ Vmag $<$ 3
7	3 $\geq$ Vmag $<$ 3.5
8	3.5 $\geq$ Vmag $<$ 4
9	4 $\geq$ Vmag $<$ 4.5

Table 5.2 – *Interval of Visual Magnitude in figure 5.3*

Additionally, the following histogram (figure 5.4) shows the number of *guide stars* per image. As the reader could see, all of the one thousand of the image created has at least 5 *guide stars* present, then theoretically the algorithm designed could find the star pattern in each image.

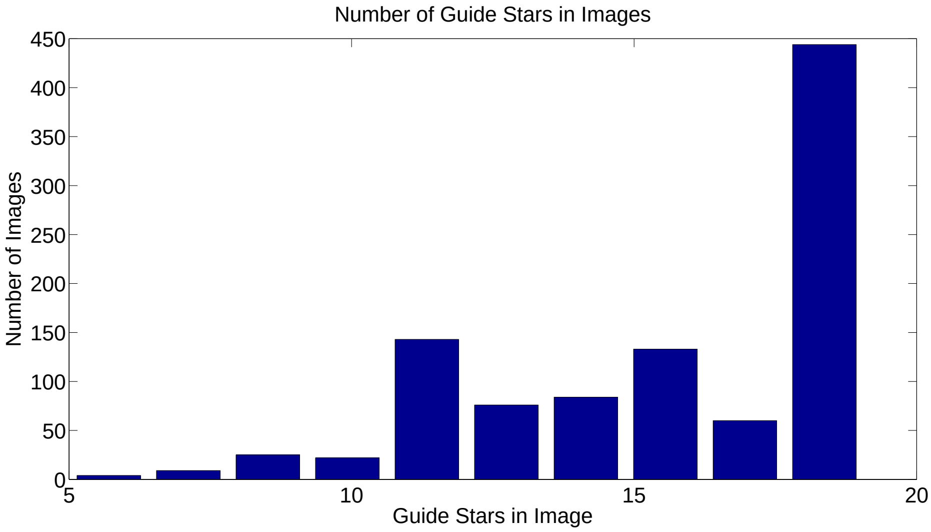


Figure 5.4 – *Number of guide stars per image*

5.3 Centroiding Accuracy

The artificial images created in section 5.1 can be used to evaluate the accuracy of the centroiding algorithm proposed in section 4.2.1. The $\left(u^{'}, v^{'}\right)$ coordinates obtained with the centroid algorithm are compared with the real coordinates (u, v) from the artificial image. The absolute error in pixel units is computed through Eq 5.3.1.

$$e_{pixel} = \sqrt{(u' - u)^2 + (v' - v)^2} \tag{5.3.1}$$

The following figure 5.5 shows the the histogram of the absolute error obtained with the centroiding parameters $pixel_{threshold} = 180$, ROI=7.

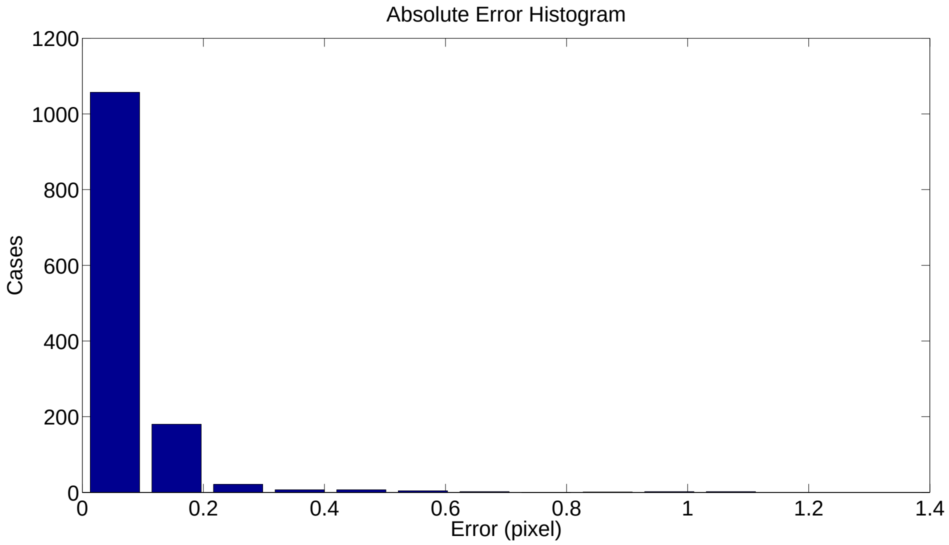


Figure 5.5 – *Histogram of the Centroid algorithm Absolute Error*

The following table 5.3 shows the evolution of the mean (m) and standard deviation (σ) of the error with different values of ROI.

ROI	m	σ
5	0.113	0.065
7	0.080	0.046
9	0.077	0.050
11	0.082	0.060
13	0.091	0.075
15	0.123	0.145
17	0.153	0.192
19	0.177	0.211

Table 5.3 – *Evolution of the Absolute Error with ROI*

5.4 Obtaining ϵ parameter

The ϵ parameter is the measured relative error of the angular distance between stars. The following method was used to determine it.

1. Create a number of artificial images.
2. Analyse them with the Centroiding algorithm. After this process, n centroids are obtained for each image.
3. Use the text file with the information of the stars in the artificial image to correspond

each centroid detected with a star.

4. Measure the angular distance as it appears in algorithm 7.

Algorithm 7 Measuring angles

```
for $i = 1$ to $i = DetectedCentroids - 1$ do
    get the unitary vector $\vec{u_i}$ in the image frame associated with $Centroid(i)$
    for the star $S_i$
    for $j = i + 1$ to $i = DetectedCentroids$ do
        get the unitary vector $\vec{u_j}$ in the image frame associated with $Centroid(j)$
        for the star $S_j$
        get the angular distance $d_{ij} = (\vec{u_i} \cdot \vec{u_j})$
        obtain the vectors $r_i, r_j$ in the ECI reference frame for the stars $S_i, S_j$
        get the real angular distance $r_{ij} = (\vec{r_i} \cdot \vec{r_j})$
        obtain the absolute value of the relive error $\epsilon_{ij} = \frac{|r_{ij} - d_{ij}|}{r_{ij}} 100$
        store the value $\epsilon_{ij}$
    end for
end for
```

The following figure 5.6 shows the histogram of the relative error of the angular distance when no additional error is introduced but the Centroiding algorithm itself. A total of 100 images were created and 1912 angles could be obtained. The unitary vectors of each centroid were obtained considering ideal parameters, these are f=12 mm, $x_0 = 640$, $y_0 = 480$

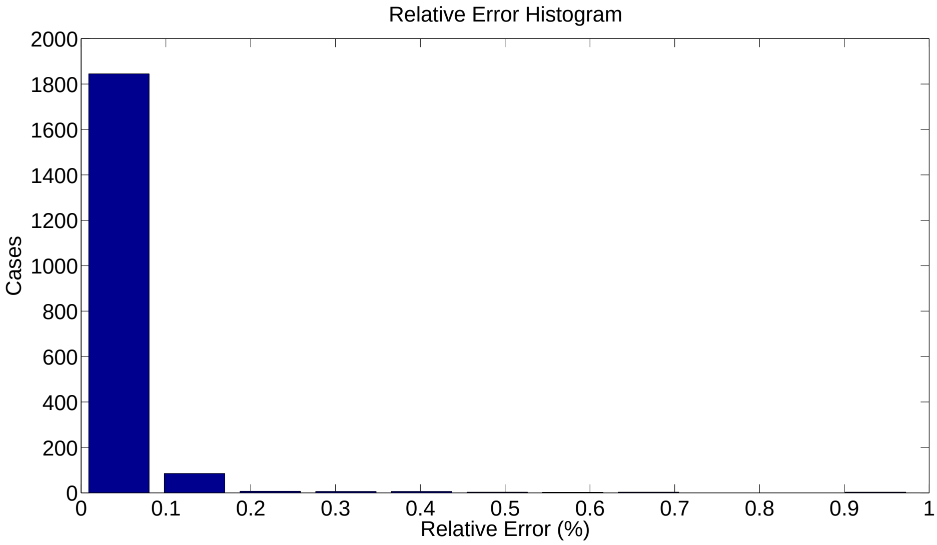


Figure 5.6 – *Relative Error Histogram*

The following image 5.7 shows the histogram of the relative error between angles for real images taken with the DMK 41BU02. As we can see, the relative error is around 3%, far away from the error obtained for the artificial images, which is less than 0.2% for the 93% of the angles measured. This disadjustement in the relative error comes from the non-ideal camera parameters (f, u_0, v_0). To solve this issue, a camera calibration was done. The procedure followed is explained in section 5.4.1.

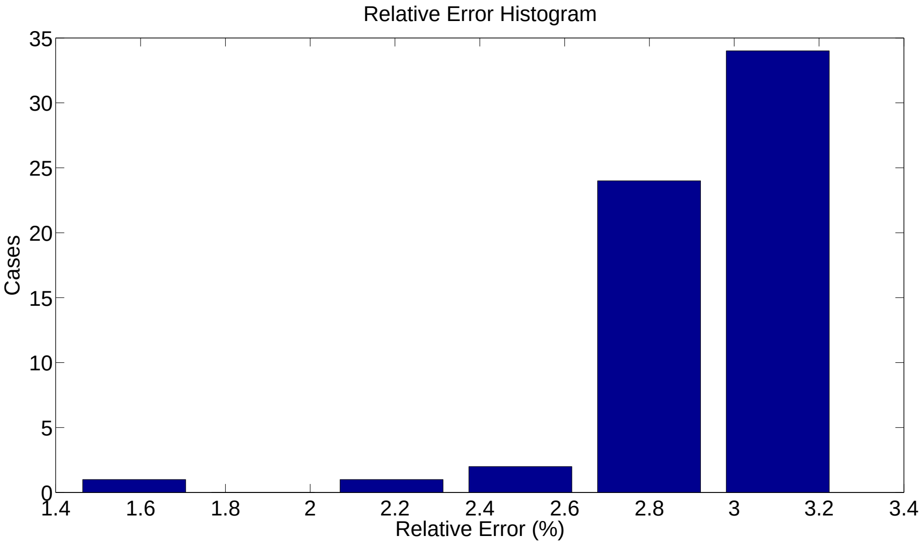


Figure 5.7 – *Uncalibrated Camera Error*

5.4.1 Camera Calibration

As we can extract from figure 5.7, we need to calibrate our camera to obtain a smaller relative error. Equation 5.4.1 shows the relationship between a 3D point $M = [X, Y, Z]$ and its image projection $m = [u, v]$ in a pinhole system[59].

$$s\hat{m} = A \begin{bmatrix} R & t \end{bmatrix} \hat{m} \qquad A = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.4.1}$$

Where $\hat{m} = [u, v, 1]$, $\hat{M} = [X, Y, Z, 1]$, s is an arbitrary scale factor, (R, t) are the *extrinsic parameters* (i.e the rotation and translation matrix which relates the world coordinate system to the camera coordinate system), and A is the camera *intrinsic matrix*, (u_0, v_0) are the coordinates of the principal point, $\alpha = f_x a$ and $\beta = f_y b$ the scale factors in image, f is the focal length and γ describes the *skew* of the two image axes.

During the calibration, the *intrinsic* parameters were found. As the calibration of a camera could be a very tedious process, we use a *toolbox*[6] for MATLAB which allows us to find the intrinsic parameters almost automatically. Figure 5.8 shows the twenty calibration images employed to calibrate the DMK 41BU02 camera.

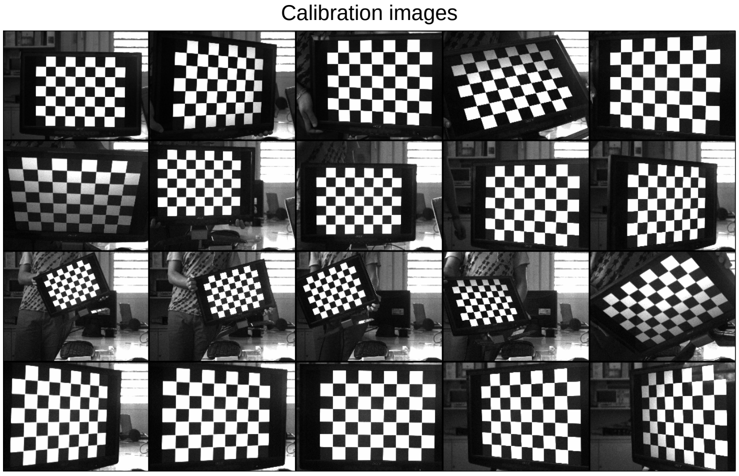


Figure 5.8 – *Calibration Images*

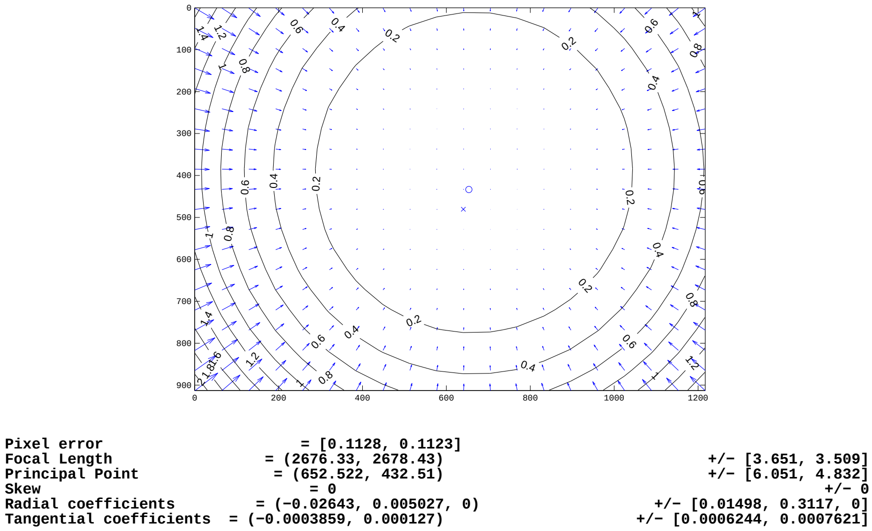
The calibration *toolbox* only needs the size in *mm* (360 mm × 360 mm in our case) of each square of the chess table to determine the *intrinsic* parameters as well as the *radial* and *tangential* distortion parameters. The *radial* distortion displaces a point *radially* in the image frame. The cause of the *tangential* distortion is that the surfaces of the lens are not strictly collinear[19]. The altered location u_i, v_i in the image frame due to both kinds of distortion can be modeled as follows:

$$u_i = \overbrace{\hat{u}\left(1 + k_1 r_i^2 + k_2 r_i^4\right)}^{\text{radial distortion}} + \overbrace{2p_1 \hat{u}_i \hat{v}_i + p_2\left(r_i^2 + 2\hat{u}_i{}^2\right)}^{\text{tangential distortion}} \tag{5.4.2}$$

$$v_i = \overbrace{\hat{u}\left(1 + k_1 r_i^2 + k_2 r_i^4\right)}^{\text{radial distortion}} + \overbrace{2p_2 \hat{u}_i \hat{v}_i + p_1\left(r_i^2 + 2\hat{v}_i{}^2\right)}^{\text{tangential distortion}} \tag{5.4.3}$$

Where k_1, k_2 are the coefficients for the radial distortion, $\hat{u}_i$, $\hat{v}_i$ are the non-displaced pixels, p_1, p_2 are the coefficients for the tangential distortion and $r_i = \sqrt{\hat{u}_i + \hat{v}_i}$.

Once the calibration *toolbox* has analyzed the calibration images, it builds a calibration model which is shown in figure 5.9.



```
Pixel error                    = [0.1128, 0.1123]
Focal Length               = (2676.33, 2678.43)                          +/- [3.651, 3.509]
Principal Point             = (652.522, 432.51)                          +/- [6.051, 4.832]
Skew                            = 0                                                   +/- 0
Radial coefficients        = (-0.02643, 0.005027, 0)                +/- [0.01498, 0.3117, 0]
Tangential coefficients  = (-0.0003859, 0.000127)            +/- [0.0006244, 0.0007621]
```

Figure 5.9 – *Camera Distortion Model*

As we can extract from figure 5.9, the camera matrix would be the following, the *radial* and *tangential* parameters were found negligible:

$$A = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2676.33 & 0 & 652.522 \\ 0 & 2678.43 & 432.51 \\ 0 & 0 & 1 \end{bmatrix} \qquad (5.4.4)$$

The parameters are given in pixel units, then, in the case of the α and β parameters they need to be scaled to the image sensor pixel size (4.65 μm) to have the estimated real focal length for the camera which is 12.4 mm regarding to the calibration. If we recompute the relative error obtained for real images with the *intrinsic* we will get the results shown in figure 5.10:

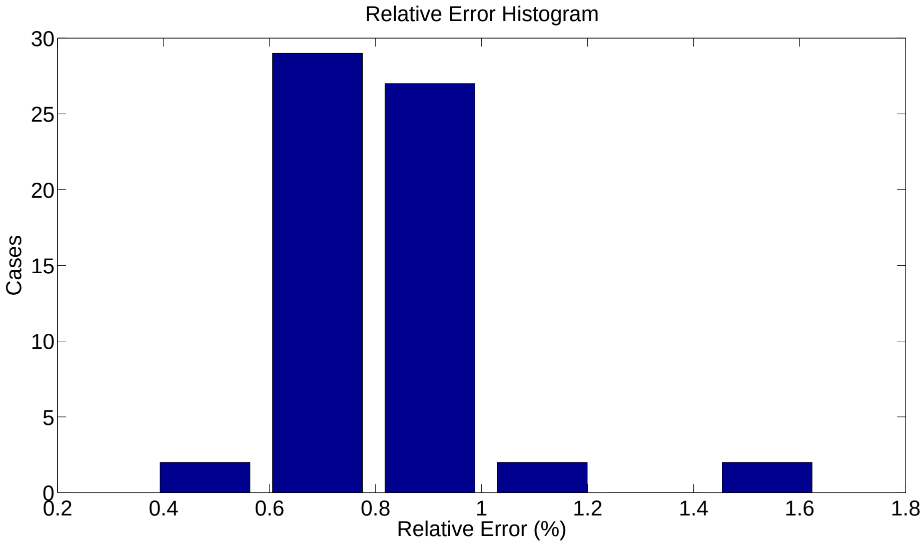


Figure 5.10 – *Relative Error After Calibration*

As we can see, the relative error has changed from around 3% to 0.8%. Even though this results are better than these results without camera calibration, we tuned the parameters iteratively in order to check if we can improve them even more, this process is shown in figure 5.11

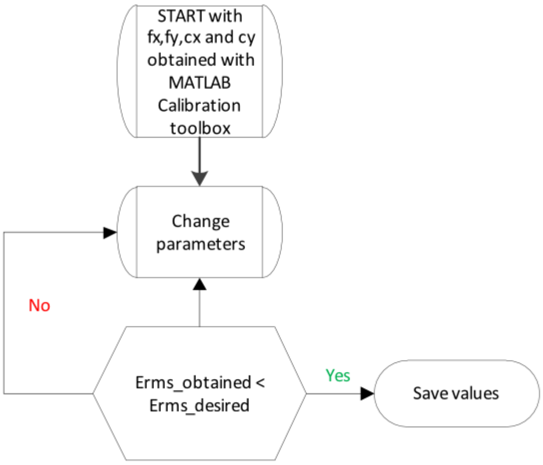

START with fx,fy,cx and cy obtained with MATLAB Calibration toolbox
Change parameters
No
Erms_obtained < Erms_desired
Yes
Save values

Figure 5.11 – *Parameters Calibration Tunning*

Finally, the final *intrinsic* parameters were obtained. The histogram of the final relative error obtained is shown in figure 5.12:

$$A = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2657.33 & 0 & 666 \\ 0 & 2657.33 & 440 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.4.5}$$

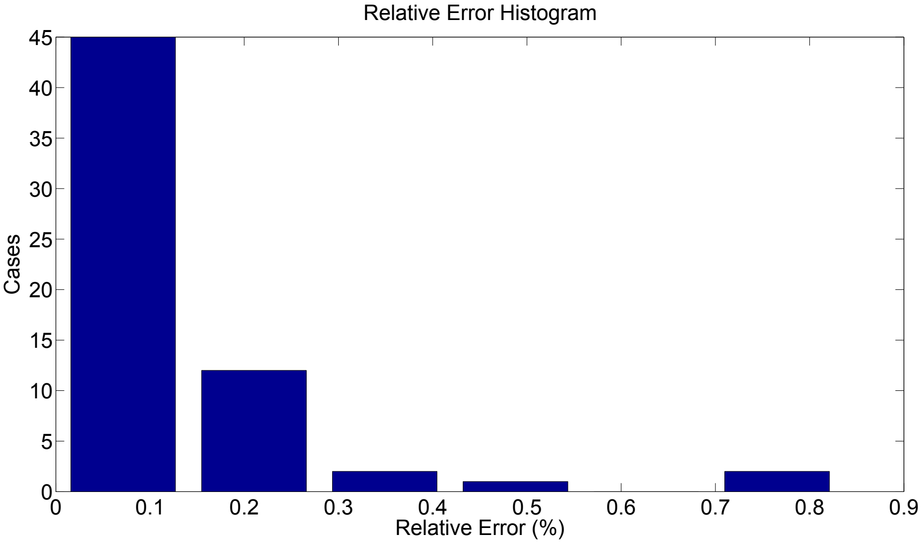


Figure 5.12 – *Final Relative Error*

5.5 Star Tracker Accuracy

The same one thousand artificial images created in section 5.2 were used in this section to find the accuracy of the LIS routine in laboratory conditions, the real accuracy of this routine will be obtained in section 5.6.1.

First, we evaluated the LIS routine proposed in section 4.2 making the software analyse the set of images created, as it is shown in table 5.4, the rate of success found was 97%.

Images Created	1000
Correctly Analysed Images	970
Incorrectly Analysed Images	30

Table 5.4 – *LIS Routine Success Rate*

Once the Attitude matrix A has been obtained as it appears in 4.2.4, the error in each x, y, z axis is computed as it appears in Eq. 5.5.1, where A is the computed attitude matrix and R is the real attitude matrix (Eq. 5.1.1):

$$Matrix_{error} = A^{-1}R = \begin{bmatrix} \approx 1 & -\theta_z & -\theta_y \\ \theta_z & \approx 1 & -\theta_x \\ \theta_y & \theta_x & \approx 1 \end{bmatrix} \tag{5.5.1}$$

The absolute error in arcsecs for each axis is computed as follows:

$$x_{err} = 3600 \left(90 - cos^{-1}\left(\theta_x\right)\right) \tag{5.5.2}$$

$$y_{err} = 3600 \left(90 - cos^{-1}\left(\theta_y\right)\right) \tag{5.5.3}$$

$$z_{err} = 3600 \left(90 - cos^{-1}\left(\theta_z\right)\right) \tag{5.5.4}$$

Figure 5.13 shows the histogram of the error for each axis using the artificial images created, as we can see, the error is between the range $[-50, 50]$ arcsecs.

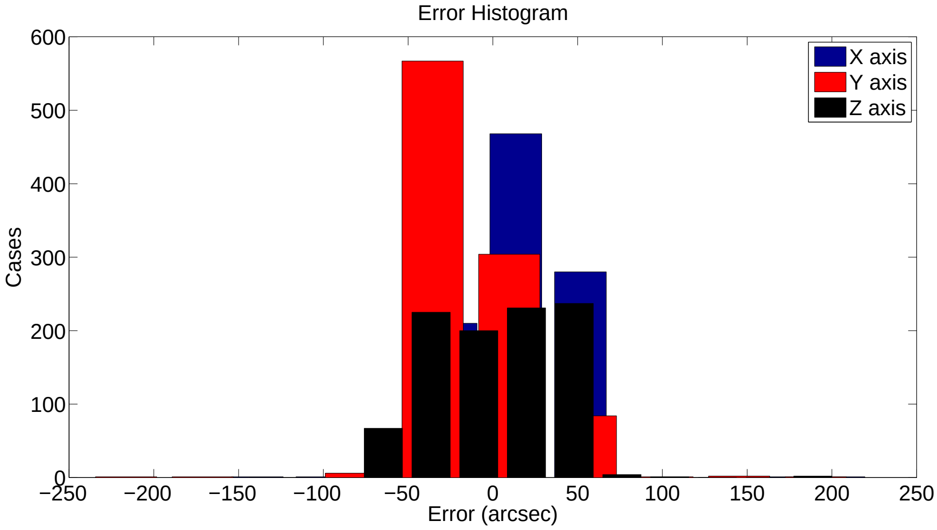


Figure 5.13 – *Attitude Error Histogram*

5.6 Flight

The BEXUS18/19 launch campaign took place in the SSC in Kiruna(Sweden) from the 3^{rd} to the 13^{th} of October, 2014. During these days, 9 teams from countries such as Germany, Italy, Spain and Sweden (figure 5.15) tested their experiments in a real flight, the BEXUS balloon is shown in figure 5.14. Specifically, the experiments and teams involved were:

- **A5-UNIBO** (Bologna, Italy): this experiment studied the relationship between the cloud formation and the atmospheric aerosols.

- **Polaris** (Padova, Italy): this experiment tested a new actuator to control the thermal properties of a radiator.

- **TORMES 2.0** (Barcelona, Spain): this experiment tested a new reflectometer called *PYCARO*(P(y) & C/A ReflectOmeter). It can be used to build topographic maps through reflections of GPS signals.

- **GranaSAT** (Granada,Spain): this team tested three Attitude Determination Systems: Star Tracker, Horizon Sensor and Magnetometer.

- **ADAM** (Kiel, Germay): this team tested the angular distribution of charged particles generated due to cosmic rays in the atmosphere.

- **COUGAR** (Louleå, Sweden): this team tested a radio link to control a *rover* from the Ground Station.

- **AFIS-P** (München, Germany): this team tested a prototype of the Antiprotron Flux In Space (AFIS) detector.

- **ARCA** (Jena, Germany): This team tested a device able to receive and decode ADS-B signals from airplanes as far as 1000 km.

Table 5.5 shows the dates of each flight and the experiments carried by the balloon. In figure 5.16, shows the flight trajectory for each balloon. As they can see, they were very similar in length and float altitude. In this document, we will focus our attention on the relevant data obtained for the Star Tracker subsystem in the BEXUS19 balloon. The results will be discussed in section 5.6.1.

	BEXUS18	**BEXUS19**
DATE	October 9^{th}, 2014	October 8^{th}, 2014
Payload	COUGAR	GranaSAT
	AFIS-P	TORMES2.0
	A5-UNIBO	ADAM
	ARCA	TAMA-OS
	POLARIS	

Table 5.5 – *BEXUS18/19 Dates and Payloads*

5.6.1 Flight Data Analysis

As it appears in table 5.5, the balloon BEXUS19 was launched on October, the 8^{th}. A nominal BEXUS flight has the phases explained below, its altitude profile is sketched in figure 5.17. The description of each phase has been obtained from [13]:

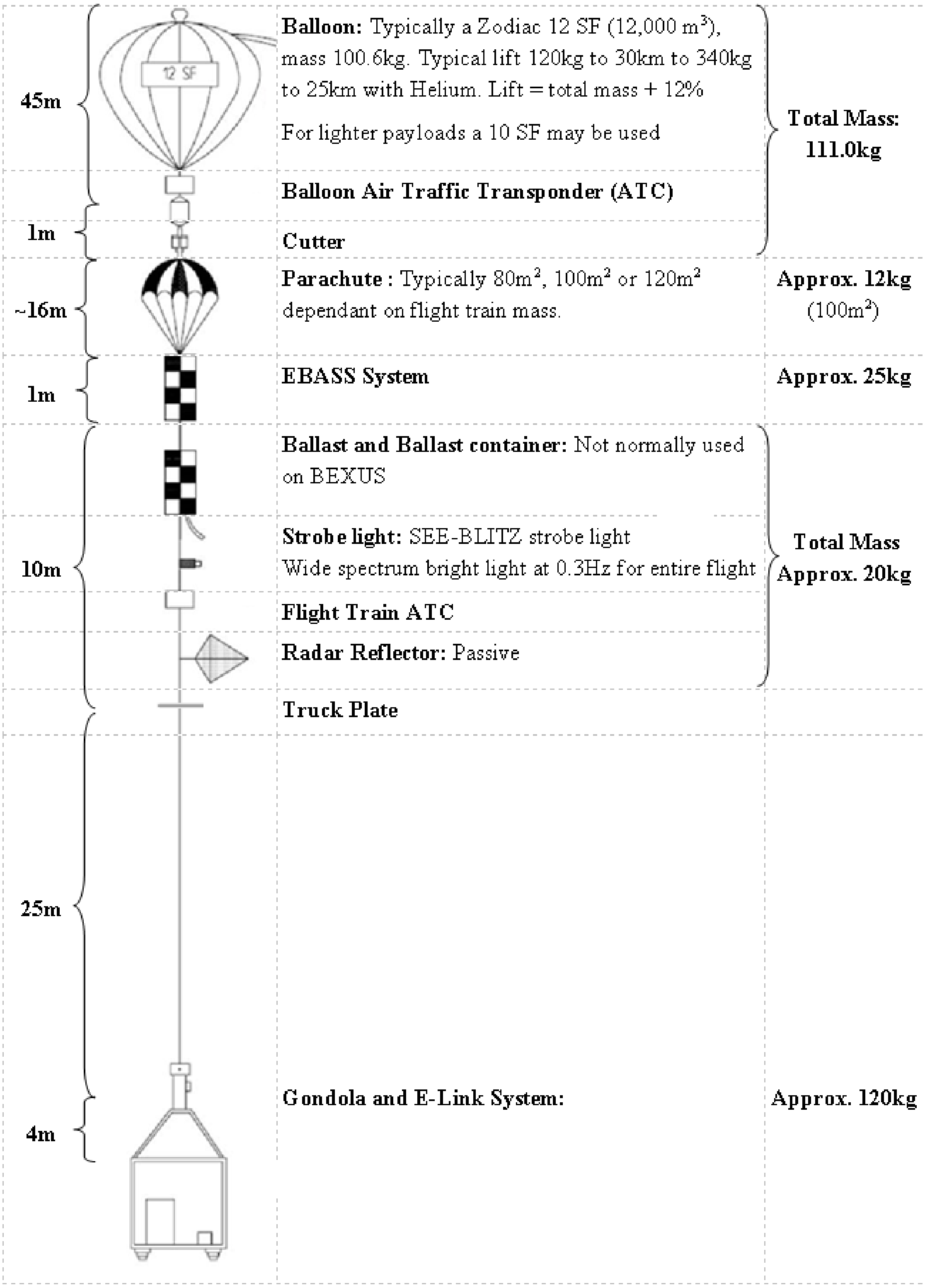
- **Launch**: the payload is released when the balloon inflation with Helium is completed.

- **Ascending Phase**: The nominal ascent speed is 5 m/s. Depending on float altitude and variations in speed, this phase takes approx. 1.5 hours. A slight oscillating movement is experienced. Expect an initial drift above ground of 5-10 m/s.

- **Float Phase**: when the total mass of the system and the buoyancy of the gas reaches equilibrium, the ascent phase stops. During float there are only minor changes in altitude (±200 m). If the sun sets during flight, the balloon will begin to descend due to the cooling of the gas.

  The payload mass influences the maximum altitude. The final altitude is calculated shortly before launch and may vary between 25 and 30 km. The nominal flight time is one to five hours.

- **Descend**: to end the flight, the cutter is activated, causing the balloon to separate from the rest of the flight train and rip open. There is a parachute system that brings down everything below the cutting device.



Balloon: Typically a Zodiac 12 SF (12,000 m^3), mass 100.6kg. Typical lift 120kg to 30km to 340kg to 25km with Helium. Lift = total mass + 12%

For lighter payloads a 10 SF may be used

Balloon Air Traffic Transponder (ATC)

Cutter

Total Mass: 111.0kg

Parachute : Typically 80m^2, 100m^2 or 120m^2 dependant on flight train mass.

Approx. 12kg (100m^2)

EBASS System

Approx. 25kg

Ballast and Ballast container: Not normally used on BEXUS

Strobe light: SEE-BLITZ strobe light
Wide spectrum bright light at 0.3Hz for entire flight

Flight Train ATC

Radar Reflector: Passive

Total Mass Approx. 20kg

Truck Plate

Gondola and E-Link System:

Approx. 120kg

Figure 5.14 – *BEXUS Balloon*

Figure 5.15 – *BEXUS18/19 Campaign. Teams and Organisers. Photo: Rocket/Balloon Experiments for University Students (Facebook's page)*

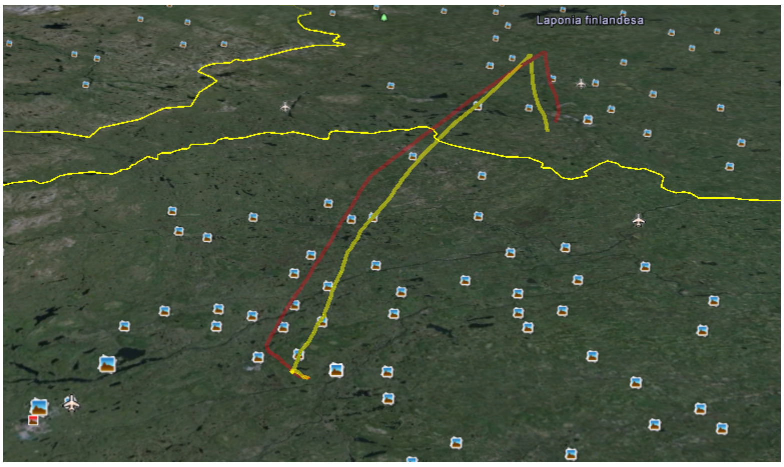


Figure 5.16 – *Balloon's trajectory*

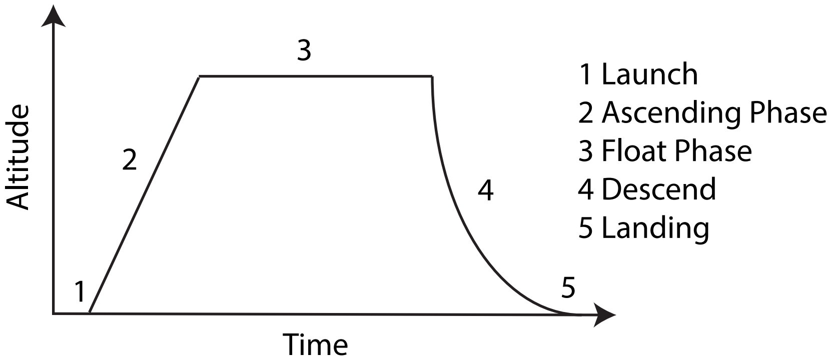


Figure 5.17 – *Float Phases*

A small period of reduced gravity will be experienced, but the gondola may tumble and it's suggested that this is not particularly suitable for microgravity experiments. The descent speed is high from the start, due to the thin atmosphere. Closer to the ground, it will stabilize at approximately 7-8 m/s.

- **Landing**: landing is always planned to be in sparsely-populated areas, preferably without any lakes. The landing velocity is approximately 7-8 m/s. This is equivalent to a drop from approximately 3 m. There is a shock-absorbing material at the bottom of the gondola that lowers the shock load at landing. Nominally, the landing is gentle with no damage to the experiments.

  On rare occasions we have seen landing shocks up to 35 g when landing in rocky terrain. A water landing is softer but comes with another problem, since the gondola is not watertight.

  Orientation is also not guaranteed and the gondola may be on its side or upside down at landing.

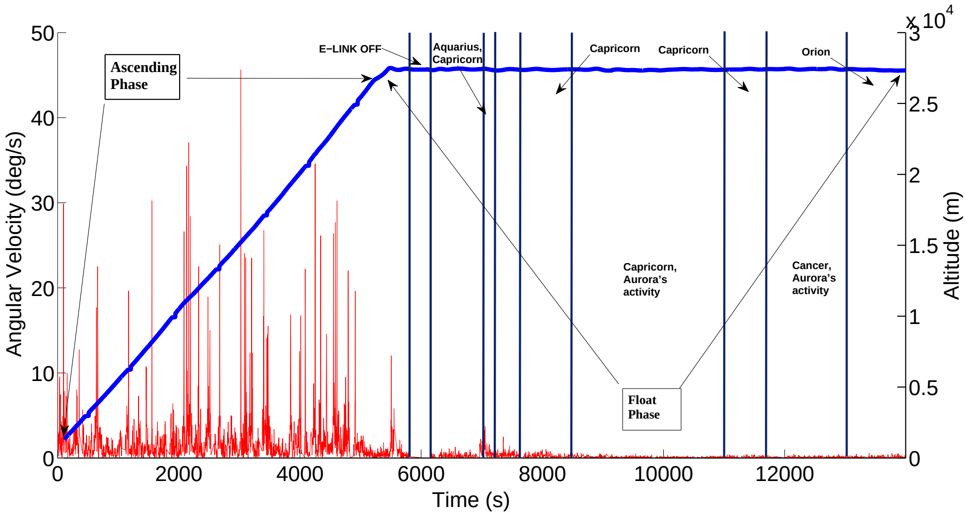


Figure 5.18 – *Angular Velocity and Altitude Profile in BEXUS19*

Table 5.6 showns the relevant events for the BEXUS19 balloon from the Star Tracker point of view. In addition, figure 5.17 graphically shows where the altitude profile for this flight, as well as the angular velocity of the gondola. The angular velocity (α) was computed as it appears in figure 5.19. It was obtained considering the direction of the Earth's Magnetic Field measured with the LSH303 magnetometer at two instants $T1$ and $T2$, each one corresponds with the instant when two images are taken.

As we can extract from table 5.6, four constellations were observed. Figure 5.20, shows their approximated locations over the horizon during the flight. We have coloured in green the constellations where our results were satisfactory. Cancer constellation is coloured in red because the attempts to analyse the images obtained from it were unsatisfactory as we will discuss in section 5.6.1.1.

	UTC Time
Launch	17:55:47
First Image Registered	17:59:11
End of Ascending Phase	19:28:58
E-LINK Turned OFF	19:36:02
E-LINK Turned ON	19:39:50
Aquarius and Capricorn Constellation Observed	19:55:56 - 20:01:22
Capricorn Constellation Observed	20:09:36 - 20:21:29
Aurora's Activity	20:21:29 - 21:03:26
Capricorn Constellation Observed	21:03:26 - 21:14:05
Aurora's Activity, Cancer Constellation Observed, Jupiter Observed	21:14:05 - 21:38:29
Orion Constellation	21:38:26 - 21:52:29
Experiment Turned OFF	21:52:29
Beginning of Descend Phase	21:55:15
Last Data Registered by the EBASS	22:24:11

Table 5.6 – *Relevant Events in BEXUS19*

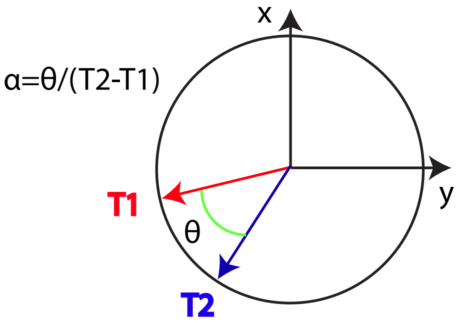


Figure 5.19 – *Angluar Velocity Computation*

Tables 5.7, 5.8, 5.9 and 5.10 show the general results for each one of Aquarius, Capricorn and Orion constellations. Once an image has been taken, it was analysed by the Star Tracker LIS routine as it appears in figure 5.22. Due to the high exposition time and gain value required to make the stars appearing in the image, for all images taken from Capricorn and

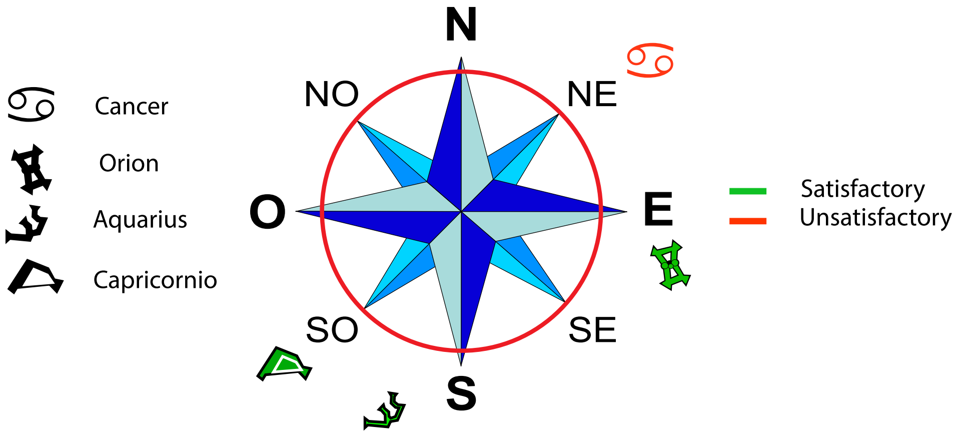
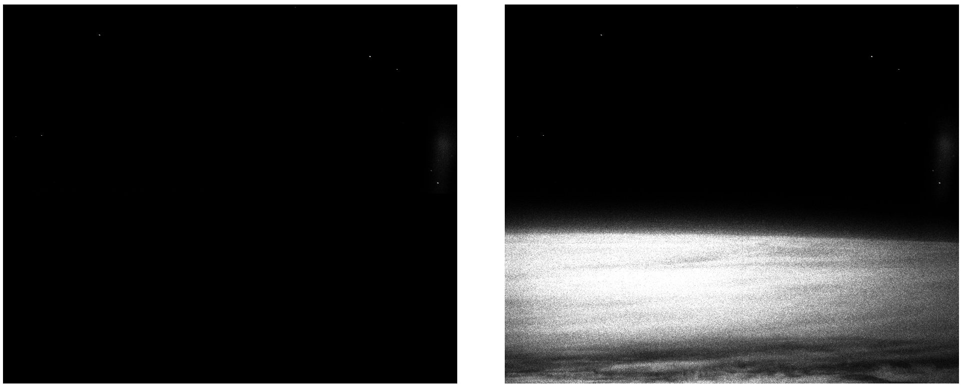


Figure 5.20 – *Approximated Constellation's location over the Earth's Horizon*

Aquarius constellation the Earth's Horizon was to be removed, because of its brightness was comparable to the stars' brightness. Figure 5.21b shows an image of Capricorn constellation where the horizon makes this image unable to be analysed by the LIS routine, in figure 5.21a the Earth's horizon has been removed.



(a) *Image Of Capricorn Constellation With The Earth Horizon Removed*

(b) *Image Of Capricorn With Earth Horizon*

Figure 5.21 – *Removing Horizon*

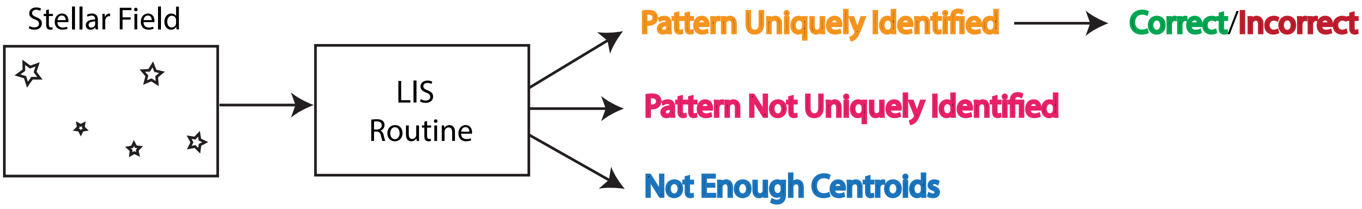


Figure 5.22 – *Possible Outputs of LIS Routine*

Total Images	94
Stellar Fields Uniquely Identified	38
Correct / Incorrect Stellar Fields	34/4
Not Enough Centroids	27

(a) *Aquarius-Capricorn Images*

Brightness	1
Gamma	20
Gain	1023
Exposure Time (s)	0.63

(b) *Camera Parameters*

Table 5.7 – *Aquarius-Capricorn 19:55:56 - 20:01:22*

Total Images	176
Stellar Fields Uniquely Identified	154
Correct / Incorrect Stellar Fields	154/0
Not Enough Centroids	4

(a) *Capricorn Images*

Brightness	1
Gamma	20
Gain	1023
Exposure Time (s)	1

(b) *Camera Parameters*

Table 5.8 – *Capricorn 20:09:36 - 20:21:29*

Total Images	129
Stellar Fields Uniquely Identified	93
Correct / Incorrect Stellar Fields	87/6
Not Enough Centroids	9

(a) *Capricorn Images*

Brightness	1
Gamma	35
Gain	1023
Exposure Time (s)	1.2 - 1.7

(b) *Camera Parameters*

Table 5.9 – *Capricorn 21:03:26 - 21:14:05*

Total Images	172
Stellar Fields Uniquely Identified	170
Correct / Incorrect Stellar Fields	170/0
Not Enough Centroids	2

(a) *Orion*

Brightness	1
Gamma	50
Gain	923
Exposure Time (s)	0.8

(b) *Camera Parameters*

Table 5.10 – *Orion 21:38:26 - 21:52:29*

As a summary, table 5.11 shows the statistics of the accuracy for the LIS routine developed. As we can see, the LIS routine was able to identify uniquely the 91% of the images not discarded because of the absence of enough centroids in the image. The 97.80% of these uniquely identified star fields were correctly matched. Consequently, the performance of the LIS routine for the Star Tracker is considered a success in absence of non-stars objects. We also show in figure 5.24 the histogram of the relative error of the angles measured in each constellation. As we can see, it validates our calibration too, since the error obtained is always bellow the 1% obtained in 5.4.1.

The attitude error for each star field is obtained once the stars appearing in the image has been identified. As explained in 4.2.4, the attitude matrix is obtained using the SVD procedure. Figure 5.23 shows the histogram of the error in arcsecs obtained for each one of the constellations analysed. It have been computed for each correctly identified stars as it appears in equation 5.6.1

$$e_i = 3600 \left| cos^{-1} \left(r_i \cdot Ab_i \right) \right| \tag{5.6.1}$$

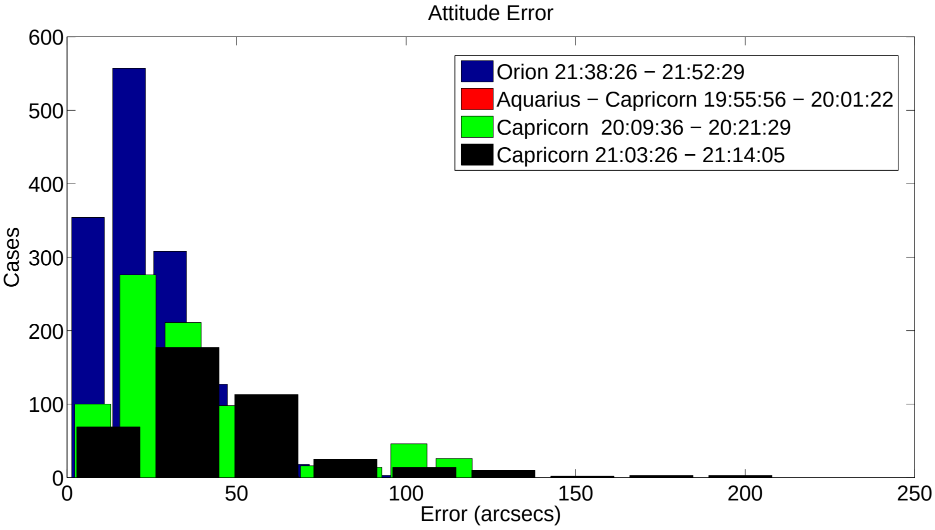


Figure 5.23 – *Histogram of the Attitude Error in Arcsecs*

Where e_i is the error obtained, r_i are the coordinates of a star in the ECI reference frame, A is the attitude matrix and b_i is the coordinates of the star in the camera frame.

Even though these results validates our design, we outlined in figure 5.20 that an entire constellation (Cancer) could no be matched properly. In section 5.6.1.1 we will analyse these situations which makes our LIS routine to fail.

Total of Images Analysed	571
Total of Uniquely Identified Star Fields **(Correct/Incorrect)**	455 445/10
Total of Not Uniquely Identified Star Fields	45
Total of Images Discarded **(Not Enough Centroids)**	71

Table 5.11 – *Summary of LIS Rutine Results*

5.6.1.1 Failure Analysis

During the flight, two main reasons made the Star Tracker software to fail in the identification of the star field in the image. The first one is due to the gondola motion, in figure 5.26 we can see the angular velocity for an approximated time of 7 minutes for each one of the previous constellations analysed in section 5.6.1.

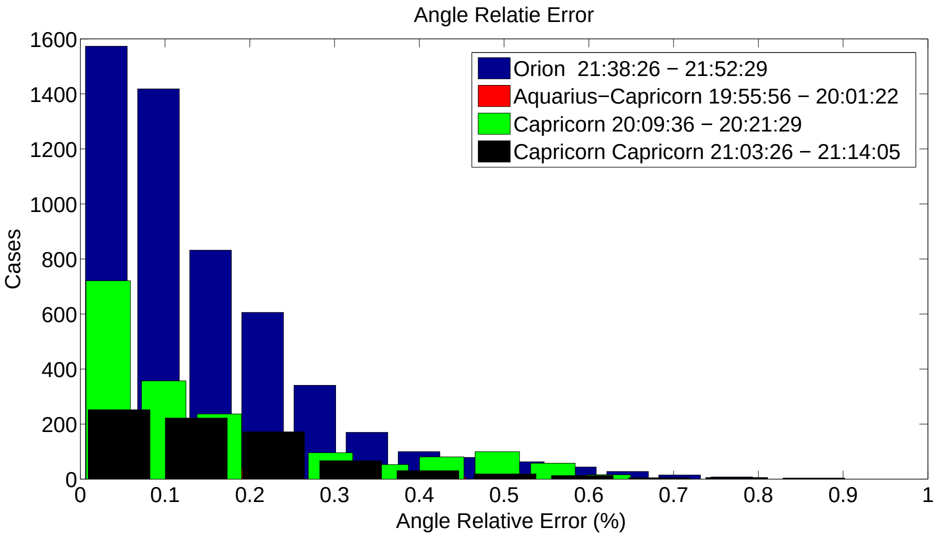
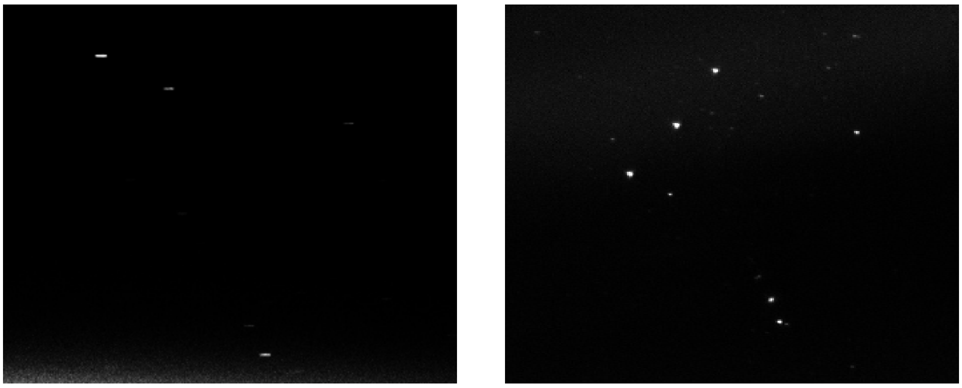


Figure 5.24 – *Sumary of Relative Error Measured*

Table 5.12 shows the mean of the angular velocity for each period in figure 5.26. As we can extract from it, the two periods of time with the larger angular velocity coincide with the two constellations in section 5.6.1 with incorrect identified stellar fields. As an example of how the gondola motion affects the quality of the images taken, figure 5.25b shows a picture of Orion constellation, as we can see the stars are quite well defined. In contrast, figure 5.25a shows an image of Capricorn constellation with blurred stars.

The LIS routine proposed did not fail with all the blurred images, but we found that 59 out of 74 not uniquely identified star fields were blurred.



(a) *Blurred Capricorn*

(b) *Orion Constellation*

Figure 5.25 – *Effects of the Gondola Motion*

The routine to identify the stars proposed in section 4.2.3 implies that the three most bright stars in the image have to be present in the GC. Otherwise the star identification will fail. The biggest example of this weakness was found from 21:14:05 -21:38:29 when Cancer

Perdiod	Mean of Angular Velolcity (deg/s)
19:55:56 - 20:01:22	0.69
20:09:36 - 20:15:25	0.13
21:03:26 - 21:11:20	0.27
21:38:26 - 21:46:16	0.13

Table 5.12 – *Mean of Radial Velocity for Periods in 5.26*

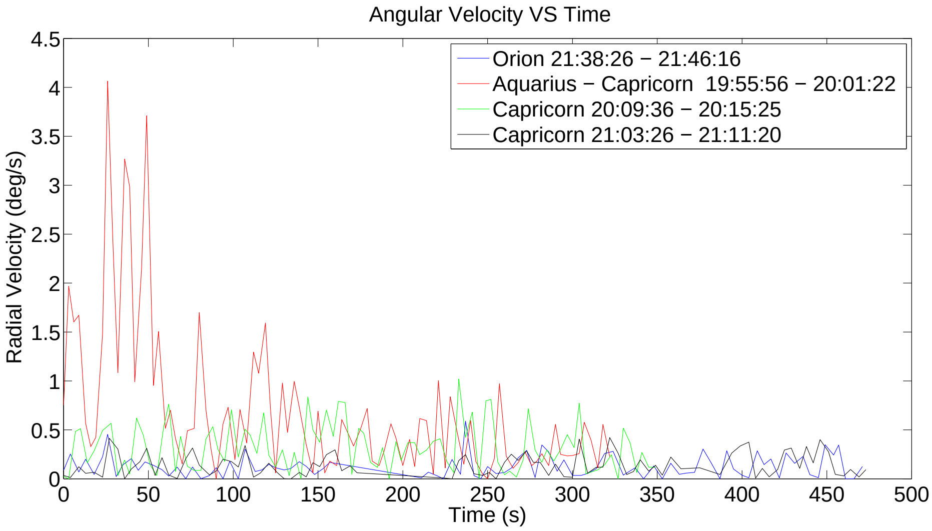


Figure 5.26 – *Radial Velocity*

constellation could be observed. However, as it is shown in figure 5.27, Jupiter was right in the middle of this constellation as the "the brightest star". For this reason, the algorithm was unable to identify properly none of the Cancer images.

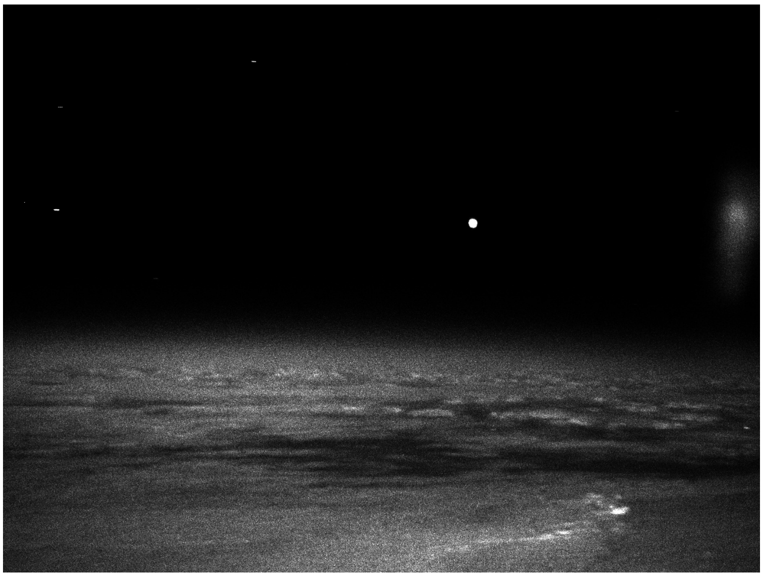


Figure 5.27 – *Jupiter as "the brightest star"*

Finally, during the flight we experimented long periods of time with high aurora activity (figure 5.18). In this case, its position in the image is impossible to predict as we did in the case of the Earth's Horizon. Figure 5.28 shows a beautiful, but useless image for the Star Tracker as the Aurora brightness is to high.

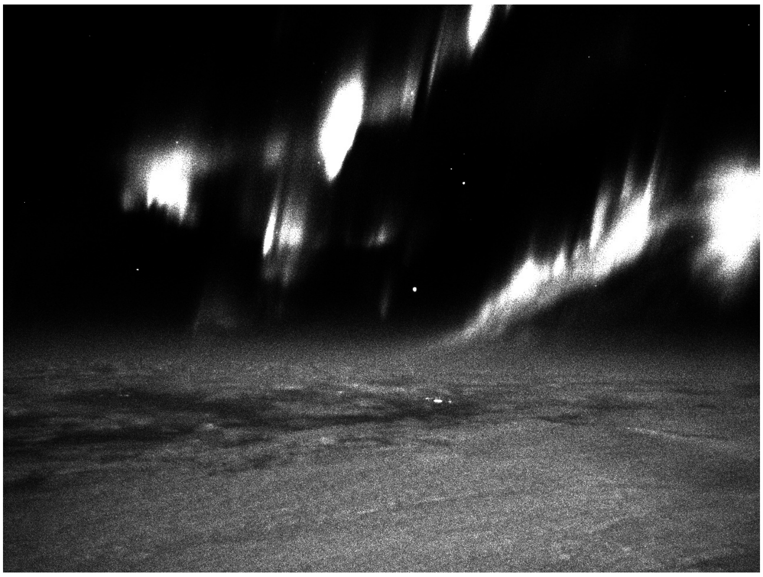


Figure 5.28 – *Aurora Borealis*

CHAPTER

6 CONCLUSIONS AND FUTURE LINES

In this document we have presented the development of the Star Tracker used in the GranaSAT experiment. This subsystem, for its complexity, is the most demanding of the three ADS implemented. Firstly, we had to deal with a complete new field as is the aerospace one. For a Telecommunication Engineering student as the author of this document is, it was arduous to get used with all the attitude concepts, the star trackers, the star pattern recognition algorithms and its implementation to make then efficient and reliable.

Not only the academic problems affected to the development of this work, in our case, the budget restriction was the main bone of contention. It affected all the stages of the project, i.e., from the material acquisition to the travel expenses to Holland and Kiruna. In spite of that, the whole team is proud of the work carried out. We have proven that the greatest hits can be achieved with effort rather than a huge pocket.

In this entire year working in an aerospace project, we have learn a little of how this industry works, its standards and how important a good documentation is, as well as we have learnt how to work in team. Usually students tend to think that they can work in groups, however, we all realised that it was an illusion. Only after getting involved in such a demanding project as the REXUS/BEXUS is you can say that you have learnt to work in groups.

The Star Tracker developed is far away from its implementation for a student designed nano-satellite. However, we think we have lay the groundwork for this purpose. The LIS algorithm implemented has proven reliable, it returned a 97.80% of accuracy in identifying stellar fields. The non-correct ones are within the expected, as the reason that make the Star Tracker LIS routine to fail cannot be controlled by us, i.e, gondola motion, aurora borealis and other objects as planets.

A large future work could be done with this work as basis. From the author's point of view, it would be interesting to implement an algorithm to discard objects like planets and other spacecraft if they are enough brighter to appear within the three most brilliants object in the image. We have seen our algorithm working well in abscense of this objects. It also can discard them in the previous stages of the algorithm, however, a future LIS algorithm should implement a procedure to discard them as stars in the previous stage of the algorithm, when the first triplet of stars are found.

The implementation of the tracking algorithm the next step for the true implementation of this star tracker. In this document we have focused our effort in the programming and testing of the LIS routine as ,well as one of the attitude determination method used, the SVD. The tracking routine will complete this work, unfortunately, its implementation was impossible due to scheduling constraints.

We want conclude this discussion remarking the support of the ESA, SNSB and DLR organizations for supporting and sponsoring the REXUS/BEXUS programme. This experience has been the most rewarding one for 10 students of the UGR, in which is included the author of this document. Thank you all for sharing your knowledge, experience and invaluable time.

REFERENCES

[1] Aptina Imaging. *MT9P031: Monochrome Addendum Introduction.* Available at https://www.google.es/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CCMQFjAA&url=http%3A%2F%2Fwww.aptina.com%2Fassets%2FdownloadDocument.do%3Fid%3D896&ei=4KpkVMmKJMPea0XmgvgE&usg=AFQjCNE-hccUhGlqi_S8-nIg0Gj8ATs7Qg&sig2=61pUQVpZxnKqC_wevdcb0w Accessed 15-February-2014.

[2] Aptina Imaging. *1/2.5-inch 5Mp CMOS Digital Image Sensor. MT9P031 Data Sheet, Rev. G*, 2005. Available at www.aptina.com Accessed 5-February-2014.

[3] Ball Aerospace & Technologies Corp. *CT-633 Stellar Attitude Sensor*. Available at http://www.ballaerospace.com/file/media/D0399_CT-633.pdf Accessed 7-May-2014.

[4] Banguert, J., Puata, W., Kaplan, G., Bartlett, J. H., W., F., and A., M. *User's Guide to NOVAS Version C3.1.* USNO, 2011.

[5] BeagleBoard.org Foundation, 2014. Available at http://beagleboard.org/ Accessed 30-November-2014.

[6] Bouguet, J. Camera calibration toolbox for matlab, 2013. Available at http://www.vision.caltech.edu/bouguetj/calib_doc/ Accessed 8-September-2014.

[7] Coley, G. Beaglebone black system reference manual, April 2003. Rev A5.S.

[8] de Données astronomiques de Strasbourg (CDS), C. Vizier service. Available at http://vizier.u-strasbg.fr/viz-bin/VizieR Last access March 2014.

[9] Digi-Key Corporation. Available at http:www.digikey.com/us/ Accessed 8-February-2014.

[10] elinux.ogr, 2014. Available at http://elinux.org/RPi_Hardware Accessed 24-March-2014.

[11] Enright, J., Sinclair, C., Grant, G., McVittie, G., and Dzamba, T. Towards star tracker only attitude estimation. In *24th Annual AIAA/USU Conference on Small Satellites* (2010).

[12] ESA. The hipparcos and tycho catalogues. *ESA SP-1200 1* (1997). Available at http://www.rssd.esa.int/index.php?project=HIPPARCOS&page=Overview Accessed 6-December-2014.

[13] EuroLaunch. *BEXUS user manual.* EuroLaunch, June 2014. v6-11, Available at http://www.rexusbexus.net/index.php?option=com_content&view=article&id=51&Itemid=63 Accessed 6-December-2014.

[14] Feissel, M & Mignard, F. The adoption of icrs on january 1998: meaning and consequences. *Astronomy and Astrophysics 331* (1998), 33–36.

[15] G.D Padfield, B. L. The birth of the flight control: An engineering analysis of the wright brothers' 1902 glider. *The Aeronautical journal* (2003), 697–718.

[16] Granada Imprusa, 2013. Available at http://granadaimprusa.org/ Accessed 2-December-2013.

[17] Hall, C. *Spacecraft Attitude Dynamics and Control.* Virginia Tech, 2003.

[18] Harold, B. A passive system for determining the attitude of satellite. *AIAA Journal 2(7)* (1964), 1350–1351.

[19] Heikkila, J., and Silven, O. A four-step camera calibration procedure with implicit image correction. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on* (Jun 1997), pp. 1106–1112.

[20] I-Nova. *PLA-MX Multi-purpose Camera*, 2012. Available at http://www.inova-ccd.fr/pub/PLA-Mx%20June%202014%20-%20English.pdf Accessed 7-February-2014.

[21] ICRS. *The International Celestial Reference System and Frame*, icrs note no.34 ed. 61 Av. de l'Observatorie,75014,Paris,France, 2006. Available at http://www.iers.org/MainDisp.cls?pid=46-1100077 Accessed 12-October-2014.

[22] The Imaging Source. *DKM 21B618 Monochrome Camera*, June 2014. Available at http://www.theimagingsource.com/en_US/products/cameras/usb-ccd-mono/dmk41bu02h/ Accessed 23-September-2014.

[23] The Imaging Source. *DKM 41BU02 Monochrome Camera*, June 2014. Available at http://www.theimagingsource.com/en_US/products/cameras/usb-ccd-mono/dmk41bu02h/ Accessed 23-September-2014.

[24] Kolomenkin, M., Pollak, S., Shimshoni, I., and Lindenbaum, M. Geometric voting algorithm for star trackers. *Aerospace and Electronic Systems, IEEE Transactions on 44*, 2 (April 2008), 441–456.

[25] Liebe, C. Star trackers for attitude determination. *Aerospace and Electronic Systems Magazine, IEEE 10*, 6 (Jun 1995), 10–16.

[26] Liebe, C. Accuracy performance of star trackers - a tutorial. *Aerospace and Electronic Systems, IEEE Transactions on 38*, 2 (Apr 2002), 587–599.

[27] Ltd., R. C., 2014. Available at http://uk.rs-online.com/web/ Accessed 10-February-2014.

[28] Markley, F. L. Attitude determination using vector observations and the singular value decomposition. *The Journal of the Astronautical Sciences 36*, 3 (1988), 245–258.

[29] McBryde, C., and Lightsey, E. A star tracker design for cubesats. In *Aerospace Conference, 2012 IEEE* (March 2012), pp. 1–14.

[30] McCollough, E. *"Photographic Topography" Industry: A Monthly Magazine Devoted to Science, Engineering and Mechanic Arts.* Industrial Publishing Company, San Francisco, 1893.

[31] Mortari, D., and Neta, B. k-vector range searching techniques. In *AAS 00-128,AAS/AIAA Space Flight Mechanics Meeting* (January 2000).

[32] Na, M., and Jia, P. A survey of all-sky autonomous star identification algorithms. In *Systems and Control in Aerospace and Astronautics, 2006. ISSCAA 2006. 1st International Symposium on* (Jan 2006), pp. 6 pp.–901.

[33] Padgett, C., and Kreutz-Delgado, K. A grid algorithm for autonomous star identification. *Aerospace and Electronic Systems, IEEE Transactions on 33*, 1 (Jan 1997), 202–213.

[34] pandaboard.org. *Pandaboard Wiki.* Available at http://www.omappedia.com/wiki/PandaBoard Accessed 11-February-2014.

[35] PHOTOMETRICS. *CoolSNAT cf2 Monochrome Datasheet*, 2010. Available at http://www.photometrics.com/products/datasheets/cf2.pdf Accessed 5-February-2014.

[36] Raspberry PI Foundation UK Registered Charity 1129409, 2014. Available at http://www.raspberrypi.org/ Accessed 7-February-2014.

[37] REXUS/BEXUS. Rocket & ballon experiments for university students, 2013. http://www.rexusbexus.net/ Accessed 23-June-2014.

[38] RICOH, 2014. Available at http://www.ricoh.com/ Accessed 7-December-2014.

[39] RS COMPONENTS. *Raspberry Pi Model B.* Available at http://www.rs-components.com/raspberrypi Accessed 23-September-2014.

[40] Saft Specialty Battery Group. *LSH20 Primary Lithium Battery.* Doc. Nº31015-2-1006, Available atwww.saftbatteries.com/force_download/LSH20.pdf Accessed 4-July-2014.

[41] Samaan, M., and Theil, S. Development of a low cost star tracker for the {SHEFEX} mission. *Aerospace Science and Technology 23*, 1 (2012), 469 – 478. 35th ERF: Progress in Rotorcraft Research.

[42] SELEX ES. *A-STR & AA-STR Autonomous Star Trackers.* Available at www.selex-es.com Accessed 7-May-2014.

[43] Shuster, M., and OH, S. Three-axis attitude determination from vector observations. *Journal of Guidance and Control 4*, 1 (1981), 70–77.

[44] SODERN. *HYDRA-M CMOS Star Tracker.* Available at http://www.sodern.com Accessed 7-May-2014.

[45] Sony. *Diagonal 4.5mm (Type 1/4) Progressive Scan CCD Image Sensor with Square Pixel for B/W Cameras*, 2005. Available at http://s1.dl.theimagingsource.com/api/2.0/packages/publications/sensors-ccd/icx618ala/d4203d08b4/icx618ala_1.2.en_US.pdf Accessed 7-February-2014.

[46] Sony. *ICX205AL.Diagonal 8mm (Type 1/2) Progressive Scan CCD Image Sensor with Square Pixel B/W Cameras*, 2005. Available at http://www.ccd.com/pdf/ccd_205.pdf 7-February-2014.

[47] STMicroelectronics. *LSM303DLHC:Ultra-compact high-performance eCompass module:3D accelerometer and 3D magnetometer*, November 2013. Available at http://www.st.com/web/en/resource/technical/document/datasheet/DM00027543.pdf Accessed 15-May-2014.

[48] Team, G. *GranaSAT Student Experiment Documentation.* University of Granada, September 2014. v4-1.

[49] TERMA Space. *Star Tracker HE-5AS.* Available at www.sodern.com Accessed 7-May-2014.

[50] The Imaging Source, 2014. Available at http://www.theimagingsource.com/ Accessed 7-December-2014.

[51] The MathWorks, I. Matlab 2013b.

[52] TIGAL KG. Available at http:www.tigal.com Accessed 8-January-2014.

[53] The University of Wisconsin-Madison Space Astronomy Laboratory. *ST5000*. Available at http://www.sal.wisc.edu/st5000/intro/st5000_final2007.pdf Accessed 7-May-2014.

[54] USNO. Naval observatory vector astrometry software (novas) version 3.1 fortran, c, and python editions, 2013. Available at http://aa.usno.navy.mil/software/novas/ Accessed 20-October-2014.

[55] van Bezooijen, R. W. H. True-sky demonstration of an autonomous star tracker. In *Acquisition, Tracking, and Pointing VIII* (1994), vol. 2221, pp. 156–168.

[56] Wahba, G. Problem 65-1: A least squares estimate of spacecraft attitude. *SIAM Review 7(3)* (1965), 409.

[57] Wu, A., and Hein, D. Stellar inertial attitude determination for leo spacecraft. In *Decision and Control, 1996., Proceedings of the 35th IEEE Conference on* (Dec 1996), vol. 3, pp. 3236–3244 vol.3.

[58] Xie, J., Tang, X., Jiang, W., and Fu, X. An autonomous star identification algorithm based on the directed circularity pattern. *star 4* (2012), 5.

[59] Zhang, Z. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 22*, 11 (Nov 2000), 1330–1334.