This Master's Thesis addresses the design of a multidisciplinary 1U CubeSat Simulation Platform, composed of three blocks: a mechanical simulation platform, a Ground Station management software and a CubeSat prototype that includes different subsystems: OBC, ADCS and part of the OBDH and the EPS. This prototype is the base for the future GranaSAT-I.

The project is approached from a double perspective: on the one hand, developing an academic Simulation Platform that gets students closer to CubeSats; on the other hand, providing with an integrated environment for researchers to test new technologies and algorithms before launching.

This wide scope requires applying professional System Engineering methodologies, which minimizes the risk and culminates with the successful completion of the project.

José Carlos Martínez Durillo is a Telecommunication engineer from Bailén, Spain. He was awarded twice for his Bachelor's Thesis, including the prize of the *Colegio Oficial de Ingenieros Técnicos de Telecomunicación* to the best Thesis. With this ambitious Master's Thesis he starts a fascinating new research line in the GranaSAT Project and finalizes his MEng.

Andrés María Roldán Aranda is the academic head of the present project, and the student's tutor. He is a professor in the Departament of Electronics and Computers Technologies.

MASTER'S THESIS

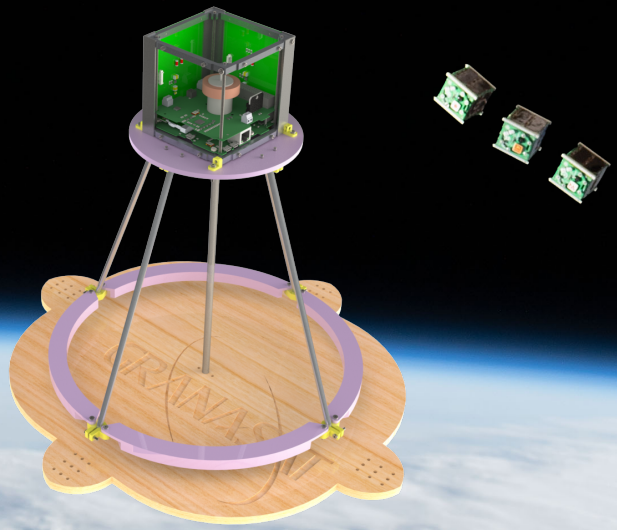
Design of a multidisciplinary 1U CubeSat Simulation Platform

José Carlos Martínez Durillo

TELECOMMUNICATION ENGINEERING

UNIVERSITY OF GRANADA

Master in Telecommunication Engineering



Design of a multidisciplinary 1U CubeSat Simulation Platform

José Carlos Martínez Durillo

2018/2019

Tutor: Andrés María Roldán Aranda

Credits for the cover: **NASA**.
Printed in Granada, September 2019.

“Design of a multidisciplinary 1U CubeSat Simulation Platform”

MASTER IN
TELECOMMUNICATION ENGINEERING

Master's Thesis

"Design of a multidisciplinary 1U CubeSat Simulation Platform"

ACADEMIC COURSE: 2018/2019

José Carlos Martínez Durillo

MASTER IN TELECOMMUNICATION ENGINEERING

"Design of a multidisciplinary 1U CubeSat Simulation Platform"

AUTHOR:

José Carlos Martínez Durillo

SUPERVISED BY:

Andrés María Roldán Aranda

DEPARTMENT:

Electronics and Computers Technologies

D. Andrés María Roldán Aranda, Profesor del departamento de Electrónica y Tecnología de los Computadores de la Universidad de Granada, como director del Trabajo Fin de Máster de D. José Carlos Martínez Durillo,

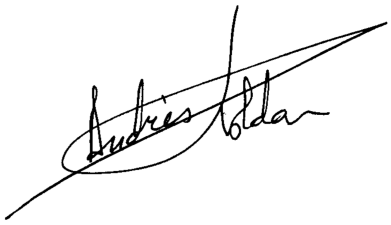
Informa:

Que el presente trabajo, titulado:

"Design of a multidisciplinary 1U CubeSat Simulation Platform"

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

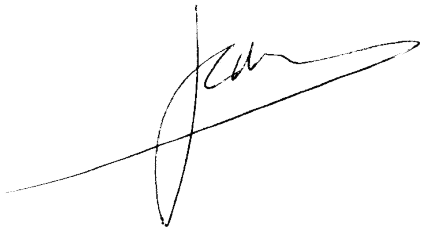
Granada, a 2 de Septiembre de 2019

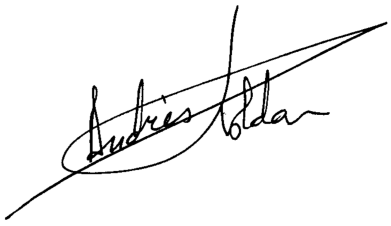


Fdo. Andrés María Roldán Aranda

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

Granada, a 2 de Septiembre de 2019





Fdo. José Carlos Martínez Durillo

Fdo. Andrés María Roldán Aranda

Design of a multidisciplinary 1U CubeSat Simulation Platform

José Carlos Martínez Durillo

KEYWORDS:

CubeSat, Altium Designer® 19, Aerospace design, SolidWorks®, Ground Station, EDA, Electronic, OBC, ADCS, EPS, OBDH, PCB Design, MATLAB®.

ABSTRACT:

The main purpose of this project is developing a multidisciplinary Simulation Platform for **CubeSats**. It will be composed of three differentiated blocks, around which the project is structured: a mechanical simulation platform, a Ground Station management software and a CubeSat prototype that will be the base for the future **GranaSAT-I**.

This Master's Thesis is addressed from a double perspective: on the one hand, the development of a Simulation Platform of great usefulness in an academic environment, as a way to get students from multiple degrees closer to the aerospace world, and particularly to CubeSats, given its current context of peak, being fostered by institutions such as **European Space Agency** (ESA); on the other hand, in a research environment, providing with a mean to implement new communication algorithms, orbit controllers, and generally speaking, for the development and test of new technologies and techniques, before launching.

The development and implementation of this project is performed following methodologies of System Engineering contrasted in the aerospace industry, giving realism and getting the student closer to professional techniques, widely recognized in the job market. Furthermore, the complexity and multidisciplinary scope of this Master's Thesis allows covering not only the different specialties of the Master in **Telecommunication** Engineering but also acquiring knowledge and transversal abilities from other fields of the Engineering, such as **Mechanical** or **Aerospace**. Besides specific software of each of the mentioned areas, advanced techniques of **machining** (aluminum milling), **manufacturing** (solder reflow) or **characterization** of different devices (lithium batteries, silicon solar cells...) among others, have been analyzed and applied.

The result of the exposed culminates with the obtention of a complete and functional simulation environment, which complies with the requirements defined in the preliminary stages, and supposes the finalization of the Master.

Design of a multidisciplinary 1U CubeSat Simulation Platform

José Carlos Martínez Durillo

PALABRAS CLAVE:

CubeSat, Altium Designer® 19, Diseño aeroespacial, SolidWorks®, Ground Station, EDA, Electrónica, OBC, ADCS, EPS, OBDH, Diseño de PCB, MATLAB®.

RESUMEN:

El objetivo principal del presente proyecto es desarrollar una Plataforma de Simulación multidisciplinar de **CubeSats**. Estará compuesta de tres bloques diferenciados, en torno a los cuales pivotará el proyecto: una plataforma de simulación mecánica, un software de gestión de Ground Station y un prototipo de CubeSat, que constituirá la base del futuro **GranaSAT-I**.

Este Trabajo Fin de Máster se aborda desde una ambiciosa doble perspectiva: por un lado, el desarrollo de una Plataforma de Simulación de amplia utilidad en el ámbito académico, como medio para el acercamiento del alumnado de múltiples titulaciones al mundo aeroespacial y en concreto a los CubeSats, en el contexto de auge actual, fomentado por instituciones como la **Agencia Espacial Europea** (ESA); en segundo lugar, en el ámbito de investigación, proveyendo de un medio para la implementación de nuevos algoritmos de comunicación, de control orbital y, en general, para el desarrollo y testeo de tecnologías y técnicas novedosas, de manera previa a su lanzamiento.

El desarrollo e implementación de este proyecto se lleva a cabo siguiendo metodologías de **Ingeniería de Sistemas** contrastadas y asentadas en la industria espacial, dotándolo de realismo y acercando al alumno a técnicas profesionales de amplio reconocimiento en el mercado de trabajo. Asimismo, la complejidad y ámbito multidisciplinar de este Trabajo Fin de Máster le permite cubrir, no sólo las diferentes especialidades del Máster de Ingeniería de **Telecomunicación**, sino también adquirir conocimientos y habilidades transversales o específicos de otros campos de la Ingeniería, como la **Mecánica** o la **Aeroespacial**. Así, además de software especialista de cada uno de los campos mencionados, se han analizado y aplicado técnicas avanzadas de **mecanizado** (fresado de aluminio mediante control numérico), **fabricación** (soldadura utilizando técnicas de *reflow*) o **caracterización** de diferentes dispositivos (baterías de litio, células solares de silicio...), entre otros.

El resultado de todo lo expuesto culmina con la obtención de un entorno de simulación completo y funcional, que cumple con los requisitos definidos en etapas iniciales, y con el cual se cierra la etapa universitaria de Máster.

'Tough and competent'

1969 — 2019
50th anniversary of the first human landing on the moon

Science-fiction yesterday, fact today, obsolete tomorrow
Otto O. Binder

Acknowledgments:

This work has been possible thanks to a very small amount of people, but to whom I owe a great acknowledgment; although the one I can show you in these lines will undoubtedly be insufficient, serve this brief space as such.

The first of them must be for my family, my parents, José and Amparo, and my brother Francisco Javier. Ever since I can remember, their unconditional support in every aspect has been of primary importance for every of my undertakings to come to fruition. The support provided during this project has been only another proof of this.

To Luis and Alberto, for the technical support, the laughs, endless coffees and warnings to stop doing what you are doing. To Pablo, for the countless hours of advice, for the vision where mine got clouded, for the important things in life, for the wine and the debate near a bar.

This Master's Thesis would not have been born without the persistence of my tutor, Andrés Roldán Aranda, in the face of my initial reticence. His ambition has got me closer to numerous projects throughout all these years, and this work unbeatably closes this unforgettable stage.

And of course, to Natalia, for being with me during the course of this long stage, and keeping supporting me at the worst moments, for the understanding and the patience, thank you very much.

Everyone of the mentioned is part of this Thesis. To all of them, and to those who could not see it, thank you.

Agradecimientos:

Este trabajo ha sido posible gracias a un número muy reducido de personas, pero a las cuales debo un gran agradecimiento; aunque el que pueda mostrarle en estas líneas será sin duda insuficiente, sirva este breve espacio como tal.

El primero de ellos ha de ir destinado a mi familia, mis padres, José y Amparo, y mi hermano Francisco Javier. Desde siempre, su apoyo incondicional en todos los aspectos ha resultado de importancia capital para que toda empresa acometida llegara a buen puerto. El apoyo mostrado durante el periodo de desarrollo de este proyecto no ha sido sino una muestra más del mismo.

A Luis y Alberto por el apoyo técnico, las risas, los cafés interminables y los avisos para dejar lo que estás haciendo. A Pablo, por las incontables horas de asesoramiento y visión allí donde la mía se nublaba, por las cosas importantes de la vida, por el vino y el debate cerca de una barra.

Este Trabajo Fin de Máster no habría nacido sin la persistencia de mi tutor, Andrés Roldán Aranda, frente a mi reticencia inicial. Su ambición me ha acercado a numerosos proyectos a lo largo de todos estos años, y este trabajo cierra de manera inmejorable esta inolvidable etapa.

Y por supuesto, a Natalia, por acompañarme durante el transcurso de esta larga etapa, y seguir aún sirviendo de apoyo en los peores momentos, por la comprensión y la paciencia, muchas gracias.

Cada uno de los mencionados son partícipes de este Trabajo. A todos ellos, y a los que no pudieron verlo, muchas gracias.

Index

List of Figures

List of Videos

List of Tables

Glossary

Altium Designer® 19 EDA software used to design PCB from schematics. It allows 3D Design, as well as electronics simulation.

Blackbody Idealized physical body which absorbs all the electromagnetic radiation which falls upon it, regardless of angle of incidence or frequency [59].

CAD Uso de computadores en el proceso de diseño y documentación de un bien o servicio.

CubeSat Miniaturized satellite normally for space research, with dimensions of 1 dm^3 and mass lower than 1.33 kg per unit.

Eigenvector In linear algebra, non-zero vector of a linear transformation that changes by only a scalar factor when the linear transformation is applied to it [59].

Engineering model Representation of a real object or system in order to ease physical visualization or behavior.

GranaSAT GranaSAT is an academic project from the University of Granada originally consisting in designing and developing a picosatellite (CubeSat). Coordinated by the Professor Andrés María Roldán Aranda, GranaSAT is a multidisciplinary project with students from different degrees, where they can acquire and enlarge the knowledge necessary to face an actual aerospace project.

Ground Station Facilities in which instruments and devices necessary to establish a radio - link communication are normally located. Also used to control and monitor antenna system.

Housekeeping Telemetry data associated to the health state of the instruments and devices of the spacecraft, in contrast the so-called **space telemetry** or simply telemetry, which gathers the real observations.

Jitter Deviation from presumably periodicity in relation to a reference clock signal.

Magnetic moment Magnetic strength and orientation of a magnet or other object which produces a magnetic field [59].

Magnetorquers Satellite system used for the attitude control by generating a certain magnetic moment using electromagnetic coils, whether in winding or printed format, when an alternating current is applied to them.

Memory effect Associated to NiCd and NiMH batteries, it results in a lose of the maximum capacity when recharged unnecessarily.

Outgassing Release of a gas previously contained in some material. In space, it is undesirable as it can condense onto optimal elements or solar cells, so low-outgassing materials are recommended.

Payload Carrying capacity of an aircraft, normally measured in terms of weight. More specifically, it can be referred to the equipment carried for the performance of a certain mission, e.g. an camera or an star tracker.

Relative Permittivity Regarding a material, said of its permittivity expressed as a ratio relative to permittivity of vacuum.

Seebeck effect Also called thermoelectric effect, it is the conversion of temperature differences to electric voltages and vice versa using a thermocouple.

Single Page Application Web site which, once downloaded, allows dynamic interaction with the user by rewriting the page instead of requesting the whole page to the server each time.

SolidWorks® CAD Software from Dessault Systèmes for 3D Mechanical Design.

Stakeholders According to ISO21500, person, group or organization that has interests in, or can affect, be affected by, or perceive itself to be affected by, any aspect of the project.

Star tracker In spacecrafts, optical device which calculates the positions of stars using a camera, normally to estimate its attitude with respect to the stars.

Stencil Stainless steel foil in which there are several openings for every SMD on the board. Once it is aligned with the PCB, it is used to transfer the solder paste to it.

Torque Rotational equivalent of linear force.

Tumbling Normally referred to the situation after deployment, when the CubeSat is spinning uncontrolled, in an unusable situation until the angular rate is reduced.

Websockets Full-duplex communications protocol which allows multiple channles over a single TCP connection (ports 80 or 443). Although it is not mandatory, it is usually used along with HTTP. It enables interaction with a web site (typically a SPA) with low overhead, easing real-time data transfer in both ways [59].

Acronyms

AC Altern Current.

ADC Analog to Digital Converter.

ADCS Attitude Determination & Control System.

AM Air Mass.

AOCS Attitude & Orbit Control System.

BOL Beginning Of Life.

CCSDS Consultative Committee for Space Data Systems.

COB Chip On Board.

COMMS Communications.

COTS Commercial Of The Shelf.

CPU Central Processing Unit.

DC Direct Current.

DET Direct Energy Transfer.

DoD Depth of discharge.

DUT Device Under Test.

ECSS European Cooperation for Space Standardization.

EDA Electronic Design Automation.

EDP Engineering Design Process.

EMC ElectroMagnetic Compatibility.

EMF Electromotive Force.

EMI Electromagnetic Interference.

EOL End Of Life.

EPS Electrical Power System.

ESA European Space Agency.

FPGA Field-Programmable Gate Arrays.

GPIO General Purpose Input Output.

HCI Host Controller Interface.

HID Human Interface Device.

HTTP HyperText Tranfer Protocol.

I2C Inter-Integrated Circuit.

I2DOS Inertial 2D Orbit Simulator.

IC Integrated Circuit.

IEEE Institute of Electrical and Electronics Engineers.

IMU Inertial Measurement Unit.

ISO International Organization for Standardization.

ISS International Space Station.

JAXA Japan Aerospace Exploration Agency.

LAN Local Area Network.

LDR Light Dependent Resistor.

LED Light-Emitting Diode.

LUT Lookup Table.

MATLAB® MATrix LABoratory.

MPPT Maximum Power Point Tracking.

NASA National Aeronautics and Space Administration.

OBC On-board Computer.

OBDH On-board Data Handling.

OS Operative System.

P-POD Poly-Picosatellite Orbital Deployer.

PCB Printed Circuit Board.

PID Proportional Integral Derivative.

PLA Polylactic Acid.

PWM Pulse Width Modulation.

RF Radio Frequency.

RMS Root Mean Squared.

RTC Real Time Clock.

RTG Radio-isotope Thermoelectric Generators.

RTOS Real-Time Operative System.

SBC Single-Board Computer.

SD Secure Digital.

SMD Surface Mount Devices.

SNR Signal to noise ratio.

SoC System-on-Chip.

SoC State of Charge.

SPA Single Page Application.

SPI Serial Peripheral Interface.

SSH Secure SHell.

STL Standard Triangle Library.

TC&C Telecommand & Telemetry System.

TCP Transmission Control Protocol.

THS Thermal Subsystem.

TVAC Thermal VAcuum Chamber.

UART Universal Asynchronous Receiver-Transmitter.

UDP User Datagram Protocol.

USB Universal Serial Bus.

VHDL VHSIC Hardware Description Language.

XTCE XML Telemetric and Command Exchange.

Chapter 1

Introduction

This Master's Thesis is presented as a compilation of the knowledge acquired throughout the years of the bachelor's and master's degree and specially, during this project period. It aims to reflect the engineering process behind the design, development, prototyping and verifying stage of a product. The overall goal of the project is developing an integrated 1U CubeSat Simulation Platform, to be used as a testing and training environment.

This ambitious project has been performed seeking a **double perspective** which will be detailed later, applying professional standards and methodologies used in space sector, in order to successfully overcome the project, minimizing risks and costs. It has supposed not only a hard design process but also a great labour of integration in order to maximize the use of existing equipment and procedures, available at GranaSAT laboratory. The final result is a fully-integrated simulation environment useful in a variety of situations: algorithms implementation, physical measurement or hardware characterization, among many others; its flexibility and expansion possibilities are countless. To make it possible, a conjunction of different fields of knowledge has been necessary; not only related to Telecommunication Engineering but also with completely different fields such as Mechanical or Aerospace Engineering. In addition to an academically and experimentally interesting platform, this conjunction has allowed developing a potentially marketable product, implying additional considerations to be taken into account.

In order to keep an structure, the project has been developed considering the different parts of the platform, i.e., **Mechanical Platform, Ground Station and Simulation CubeSat**. This document also follows that division, which eases understanding and allows a natural progress to the reader.

There is a generalized consensus regarding the best practices and methodologies

1

around the design process in engineering; this project will follow one of the most extended methodologies in space sector, proposed by Wertz and Everett [94], simplified when needed. As usual, it is characterized by an iterative refinement of the requirements which leads to the design and finally the verification of the objectives.

This Bachelor's Thesis fits within GranaSAT Project, a multidisciplinary project which gathers people from a variety of fields who are committed to acquiring new knowledge related to Electronics and Aerospace Engineering. Since its origins, one of its main purposes has been getting a CubeSat in orbit, one of the reasons of this project; however, today its goals goes far beyond, and a wide range of devices and projects are developed in collaboration with different students and enterprises.

Figure 1.1 – *GranaSAT Logo*

1.1 Motivation

After investing almost three years working on GranaSAT Project, dealing with hundred of problems and a variety of technologies, as well as being aware of its participation in Bexus Project [19] in 2014, I had an overall knowledge of the CubeSat concept, main subsystems and typical objectives. However, after being selected to attend CubeSats Workshop organized by **European Space Agency ESA** in Belgium, I realized that I needed to go in depth in the study of these fascinating satellites. Figure 1.2 shows a photo of that great experience.

Attending the aforementioned workshop made me realize the usefulness of using a simulated environment to acquire new concepts and increase technical knowledge, which is particularly interesting in an academic atmosphere such as the University. So my first motivation to develop this Master's Thesis was to provide the GranaSAT laboratory with a multidisciplinary Simulation Platform for CubeSats, taking the advantage of their little size, in contrast with bigger satellites, much more complex to test in a non-professional environment; as it is widely known, after Bexus Project (see [19]) GranaSAT Group has kept working on real developments and missions in collaboration with ESA, particularly around CubeSats, with the first CubeSat of the

Figure 1.2 – *At ESA CubeSats Workshop in Redu, Belgium (2018)*

University of Granada, **GranaSAT-I** intending to participate in the ESA programme, ***Fly Your Satellite!*** [13], introduced in Subsection 1.2.2.

For all these reasons, I decided to focus my Master's Thesis on a **double perspective**: on the one hand, a multidisciplinary academic-oriented **Simulation Platform** for CubeSats, which allows testing its different subsystems as well as implementing innovative solutions in a physical scenario; on the other hand, a complete project-oriented platform to be used by future GranaSAT members on its way to the *Fly Your Satellite!* programme [13], including a CubeSat Engineering model with different subsystems to be tested and fully integrable with the countless technology modules and procedures which GranaSAT Project has been provided with throughout the years, as result of the work of voluntary students and researchers; precisely, some of them are used in this Master's Thesis, so interoperability and cooperation with the rest of the technical staff of the group have also been important incentives to me; this combination of existing technologies and equipment is intrinsic to complex project and aerospace design in particular: a large group of professionals working towards a same goal throughout time, in this case, **flying a CubeSat**. Figure 1.3 depicts these perspectives.

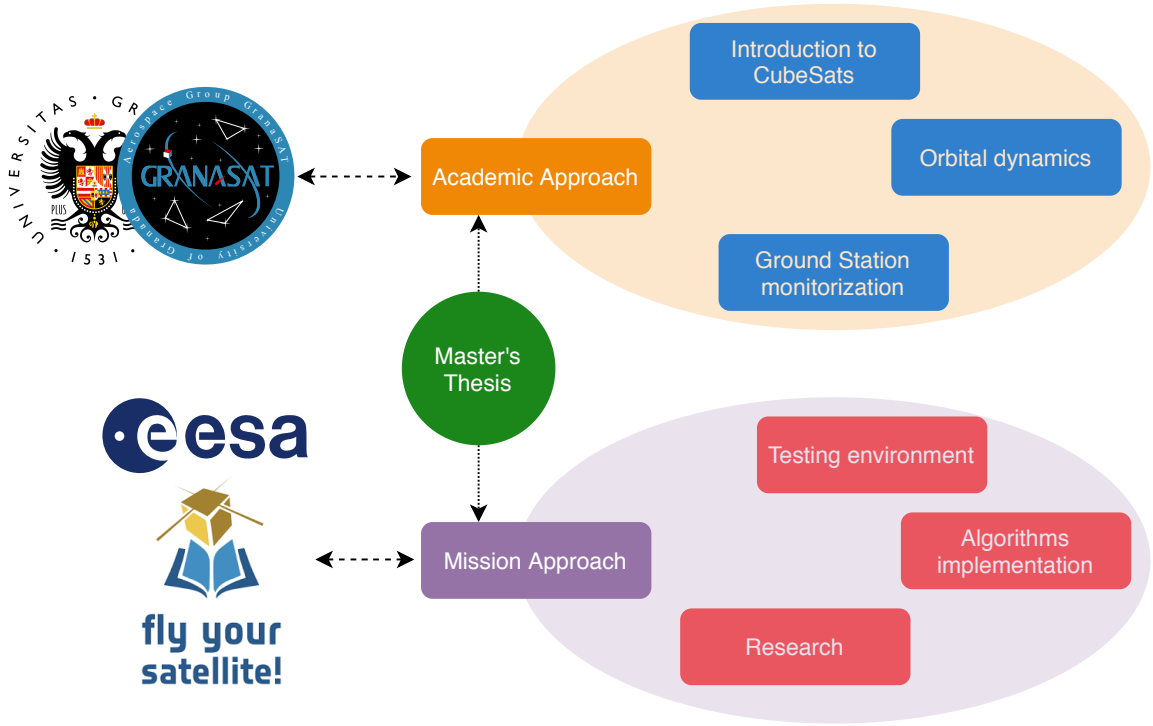


Figure 1.3 – *Double orientation of the project*

This double orientation of the project will entail innumerable mentions through this document, as some decisions will have to be taken in order to comply with both perspectives. Generally speaking, my intention has been keeping an adequate balance which allows the Simulation Platform to **satisfy both testing and academic needs.**

After an arduous Bachelor's Thesis [71] **awarded twice** by the *Colegio Oficial De Ingenieros Técnicos De Telecomunicación* [6] and *Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación* at the University of Granada [14], this project has supposed a definitely larger and exciting challenge, substantially increasing my professional experience and knowledge in aerospace field by dealing with space standards, learning different systems architecture and putting into practice numerously experimental techniques.

The Simulation Platform which emerges from this Master's Thesis is the result of a painstaking process of engineering, study and comparison of the cutting-edge technology used in space which I sincerely hope that can be useful for the GranaSAT Project in the future.

1.2 Prior art. Problem Statement

1.2.1 Background study: CubeSats

CubeSat emerged as a result of increasing costs in space field, incredible amounts of money had to be spent in missions with a considerable risk. They allowed to trade some functionalities off in exchange for a significant reduction in costs and therefore a greater number of missions; higher risk of failure is accepted because of the possibility to perform more and more frequent launches, as illustrated in Figure 1.4.

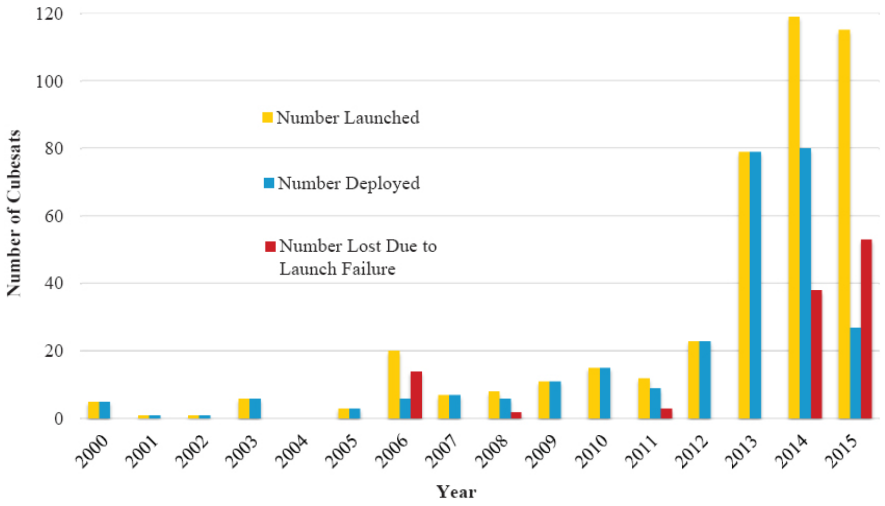
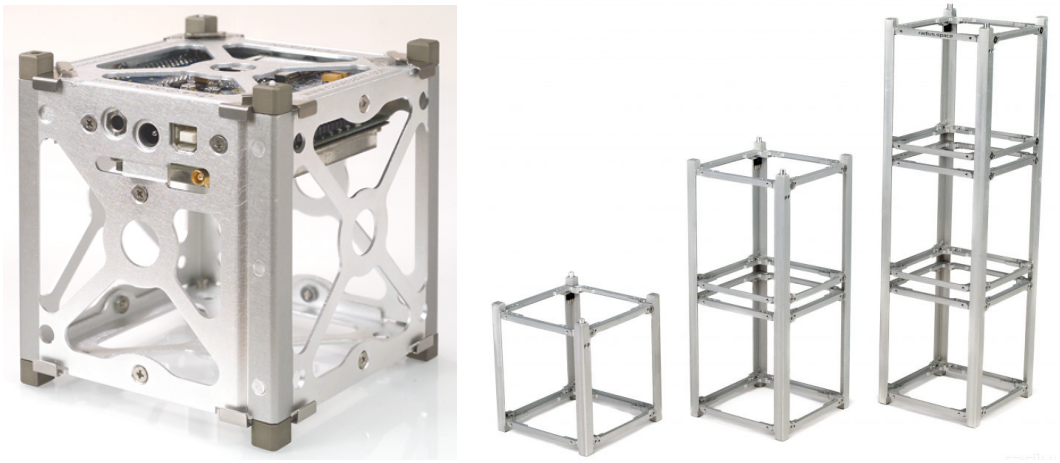


Figure 1.4 – *Number of CubeSats launched over the last years [60]*

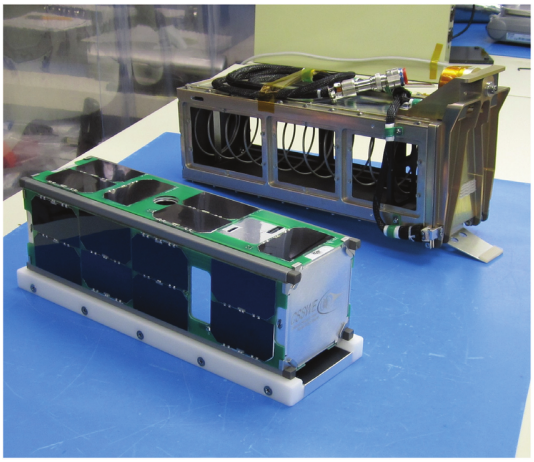
As for the academic scope, CubeSat allowed people with little or no experience in the design of space missions to incorporate new ideas and concepts into missions with fewer restrictions. Within university projects, the challenge is not only accomplishing a successful mission but also get it at the lowest cost. All of which, of course, supposes the risk to be inherently high, but it is not a problem itself.

CubeSat concept were first developed at Stanford and California Polytechnic universities in 1998. It was precisely an Spanish engineer, **Jordi Puig-Suari** along with Robert Twiggs, who created them. The primary intent of the development of the CubeSat was to provide a standard set of dimensions for external physical structure, in order to make it compatible with a standardized orbital deployer. The final concept of the CubeSat structural standard is shown in Figure 1.7. It is based on a **10 x 10 x 10 cm cube and a mass of 1.33 kg** maximum (see Figure 1.5a) and is referred to as a one unit '1U' and is scalable in 1U increments, as shown in Figure 1.5b. This standardization facilitates a wide variety of designs to be launched and deployed using a common system; the most used is Poly-Picosatellite Orbital Deployer (P-POD), having launched all CubeSats since 2007. Figure 1.6a shows a typical P-POD launcher while Figure 1.6b depicts three 1U CubeSats being launched.



(a) *1U standard* **(b)** *1U, 2U and 3U standards comparison*

Figure 1.5 – *CubeSat physical structure [33]*



(a) *3U capacity example [97]*

(b) *CubeSat launching [33]*

Figure 1.6 – *P-POD Launcher*

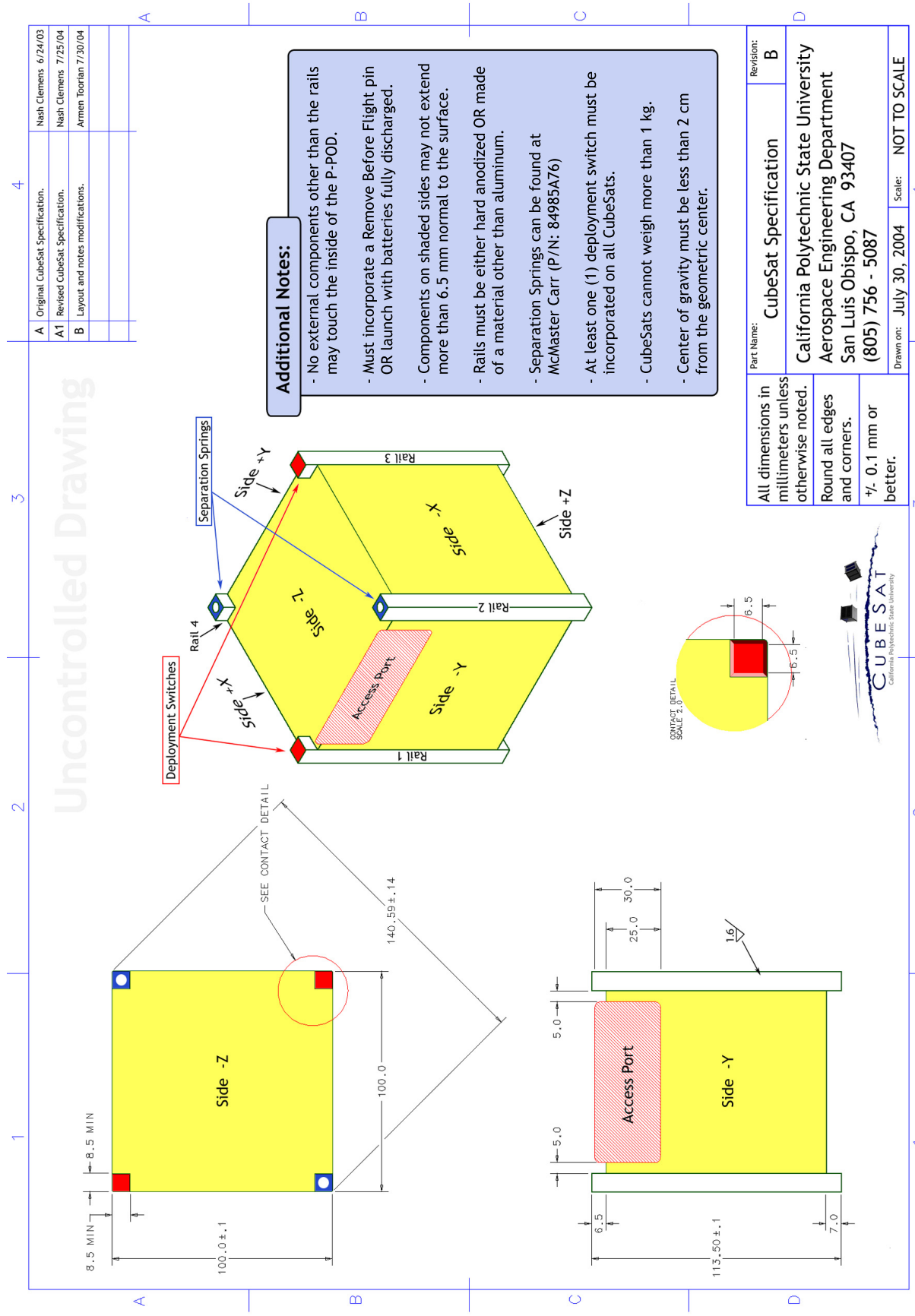


Figure 1.7 – *Standard for CubeSat Mechanical Structure [5]*

1

CubeSats are typically composed of different subsystems, which are more or less standard in literature. This project will follow the division used by ESA, depicted in Figure 1.8; besides, the most relevant components and functions of each subsystem are also included. These subsystems are briefly analyzed next.

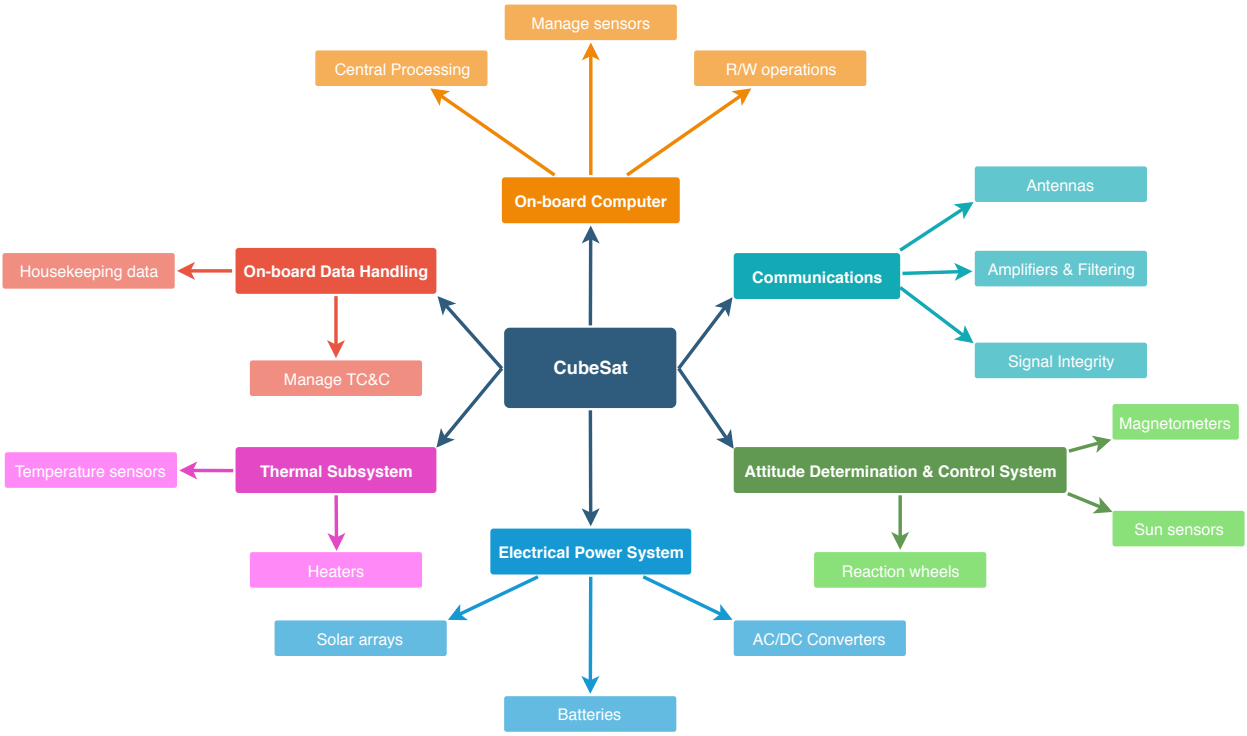


Figure 1.8 – *CubeSat typical subsystems*

- **On-board Computer (OBC):** the main subsystem of a CubeSat, it is in charge of the core processing, reading and polling sensors, R/W operations on memory, data management, etc. It is typically based on a central microprocessor or computing core.

- **Communications (COMMS):** in charge of sending and receiving all the information related to a mission. It is sometimes taken as the most critical subsystem, given that **without contact the mission is over,** even if the rest of the system functions flawlessly. It is composed of the transmission and receiving antennas, amplifiers, RF electronics, filtering, signal couplers, among many others elements.

- **Attitude Determination & Control System (ADCS):** its main purpose is assuring the correct orientation of the CubeSat and dealing with external torques. Composed of sensors and actuators such as Magnetorquers, it usually implements an independent processing unit. As for its software, it can be based on different

control laws and predictive filters such as **Kalman filter**.

- **Electrical Power System (EPS):** this module deals with the generation, distribution and control of the spacecraft electrical power. It must supply a continuous source of power in adequate condition for each element of the system, and deal with possible faulty situations. It is composed of a long chain of elements, from solar arrays to voltage regulators or batteries.

- **Thermal Subsystem (THS):** in charge of setting and maintaining the temperature range of the CubeSat and its components, which are all designed to operate over a defined temperature range. This is accomplished by balancing the flow of heat energy. Generally, in CubeSats it is not required an active thermal control system.

- **On-Board Data Handling (OBDH):** it handles all data sent and received by the CubeSat, including TC&C. Although it is typically implemented on the OBC and connected to the COMMS subsystem, because of its increasing importance during the last few years, it is nowadays considered a proper subsystem by itself.

Additionally, sometimes the **payload** is taken as another subsystem of the CubeSat. It is actually the most important part of it given that it constitutes the reason of the mission itself: **the payload provides the expected results and measurements**. There is a huge variety of payloads, with equal amount of purposes; cameras, sensors, radiometers... the list is endless. In sum, anything able to provide the team with some kind of result or just even functioning correctly is liable to be a payload.

This project will focus on OBC, ADCS and in the power source area of the EPS. Also, some basic functions related to OBDH will be implemented. Section 3.4 deeply analyzes these subsystems.

1.2.2 ESA *Fly Your Satellite!*

Because of the increasing interest within university students in CubeSats, ESA decided to boost this initiative in 2013 by organizing the first *Fly Your Satellite!* program in collaboration with European Universities, with the goal to be a complementary preparation for the future professionals of the space sector.

During the program, the accepted student teams are supported in the development of their CubeSat and their previously defined mission. ESA experts support and mentor the teams through the different phases, focusing on assembly, integration, verification, testing and operations. At a point, the students are allowed to use the Agency test facilities and take part in training sessions. This priceless opportunity allows the students to gain great experience in the lifecycle of an actual space project,

using working methods and standards from professional space programmes which significantly help to build their own professional skills [17].

ESA *Fly Your Satellite!* programme is divided into different phases with pre-defined objectives and reviews conducted by ESA specialists. Gaining access to the next phase requires to successfully complete the previous one. These phases are depicted in the flow diagram of Figure 1.9.

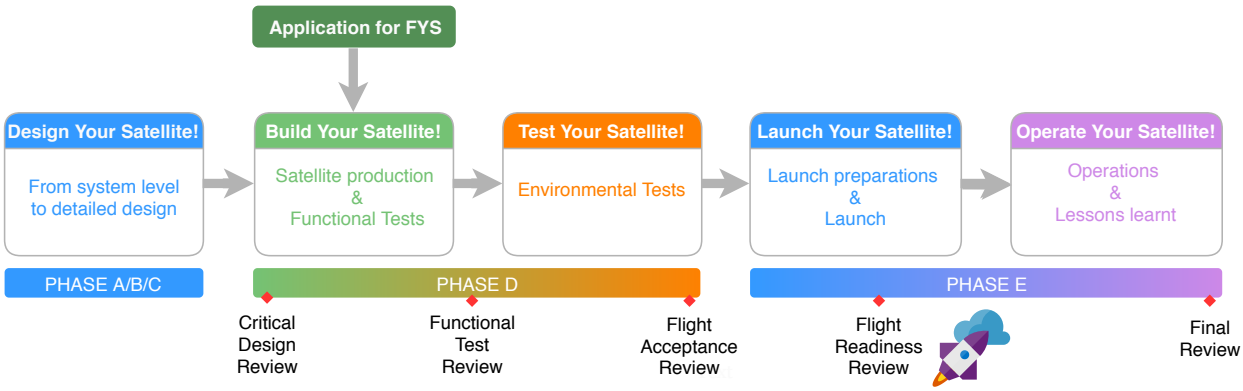


Figure 1.9 – *Fly Your Satellite! phases*

As illustrated in the diagram, application to the program is constrained to those teams which already have a complete detailed design. As previously mentioned, one of the main goals of the GranaSAT Project is successfully participating in this program. This Master's Thesis lays the foundations for a future CubeSat design and application to the *Fly Your Satellite!*.

1.2.3 European Cooperation for Space Standardization

The European Cooperation for Space Standardization (ECSS) is an initiative established to develop a coherent, single set of user-friendly standards for use in all European space activities. It started officially in the autumn of 1993, when the European space community realized that differences in space standards resulted in higher costs, lower effectiveness and a less competitive industry.

Figure 1.10 – *ECSS logo [82]*

Numerous ECSS standards have been released with the aim to be applied altogether for the management, engineering and product assurance. Standards and recommendations for the development of the space technology are split up in four disciplines: *Space project management branch (M)*, Space product assurance branch (Q), Space engineering branch (E), and Space sustainability branch (U). Figure 1.11 shows the different ECSS disciplines [73].

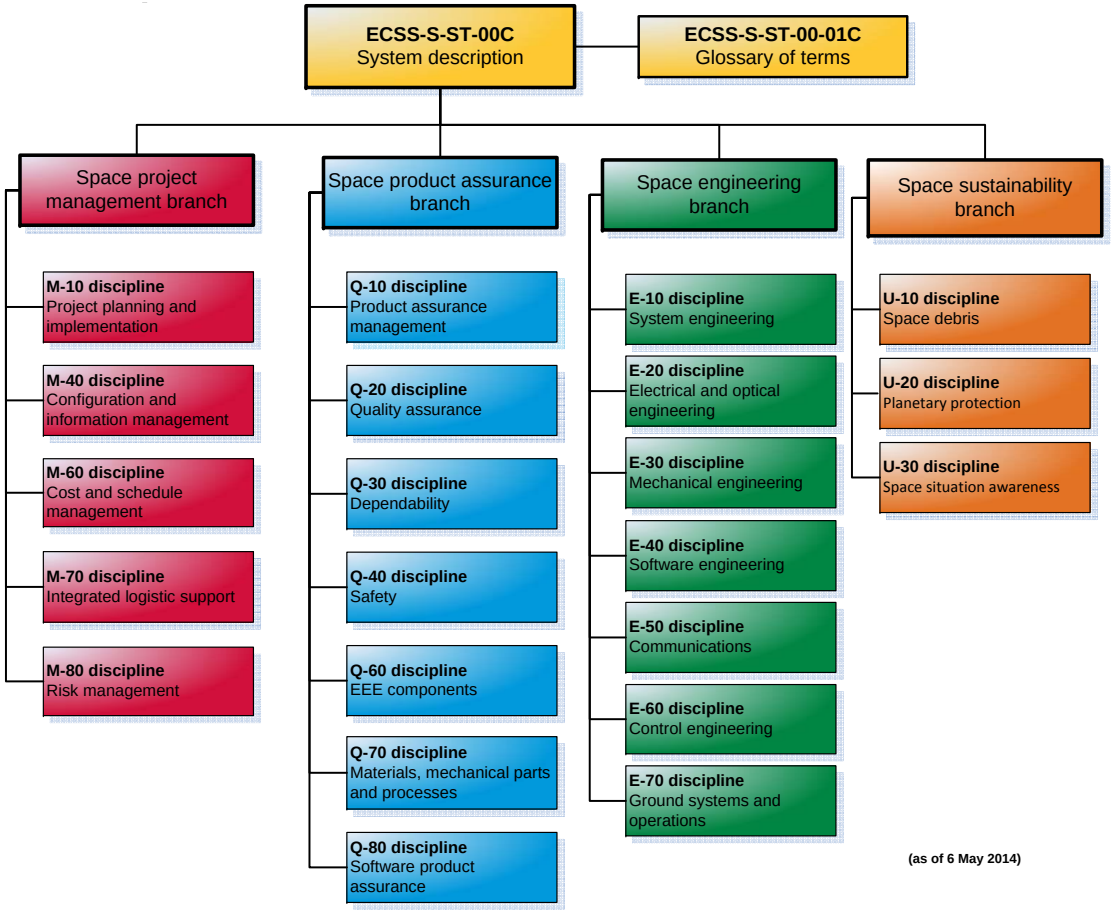


Figure 1.11 – *ECSS Document Tree [12]*

As shown in Figure 1.12, the documentation is organized into four branches, each one defined to the detail level required to differentiate major functions, disciplines and activities. Besides, four hierarchical levels are considered:

- **Level 0 (ECSS-P-00)** – The document at Level 0 describes the policy and objectives of the ECSS system and its architecture as well as the principle rule for the creation, validation and maintenance of documents.
- **Level 1 (ECSS-M-00, ECSS-Q-00, ECSS-E-00, ECSS-U-00)** -The documents at Level 1 describe the strategy in the specific domain. They give a global view of the

requirements and outline the interfaces between the elements (and the documents) at Level 2.

- **Level 2 (ECSS-M-10, ECSS-Q-10 ...)** – The documents at Level 2 describe the required objectives and functions for all aspects in the individual domain (project organization, quality assurance, system engineering, etc.).
- **Level 3** – The documents at Level 3 describe methods, procedures and recommended tools to achieve the requirements of Level 2 documents. In addition they define the constraints and requirements for interfaces and performance of the specified product or activity. The Level 3 documents are guidelines and are allowed to be adapted to the projects' needs.

Also, as a reference, and given that a large number of them will be considered for this project, space engineering branch is broken down and included in Figure 1.12.

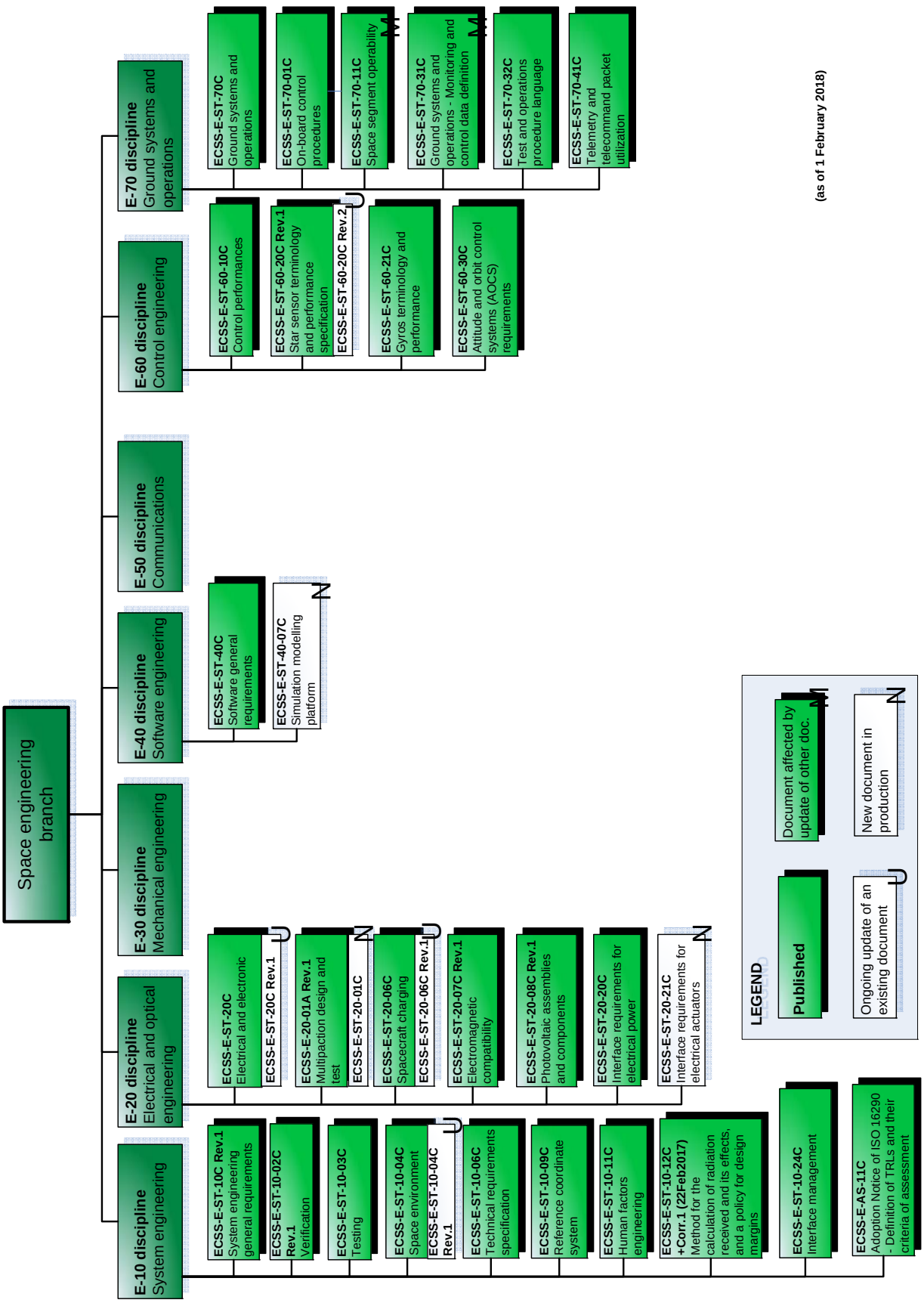


Figure 1.12 – *ECSS Engineering branch breakdown [12]*

1.3 Introduction to System Engineering. EDP in Space.

1

Systems engineering is an interdisciplinary field which involves engineering management and focuses on how to design and manage complex systems over their life cycles [59].

A system is an integrated set of elements to accomplish a defined objective. These elements include hardware, software, firmware, human resources, information, techniques, facilities services, and other support elements.

Figure 1.13 shows, according to ECSS-E-ST-10C [76], the boundaries of system engineering, its relationship with production, operations, product assurance and management disciplines and its internal partition into the following system engineering sub-functions:

- **Requirement engineering**, which consists on requirement analysis and validation, requirement allocation, and requirement maintenance;
- **Analysis**, which is performed for the purpose of resolving requirements conflicts, decomposing and allocating requirements;
- **Design and configuration**, which results in a physical architecture, and its complete system functional, physical and software characteristics;
- **Verification**, which objective is to demonstrate that the deliverables conform to the specified requirements;
- **System engineering integration and control**, coordinating the various engineering disciplines and participants;

Through this process the system engineering function performs a multidisciplinary functional decomposition to obtain logical lower level products.

Along with the variety of disciplines in engineering, several design processes have been proposed throughout time whether in a general way or a field-specific. Space field is not an exception to that and there are different models applicable in order to accomplish a systematic and effective design process. It is the so-called **Engineering Design Process** or EDP.

Within space sector, the highest level is often called ***mission level***, which usually consists of one or more segments (space, ground, user...). From this perspective, requirements originate from the next upper level (the customer) and elements are procured from the next lower level (the suppliers).

This project will apply one of the most extended design methodology, proposed by Wertz and Everett, see [94]. It divides the design process into a series of phases or steps,

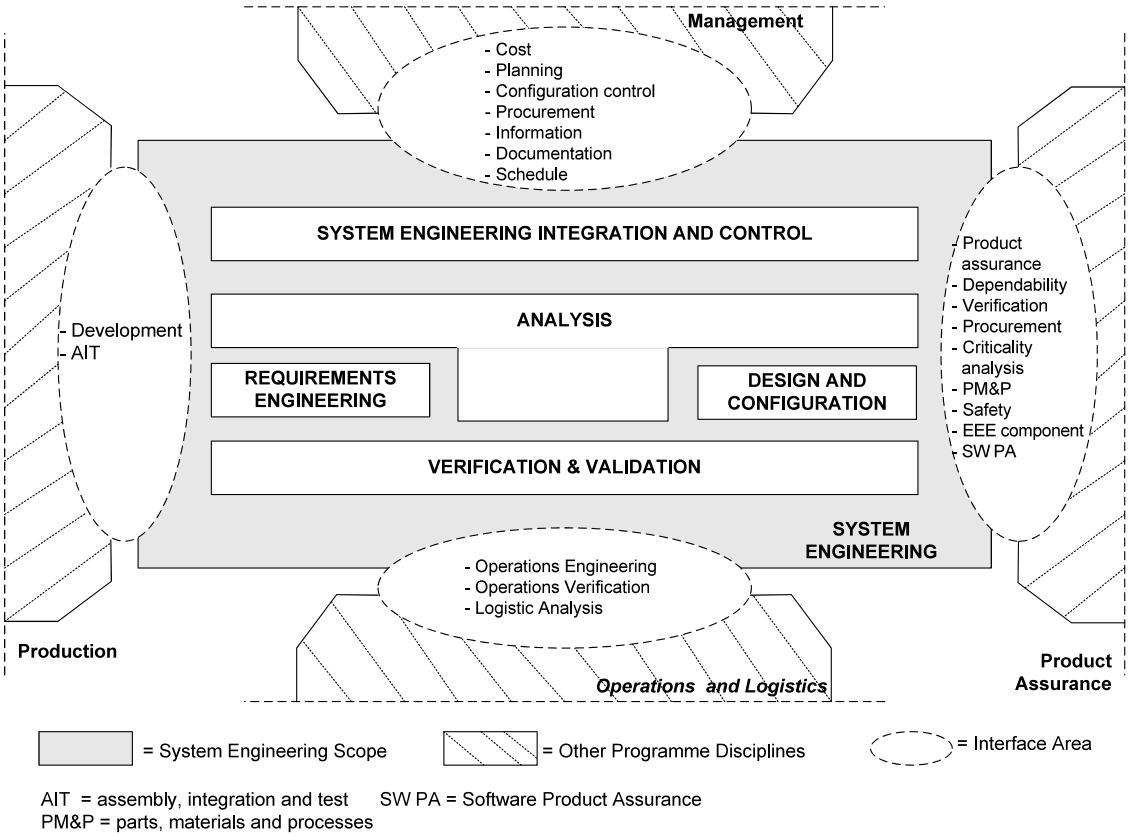


Figure 1.13 – *System engineering, sub-functions and boundaries [76]*

which coincides with the structure followed not only during the development of this project but also in this document itself.

As previously stated, any design process is in some way focused on **requirements**. Requirements definition is a complex and iterative process which begins with one or more objectives and continues defining a system which meets them as much as possible, with the **minimum cost and risk**. While overall objectives normally remain the same through the whole process, they can be accomplished in many different ways according to the technology or solution applied. The Design process followed in Space field breakdowns into different phases:

- **Objectives/goals definition.** Expectable results from the project, not only in technical terms but also economic, social, political, etc. This is the very first stage of the design process and defined goals here are not supposed to vary during the process. They are treated in Section 1.4.

- **Space Mission Engineering.** Inherently iterative process in charge of refining both the requirements and methods to accomplish them. Each iteration will lead to a more detailed and defined mission concept. Primary and secondary

objectives are specified, Stakeholders interests are considered, the timeline is defined, etc. It is treated in Chapter 2.

- **Space System Engineering.** The previously preliminary requirements defined are translated into a set of specific requirements on every element of the system. This translation will determine performance, cost and risk. It is extensively treated in Chapter 3.

- **System Design.** The project is designed according to the requirements stated in previous stages, comparing different technologies and implementing the optimal one for each subsystem. It is treated in Chapter 4.

- **Verification & testing.** The project is concluded by conducting different functionality tests which verifies the results and checks the compliance with the requirements. It is exposed in Chapter 5.

Figure 1.14 briefly illustrates this flow. This project will follow a methodology as close as possible to this system engineering perspective, according to [82] standards.

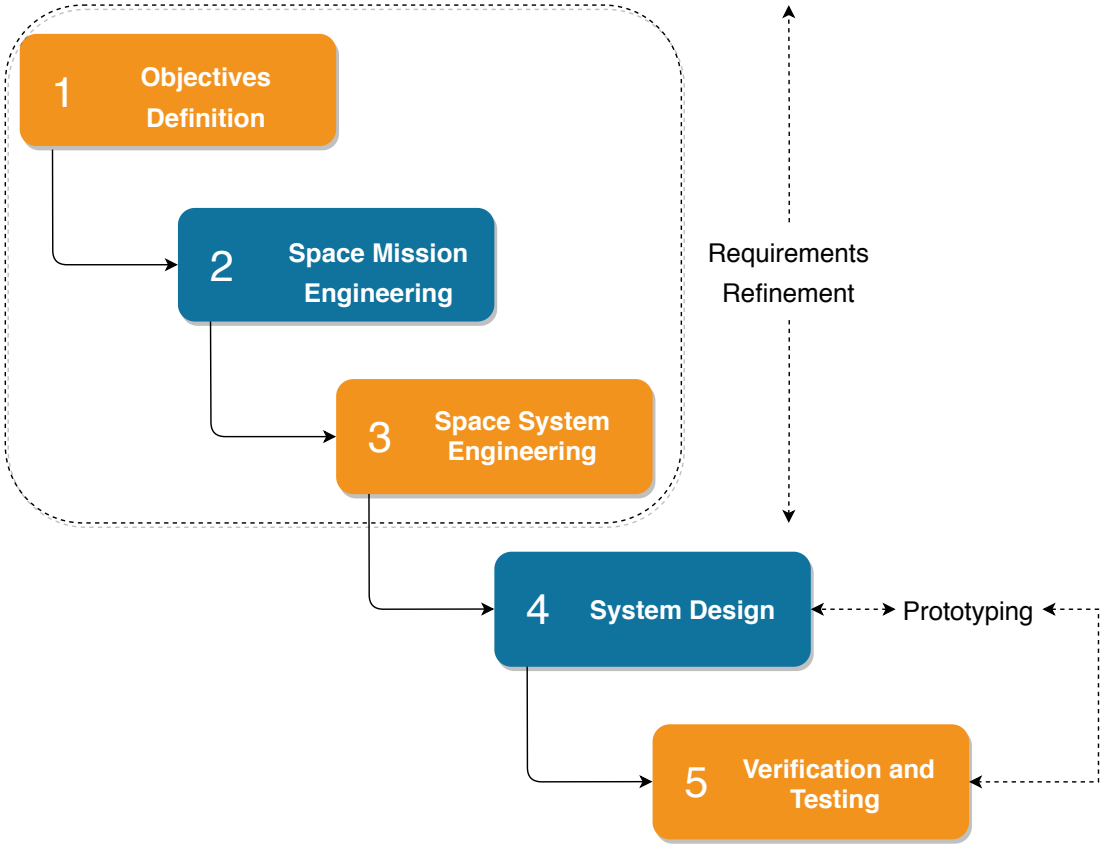


Figure 1.14 – *Engineering Design Process (EDP) in space*

1.4 Project Goals and Objectives

After introducing the problem statement which motivates this Master's Thesis, this section outlines the main top-level non-technical goals of the project. The reason to limit this section to those is to stick to the **Engineering Design Process** described in Section 1.3 and followed throughout this Master's Thesis, which considers objectives definition part of a much more complex and iterative process called *Requirements Definition*. Besides, considering again the double perspective (technical testing and academic) explained in previous sections, and the substantial extent of this project, I decided to breakdown the objectives and functional requirements of each subsystem in Chapter 2 **(Mission Engineering)**, while formal requirements are addressed in Chapter 3 **(System Engineering)**. Then, Chapter 4 addresses the design of the system in accordance with the exposed in previous chapters.

Therefore, objectives listed in Table 1.1 must be understood as the author's expected results in academic and professional terms of the execution of this project.

Ref.	Objective
Obj.1	To gain experience with different standards typically used in aerospace field such as ECSS or CCSDS.
Obj.2	To familiarize with professional Requirements Refinement techniques by applying the Engineering Model Process, particularly oriented to space field.
Obj.3	To generate enough and clear documentation of the whole process, which may be required during the same project or just useful for future references or designs.
Obj.4	To dynamically acquire multidisciplinary knowledge of different specialties as needed during the execution of the project, featuring Mechanical and Aerospace Engineering concepts and techniques.
Obj.5	To demonstrate the knowledge acquired during both, Bachelor's and Master's degree in Telecommunication Engineering, as well as multidisciplinary abilities gathered during the execution of this Master's Thesis.
Obj.6	To successfully overcome the subject 'Master's Thesis'.

Table 1.1 – *Top-level objectives of the project*

1.5 Project Structure

This project, divided into six chapters and an addendum, progressively analyzes the system under development from different points of view, addresses the design tasks and finalizes with the successful completion of the product.

These chapters are:

- **Chapter one**. This chapter, which is intended to be an introduction and show the general objectives and the reasons which motivate this project. The prior art, as well as a brief introduction to the system engineering methodology applied in this development, are also included in this chapter.
- **Chapter two**. The second chapter addresses the so-called stage *Space **Mission** Engineering*, in which the technical objectives of the system are remarked from a high abstraction level. These objectives lead to the **Functional Requirements Definition**.
- **Chapter three**. According to the system engineering methodology applied, the third stage of the project corresponds to the *Space **System** Engineering*, so it is treated in this chapter. It is included a short introduction to the testing procedures needed to comply with a real mission constraints, including several ECSS standards. Besides, in this chapter the project is breakdown into three separated blocks; each of them is analyzed, and the **Formal Requirements** are defined, as a previous step to the design stage.
- **Chapter four**. The fourth chapter deals with **system design**. It translates the technological solutions analyzed in the previous chapter to actual systems able to execute the tasks required. The blocks structure of the project introduced in the previous chapter is followed again, and each of them is extensively treated, including details at all levels of the design task.
- **Chapter five**. Once the design is completed, chapter five addresses a series of validation tests, in order to check that the system meets the **Formal Requirements** defined and, consequently, the Functional Requirements. It is worth noting that this validation is performed using one of the tools developed within this project.
- **Chapter six**. Finally, chapter six includes the main conclusions extracted from this Master's Thesis, as well as some future lines of work which have naturally emerged during the design process.
- **Addendum**, in which the budget and associated cost of this project are detailed.

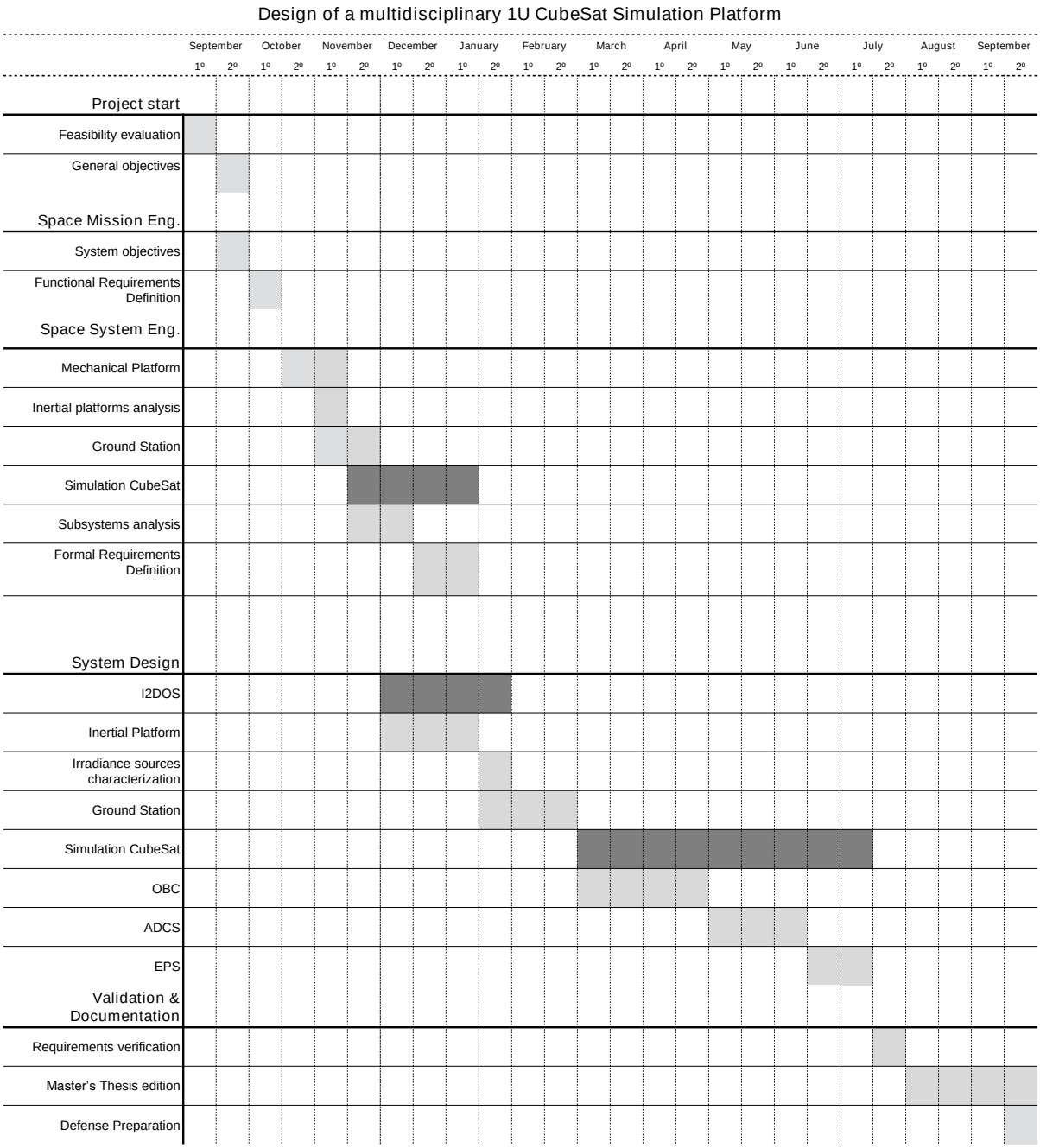


Figure 1.15 – *Gantt Chart of the Project*

1

Chapter 2

Space Mission Engineering

Space Mission Engineering is a broad process in charge of determining the needs of the *client* or goals of the system. Those initial needs are not likely to change and can be considered the **objectives of the mission**. In that line, this chapter will address top-level objectives, analyzing them and culminating with the task **Functional Requirements Definition** for each of the subdivisions of the system.

Furthermore, a brief introduction to this stage of the project is included, in relation with both, generally speaking missions and this particular project. Regarding this point, it may be needed to remember that the double scope of this Master's Thesis is at the halfway between a real mission design and the development of a simulation platform, which will imply taking balanced decisions when an excessively mission-oriented requirement is needed, i.e., space-qualified materials will not be treated in this project. Those assumptions shall be taken into account for future developments specifically intended to fly in orbit.

2.1 Introduction to Mission Engineering

Mission Engineering typically begins with one or more general objectives and constraints which build up the base to later define the functional requirements of the system, i.e., which tasks should the system be able to perform, from the *client* (mission) point of view. It is usually an iterative process, consisting in refining the requirements expected, being aware of the **capabilities** and **constraints** of the technical team and equipment. Documenting the results of this iterative process must also be part of the philosophy of work, so decisions can be reexamined and justified if needed.

In a real mission, setting excessively optimistic or ambitious objectives (with the aim of getting the greatest amount of results, for example) may be counterproductive, taking

to a delayed delivery of the system (increasing costs) or, in the worst scenario, to a failure of the project. Indeed, not only too optimistic goals are risky, overdoing details of the mission in any sense may lead to unexpected problems once the mission is analyzed as a whole. At this point, the main focus must be to translate the needs and objectives of the end user into functional requirements.

As analyzed in [94], this stage typically implies a series of steps through which the team must wonder themselves the questions that will lead to the definition of the needs and requirements. Some of these are:

- What are the qualitative goals seeked and why?
- Who are the Stakeholders?
- Which is the timelime?
- Considering needs, available technology, previous know-how and so forth, how well should the broad objectives be achieved?

Therefore, Mission Engineering is a crucial stage of the **Engineering Design Process** (EDP) in space, in order to get the most performance possible for the inversion made. Its importance is directly related to the early point of the project in which they are defined; because they are stated as the goals to achieve, the design will point to those from the beginning, implying therefore seeing a significant cost increase in the project as a result of any eventual change to them. as well as additional, avoidable risk. All in all, the final goal is to successfully fulfill the needs of the mission by meeting the end-clients' objectives in time at minimum risk and cost.

Considering the double perspective of this project, as previously exposed in Chapter 1, these recommendations apply specially for the prototype-oriented focus, as developing a system intended to be in orbit. While the Simulation Platform branch allows relaxing them in some ways, being an academic project with a certain **deadline** makes it recommendable to keep risks under control in order to stick to the schedule and conclude it in time. In any case, as stated in Subsection 1.2.1, because of the nature of CubeSats itself, a certain amount of risk is accepted in these kind of projects; trying new ideas or concepts inherently **includes that risk**, even more so when low cost is intended. Accomplishing the mission under that condition will be the real challenge.

2.2 System Objectives. Functional Requirements Definition

After introducing the foundations of the Mission Engineering Process, the technical objectives of the system are defined, in contrast with the ones defined in Section 1.4. They fit the top-level needs and will lead to the definition of the **functional requirements**

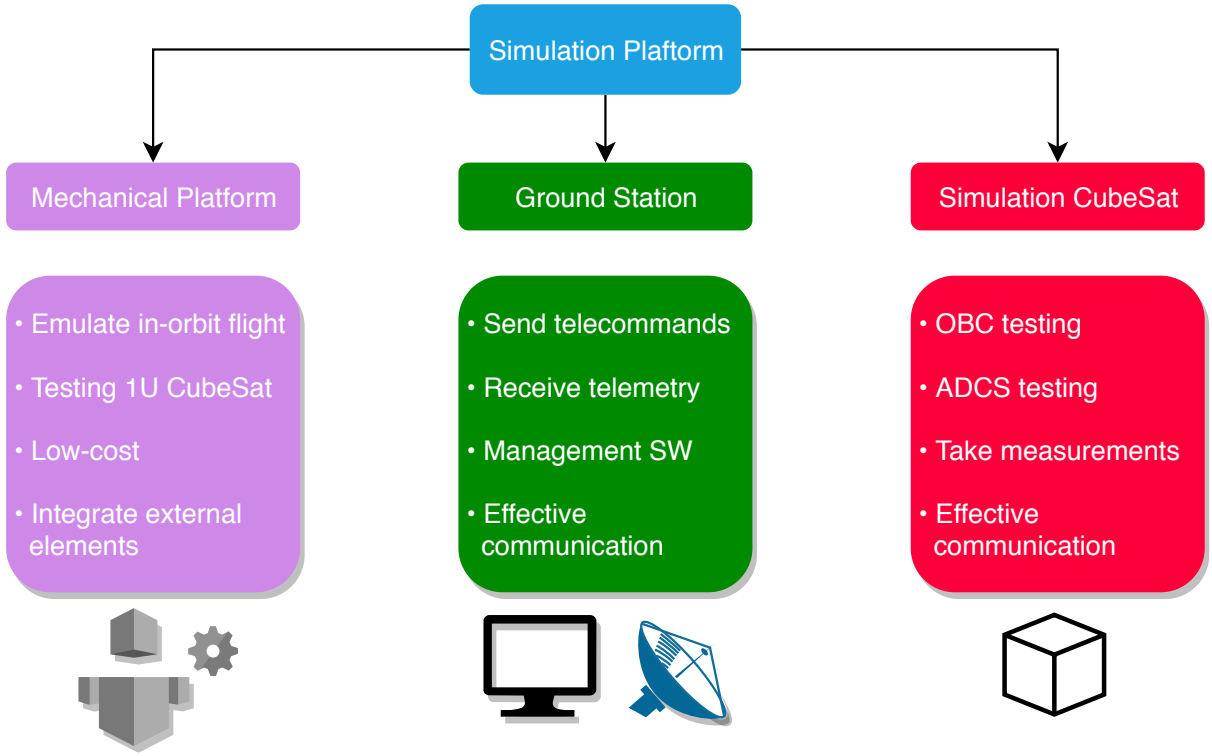


Figure 2.1 – *Simulation Platform Subsystems*

of this project, after performing the iterative process stated in Section 2.1; firstly, the breakdown of the parts of this project is shown. The proposed simulation platform will be composed of three different subsystems, illustrated in Figure 2.1, in which the main purposes and functions of each are featured.

As seen in Figure 2.1, this project involves three clearly differentiated parts, which will be broken down in later chapters. Following the philosophy of this stage, now they are considered from a top-level mission's view, determining the technical objectives of the project for each one.

2.2.1 Mechanical Platform

Although, as illustrated in Figure 2.1 this project as a whole has the final purpose to develop a complete Simulation Platform framework, one of the main differentiated parts is specifically a physical simulation platform which allows testing a CubeSat prototype, as well as interacting with it in every possible way. Therefore, this part of the platform will require an extensive work of **mechanical design**. Also, this subpart of the system answers both orientations of this Master's Thesis: on the one hand, it allows technical testing of functionalities, which can be useful at early stages of the development, before going through official testing (see Section 3.1) so the approximate

performance of the prototype can be analyzed; additionally, it allows determining the capabilities of the prototype, feasibility of the design, changes needed, etc. On the other hand, it also satisfies the educational aspect of this project, by allowing students to implement different solutions in a space-oriented platform, easing the acquisition of specific concepts and putting it into practice. Taking this into account, the set of objectives related to the mechanical platform is shown in Table 2.1. Note that both objectives and functional requirements are identified by a code which eases a quick reference when needed; this notation will be used through all this project when dealing with requirements.

Technical Objectives	
Ref.	**Primary**
MP.Obj.1	To allow testing at least 1U CubeSat, physically rotating, emulating in orbit flight.
MP.Obj.2	To allow integration of external elements to the platform, such as one or more lightning source emulating the Sun, magnetic field presence, etc.
Ref.	**Secondary**
MP.Obj.3	To keep it as lightweight as possible, maximizing portability.
MP.Obj.4	To build it using a low-cost philosophy, specially in prototyping stage.
MP.Obj.5	The subsystem shall use inexpensive materials in prototyping stage.

Table 2.1 – *Mechanical Platform - Technical Objectives*

While **Primary Objectives** are those to be fulfilled in order to consider a certain level of success, **Secondary Objectives** are additional goals which would complement the primary list. All in all, they express the needs of the client and/or mission. From these, functional requirements can be extracted, as shown in Table 2.2.

Ref.	Functional Requirements
MP.FR.1	The subsystem shall have enough capacity to host at least 1U CubeSat.
MP.FR.2	The subsystem shall have low friction in order to emulate in orbit conditions.
MP.FR.3	The subsystem shall weight as less as possible (without CubeSat unit).

Table 2.2 – *Mechanical Platform - Functional Requirements*

2.2.2 Ground Station

When talking about a satellite link, Ground Station is the Earth's-end in charge of dealing with the communications tasks from and to the spacecraft. In the same way, this project will make use of the Ground Station to establish a communication link with the CubeSat under test, allowing testing TC&C System, but will also function as central server of a hypothetical network of CubeSats, which would be certainly useful in a classroom environment, in which each student would have their CubeSat but only one Ground Station is needed. Once again, this subpart satisfies both orientations of this Master's Thesis. Table 2.3 and Table 2.4 shows the technical objectives and functional requirements of this part, respectively.

Technical Objectives	
Ref.	**Primary**
GS.Obj.1	To allow Ground Station operators to effectively communicate with GranaSAT-I CubeSat.
GS.Obj.2	To allow sending control commands to GranaSAT-I.
GS.Obj.3	To allow receiving telemetry and payload data from GranaSAT-I.
GS.Obj.4	To allow defining new control commands and telemetry packages.
Ref.	**Secondary**
GS.Obj.5	To function as central Ground Station for a number of compatible CubeSats, potentially a swarm, in a local environment. It may be useful as approach to the real case stated in [104] in which I collaborated with Professor Stakem from **Goddard Space Flight Center** (NASA).
GS.Obj.6	To count with software analysis in charge of managing transmitted and received data.
GS.Obj.7	To communicate with GranaSAT-I using industry standard formats.

Table 2.3 – *Ground Station - Technical Objectives*

2.2.3 Simulation CubeSat

The third top-level subpart of the system is the simulation CubeSat itself, the GranaSAT - I. It is intended to be used as the basis to keep working in the development of an university CubeSat in the University of Granada. As for this project, it must be able to integrate within the physical simulation platform, described in Subsection 2.2.2, as well as communicating with it in the terms stated before. It must comply with CubeSat standard [5] and being designed in such a way future designs and subsystems can be integrated. Particularly, this project will deal with **On-board Computer** (OBC),

Ref.	Functional Requirements
GS.FR.1	The subsystem shall have a communication system compatible with CubeSat GranaSAT-I, in terms of frequency, technology and whatever other aspect needed to accomplish an effective communication.
GS.FR.2	The subsystem shall have a software module in charge of packetization and de-packetization.
GS.FR.3	The subsystem shall have a software module in charge of keeping logs, plotting received data, managing alerts and any other capability liable to implemented. It shall allow data analysis in both, real time and post-processing stage.
GS.FR.4	The subsystem shall be able to function in a distributed local network of compatible CubeSats, managing communications, connections, etc.
GS.FR.5	To communicate with GranaSAT-I using CCSDS (see 3.3.1.1).

Table 2.4 – *Ground Station - Functional Requirements*

Attitude Determination & Control System (ADCS) the photovoltaics module of the **Electrical Power System** (EPS) and basic **On-Board Data Handling** OBDH functions, integrated in the OBC. Additionally, as part of the software subsystem of the CubeSat, at least OBC and ADCS shall have a minimum working set of functions, which allow testing basic functions of the platform (mission simulation approach) and can be used as guide for students to implement and try new features (academic approach).

This is the largest and most complex subpart of the whole system. It involves tasks belonging a wide range of the technical spectrum, from **Mechanical Engineering** (GranaSat-I physical design and characterization and inertial orbit simulator design) to **Telecommunication Engineering** (electronic design, communication, software development) or **Aerospace Engineering** (in orbit dynamics, space representations). This fact makes this project an actual representation of what a mission design or space-related project is like: a multidisciplinary challenge which combines work and knowledge of completely different fields in order to reach a common objective.

These end user objectives yields to the functional requirements listed next. They contain not only those derived from the iterative analysis of the mission objectives, but also taking into account ECSS applicable standards, featuring ECSS-E-ST-60-30C [78] or ECSS-E-ST-70-41C [79] among others. Once more, the intention is keeping the design process as close as possible to a real mission design, easing future improvements of this project intended to be in orbit.

Table 2.5 lists Technical Objectives for the Simulation CubeSat.

Technical Objectives	
Ref.	**Primary**
SC.Obj.1	To be an Engineering model for GranaSAT-I design, which allows testing both provided functionalities and new ones added in subsequent projects.
SC.Obj.2	To allow the complete testing possible of OBC module developed, from management of the GranaSAT-I electronics, to dealing with the different functioning modes, or coping with critical situations.
SC.Obj.3	To allow the complete testing possible of ADCS module developed, from management of the GranaSAT-I maneuvering system, including motors or Magnetorquers, detumbling mode, send sensors data to the OBC dynamically, etc. etc.
SC.Obj.4	To allow space and magnetic attitude determination, and control, if needed.
SC.Obj.5	To allow photovoltaics module testing, when used with a proper EPS. This module must be able to provide enough energy to the GranaSAT-I in every necessary point of the mission/simulation.
SC.Obj.6	To allow wired communications, when used in the simulation platform, in order to change any internal configuration or download data.
Ref.	**Secondary**
SC.Obj.7	To keep it as lightweight as possible, maximizing portability.
SC.Obj.8	To measure different ambient conditions, such as pressure, temperature or similar.
SC.Obj.9	To allow wireless communications, when used in the simulation platform, in order emulate radio-link communication used in a real in orbit mission.
SC.Obj.10	To deal with the changing conditions *in orbit* and act consequently.
SC.Obj.11	To determine its angular velocity.

Table 2.5 – *Simulation CubeSat - Technical Objectives*

Table 2.6 shows the Functional Requirements determined.

Ref.	Functional Requirements
SC.FR.1	The subsystem shall follow applicable interconnection standards, so OBC and any other subsequent can be tested.
SC.FR.2	The subsystem shall have movement, magnetic and ambient conditions sensors, as well as a tachometer-like sensor.
SC.FR.3	The subsystem shall have integrate ADCS and lightning sensors capabilities in order to determine power source attitude or irradiation level.
SC.FR.4	To ensure that the attitude guidance and pointing specified by the mission requirements, during the mission operational phase, are met. This includes tracking of a fixed point or Sun pointing, as well as any other specific attitudes needed for system purposes, like communications for instance.
SC.FR.5	To perform the attitude measurement, estimation, guidance and control needed for the mission, autonomously.
SC.FR.6	The subsystem shall count with IEEE802.11b [91] connection capability, integrated with the OBC.
SC.FR.7	The subsystem shall count with IEEE802.3 [92] (Ethernet) connection capability, integrated with the OBC.
SC.FR.8	The subsystem shall count with USB [108] connection capability, integrated with the OBC.
SC.FR.9	The whole system shall provide Housekeeping telemetry to enable the verification of the nominal behaviour of sensors, actuators and on-board functionalities, on ground.

Table 2.6 – *Simulation CubeSat - Functional Requirements*

Chapter 3

Space System Engineering

This third chapter tackles another phase related to analysis and requirements definition. It resembles the more general 'System Analysis' and will cover the whole Platform, as well as the most common **testing procedures** demanded in space according to ECSS. The Platform will be analyzed following the previously defined structure, divided into Mechanical Simulation Platform, Ground Station and Simulation CubeSat.

After deeply studying every subsystem and technologies associated to each, **formal requirements** are defined (in contrast with the objectives and preliminary requirements defined in Chapter 1 and Chapter 2) according to the Design Process followed in this project, described in Section 1.3. At that point, the project may go through the **System Design** process, detailed in Chapter 4.

3.1 Testing procedures

CubeSats intended to be in orbit must pass a series of comprehensive testing programs in order to get the qualification to fly. Through this analysis, the different tests performed are introduced, so the constraints can be taken into account in Chapter 4, and especially in future designs. In a real mission project, in which a qualification to fly is needed, it is vital to know which are the technical goals the system shall comply with as soon as possible.

This project does not intend to perform a complete testing-oriented design, because its double perspective does not allow to set space-related requirements specifically, needed to pass the demanding testing stage. Instead, the most relevant tests are introduced; they can be considered when setting the requirements and the design can stick to them as much as possible, enabling to design in complete accordance with them in the future;

the aforementioned work will suppose less requirements change and their associated costs when that re-engineering process is performed.

Testing environment and requirements are given by standard ECSS-E-ST-10-03C [75]. Particularly, this standard splits the process into two levels of decomposition: requirements for **space segment equipment test** and **space segment element test**. While *space segment* refers to a part of a space system intended to fulfil the goals of the mission, *space segment elements* are elements within a space segment.

3.1.1 Space segment equipment test requirements

3.1.1.1 General tests

3.1.1.1.1 Humidity test

In order to prevent the space segment of the system from the humidity effects (see more information on this in ECSS-Q-ST-70-01 [80]). This test shall be performed if space segment equipment can be exposed to humidity level above 65 % during its life time. To qualify humidity test, the space segment equipment shall be installed in a chamber with temperature at room ambient conditions. Temperature shall be increased to +35°C and humidity to at least 95 % over an hour. This process must be repeated during 4 different cycles according to the mentioned standard. Finally, the equipment must be visually inspected for deterioration or damage. An example of humidity chamber is shown in Figure 3.1.

Figure 3.1 – *Temperature and humidity chamber [11]*

3.1.1.1.2 Life test

The life test for space segment equipment qualification shall be designed to demonstrate the ability of the space segment equipment to withstand the maximum operating time and the maximum number of predicted operational cycles during the "product lifetime" by providing the required performance at the end of life. Obviously, the test must be performed under the environmental conditions expected during actual operation, including ambient, thermal, vacuum or a combination of these.

3.1.1.2 Mechanical and structural integrity tests

3.1.1.2.1 Physical properties measurements

Mechanical integrity is one of the most crucial aspects in every spatial system. Therefore, according to ECSS-Q-ST-70-01 [80], there must be determined the following physical properties:

- Dimensions and interfaces
- Mass
- Centre of gravity with respect to a given coordinate system for three mutually perpendicular axes
- Momentum of inertia with respect to the given coordinate system

Because of the intermediate character of the simulation platform designed in this project, between real simulation and training platform, this testing will be applied not only to the CubeSat itself, but to the whole physical simulation platform, as widely analyzed in Chapter 4.

3.1.1.2.2 Acceleration test

In order to ensure uniform force distribution on the space segment equipment, it is centrifuged with an arm whose length shall be at least five times the dimension of the space segment equipment measured along the arm.

3.1.1.2.3 Random and sinusoidal vibration test

Random and sinusoidal tests shall be both conducted in the launch configurations for all axes. In order to evaluate the space segment equipment integrity, a resonance search

shall be performed before and after both tests. The equipment shall not suffer a shift in frequency above 5 % and 10 % for amplitude shift, respectively. Figure 3.2 shows a typical vibration test equipment from ESA.

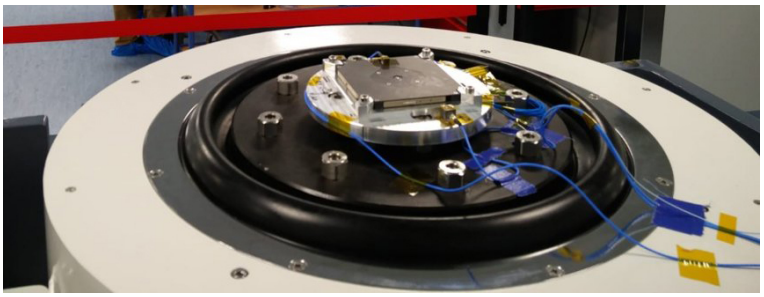


Figure 3.2 – *Vibration test [17]*

3

3.1.1.2.4 Shock test

The shock tests demonstrate the ability of the space segment equipment to withstand the shocks encountered during its lifetime, e.g.: fairing separation or solar arrays and antennas deployment. The equipment shall be powered during the test and the selected shock test method shall achieve the specified **Shock Response Spectrum** with a representative transient, comparable in shape and duration to the expected in-flight shock.

Once again, visual inspection and hardware integrity review shall be performed after test. Figure 3.3 shows a standard shock test equipment.

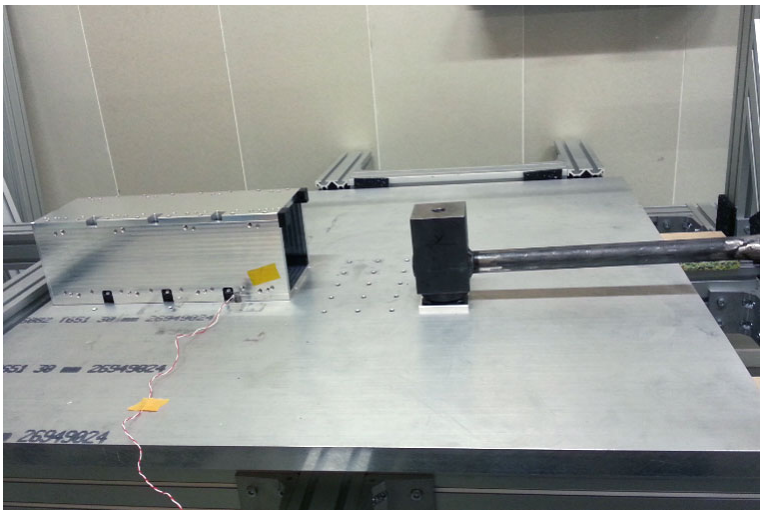


Figure 3.3 – *Shock test equipment [24]*

3.1.1.2.5 Leak test

The leak test shall demonstrate the ability of sealed or pressurized space segment equipment to conform to the leak rates stated in the specifications. They are performed only on sealed or pressurized space segment equipment, sensitive to loss of pressure or vacuum. Additionally, leak tests shall be performed prior to and following the completion of space segment equipment thermal and mechanical tests. Figure 3.4 shows a leaking test performed to a CubeSat at ESA.

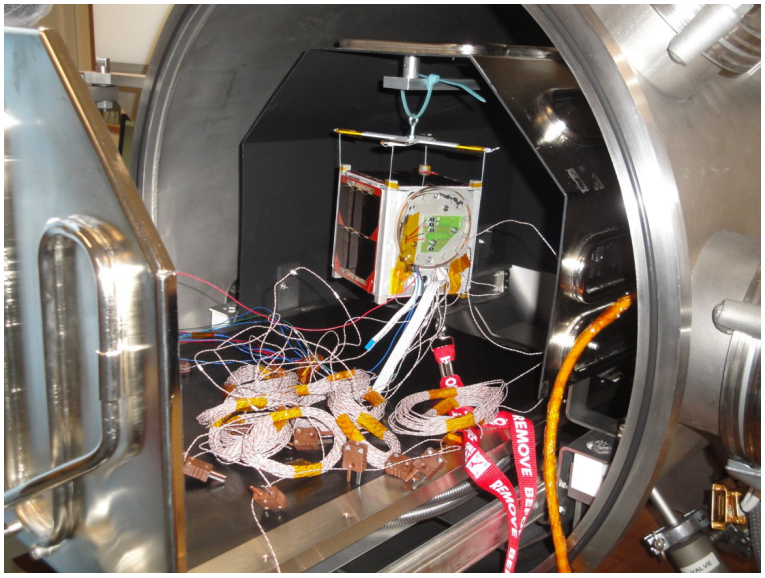


Figure 3.4 – *Leak test [17]*

3.1.1.3 Thermal-Vacuum tests

While ECSS-E-ST-10-03C [75] is the general standard for testing, ECSS-E-HB-31-03A [81] and ECSS-Q-ST-70-04C [74] particularly address Thermal-Vacuum testing and analysis; this kind of test subjects the system to a series of temperature cycles in vacuum conditions as present in space. The latter of them include among the deleterious effects to be anticipated during the thermal cycling test under vacuum:

- Outgassing
- Cracking, contraction or fracture of materials or assemblies due to sudden dimensional changes by expansion.
- Overheating of materials or assemblies due to change in convection and conductive heat transfer characteristics.

This standard also specifies conditions for work area, such as the need to be nominally clean with minimum dust, filtered ventilation or ambient conditions relative to temperature (25 °C ± 3) or humidity (25 % ± 10). A typical thermal cycling test has the following steps:

- The initial characteristics are analyzed at (22 °C ± 3) or humidity (55 % ± 10) within six hours prior to the beginning of the thermal cycling.
- Thermal cycling begins after a working vacuum of 1×10^{-5} Pa to ± 5 % has been reached.
- The chamber is thermally cycled between temperatures of (±100 °C ± 5) unless otherwise specified, at a nominal heating or cooling rate of +(10 °C ± 2) per minute.
- A minimum of 100 thermal cycles shall be performed on each sample.
- Within six hours after the completion of thermal cycling in vacuum, final inspection and testing of the sample shall be conducted at the initial ambient conditions.

This test is commonly performed in a so-called **Thermal Vacuum Chamber** or TVAC, as shown in Figure 3.5.

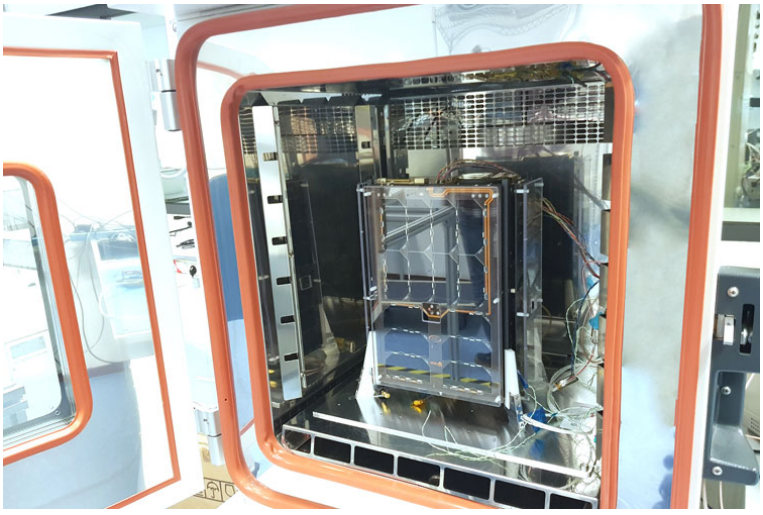


Figure 3.5 – *Thermal Vacuum Chamber [24]*

GranaSAT laboratory counts with a self-made TVAC with a full *in-house* design; details are available at [107]. Figure 3.6 shows some photos of the chamber. In the future, the testing procedures described above will be performed using that equipment, allowing to determine the accuracy of a certain system to the standard, before going through the official tests of the *Fly Your Satellite!* [13].

(a) *Front*

(b) *Side*

Figure 3.6 – *Thermal Vacuum Chamber at GranaSAT laboratory*

3.1.1.4 Electrical/RF Tests

The following Electrical/RF Tests are crucial for an adequate functioning of the whole system and are specified in ECSS-E-ST-20-07C Rev. 1 [77].

3.1.1.4.1 EMC

Electromagnetic compatibility (EMC) of a space system or equipment is the ability to function satisfactorily in its electromagnetic environment without introducing intolerable electromagnetic disturbances to anything in that environment. For acceptance stage, the space segment equipment shall be subjected to the following tests:

- **Bonding verification:** also called ground continuity test, consists in testing whether the ground points of a device are well connected in between each other.
- **Power lines isolation.**
- **Inrush current:** consisting in the surge or momentary burst of current that flows into the device when powered on, due to the high initial currents required to charge the capacitors and inductors or transformers. These currents can be as high as 20 times the steady state currents. Even though it only lasts for about 10 ms it takes between 30 and 40 cycles for the current to stabilize to its steady state. It must be limited in order to prevent the equipment from any damage. Figure 3.7 illustrates this phenomena.

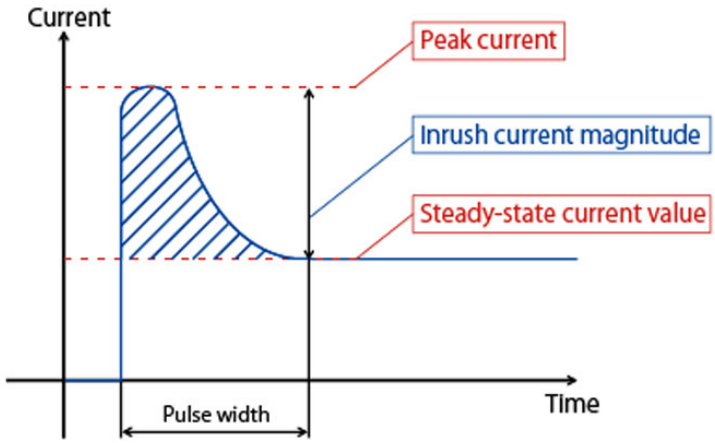


Figure 3.7 – *Inrush current phenomena [51]*

- Conducted emission time & frequency domain on power lines in the operating mode.

3.1.1.4.2 Magnetic test

This test is performed in order to obtain an estimation of the Magnetic moment of the device under test. A common constraint is measuring the magnetic field at distances typically more than three times the size of the device, as well as assigning to its geometric center a right-handed orthogonal coordinate system XYZ. In order to perform this test, the device should be placed in an **earth field compensated area** providing zero-field conditions for the intrinsic moment determination i.e., a set-up which is able to *cancel* Earth's magnetic field, so it does not alter the measurement. This is usually ensured by using a **Helmholtz Cage**. GranaSAT Laboratory counts with one of these systems developed by one of its members throughout the years, see [85]. Figure 3.8a shows the developed Helmholtz Cage while Figure 3.8b shows the detail of the wires along the cage.

(a) *Developed Helmholtz Cage* **(b)** *Wires detail*

Figure 3.8 – *GranaSAT Helmholtz Cage*

Once the setup is ready, the following test sequence must be followed:

- DUT not operating, initial measurements on the six semi-axes at the reference distances.
- Deperm or demagnetization, in order to get an earth field compensated area in the Helmhotz Cage, at a frequency of 3 Hz, maximum amplitude between 4000 µT and 5000 µT, successively on each XYZ axis of the DUT.

- It is applied a perm field of 300 µT on each axis and measurements are repeated.

Once again, this procedure can be tested using our own equipment, shown before. Figure 3.9 shows graphically how this procedure shall be performed. Once the device is inside a demagnetizated area, measurements are taken at reference points.

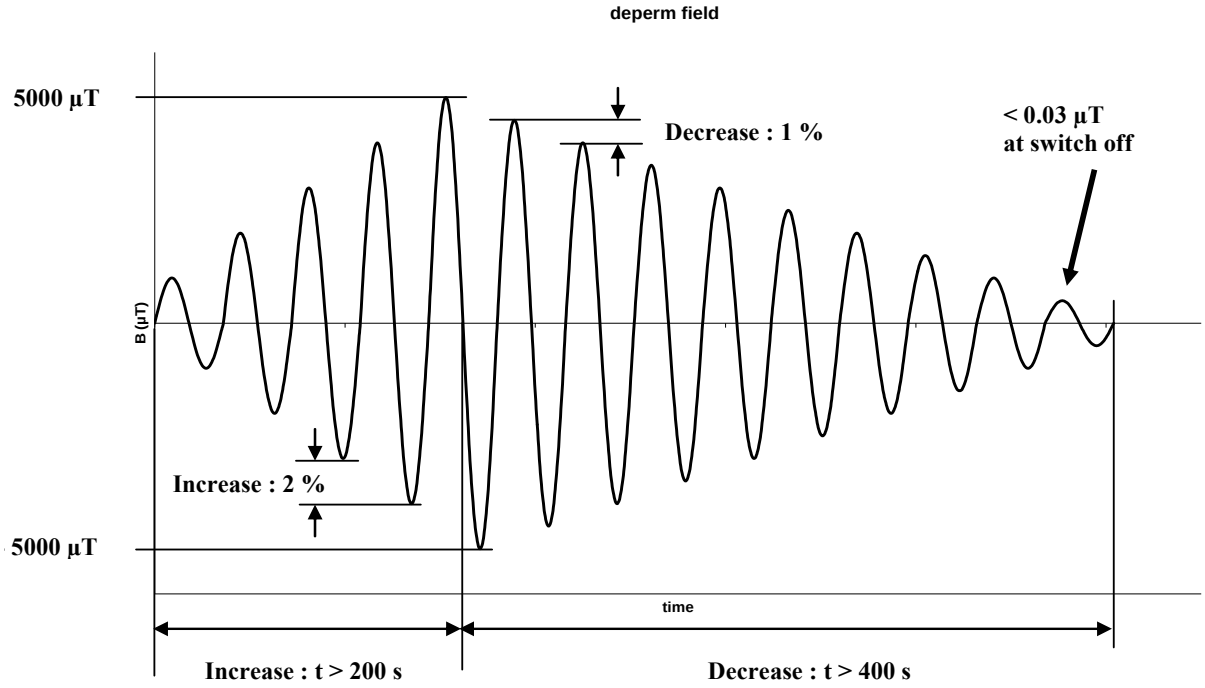


Figure 3.9 – *Smooth deperm procedure [77]*

3.1.2 Space segment element test requirements

Sometimes, a satellite is used as service module on one end and as payload module test on the other. In these cases when it is not feasible to test a space segment element as a single entity, it may be tested individually. On the other hand, sometimes testing as *segment* is not possible just due to the size of the DUT, which exceeds the capacities of the testing facility needed. When some of the described situations apply, instead of performing a **segment equipment testing**, an **element equipment testing** is chosen. As far as the scope of this project concerns, descriptions of Subsection 3.1.1 apply.

To finish this section, allowable tolerances and test accuracies allowed by this standard are included for the most representative tests, Figure 3.10 and Figure 3.11.

Test parameters	Tolerances
1. Temperature	Low High
above 80K	Tmin +0/-4 K Tmax -0/+4 K
T< 80 K	Tolerance to be defined case by case
2. Relative humidity	± 10 %
3. Pressure (in vacuum chamber)	
> 1,3 hPa	± 15 %
1,3 10-3 hPa to 1,3hPa	± 30 %
< 1,3 10-3 hPa	± 80 %
4. Acceleration (steady state) and static load	-0 / +10 %
5. Sinusoidal vibration	
Frequency (5 Hz to 2000 Hz)	± 2 % (or ±1 Hz whichever is greater)
Amplitude	± 10 %
Sweep rate (Oct/min)	± 5 %
6. Random vibration	
Amplitude (PSD, frequency resolution better than 10Hz)	
20 Hz - 1000 Hz	-1 dB / +3 dB
1000 Hz - 2000 Hz	± 3 dB
Random overall g r.m.s.	± 10 %
7. Acoustic noise	
Sound pressure level, Octave band centre (Hz)	
31,5	-2 dB /+4 dB
63	-1 dB /+3 dB
125	-1 dB /+3 dB
250	-1 dB /+3 dB
500	-1 dB /+3 dB
1000	-1 dB /+3 dB
2000	-1 dB /+3 dB
Overall	-1 dB /+3 dB
Sound pressure level homogeneity per octave band	+/- 2 dB
8. Shock	
Shock level	3 dB/ + 6 dB

Figure 3.10 – *Allowable tolerances [75]*

Test parameters	Accuracy
1. Mass	
Space segment equipment and space segment element	± 0,05 % or 1 g whatever is the heavier
2. Centre of gravity (CoG)	
Space segment equipment	Within a 1 mm radius sphere
Space segment element	± 2,5 mm along launch axis ± 1 mm along the other 2 axes
3. Moment of inertia (MoI)	
Space segment equipment and Space segment element	± 3 % for each axis
4. Leak rate	One magnitude lower than the system specification, in Pa m^3 s^{-1} at standard conditions (1013,25 Pa and 288,15 K).
5. Audible noise (for Crewed Element only)	
32,5 Hz to 160 Hz	± 3 dB
160 Hz to 16 kHz	± 2 dB
6. Temperature	
above 80 K	± 2 K
T< 80 K	Accuracy to be defined case by case
7. Pressure (in vacuum chamber)	
> 1,3 hPa	± 15 %
1,3 10^{-3} hPa to 1,3 hPa	± 30 %
< 1,3 10^{-3} hPa	± 80 %
8. Acceleration (steady state) and static load	± 10 %
9. Frequency for mechanical tests	± 2 % (or ±1 Hz whichever is greater)
10. Acoustic noise	± 0,1dB
11. Strain	± 10 %
12. EMC	See ECSS-E-ST-20-07 clause 5.2.1.
13. ESD	See ECSS-E-ST-20-06 See ECSS-E-ST-20-07 clause 5.2.1 for ESD test on space segment equipment.

Figure 3.11 – *Tests accuracies [75]*

This finalizes the analysis of the most relevant testing procedures needed to qualify a design to fly, according to ECSS. It must be used as basis not only for this project but especially for future designs, which shall be performed sticking to them as much as possible since the early stages of the design, taking advantage of the previous work developed here.

At this point, where the main challenges to overcome for the design have been outlined, the main subsystems that integrate this project are analyzed next, following with the formal requirements definition.

3.2 Mechanical Platform Analysis

As seen in Section 3.1 CubeSats can be tested in a large variety of situations and circumstances. Since their outbreak, different kinds of non-professional platforms have emerged, which allow enthusiasts and university developers to test their designs before official assessment, as seen in previous section. As commented before, this Master's Thesis proposes a planar platform to emulate in orbit dynamics in a 2D plane; the formal requirements will be treated along this section. Previously, some of the most featured kinds of platforms commented are reviewed, by way of *Prior Art* in this particular subsystem.

3.2.1 Air bearing

Air bearing platforms have been used for satellite ADCS testing for decades. The primary objective of air bearing tests is the faithful representation of spacecraft dynamics, so they try to get a minimal-torque environment, as in space, which is particularly difficult to duplicate. This frictionless nature partially simulates a zero-g environment, allowing the pitch, roll, and yaw (see section 3.4.2.1.1.2) control systems of the satellite to work as they would in space.

It allows manipulating the system under more realistic conditions. It is one of the most popular technologies for this goal. Figure 3.12 illustrates a 3D model example of an air bearing testbed.

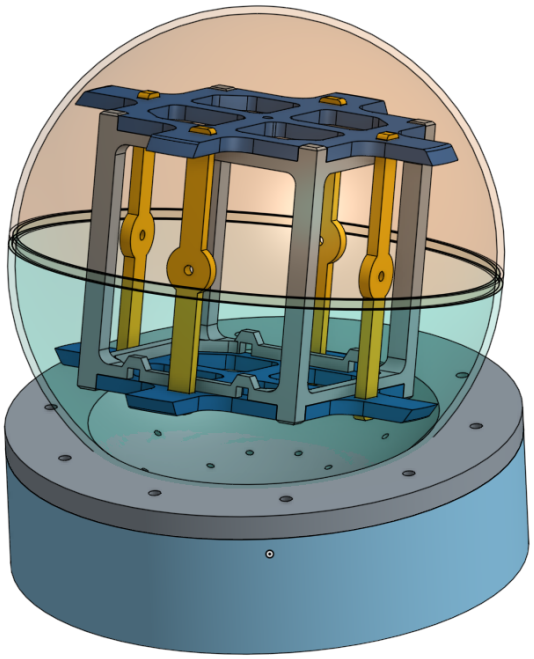


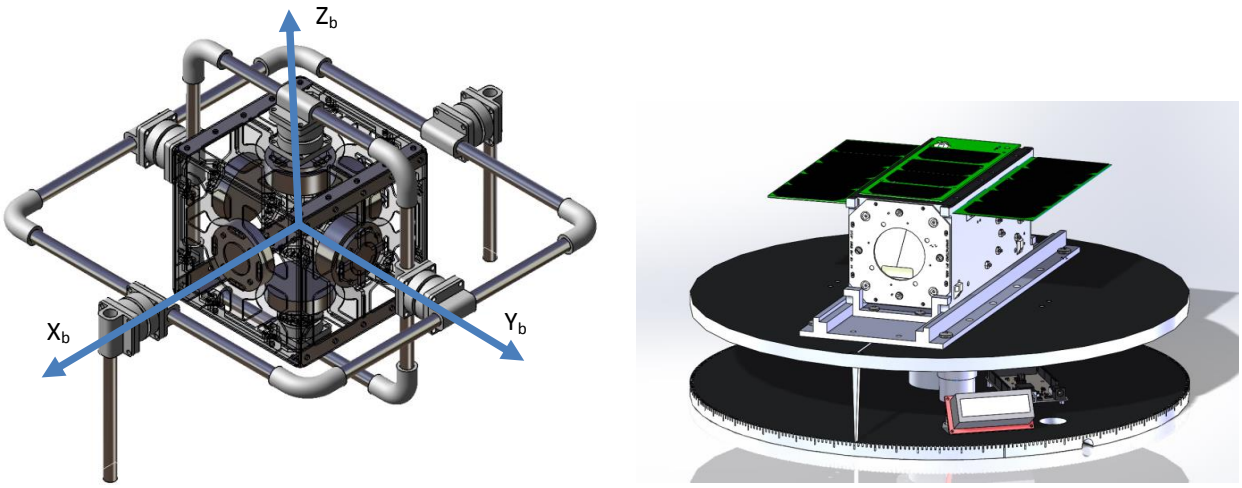
Figure 3.12 – *Spherical air bearing testbed 3D model [67]*

Additionally, air bearings are intended to enable DUT to experience some level of rotational and translational freedom. Pressurized air passes through small holes in the grounded section of the bearing and establishes a thin film that supports the weight of the moving section. These planar air-bearing systems provide one rotational and two translational degrees of freedom [83].

One of the most common types of air bearing is the **spherical**, as the one depicted in Figure 3.12. As shown, the two sections of the bearing are portions of concentric spheres, one of which rotates on an air film bounded by the other section in three degrees of freedom. Another GranaSAT member developed one of these systems, see in detail in [87]. Figure 3.13 shows that implementation.

3.2.2 Inertial platforms

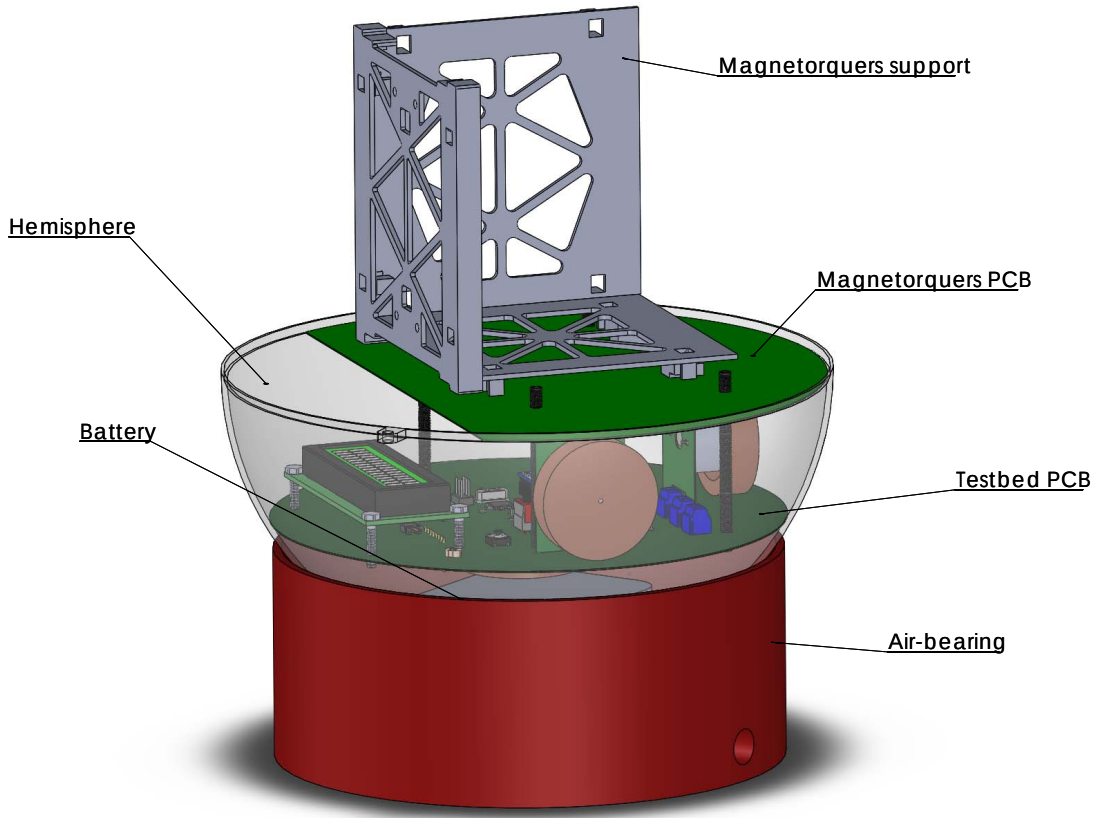
Another technology to simulate orbital dynamics is an inertial platform. It must allow CubeSat to rotate in three axes. Just as air bearings could be spherical or other type, inertia platforms design can also vary. One of the most extended is the so-called gyroscopic mechanical platform, because of its simplicity. Figure 3.14a depicts a 3D model example. The platform is designed and constructed to be adjusted to the CubeSatsize.



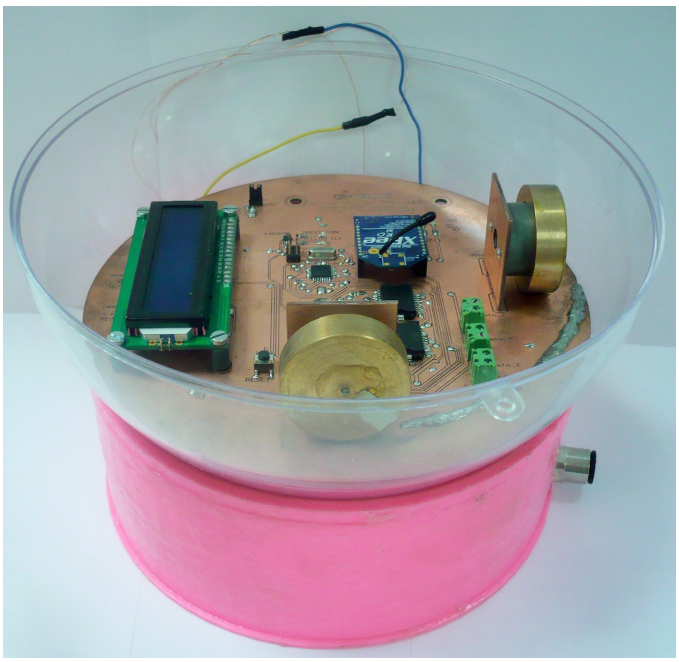
(a) *Gyroscopic mechanical inertia platform [84]* **(b)** *Rotational inertia platform [102]*

Figure 3.14 – *Inertial platforms examples*

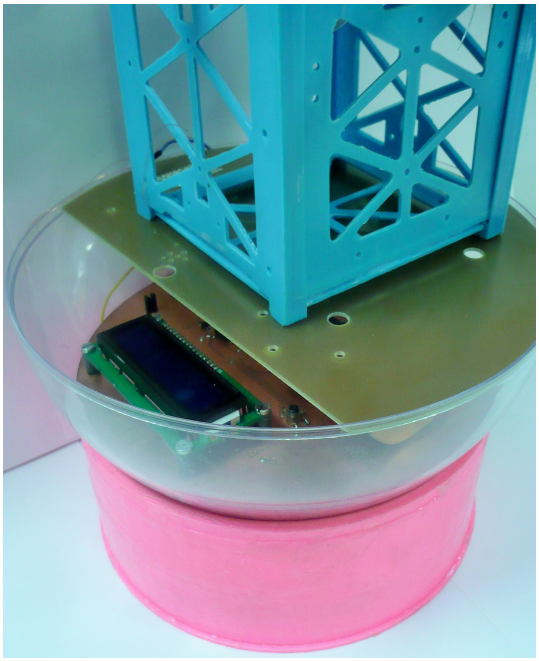
Another typical inertia platform is the **rotational** one. During testing, the CubeSat is aligned with the centre of the rotation axis of the board. Figure 3.14b shows another 3D model example.



(a) *3D model*



(b) *Front*



(c) *With CubeSat structure*

Figure 3.13 – *Air bearing testbed developed at GranaSAT [87]*

3.2.3 Formal Requirements Definition

As described when analyzing EDP, after an iterative refinement of the preliminary and technical requirements stated in previous chapters, the formal requirements can be defined. These are the ones which will allow the system to comply with the needs of the mission and therefore, satisfy the objectives of all the Stakeholders. This section is included for each subsystem. Once again, they are shown in tabular format, see Table 3.1.

Ref.	Formal Requirements
MP.FoR.1	The platform shall be 10x10x10 cm dimensions and have adequate attachment capabilities, in order to secure the CubeSat.
MP.FoR.2	The platform shall allow a consistent and balanced rotation movement, initiated by hand.
MP.FoR.3	The platform shall weight less than 3 kg without the CubeSat unit.
MP.FoR.4	The platform shall take up a surface inferior to 0.4 m^2.
MP.FoR.5	The platform shall have three pair of holes separated by 1.50 cm, in order to incorporate external elements. The nearest to the center shall be at least 20 cm far.
MP.FoR.6	The rotating part of the platform shall exhibit a diagonal inertia matrix.

Table 3.1 – *Mechanical Platform - Formal Requirements*

3.3 Ground Station Analysis

Every mission must count with a Ground Station. As described in section 2.2.2, in a real mission, it is in charge of communicating with the spacecraft, assuring the correct functioning of the system and so forth. As for this project, its tasks will be similar, but in the local environment designed for the simulation platform. Besides, the system must be expandable enough so it can be used in a real mission.

The concept Ground Station is so wide that this project cannot go into detail as it comprises a huge amount of equipment, software, procedures, communications system and so on. Instead, formal requirements will be defined based on the expected objectives and technical requirements stated. This is one of the examples in which this Master's Thesis must focus on its academic orientation; nevertheless, as mentioned, the designed Ground Station shall suppose the basis for future improvements which allow to comply with the requirements for a real mission scenario. Additionally, a couple of relevant standards are briefly introduced next.

3.3.1 Data-management standards in Ground Station

The packetization format of the information exchanged between the spacecraft and the CubeSat is also standardized. Indeed, not only the packet format is based on a standard but also the mechanism to exchange the definition of the control commands and telemetry itself. Both are addressed next.

3.3.1.1 Consultative Committee for Space Data Systems

This is a governmental organism founded in 1982 to develop standards for space data and foster interoperability. It is composed of the most important space agencies in the world, NASA, ESA, JAXA...

Figure 3.15 – *CCSDS logo*

One of its most important contributions is the so-called space packet procotol, typically known just as the organism, CCSDS. It specifies a communications protocol to

be used by space missions to transfer space application data over a network that involves a ground-to-space or space-to-space communications link. It is defined in [69], [68] and [70]. Figure 3.16 shows an example of CCSDS telemetry packet format.

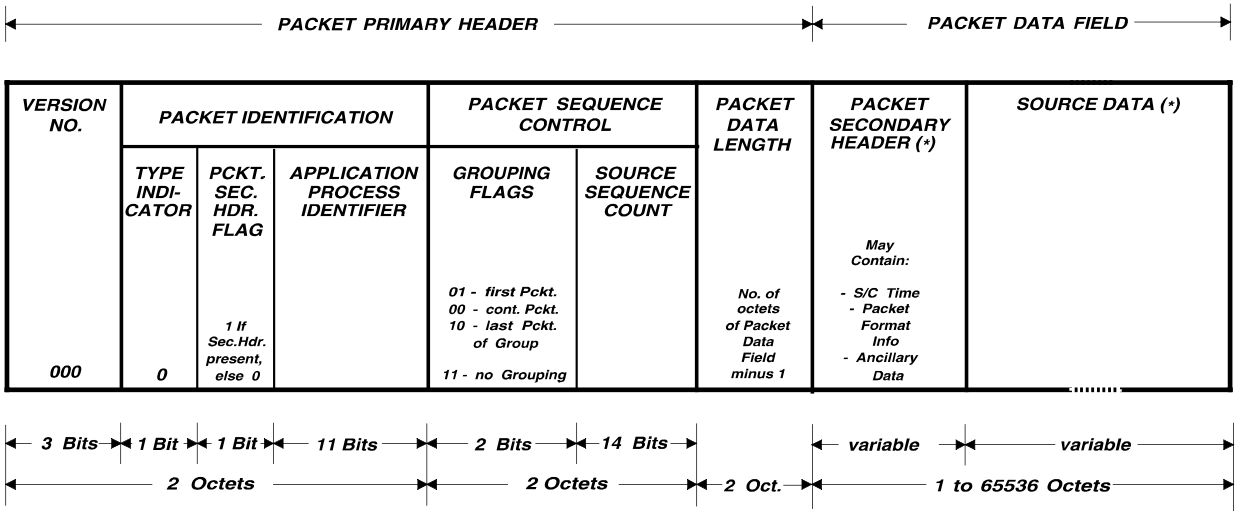


Figure 3.16 – *CCSDS format example [68]*

3.3.1.2 XML Telemetric and Command Exchange

XML Telemetric and Command Exchange is an XML based data exchange format and spacecraft telemetry and command meta-data [59]. For a given mission there are a number of lifecycle phases supported by a variety of systems and organizations; telemetry and command definitions must be exchanged among all of these phases, systems, and organizations.

A typical example of this process is between the spacecraft manufacturer and spacecraft-operating agency. The first defines the telemetry and command data in a format much different than the one used in the ground segment. This creates the need for database translation, increasing customization and probability of error. Using a common exchange format streamlines the process of transferring definitions from the satellite integrator to the operations team and between ground systems supporting the same satellite. This reduces the need to develop mission-specific database import/export tools and enables the reutilization of command and telemetry database tools [89]. XTCE has also been adopted as a recommendation by the CCSDS.

3.3.2 Formal Requirements Definition

Ref.	Formal Requirements
GS.FoR.1	The Ground Station shall have a centralized system CCSDS and XTCE compatible.
GS.FoR.2	The Ground Station shall allow connectivity with standard radio stations.
GS.FoR.3	The Ground Station shall be able to use both TCP or UDP for the local environment operation.
GS.FoR.4	The Ground Station shall have a text-based logging system which continuously registers events, along with timing information.
GS.FoR.5	The Ground Station shall allow connectivity with both IEEE802.3 and IEEE802.11 standards.
GS.FoR.6	The Ground Station shall count with a LED lamp exhibiting an irradiance of at least 200 200 W/m^2.

Table 3.2 – *Ground Station - Formal Requirements*

3.4 Simulation CubeSat Analysis

CubeSats were introduced at Subsection 1.2.1. In this section, the subsystems addressed in this Master's Thesis are profusely analyzed and just as done before, formal requirements are defined. It is important to keep in mind the big picture illustrated in Figure 1.8 when analyzing a CubeSat as a whole system. As it was specified at that point, this Master's Thesis will focus on OBC, ADCS and in the power source area of the EPS.

3.4.1 On-board computer (OBC)

On-board computer is probably the most important subsystem of a CubeSat. It is in charge of controlling the whole system, controls I/O and coordinates the different subsystems to successfully perform the tasks needed at every moment. As seen in 1.2.1, sometimes OBC inherently includes another subsystem such as **On-board Data Handling** (OBDH), while in others systems it tackles functions related to a different subsystem (e.g. TC&C) out of simplicity. In order to perform all these tasks, the OBC can be composed of more than one processing core, which can be used to free the main one. This wide scope makes this subsystem to require a strong effort of integration with different specialists.

Additionally, sensors can also be considered part of the OBC given that as a last resort it is in charge of processing the information received from them. Besides, a significant

number of them are physically placed at the OBC

Just as previous subsystems analyzed, OBC can be subdivided into different modules, which are broken down next.

3.4.1.1 Central Processing Unit

The Central Processing Unit can be seen as the microprocessor of a standard computer. It is typically an **embedded system** and the most complex module of the system. It works as interface with the rest of the modules and HID.

It is recommendable for this module to be as flexible as possible and to allow a wide range of programming possibilities. It must be able to deal with hardware interruptions, act as **Scheduler**, manage Write/Reading operations on memory, prevent the system from failures and recover it from faulty situations... in sum it is the global coordinator of the system.

Central Processing Units can be either a single SoC or a complete module COTS and in aerospace sector they are usually able to have a complete **Operating System** installed, normally Linux-based.

For all the reasons exposed, a high-level analysis of this subsystem is impossible due to the vast amount of possibilities. Therefore, although the main requirements are listed at 3.4.1.6 just as with the rest of the sections, to go deeper on the possibilities and distinctive features of this kind of modules requires the scope to be somewhat more constrained; that task will be performed and exposed in Subsection 4.3.2.

3.4.1.2 Co-processing Programmable Core

One of the featured characteristics of the cutting-edge designs in a variety of sectors is including a reconfigurable or programmable hardware module, which can be used to act as auxiliary processor, performing a variety of tasks. **Field-Programmable Gate Arrays** (FPGA) are usually used for this purpose. The task to be developed by this module is generally specified using some Hardware Description Language such as VHDL or Verilog.

An FPGA can be used to deal with any computable problem, featuring its speediness for some applications because of their parallel nature. Some of the common usages nowadays are aerospace, digital processing, wireless communications or image processing. All of those usages are potentially useful in a CubeSat like the one under development; that is the reason why Co-processing Programmable Core are increasingly implemented, its versatility allows to solve different problems with a single module.

This brief analysis will focus on FPGA because of their wide usage for this purpose, as mentioned before. The most common FPGA architecture consists of an array of **logic blocks** and routing channels with I/O capabilities. A certain application circuit must be mapped into an FPGA with the adequate resources, which may vary considerably. Generally speaking, a logic block consists of a number of logical cells which are typically composed of four input **Lookup tables** LUT, a **full adder** and **D-type flip-flop**. Figure 3.17 shows this structure. Also, as depicted in Figure 3.17, a clock signal is needed as most of the electronics inside of an FPGA is synchronous.

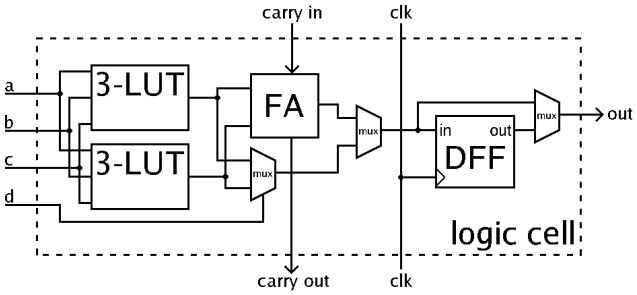


Figure 3.17 – *Typical logic cell [59]*

Among the concepts introduced before, **Lookup tables** (LUT) have special importance. They are arrays which replace runtime computation with simple array indexing which is time-saver in processing terms, as retrieving a value from memory is often faster than I/O operation [59]. LUT are in charge of storing the truth table of the any boolean function. It is a parameter sometimes used to estimate the *capacity* of an FPGA.

Regarding the programming, the user normally provides a certain design using some Hardware Description Language, as mentioned. A netlist is generated by a procedure called **synthesis**, which will be fitted to the FPGA architecture using a process called **place-and-route**. As a result of this process, the user will obtain a performance report exposed via timing analysis or simulation. Finally, the generated binary file is used to program the FPGA.

FPGA are particularly suitable for this Co-processing module because of their versatility; the variety of implementations possible is countless; for example, one of the most featured applications is **embedding a processor** inside an FPGA. It has many advantages, e.g.: specific peripherals can be chosen based on the application, mitigates obsolescence, reduces costs and allows impressive customization. For instance, there is a complete flexibility to select any combination of peripherals and controllers which can be directly connected to the processor's bus. This point allows meeting non-standard requirements, for example a COTS processor with ten UART may be impossible to find, but it is easy to implement in an FPGA embedded processor [18].

An embedded processor implementation example can be found on [38]. Another possible implementations using FPGA are generating digital interfaces such as I2C or generating PWM.

3.4.1.3 Communications

For obvious reasons, Communications subsystem or, more specifically, Telecommand & Telemetry (TC&C) are vital in CubeSats. Communications must be understood not only as the ones between the CubeSat and the Ground Station but also inside the CubeSat itself. Next, a brief analysis of the useful technologies for CubeSats is introduced.

3.4.1.3.1 Wired

Normally, wired connections are necessary even in CubeSats intended to fly to be able to program data, perform test and so forth. In this project, because of the simulation parcel, wired communications will be specially important.

3.4.1.3.1.1 Serial

This in an historical wired interface. Serial communication is based on sequentially sending data one bit at a time. Serial ports are typically identified as such which comply with RS-232 standard. One of the main drawbacks of serial communications is their slowness.

Although it is considered to be deprecated, it is still usual in many electronics systems, as debugging port or just as a contingency way of communication. In fact, these have could its usefulness in a CubeSat. Throughout years, serial ports have led to new derived technologies, such as USB. Figure 3.18 depicts DE-9 connectors, to be used with RS-232 standard.

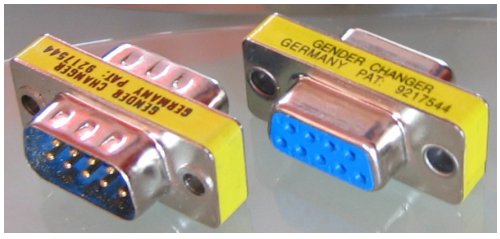


Figure 3.18 – *DE-9 connector [59]*

3.4.1.3.1.2 Universal Serial Bus

One of the most extended interfaces in industry. Derived from serial ports, they were initially intended to standardize the connection of peripherals, and finally have largely replaced its preceding interfaces.

USB has plenty of advantages, improved ease of use as it is self-configurable and hot pluggable. It is also much more faster than serial ports, reaching 5 Gb/s in the 3.0 version, one of the latest [59]. They are also extensively used as power ports. Figure 3.19 shows some of the standard connectors pinout.

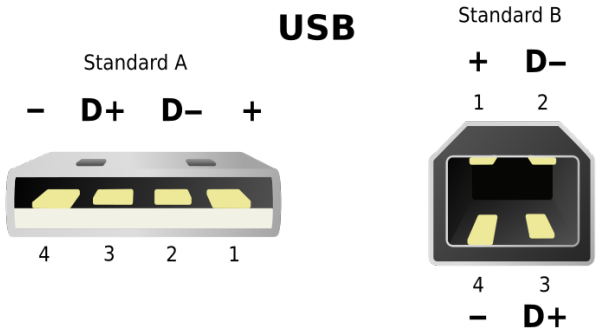


Figure 3.19 – *USB standard connectors pinout [59]*

Because of their versatility, USB interfaces may have numerous applications on CubeSats, from programming to power ports or to downloading data.

3.4.1.3.1.3 Ethernet

First standardized in 1983 as IEEE802.3 is by far the most used wired technologies in LAN. Systems communicating over this interface divide streams of data into **frames** which contain source and destination addresses and allow error-checking.

In CubeSats, Ethernet interfaces may be useful to allow communications using high-level protocols such as SSH, which ease communications and configuration of the system.

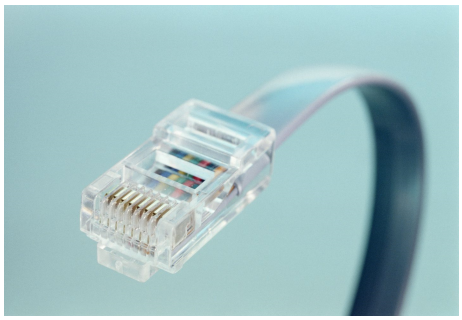


Figure 3.20 – *RJ45 connector [27]*

3.4.1.3.2 Wireless

Wireless communications are vital in real CubeSats, for obvious reasons. In orbit CubeSats communicates with Ground Station using radio-links and dedicated antennas. It is normally a whole subsystem within Communications system. It goes beyond the scope of this project analyzing that kind of implementations; extended radio-link analysis and calculation can be found on [71]. In this section, it will be briefly introduced local area wireless communications, which can be used as base for future developments focused on real long-distance links.

3.4.1.3.2.1 Radio-frequency

Communications over the air are the only choice when dealing with space communications. There is a variety of frequencies used in radio-links, depending on distance, power and numerous additional factors. Dependingo on the frequency used, the link will have certain particularities which shall be taken into account. Figure 3.21 schematically shows an example of radio path between a Ground Station based on Granada and ISS; it illustrates some of the aspects to be considered in a long-distance radio-frequency communication such as elevation angle or effective distance, see Figure 3.21.

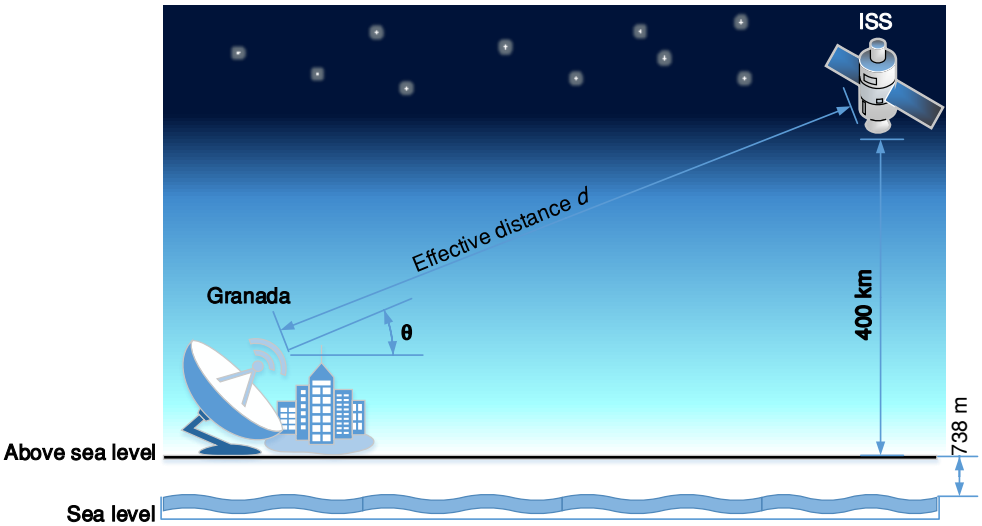


Figure 3.21 – *Radio-path basic elements [71]*

Although this project will not tackle this kind of communications, short-distance radio link may be considered useful for the Simulation Platform branch of the project. Contrary to the links designed for larger distances, short-distance communications over the air exhibits way lower complexity and cost due to the minor requirements in relation with power, losses constraints, among others.

3.4.1.3.2.2 IEEE802.11

Commonly known as Wi-Fi (a trademark from the Wi-Fi Alliance), it is based on IEEE802.11 standard and it is the most extended wireless communication protocol for local areas. Once again, this makes it specially suitable for this Simulation Platform because of its ease of use and extended use. On the other hand, it cannot be used for space radio-links.

Since its origins, back in 1997, the standard has gone through a significant number of reviews. Nowadays the most usual versions are IEEE802.11g/n/ac reaching data rates up to 1300 Mbit/s in the 5 GHz band [59].

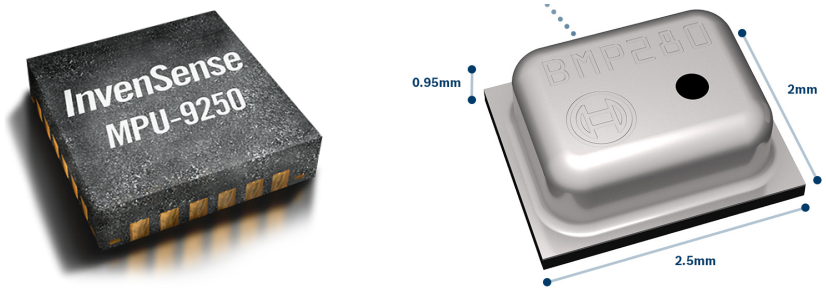
As a result of the wide use of IEEE802.11, the majority of the systems related to it are highly standardized (IC, antennas, amplification subsystems, etc.) which reduces complexity and eases integration.

3.4.1.4 Payload. Sensors.

The Payload is generally known as the amount of cargo capacity of an aircraft, including fuel and people. However, it may also refer to the equipment specifically intended to perform a certain mission while in orbit, for instance a camera or a Star tracker.

In CubeSats field, a common Payload example are **sensors**: they are typically a crucial part of any mission, not only as part of the Payload but also a necessary part for the correct functioning of the whole system. Normally they are some kind of electronic device which takes some measurement or perform a certain action depending on the inputs it receives. Some examples of sensors have already been treated before, see 3.4.2.2. Another possible sensors are: barometer, thermometer, magnetometer, lightmeter, tachometer... the list is countless. Many of them will be analyzed and implemented in Chapter 4. Figure 3.22 shows some examples of them.

While this analysis considers sensors as part of the OBC, given that eventually the data will be treated there, they can belong to a different subsystem such as ADCS, see section 3.4.2.2.



(a) *IMU* **(b)** *Barometer and thermometer*

Figure 3.22 – *Sensor examples [20]*

3.4.1.5 Flight Software. On-Board Data Handling

In charge of all the exposed, there must be a **governor**, a coordinator which controls the hardware and deals with the different stages and situations of a mission, this is the **Flight Software** (FSW). As in a significant number of the subsections analyzed through this Master's Thesis, Flight Software is such a large field that it allows a complete project focused on it. Therefore, hereby only a basic classification is introduced; this will be an important issue at the design stage and also for future improvements of the system.

As far as this project concerns, considered flight software can be subdivided into the following.

3.4.1.5.1 Non-real-time Operating System

Non-real-time Operating Systems is basically a general purpose OS to be used with personal computers, servers, etc. The main difference with RTOS is **determinism**. Non-RTOS are not deterministic as tasks will not run at a certain time and for a certain time, there are no guarantees for critical tasks, exhibits high latency because of using unpredictable virtual memory, as well as Jitter.

The previous characteristics translate into the use of **non-preemptive schedulers**. Examples of these kind of OS are the widely used Windows®, Debian, etc.

3.4.1.5.2 Real-time Operating System

In contrast with the latter, RTOS are completely deterministic, this is, how and when a task will run given whatever conditions defined for it to do so, it is **guaranteed**. They are intended to process data as it comes in, without buffer delays. There are a huge amount of systems nowadays which must use RTOS: cars, **spacecrafts**, avionics, critical systems... In sum, RTOS must be able to compute a task in a **limited or predictable amount of time** i.e., is time-bounded. However, this behaviour has nothing to do with processing speed but with a known deadline (a second, an hour or a month) and reduced Jitter.

From a technical perspective, this implies not to use virtual memory, strict scheduling (preemptive) and avoid non-deterministic elements. RTOS are usually much smaller than general purpose OS in order to ease maintenance and find sources of delay. One of the most used RTOS is **VxWorks** [57]; Linux non-preemptive kernel can also be **patched** to allow real-time behaviour.

Figure 3.23 – *VxWorks logo [57]*

3.4.1.6 Formal Requirements Definition

Ref.	Formal Requirements
OBC.FoR.1	The OBC shall have an I2C interface which allows communication with the rest of subsystems and components.
OBC.FoR.2	The OBC shall have an SPI interface which allows communication with the rest of subsystems and components.
OBC.FoR.3	The OBC shall have a programmable FPGA of at least 2500 LUT.
OBC.FoR.4	The OBC shall have one RJ-45 connector.
OBC.FoR.5	The OBC shall have at least two USB Micro-B device-ports and capability to deal with four.
OBC.FoR.6	The OBC shall be programmable with an external computer using a USB Micro-B connector.
OBC.FoR.7	The OBC shall have a LED system to check the correct functioning of the EPS
OBC.FoR.8	The OBC shall have at least two USB device-ports and capability to deal with four.
OBC.FoR.9	The OBC shall have a barometer with a resolution of at least 0.1 hPa.
OBC.FoR.10	The OBC shall have a thermometer with a resolution of at least 0.1° C.
OBC.FoR.11	The OBC shall have at least 8 ADC channels.
OBC.FoR.12	The OBC shall have 6 analog sun sensors, one per each face of the CubeSat.
OBC.FoR.13	The OBC shall have a Real Time Clock (RTC) which provides the whole system with time information.
OBC.FoR.14	The OBC shall have an SD card holder, to store telemetry data.
OBC.FoR.15	The OBC shall have an integrated IEEE802.11g device which allows wireless communications.
OBC.FoR.16	The OBC shall implement a linux-based RTOS.

Table 3.3 – *OBC - Formal Requirements*

3.4.2 Attitude Determination and Control System (ADCS)

The Attitude Determination and Control System (ADCS) (or AOCS) is the subsystem in charge of assuring that the CubeSat is correctly oriented and is able to deal with external torques and forces properly. It consists of a series of sensors and actuators, which work along with different algorithms, allowing proper pointing to the objective, like sun pointing (power) or receiving antennas pointing (communications). All these factors will be extensively treated along this and Chapter 4, but first it is necessary to analyze and put into context the need of this subsystem, as well as the basic concept regarding **Spacecraft dynamics** and mathematical tools required to deal with it.

3.4.2.1 Spacecraft dynamics

Spacecraft dynamics is again an extensive topic which cannot be addressed in a few lines in this project. Nevertheless, this section will introduce some of the most important concepts used in this Master's Thesis, vital for the design stage; particularly, rotation representation and concepts such as angular momentum or **inertia matrix** are briefly introduced. Out of simplicity, no demonstrations will be included and the analysis will be kept as simple as possible.

3.4.2.1.1 Rotation representations

The attitude of a three-dimensional body is most conveniently defined with a set of axes fixed to the body. This set of axes is generally a triad of orthogonal coordinates, and is normally called a *body coordinate frame*. The attitude of a body is thought of as a coordinate transformation that transforms a defined set of reference coordinates into the body coordinates of the spacecraft.

Below, the most important three-axis attitude frames are summarily exposed. For further details, see [88] and [109].

3.4.2.1.1.1 Direction cosine matrix

The basic three-axis attitude transformation is based on the direction cosine matrix, also called **attitude matrix**. Any attitude transformation in space is actually converted to this essential form. It has the important property of mapping vectors from the reference frame to the body frame, describing the transformation from coordinate system a to b.

This system has no singularities, which is its main advantage, but on the other hand, it supposes propagating nine elements (three unit vectors, each with three components) defining an orthogonal right-handed triad.

3.4.2.1.1.2 Euler angles

The Euler angle rotation is defined as successive angular rotations about the three orthogonal axes of the body frame. Typically, these are defined by i, j, and k, and those of the reference frame by I, J, and K. There is a multitude of order combinations by which the rotation can be performed. It is common to define the Euler **roll angle** (ϕ) as a rotation about the X body axis, the **pitch angle** (θ) about the Y body axis, and the **yaw angle** (ψ) about the Z body axis. However, any other definition is acceptable as long as it remains consistent with the analytical development. Figure 3.24 shows an illustration of this rotation representation system.

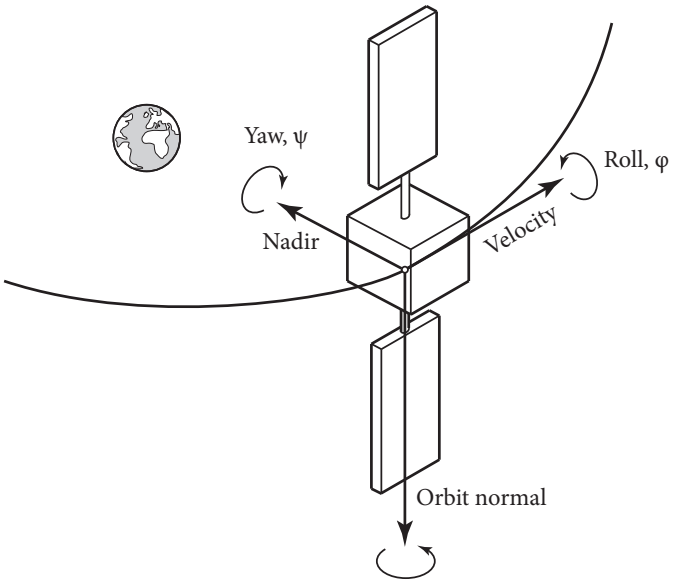


Figure 3.24 – *Three-axis Euler angles around CubeSat [17]*

Euler angles are intuitive and often relevant for requirements specification, but computer implementation is not straightforward and presents numerics singularities with 90° rotations, a phenomena called **gimbal lock**; it consists in the loss of one degree of freedom in a three-dimensional gimbal system, when the axes of two of the three gimbals are driven into a parallel configuration, *locking* the system into rotation and degenerating into a two-dimensional space [59]. Figure 3.25 illustrates this problem.

Besides, attitude rotations derived on the basis of Euler angles necessitate dealing with nine elements of the direction cosine matrix, and each element may include several trigonometric functions. These are some of the reasons for the wide use of quaternions, explained next.

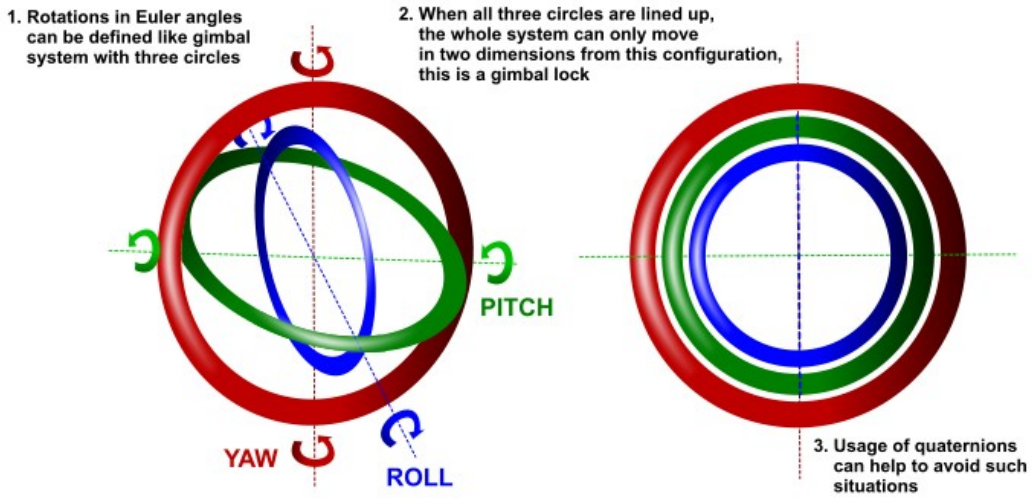


Figure 3.25 – *Gimbal lock pheonomena [106]*

3

3.4.2.1.1.3 Quaternions

Quaternions are a direct consequence of the properties of the direction cosine matrix. It can be shown with some linear algebra that a proper orthogonal 3x3 matrix has at least one Eigenvector with eigenvalue of unity. That eigenvector e_i has the same component along the body axes and along the reference frame axes. It can also be demonstrated that any attitude transformation by a series of consecutive rotation about the three orthogonal unit vectors of the coordinate system can be achieved by a single rotation about the eigenvector with unity value. The quaternion is defined as a vector as follows:

$$q = q_4 + iq_1 + jq_2 + zq_3 \tag{3.4.1}$$

With the unit vectors, i, j, k satisfying:

$$i^2 = j^2 = k^2 = -1 \tag{3.4.2}$$

The main disadvantages of quaternions are being non-linear kinematics and not very intuitive. However, they **have no singularities** and are easy to implement in software, reaching a great efficiency in computations; while representing the attitude of a body in a reference frame by a direction cosine matrix requires knowing nine parameters a_{ij}, quaternions only require four q_i parameters. Besides, the elements of the direction cosine matrix, in contrast to those of the quaternions, are trigonometric functions, which are much more cumbersome to compute. Figure 3.26 depicts the quaternions concept graphically.

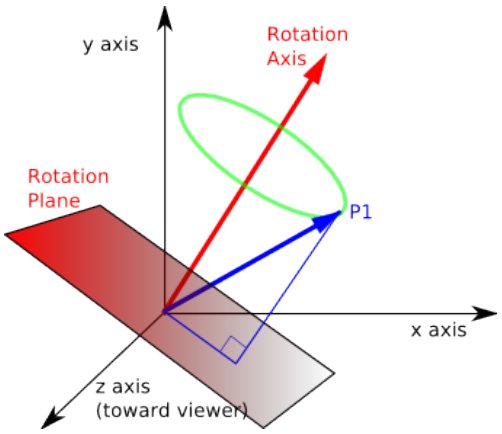


Figure 3.26 – *Quaternion graphical representation [16]*

3.4.2.1.2 Inertia matrix and angular momentum

These are concepts constantly utilized when dealing with physical designs, and are closely related to another basic concept: the **center of mass**; given a certain distribution of mass in space, it is defined as the unique point where the weighted relative position of the distributed mass sums to zero [59]. Figure 3.27 illustrates where the center of mass (C) locates in a block toy.

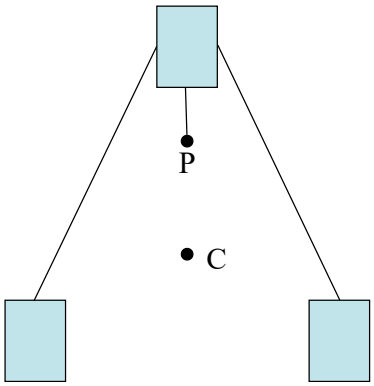


Figure 3.27 – *Center of mass example [59]*

On the other hand, the **moment of inertia** of a body determines the amount of Torque needed to reach a certain angular acceleration about a rotational axes. It depends on the body's mass distribution and chosen axis; the larger the moment, the greater the Torque needed to change the rotation rate. Depending on the number of axis of rotation, the different moments of inertia are arranged into a NxN matrix, with N that number of axis, mutually orthogonal; this is the so-called **inertia matrix**, and represents how the mass is distributed in the body. A desirable body design is that whose inertia matrix

is diagonal; non-zero values are called the **principal axes of inertia**. They are always symmetric.

These concepts are important in order to understand the basics of the rotational dynamics. Particularly, CubeSats attitude varies according to the fundamental equations of motion for rotational dynamics or **Euler equations**:

$$\dot{H} = T - \omega \times H \tag{3.4.3}$$

Equation 3.4.3 represents the equations for the conservation of the angular momentum, denoted by **H**. Angular momentum is the rotational motion of a body that will continue unless changed by a Torque, and it is calculated as the body's **moment of inertia** times its angular rate. It is clear from that expression that the body's angular momentum will remain constant in absence of external torques, **even if some parts of the body moves with respect to another**, for example, a reaction wheel (see subsubsection 3.4.2.4) spinning. Therefore, if that happens, the rest of the body will have to spin in the opposite direction in order to conserve the total angular momentum [94]. Further manipulations to Equation 3.4.3 allow understanding how attitude can change due to multiple causes, closely related to concepts as **inertia matrix**, among others, described before.

3.4.2.2 Inertial Measurement Units

If there is an element of particular importance in the ADCS, it is the **Inertial Measurement Unit** (IMU). They are electronic devices which measure gravitational acceleration, angular rate and orientation of the device, by using accelerometers, gyroscopes and magnetometers [59], normally with a set of these three per axis, see Figure 3.24. IMU are widely used in a variety of spacecrafts, including planes and satellites. However, they are also useful in everyday products such as mobiles phones or wearable devices, see Figure 3.28.

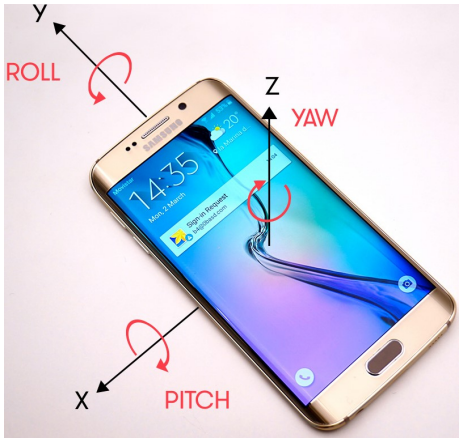


Figure 3.28 – *Mobile phones also count with an IMU [44]*

In spacecrafts field, an IMU is usually part of the Inertial Navigation System, which uses its measurements to calculate attitude or angular speed. Specifically, angular speed is normally integrated to get angular position. In combination with the gravitational acceleration given by the accelerometers, attitude can be estimated using predictors such as a Kalman filter. If that estimation is used to transform acceleration measurements into a inertial reference frame and integrated once, linear velocity can be achieved, linear position if integrated twice. As it could not be any other way, under this complex electronic system basic physics works, stated at Equation 3.4.4.

$$a_g(t) \rightarrow v_{\text{linear}}(t) = \int_0^t a_g(t)dt \rightarrow r_{\text{linear}}(t) = \int_0^t v_{\text{linear}}(t)dt \tag{3.4.4}$$

Figure 3.29 illustrates this process.

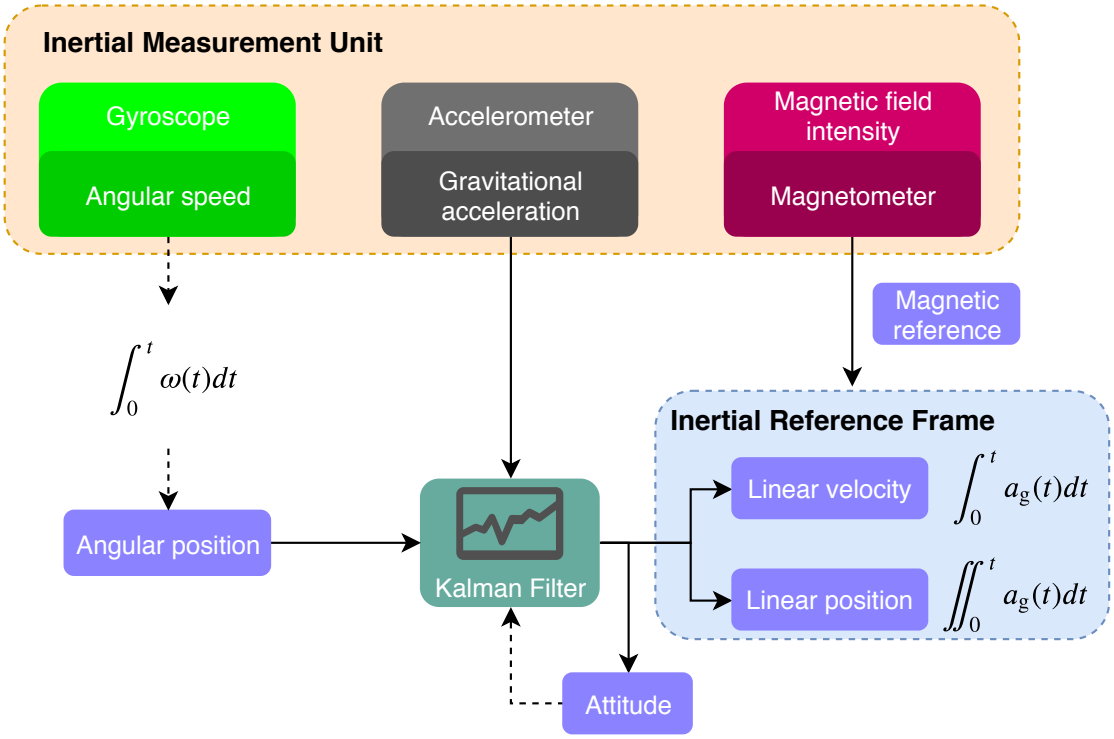


Figure 3.29 – *Position, velocity and attitude calculation process*

3.4.2.3 Control laws

As described before, the **attitude determination** process consists in combining available sensor inputs to provide an accurate solution for the attitude state as a function of time. The term **control law** is the name normally used in ADCS systems to refer to the **algorithms** which, using the data from the sensors, control the actuators available as needed in order to reach the target **attitude**. Figure 3.30 depicts this continuous process.

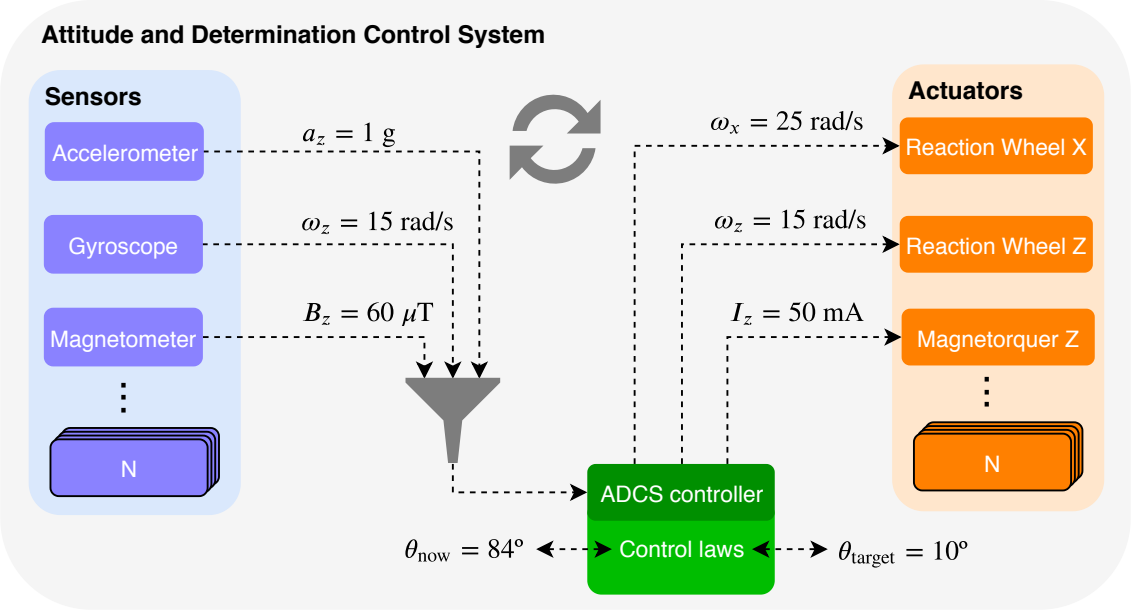


Figure 3.30 – *Control laws interaction with the rest of the CubeSat*

There is an enormous range of control laws, with different complexities and accuracies. Most spacecrafts use some kind of active control loop, as shown in Figure 3.30. It depends on the actuator used for the attitude maneuver (see section 3.4.2.4). For systems in which spacecraft rates will be small, 3-axis control can normally be decoupled into three independent axes. Two of the most used ones are briefly introduced next.

3.4.2.3.1 Proportional-Integral-Derivative Controller

In the simplest form, each axis of the CubeSat can be controlled by a **Proportional-Derivative controller**, with a control Torque given by Equation 3.4.5.

$$T_C = K_P \theta_E + K_D \omega_E \tag{3.4.5}$$

Where θ_E is attitude error angle and ω_E is the attitude rate error. The most important design parameter is K_P or proportional gain, representing the amount of control torque desired from a unit of attitude error. It determines the bandwidth of the system, closely related to the speed of response. To improve performance, sometimes an **integrator** is included, completing the Proportional-Integral-Derivative Controller or simply PID. The error signals are normally measured as **Euler angles**. One of the keys for a correct functioning is **tuning**; there are different methods in the literature to get an optimal tuning depending on the requirements.

3.4.2.3.2 B-dot Controller

It is an specific control law dedicated to the **detumbling** stage. Tumbling occurs right after deployment from the P-POD. In that moment, CubeSat is unusable because of the uncontrolled free spinning. Therefore, the first step before trying to control attitude using any control law such as the PID seen before, must be reaching a controlled rotation speed. That is the detumbling process.

B-dot is based on the usage of Magnetorquers, introduced in 3.4.2.4. They are used to generate a Torque opposed to the rotation of the CubeSat, by applying an alternating positive/negative current to the coil, which produces a certain magnetic moment. It only requires knowing the evolution of the magnetic field measured by the sensors. The functioning is simple, when the satellite is rotating around a given axis, progressively pointing in the same direction as the Earth's magnetic field, the magnetic field measured for that axis increases and so does the derivative of the magnetic field, being positive. As a response, an opposite signal to that derivative is sent through the coil, that is, a negative current. Once the magnetic field of that axis turns negative, the control signal is reversed, so the resulting torque is conserved. In sum, B-dot algorithm creates a magnetic moment in the opposite direction to the change in the magnetic field measured, expressed by Equation 3.4.6.

$$M_i = -k_i \dot{B}_i \tag{3.4.6}$$

Where i is one of the axes, M_i the generated magnetic moment, B_i the measured magnetic moment at that axis and k_i is the proportional constant calculated.

3.4.2.4 Actuators

CubeSats may implement a variety of actuators to be used to control attitude. The most important ones are **reaction wheels** and **magnetorquers**, both of which have been mentioned before. Next, they are briefly introduced.

3.4.2.4.1 Reaction wheels

Reaction wheels are a particular kind of flywheel used actively in attitude control. It is normally operated at a constant rotation rate which makes the CubeSat store a large angular momentum, tending to stabilize the satellite and allowing high pointing accuracy. They generate Torque by turning the wheel in the opposite direction of the rotation. Reaction wheels are usually controlled using PWM signals. They allow generating considerably more torque than Magnetorquers, however, current consumption is way higher and increases the weight of the CubeSat, as well as severely impacting the mass distribution, if it is not positioned correctly. Figure 3.31 shows two typical CubeSat reaction wheels.

Figure 3.31 – *Reaction wheels [8]*

GranaSAT counts with different reaction wheels designs from previous works, particularly [87]. Figure 3.32 shows the manufactured reaction wheel.

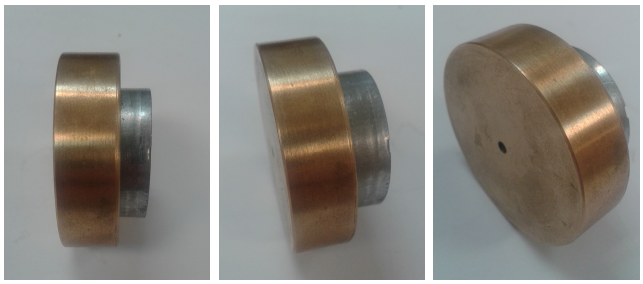


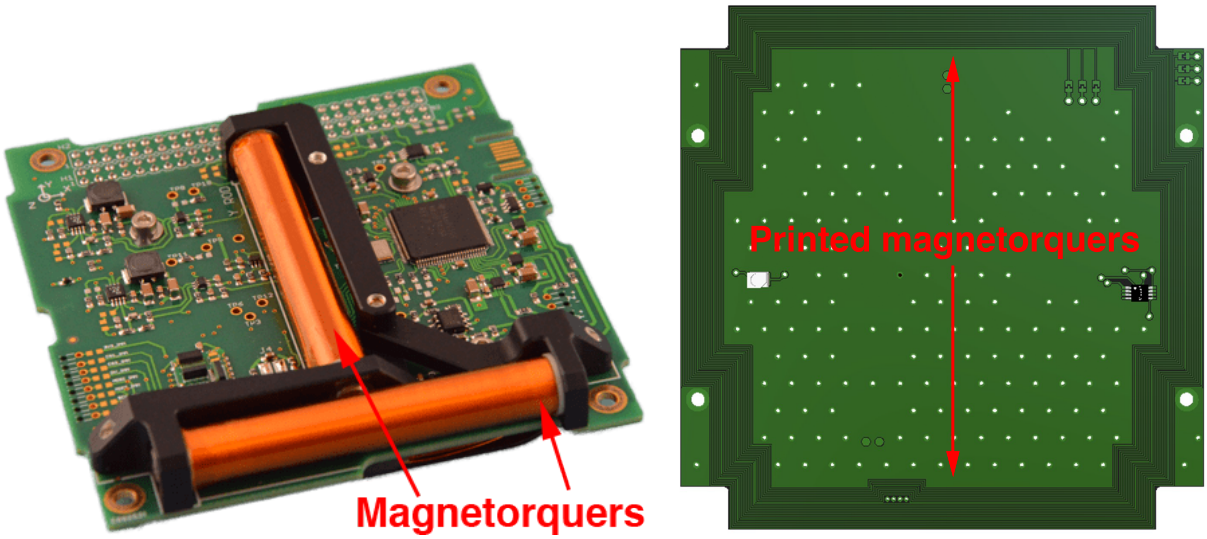
Figure 3.32 – *Reaction wheel manufactured in GranaSAT laboratory [87]*

3.4.2.4.2 Magnetorquers

Also known as torque rods, they are systems widely used to control attitude or going out of Tumbling. They are electromagnetic coils, arranged in different ways, which create a magnetic dipole and produces a certain Torque when interacting with Earth's magnetic field. Depending on the performance required and available area, it is built with a number of turns. They are reliable and energy-efficient but the Torque that Magnetorquers are able to provide is very limited. An important point is that the Torque can be generated only perpendicularly to the Earth's magnetic field vector. As advanced in 3.4.2.3.2, they are applied an alternating current depending on the desired attitude, producing a Torque τ given by Equation 3.4.7.

$$\tau = nIA \times B \tag{3.4.7}$$

Where n is the number of turns, I the current provided, A the area of the coil and B the magnetic field vector. Figure 3.33 shows two types of magnetorquers, the rod one, in which a copper wire is wrapped around a ferromagnetic core and an embedded coil based on the PCB design.



(a) *Rod format [24]* **(b)** *GranaSAT PCB design*

Figure 3.33 – *Magnetorquers*

3.4.2.5 Formal Requirements Definition

Ref.	Formal Requirements
ADCS.FoR.1	Shall have a functional IMU.
ADCS.FoR.2	Shall measure angular speed with an accuracy of at least 1°/s
ADCS.FoR.3	Shall measure gravitational acceleration with an accuracy of at least 0.1 g.
ADCS.FoR.4	Shall measure magnetic field intensity with an accuracy of at least 10 µT.
ADCS.FoR.5	Shall implement at least a PID control law.
ADCS.FoR.6	Shall count at least with a 4 cm diameter Z-axis reaction wheel.
ADCS.FoR.7	Shall include an adequate DC motor to accelerate the reaction wheel, between 3 V and 6 V supply.
ADCS.FoR.8	Shall allow generating at least 5 independent PWM signal with a minimum frequency of 10 Hz.
ADCS.FoR.9	Shall include terminals which allow connecting the system to external magnetorquers.

Table 3.4 – *ADCS - Formal Requirements*

3.4.3 Electrical Power System (EPS)

Electrical Power System (EPS) is the subsystem which provides, stores, distributes, and controls spacecraft electrical power. It is the most crucial subsystem, as a lack of energy in orbit would inevitably result in the end of the mission. Some of the most typical top-level EPS functions are listed below:

- Supply a continuous source of power during the mission life.
- Control and distribution of electrical power.
- Deal with both average and peak consumption requirements.
- Allow command and telemetry capabilities for EPS status, remote control, etc.
- Suppress transient voltages or spikes in the bus, which may damage the system.

Nowadays, the challenges around space power systems focus on maximizing

efficiency and reliability while minimizing mass and costs. In this section, a functional breakdown of a typical spacecraft EPS will be studied in detail in order to define technical requirements.

Once more, this project will address this part of the system from the double perspective mentioned (Figure 1.3). On the other hand, Figure 3.34 depicts the main components and functions of a typical EPS.

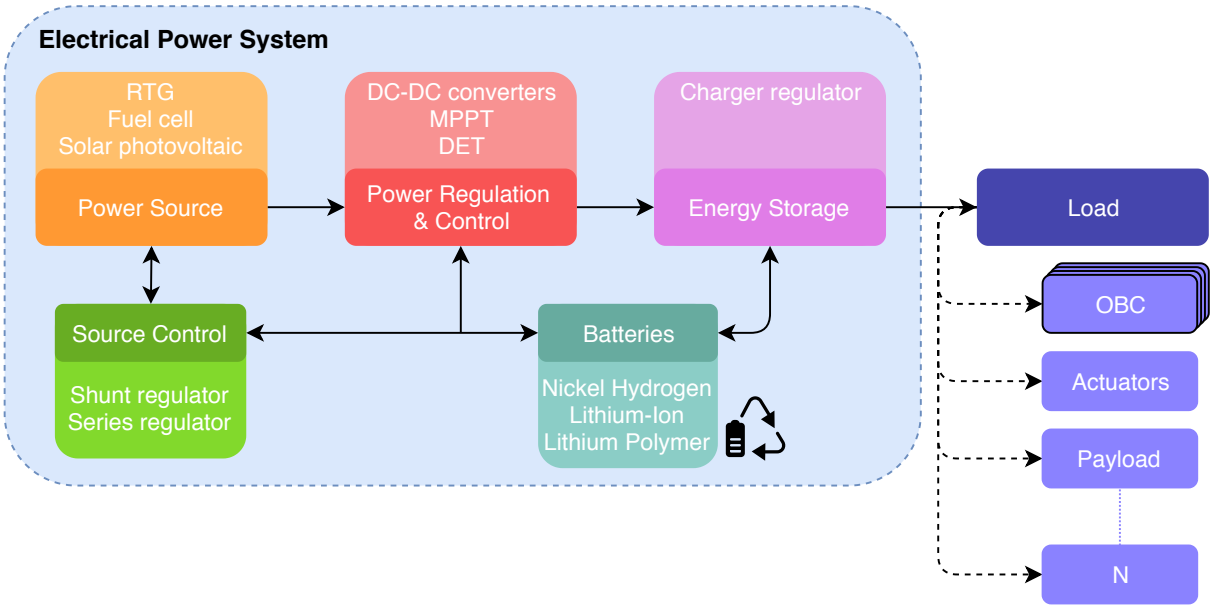


Figure 3.34 – *Electrical Power System main components and functions*

3.4.3.1 Power Source

As seen in Figure 3.34 power source is the first of the functions to be covered by EPS. As every electrical system, spacecrafts will require enough energy in order to function properly. Typically, when designing a real space system different power sources are considered depending on the energy magnitude needed for the mission, which in turn will depend on its expected duration.

Figure 3.35 illustrates that point, by plotting together the most usual power sources capabilities along with the expected duration of that technology. Some of the most common power sources depicted are briefly analyzed next, including some with an increasing interest in CubeSats sector in recent times.

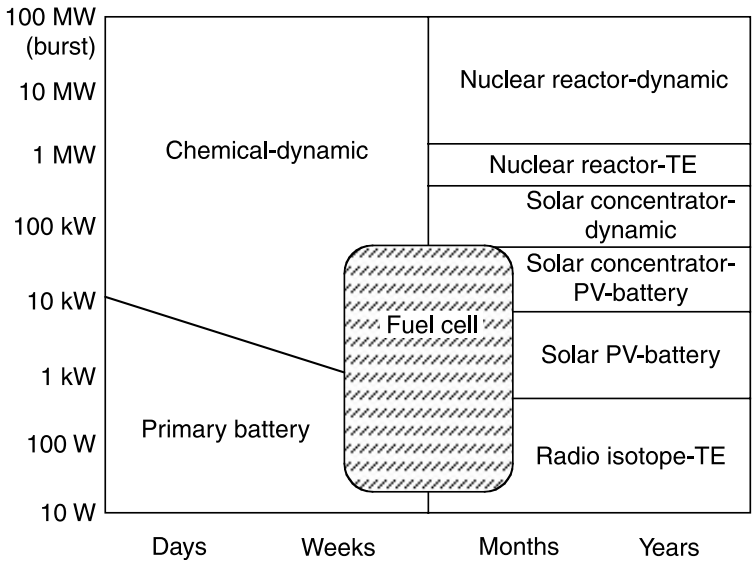


Figure 3.35 – *Optimum energy sources for various power levels and mission durations [103]*

3.4.3.1.1 Radio-isotope thermoelectric generator

Radio-isotope thermoelectric generators (RTG) can be considered similar to a battery and they are based on Seebeck effect. They are usually used when large voltages are needed in a unmaintained term longer than common batteries or fuel cells (see section 3.4.3.1.3), chemical-based, allow; therefore, they have been used as main power source in a variety of situations, from satellites to simply provide isolated facilities with electricity.

RTG, as most nuclear processes, make use of **thermocouples**, a device which is able to transform thermal energy into electrical energy, due to the Seebeck effect mentioned before; it makes this technology part of the so-called *static sources*. Two different semiconductors are mutually connected, flowing an electric current when there is a temperature gradient (produced by decay of the radioactive source) between the p-n junction of individual thermoelectric cells connected in a series-parallel arrangement to provide the desired DC electrical output from each converter. Figure 3.36 shows an example of thermocouples.

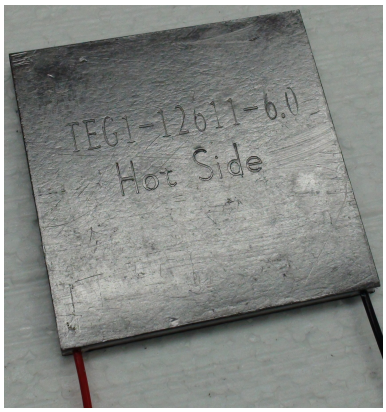


Figure 3.36 – *Thercouple [59]*

Normally, plutonium-238 is used, providing 0.54 Watts/g [100] and typically up to a few kW. RTG are really inefficient, varying between 3-7 %, and this technology is gradually falling into disuse. A notable curiosity is the numerous missions which flew to the moon using those systems[100]. Figure 3.37 shows a photography of one of these modules deployed on the Moon in Apollo 14 mission.

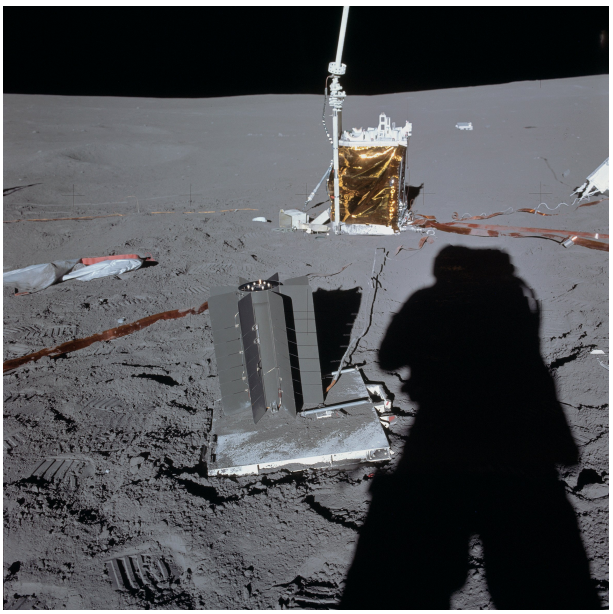


Figure 3.37 – *RTG used in Apollo 14 mission [59]*

3.4.3.1.2 Nuclear reactor

Nuclear reactor is an example of *dynamic source*, which uses a heat source and a heat exchanger to drive an engine in a thermodynamic power cycle. It is in charge of initiating a controlled self-sustained chain reaction which results into heat, transferred to a working fluid, which drives an energy-conversion heat engine. They are typically based on uranium-235 or plutonium-238 and one of its main advantages is avoiding thermal energy storage, as the source provides continuous heat.

With an average efficiency of 35 % and capability to produce up to a few hundred kW, it has been extensively used by Soviet Union and Russia, in contrast with the US.

3.4.3.1.3 Fuel cells

Fuel cells convert the chemical energy of an oxidation reaction to electricity, i.e., they are based on **REDOX reactions**, normally using hydrogen as fuel and oxygen as oxidizing agent [59]. Contrary to solar cells (see section 3.4.3.1.4) they can operate

without sunlight, but must carry their own reactant supply, which allows them to produce electricity continuously for as long as fuel and oxygen are supplied.

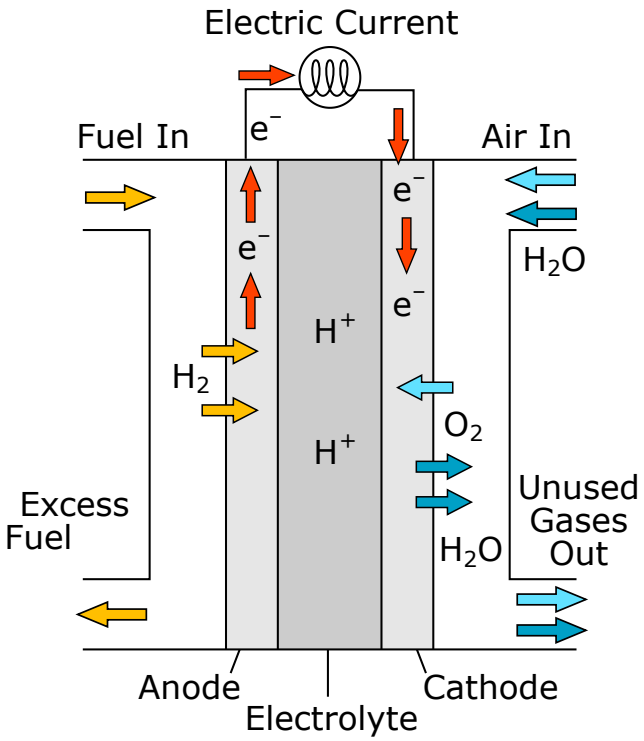


Figure 3.38 – *Fuel cell chemical basis[59]*

A typical single cell produces a voltage of 0.8 VDC, however, when combined, a fuel cell unit is able to generate up to tens of kilowatts of power with efficiencies as high as 80 %, with an average of 50-60 % for larger currents, still way higher than the rest of the power sources analyzed.

3.4.3.1.4 Solar Photovoltaic energy and Photoelectric effect

Despite of the variety of power sources commonly used in spacecrafts, so far the vast majority of CubeSats missions (although there are proposals pointing to another systems, see [72] or [101]) have utilized **Photovoltaic energy**, mainly because of size and weight constraints, as well as a lower energy requirements than bigger spacecrafts which use any of the others power sources possibilities. Therefore, with the aim of keeping this simulation platform as real as possible, it will be the one chosen. All the more reason, and contrary to the others, in this phase of the Engineering Design Process this technology is studied in depth in this section so the main drawbacks and constraints are outlined, in order to achieve more accurate formal requirements.

Solar Photovoltaic energy functioning is based on **Photoelectric effect**. It was first

described by Max Planck in 1900 and later in 1905 Albert Einstein went in depth, for which he was granted with the Nobel Prize. Basically, Photoelectric effect consists in the emission of electrons when light hits a material, with a variable kinetic energy depending on light frequency. The modern model which explains light behaviour was beyond the traditional conception of it as a wave, and proposed that light sometimes behaves as particles of electromagnetic energy called nowadays **photons**. According to Plank's equation:

$$E_{\text{photon}} = \hbar v \tag{3.4.8}$$

In Equation 3.4.8 E_{photon} is the energy of a photon, $\hbar$ is the Planck's constant and v is the frequency of the light in Hz. According to it, the energy of a photon is proportional to the frequency of the light [37]. Experimental tests show that if the incident light had a frequency lower than a certain frequency v_0 or **threshold frequency** no electrons were ejected regardless of the light amplitude, while for frequencies greater for v_0, they were. That value depends on the metal; Figure 3.39 illustrates this point.

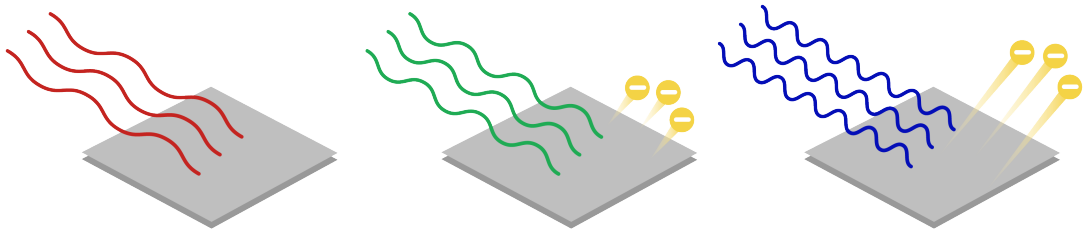


Figure 3.39 – *Threshold frequency in Photoelectric effect [59]*

In the example of Figure 3.39, the red light frequency is lower than threshold frequency of the metal, so no electrons are ejected. On the other hand, green and blue lights do produce this ejection, given its higher frequency when compared to v_0. Besides, the higher energy of the blue light makes electrons be ejected with a higher **kinetic energy** than green does.

As it will be seen next, and also during Chapter 4, this phenomena has a capital importance in functioning and performance of **solar cells**.

To conclude this subsection, Figure 3.40 depicts a duration comparison of the different technologies analyzed.

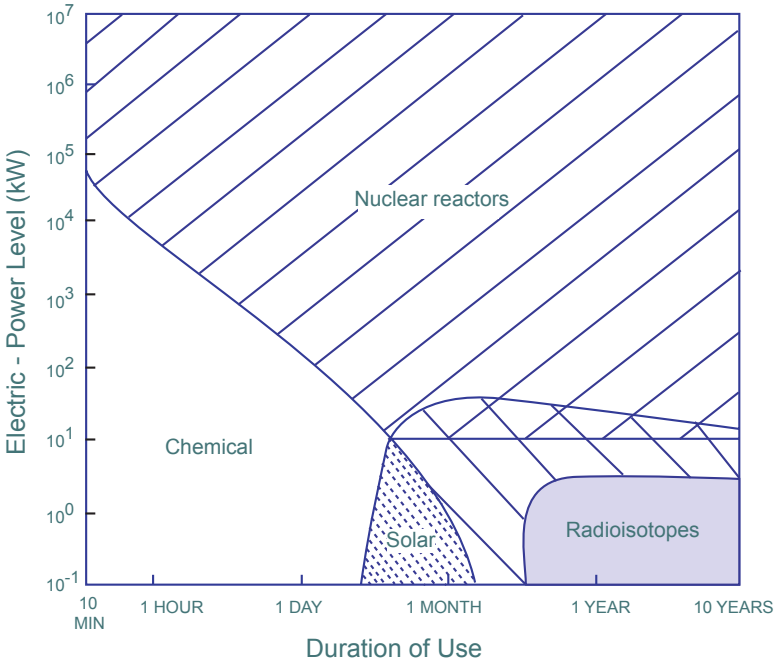


Figure 3.40 – *Duration comparison depending on technology [30]*

3.4.3.1.4.1 Solar cells

Solar cells are electrical devices that convert the energy of light directly into electricity due to the Photovoltaic effect previously described. They are composed of several P/N junctions monolithically connected in series. Briefly, in the n-type layer, there is an excess of electrons, and in the p-type layer, there is an excess of positively charged holes (which are vacancies due to the lack of valence electrons). Near the junction of the two layers, the electrons on one side of the junction (n-type layer) move into the holes on the other side of the junction (p-type layer). This creates an area around the junction, called the **depletion region**, in which the electrons fill the holes. Figure 3.41 illustrates this phenomena.

When all the holes are filled with electrons in the depletion zone, the p-type side of the depletion zone (where holes were initially present) now contains negatively charged ions, and the n-type side of the depletion zone (where electrons were present) now contains positively charged ions. The presence of these oppositely charged ions creates an internal electric field that prevents electrons in the n-type layer to fill holes in the p-type layer.

When light strikes a solar cell, electrons in the silicon are ejected, which results in the formation of 'holes' (the vacancies left behind by the escaping electrons). If this happens in the electric field, the field will move electrons to the n-type layer and holes to the p-type layer. When n-type and p-type layers are electrically connected the electrons travel

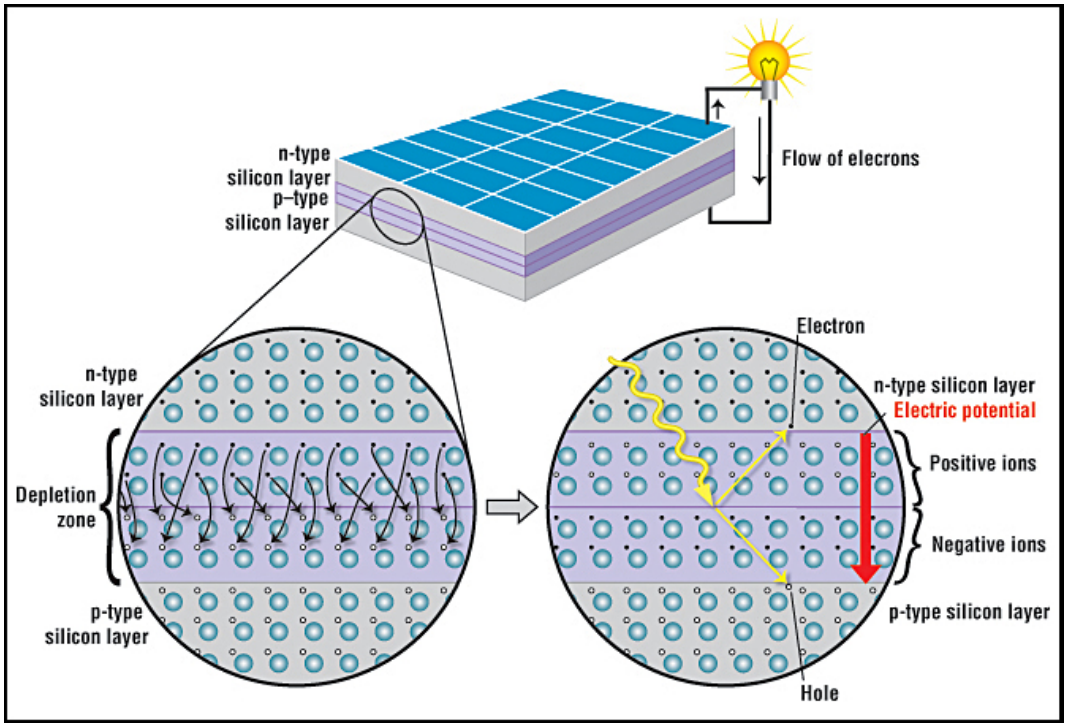
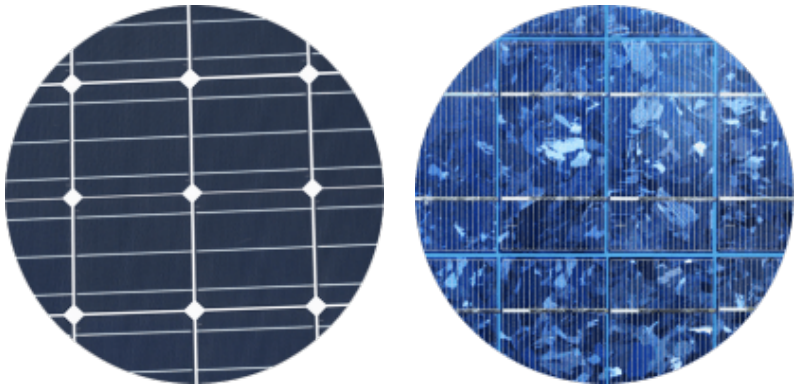


Figure 3.41 – *Solar cell functioning [21]*

from the n-type layer to the p-type layer by crossing the depletion zone and then go through the connection back to the n-type layer, creating a flow of electricity [21].

Although that is the basic behaviour, performance of solar cells varies depending on its type and material; the first noteworthy distinction is **Monocrystalline** and **Polycrystalline** Solar cells. While the first of them are composed of a single crystal which is fit into a solar panel, the second are made pouring the material into molds where it cools and solidifies. Monocrystalline are easily recognizable because of their darker colour and chopped off corners. They have a longer life compared to polycrystalline, which are also less space efficient than mono. Figure 3.42 shows an example of each of these solar cells.



(a) *Monocrystalline Solar Panel* **(b)** *Polycrystalline Solar Panel*

Figure 3.42 – *Visual aspect solar panels comparison [32]*

Another important division is considered between **single-junction** and **multi-junction** solar cells, depending on the use of one or more materials for the p-n junctions; the latter are composed of two or more junctions in layers one on top of each other, which allows to maximize the spectral range from which energy can be collected. This point is relevant because of the frequency dependency stated before, see Equation 3.4.8. Figure 3.43 graphically illustrates the advantages of this technique: by using different alloys with **Gallium**, the spectral range is increased and so does efficiency. This figure is also useful to compare the spectral response of this triple-junction solar cell with Sun spectrum, by plotting AM1.5 spectrum; the Sun as irradiance source will be analyzed in 3.4.3.1.4.2. Besides the ones mentioned, there are several more ways to classify solar cells (according to their generation, biohybrids, among others) but they are not relevant for this project.

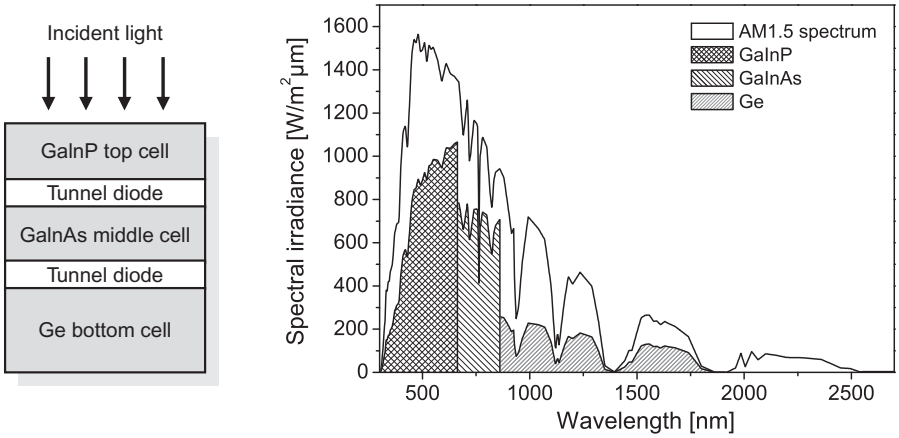


Figure 3.43 – *Structure of a GaInP-GaInAs-Ge solar cell. Spectral range covered [99]*

Finally, regarding manufacturing materials, while single-junction **silicon** with efficiencies about 24.7 % [99] are the most used, later techniques make use of materials such as **GaAs**, which due to its higher light absorption coefficient is much more efficient than silicon, specially when using multi-junction panels. However, they also have higher cost, which makes them appropriate only when high efficiency is needed, e.g. in space applications [59]. Particularly, **GaAs triple-junction** solar cells mentioned before are widely used in that sector, reaching efficiencies up to 30 % [64]. Of course, those efficiencies are related to high quality solar panels, considerably diminishing when low cost manufacturing techniques are used (see IB-3 [99]).

As a reference, Figure 3.44 shows the historical evolution of solar cells efficiency depending on the materials used, issued by the **National Renewable Energy Laboratory** in April 2019. It is worth mentioning the higher efficiencies accomplished under laboratory conditions (up to **46 %** using four-junction cells) when compared with the ones commercially available. For a more realistic perspective of the solar panels market nowadays, see Figure 3.45 [99]; it also shows some parameters which will be

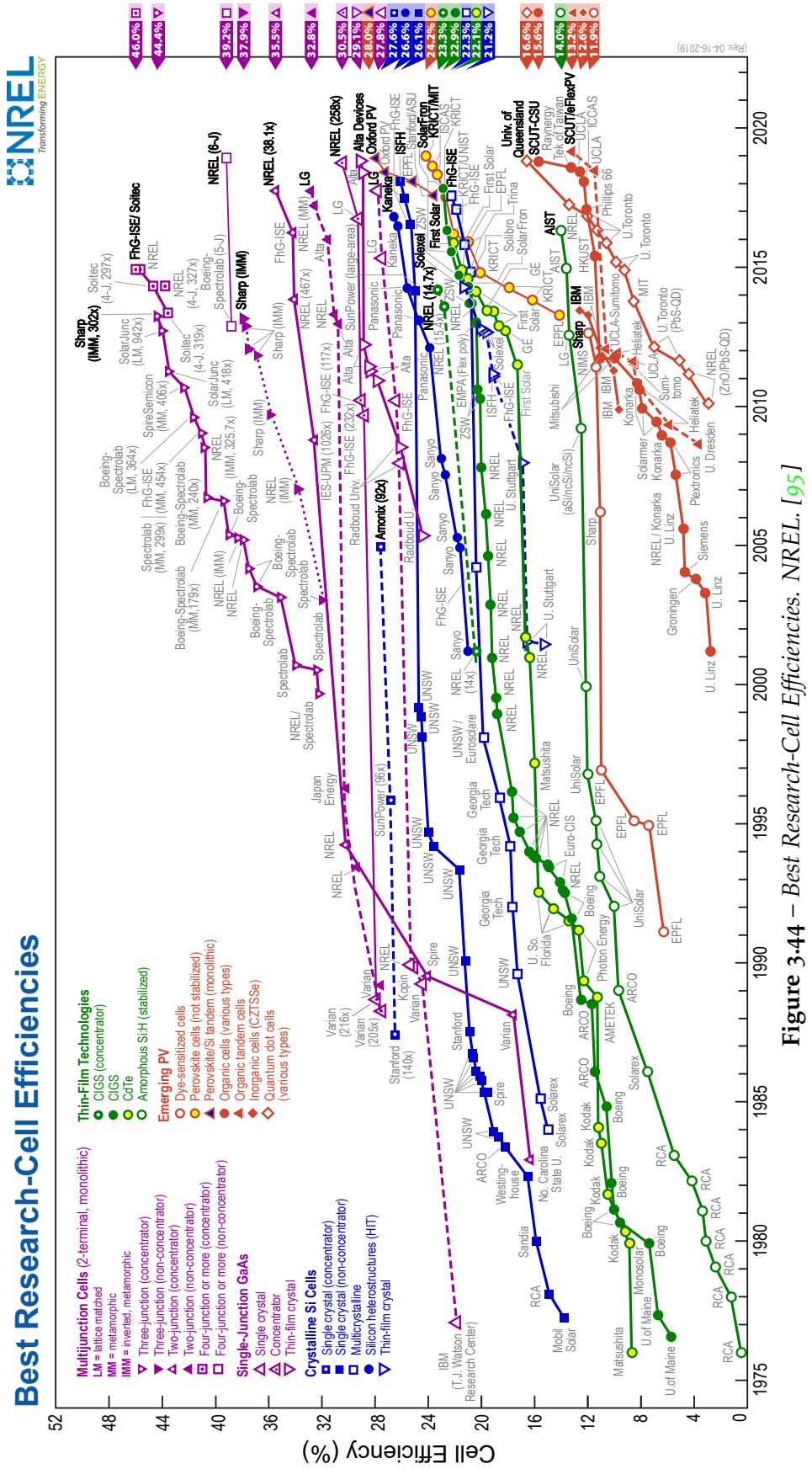


Figure 3.44 – *Best Research-Cell Efficiencies. NREL. [95]*

introduced later. Intensive research and the advance in techniques and materials assures a favourable future for photovoltaic technology.

	Efficiency (%)	J_{se} (mA/cm^2)	V_{oc} (V)	FF (%)
Crystalline: single junction				
c-Si	24.7	42.2	0.706	82.8
GaAs	25.1	28.2	1.022	87.1
InP	21.9	29.3	0.878	85.4
Crystalline: multijunction				
GaInP/GaAs/Ge tandem	31.0	14.11	2.548	86.2
Thin-film: single junction				
CdTe	16.5	25.9	0.845	75.5
CIGS	18.9	34.8	0.696	78.0
Thin-film: multijunction				
a-Si/a-SiGe tandem	13.5	7.72	2.375	74.4
Photoelectrochemical				
Dye-sensitised TiO_2	11.0	19.4	0.795	71.0

Figure 3.45 – *Typical optimum performance with the most common materials [99]*

Regardless of materials and manufacturing process, from a modelling point a view, a solar cell can be initially considered as a current source in parallel with a couple of diodes and a pair of resistors, as shown in Figure 3.46.

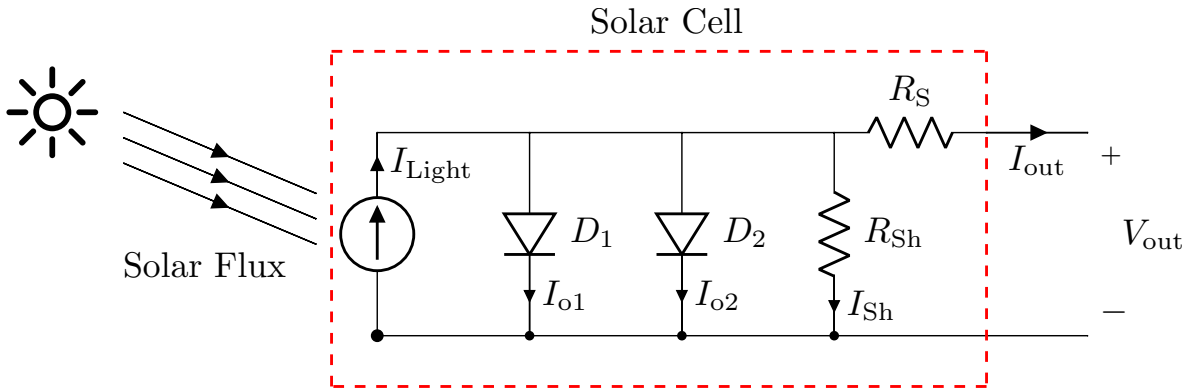


Figure 3.46 – *Solar cell circuit model*

Photons in striking light are absorbed by a semiconducting matrial, such as silicon, producing a current proportional to the amount of illumination, and adjusting the output voltage as necessary to provide that current. This is by definition the behaviour of a constant current source, so in Figure 3.46 is represented by the current source (I_{Light}),

which is diminished by parasitics effects given by diodes and resistances (I_{o1}, I_{o2}, I_{Sh}), which will be studied later. Let us analyze the circuit; by inspection, the current flowing out (I_{out}) is given by:

$$I_{out} = I_{Light} - I_{o1} - I_{o2} - I_{Sh} \tag{3.4.9}$$

Current through diodes can be substituted by **Shockley diode equation**, while I_{Sh} is found using Ohm's law. Therefore, equation Equation 3.4.9 yields to Equation 3.4.10.

$$I_{out} = I_{Light} - I_{o1}(e^{qV_{out}/kT} - 1) - I_{o2}(e^{qV_{out}/2kT} - 1) - \frac{V_{out} + I_{out}R_S}{R_{Sh}} \tag{3.4.10}$$

On the other hand, when the solar cell is short-circuited:

$$V_{out} = 0 \Rightarrow I_{out} \approx I_{Light} = I_{SC} \tag{3.4.11}$$

So finally, there are four currents: I_{SC} is the **short-circuit current**, I_{o1} and I_{o2} are **dark saturation currents**, and I_{Sh} is the **shunt current** due to ohmic losses. While the latter, as said before, is given by Ohm's law, the three first are analitycally given by complex expressions which go beyond the scope of this work (detailed demonstrations can be found at [63]); however, they are not needed in order to understand the basic functioning of a solar cell operation. Briefly, the short-circuit current is the sum of the contributions from each of the three regions: **the n-type region, the depletion region, and the p-type region**. On the other hand, the dark saturation current is the current generated due to recombination in the quasi-neutral regions (apparently neutral regions where electric field is zero). Hence, D_1, in parallel, represents the recombination current in the quasi-neutral regions ($I_{o1} \propto e^{qV/kT}$), while D_2 represents recombination current in the depletion region ($I_{o2} \propto e^{qV/2kT}$); a common and reasonable assumption is to ignore the dark current due to the depletion region I_{o2}. Figure 3.47 shows at the front a graphical representation of Equation 3.4.10 for typical values, while at the background the corresponding Power-Voltage curve.

At small applied voltages, diodes currents (I_{o1} and I_{o2}) are negligible and the solar cell behaves as a constant current source with an output current equivalent to the short-circuit current, I_{SC}, as stated in Equation 3.4.11. When the applied voltage is high enough so that diodes currents (recombination current) become significant, the solar cell current drops quickly.

Finally, shunt current is due to ohmic losses, modelled with a couple of resistors with a varying influence: as can also be seen in Equation 3.4.10, shunt resistance R_{Sh} has no

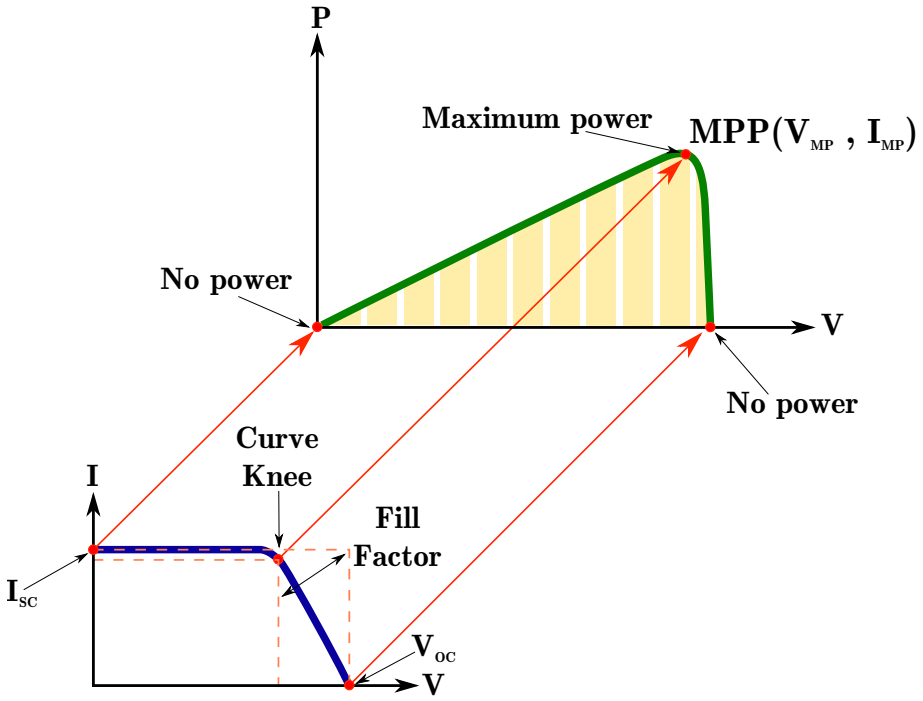
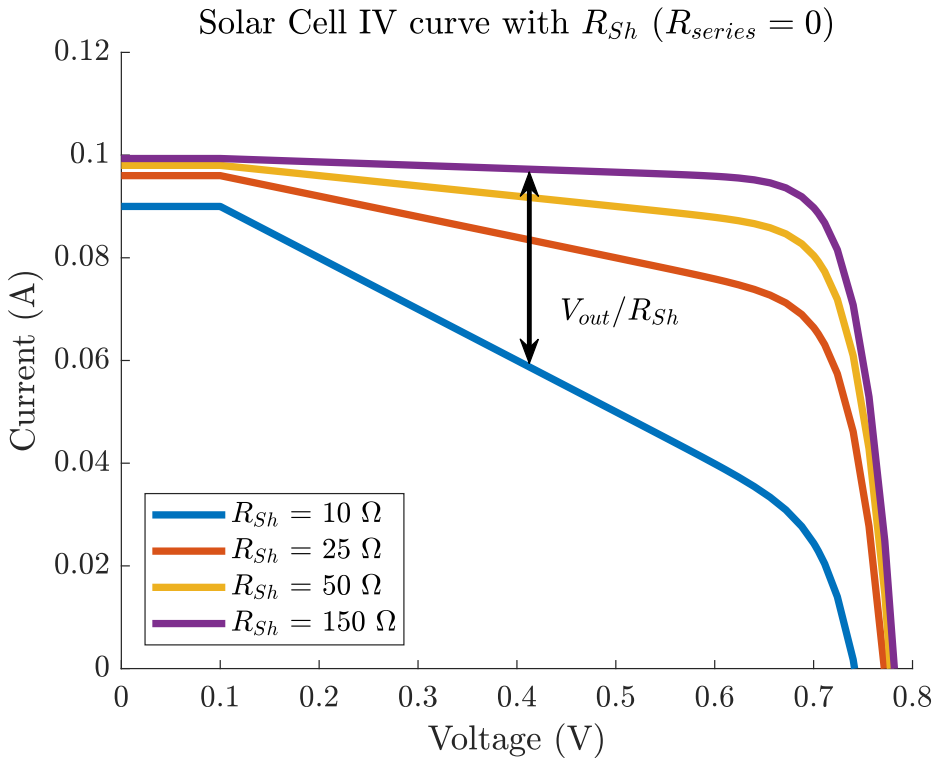


Figure 3.47 – *Typical IV curve for standard solar cell*

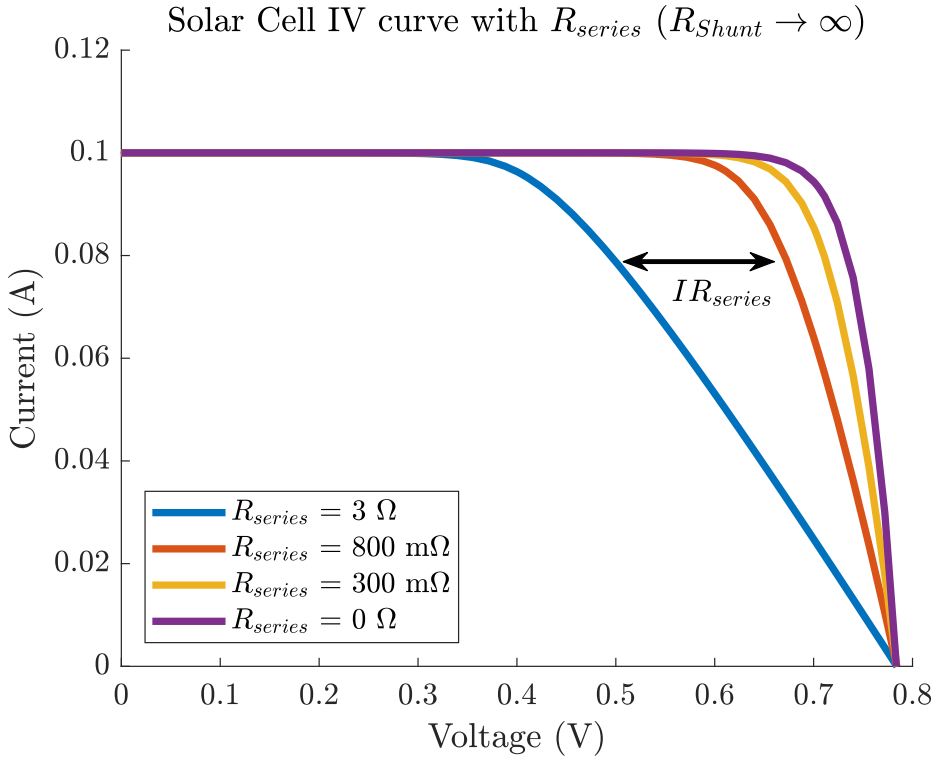
effect on the short-circuit current, but reduces the open-circuit voltage, V_{OC}. Conversely, the series resistance R_S has no effect on the open-circuit voltage, but reduces the short-circuit current, I_{SC}; sources of series resistance include the metal contacts, particularly the front grid, and the transverse flow of current in the solar cell emitter to the front grid. This phenomena is illustrated in Figure 3.48 which shows the behaviour of a solar cell depending on parasitics resistances; in 3.48a, shunt resistance value is varied assuming that series resistance is zero, while in 3.48b, it is performed an analysis the other way around. Indeed, it is immediate to see how shunt resistances shift open-voltage circuit with respect to no shunt resistor, while series resistances shift short-circuit current. Notice that some extreme values have been used to illustrate the phenomena.

Going back to Figure 3.47, it illustrates several important **figures of merit** for solar cells, which help to understand its behaviour, some of which have already been tackled.

- **Open-circuit voltage (V_{OC}):** voltage V_{out} across the output terminals when the cell is operated at open circuit. It is not possible to extract any power from the cell at this point.
- **Short-circuit current (I_{SC}):** current I_{out} at the output when the cell is operated at short-circuit. It is not possible to extract any power from the cell at this point.
- **Maximum power point ($MPP(V_{MP}, I_{MP})$):** point on the I-V curve where the power produced is at a maximum. For any given set of operational conditions, cells have a single operating point where the values of the current and voltage of the cell



(a) *Depending on* R_{Sh} ($R_{series} = 0$)



(b) *Depending on* R_{series} ($R_{Sh} \to \infty$)

Figure 3.48 – *I-V Curves analysis with parasitics effects*

result in a maximum power output. These values correspond to a particular load resistance given by Ohm's law. This point defines a rectangle whose area is the largest for any point on the I-V curve, given by $P_{MP} = V_{MP}I_{MP}$. The maximum power point is found by solving: [63]

$$\frac{\partial P}{\partial V}\Big|_{V=V_{MP}} = \frac{\partial(IV)}{\partial V}\Big|_{V=V_{MP}} = \left[I + V\frac{\partial I}{\partial V}\right]\Big|_{V=V_{MP}} = 0 \tag{3.4.12}$$

The current at the maximum power point, I_{MP}, is then found by evaluating Equation 3.4.10

- **Fill factor (FF):** measure of the *squareness* of the I-V curve, given by the ratio of the areas of the two rectangles shown in figure Figure 3.47, always lower than one. Mathematically:

$$FF = \frac{V_{MP}I_{MP}}{V_{OC}I_{SC}} = \frac{P_{MP}}{V_{OC}I_{SC}} \tag{3.4.13}$$

- **Packing Density:** refers to the area of the panel which is covered with solar cells compared to that which is empty. It affects the output power of the module as well as its operating temperature; it mainly depends on the shape (round, squared...) of the solar cells used. It is usually taken to be about 0.8.

- **Power conversion efficiency (η):** relates the power obtained at the maximum power point with the incident power P_{in}, where the incident power is determined by the properties of the light spectrum incident upon the solar cell:

$$\eta = \frac{P_{MP}}{P_{in}} = \frac{V_{MP}I_{MP}}{P_{in}} \tag{3.4.14}$$

Obviously, the goal of a system getting energy from a solar cell, will be operating at **Maximum Power Point**. Apparently, it should not a problem: as illustrated in Figure 3.47, every solar cell will have a certain point in which power is maximized, according to Equation 3.4.12; it may be higher or lower depending on the quality of the solar cell (i.e, parasitic effects importance, as seen before), but it is just a matter of finding it. The problem is, reality is not that simple; in a real scenario, there is a limit to how much voltage a certain panel can produce, i.e., it will act as a constant current source only as far as the connected load allows a voltage below that maximum. Now, let us suppose the illumination changes: according to the stated, for a given connected load, output current will vary, so does voltage; if for that load, the generated current tries to impose a certain voltage **above** the maximum the panel is able to produce, the **knee** of the curve is reached and generated power will turn unstable (see Figure 3.47).

Using the concepts exposed, the latter means that *MPP* will move constantly through the I-V curve, particularly depending on two factors: **irradiance** and **temperature**. While the first of them is obvious (as stated before, the more lightning striking the solar cell, the more current), the second is due to the temperature dependence of the intrinsic carrier concentration. It can be demonstrated ([63]) that temperature rising yields to an increase of it, which then increases the **dark saturation currents**. Because open-circuit voltage is proportional to the reverse of them, it will decrease. This is usually **the most important performance loss.** Figure 3.49 shows graphically this degradation in performance.

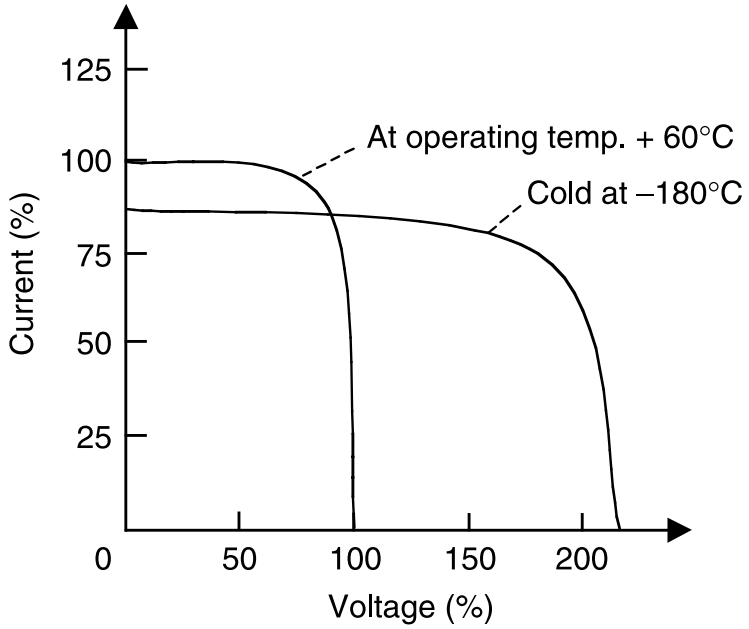


Figure 3.49 – *Standard I-V curve depending on extreme temperatures [103]*

It is immediate to see the degradation mentioned, which must be adequately taken into account when dimensioning a photovoltaic system. On the other hand, it is worth to mention the efficiency increase of a solar cell when exposed to extremely low temperatures; indeed, it is coherent with theoretical behaviour explained before: the lower the temperature, the lower dark saturation currents will be, and with them, the larger open-circuit voltage, as precisely shows Figure 3.49.

Another significant reason for performance degradation is the point in cell's life, that is, generated power is expected to be lower at its **End Of Life (EOL)** than at its **Beginning Of Life (BOL)**, as shown in Figure 3.50.

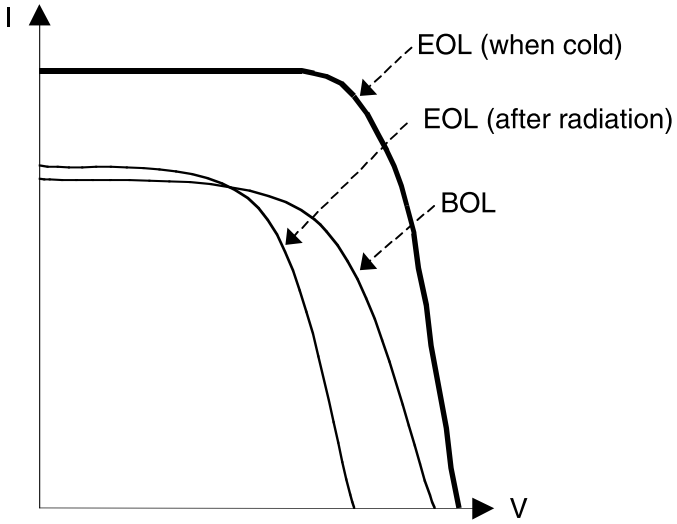


Figure 3.50 – *I-V curves with cell's life [103]*

Finally, the last noteworthy reason of performance decrease is **radiation**. The impinging particles produce defects in the crystalline structure of the PV cells. The resulting defects degrade the voltage and current outputs of the cell. Low-energy particles create damage close to the surface, and therefore lower the open circuit voltage. On the other hand, high-energy particles penetrate deeper in the base and lower the lifetime of electron hole pairs, thus decreasing the short circuit current. Figure 3.51 plots an I-V typical curve affected by different radiation doses accumulated during its life service.

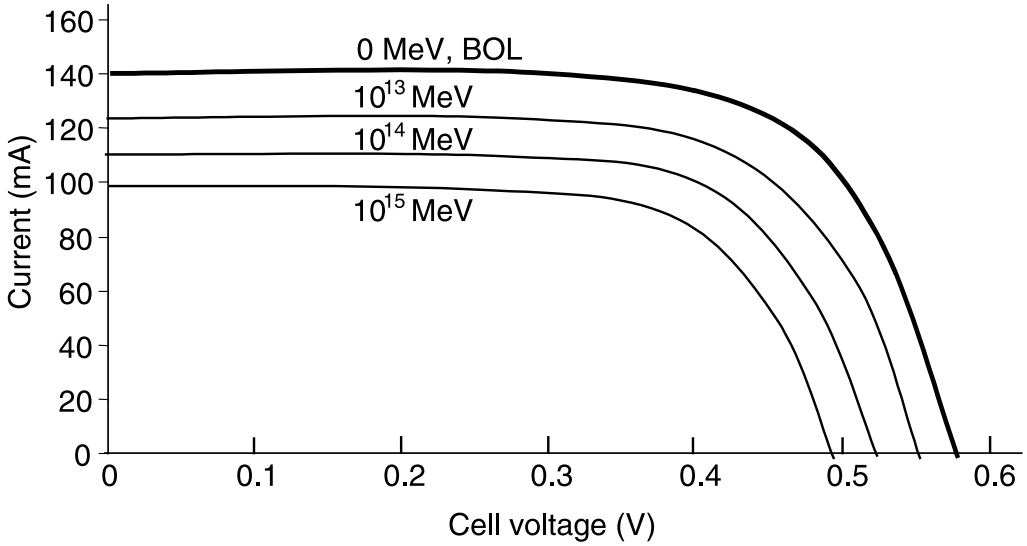


Figure 3.51 – *I-V Curve degradation with radiation doses [103]*

The power generation capability continues degrading as the radiation dose accumulates over the years [103]. Figure 3.52 depicts the consequences of radiation, showing that, although it oscillates, power generation have a decreasing tendency with time.

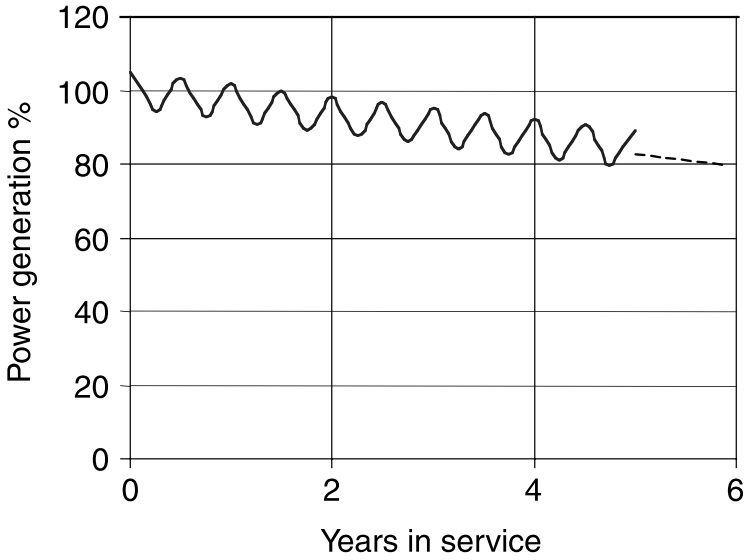


Figure 3.52 – *Losses throughout the years [103]*

To finalize this comprehensive analysis of solar cells, there is another vital point in solar cells response, apart from the systems to optimize delivered power, intrinsic efficiency of the solar panels and so forth; this is **spectral response**. Indeed, solar cells behave differently depending on manufacturing materials, given that each one may cover different sections of the **spectrum**, affecting to efficiency and extracted power, as mentioned before. Figure 3.53 shows spectral response of a typical solar cell manufactured with the most usual materials nowadays, some of which have been mentioned before. I must personally express my gratitude to Dr. Hamadani, from the **National Institute of Standards and Technology** in the US, for providing me with the data shown; additional information on that study can be found on [90].

As stated in the legend, solar cells made of different materials are characterized: average silicon Si, high efficiency silicon, gallium arsenide GaAs, cadmium telluride CdTe, copper indium gallium selenide CIGS and organic solar cell. Although some of them have not been mentioned in this Master's Thesis because of its low utilization in space sector, they are included as general reference for photovoltaic technology. As plotted, while the organic solar cell presents the shorter range **350-630 nm** the solar cell manufactured with high efficiency silicon expands its spectral response between the **350-1150 nm** range, clearly the best out of the analyzed. It allows us to conclude that even the material which shows the best spectral response by its own, behaves worse than multi-junction architectures which in the end cover a larger portion of the spectrum.

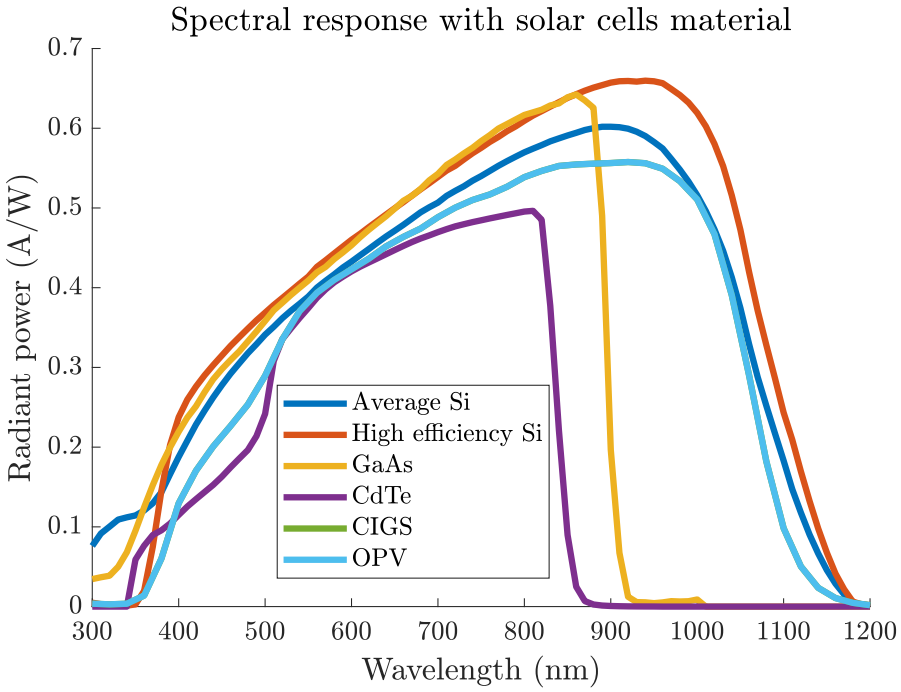


Figure 3.53 – *Spectral response depending on solar cells material*

3.4.3.1.4.2 Irradiance sources

As explained in previous section, spectral response of solar cells depends on the manufacturing material. In the same way, for a certain material, maximum efficiency will be accomplished only when the **whole spectral range of the solar cell** is covered by the irradiation source. This is usually not a problem when dealing with Sun spectrum (see AM1.5 series in Figure 3.43) as current technology allows a reasonable covering of it; however, it may not satisfy the simulation platform under development in this project, as the Sun is not likely to be used as irradiation source indoors. Therefore, it is worth considering whether solar cells behaviour will be the same when excited with a non-solar illumination source. Through this section, different irradiance sources, including the Sun, will be analyzed and compared with typical solar cells expected response.

- **Sun**

The Sun is, for obvious reasons, the most common irradiance source when dealing with photovoltaic energy. Solar irradiance is the energy emitted by the Sun as a result of its nuclear fusion reactions in the range 250-2500 nm. The Sun radiates approximately as a Blackbody at an effective temperature of 6000 K, with a total amount of energy **above the atmosphere** of about $1366\,W/m^2 \pm 6.9\%$ depending on the varying distance from the sun. That irradiance is distributed along a wide portion of the spectrum, including non-visible. However, a significant portion of solar radiation incident on the atmosphere is not received at the ground, being absorbed by atmospheric constituents such as H_2O vapour, CO_2, O_3, and O_2, and resulting on an average irradiance **at the Earth's surface**

of about 1120 W/m^2 [86]. Figure 3.54 illustrates Sun's spectral behavior depending on the measurement point.

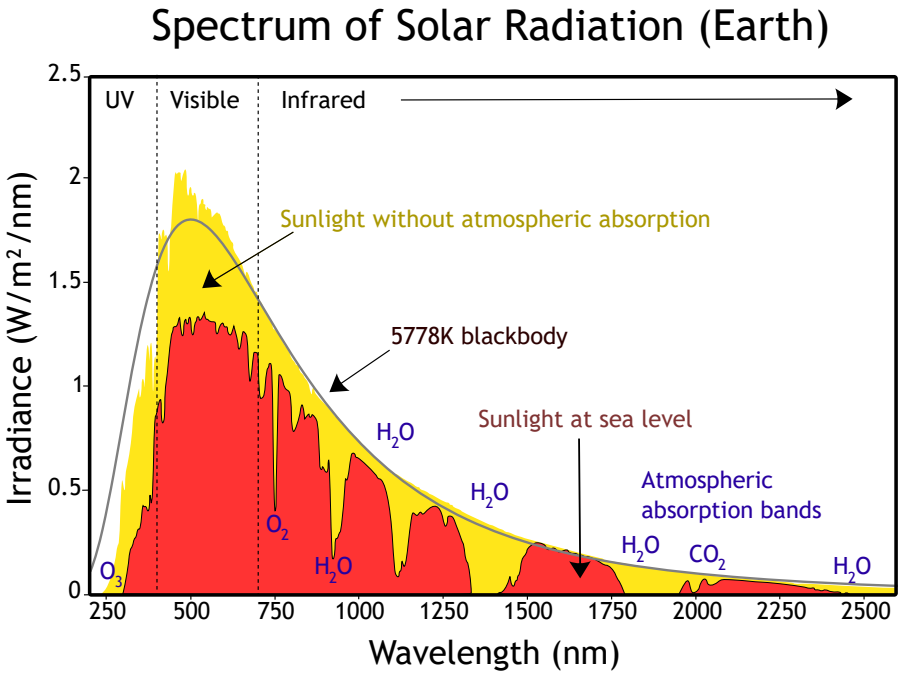


Figure 3.54 – *Solar spectrum above atmosphere and at surface [59]*

As stated, the terrestrial spectrum varies, so in order to count with a standard spectrum as representative as possible, it is commonly taken so-called **AM1.5 spectrum**. Instead of a particular measured spectrum, it is calculated from the reference AMo spectrum under representative geometric and atmospheric conditions. AM stands for *Air mass*; given that both absorption and scattering depend rather strongly on the path length of sunlight through the atmosphere, i.e. on the Sun's elevation angle above the horizon. When the sun comes closer to the horizon, its light passes through more air. AM1.5 refers to an spectrum taken with an air mass of 1.5 (Sun 41° above the horizon) with atmospheric conditions from the US standard atmosphere, representative for most middle latitudes, among others [40]. It will be used extensively during this project.

In the same topic, solar irradiance is usually measured using devices called **pyranometers**; they translate incident irradiation into a voltage, which can be later mapped into irradiation units. Briefly, there are three different types: **thermopile pyranometers**, adequate for high accuracy, exhibiting flat spectral response on large range between 300-3000 nm; they are based on the thermocouple principle already mentioned on 3.4.3.1.1. Secondly, **Silicon-based pyranometers** also known as photodiode-based or photovoltaic are a more cost-effective option, with lower accuracy specially under cloudy conditions. Its main drawback is their limited spectral range,

between 350-1100 nm. They are based on Photoelectric effect, also studied before in 3.4.3.1.4. Figure 3.55 compares AM1.5 reference with typical spectral response of each technology.

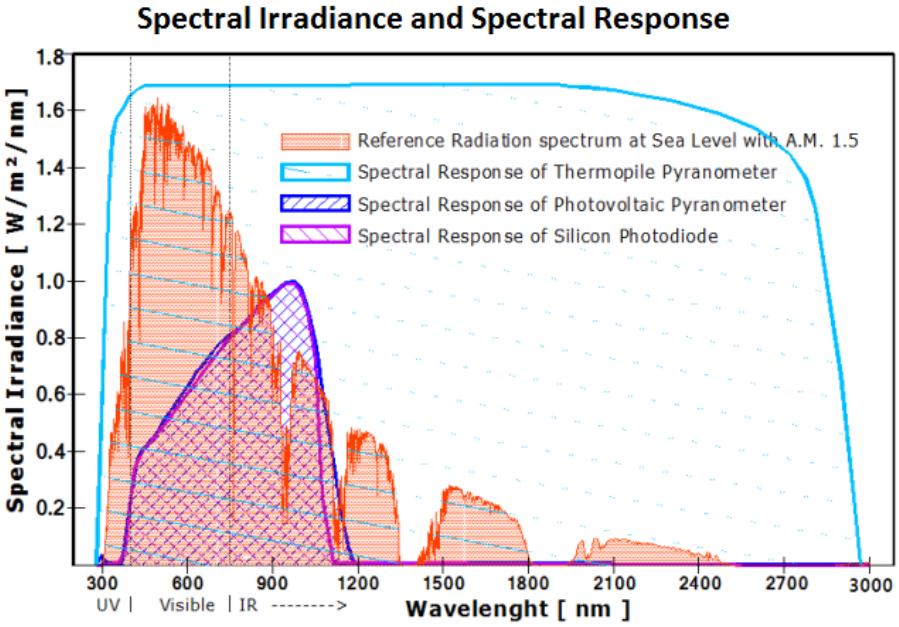


Figure 3.55 – *Solar spectrum AM1.5 reference and pyranometers expected spectral range.*

As mentioned, thermopile pyranometers exhibit a completely flat response, much wider than the rest of technologies.

- **Xenon Arc Sun Simulator**

Considering the simulation parcel of this project, it is needed to count with irradiance sources usable indoors. One of the most typically used are high power xenon lamps, as sun simulators. They are gas-discharge-based lamps which produce light when passing electricity through ionized xenon gas at high pressure [59]. Although it depends on the lamp, life duration and some additional factors, xenon arc lamp may match closely solar spectrum, as desired. Figure 3.56 illustrates the typical spectral behaviour of this kind of lamps, compared with AM1.5 standard reference again; it shows a pretty matched spectrum considering AM1.5 standard which confirms this technology as a really good candidate to simulate the Sun; however, as it will be analyzed later, that matching accuracy will depend on several factors (life of the lamp, quality, filters...) which typically will make it difficult to get such a good result.

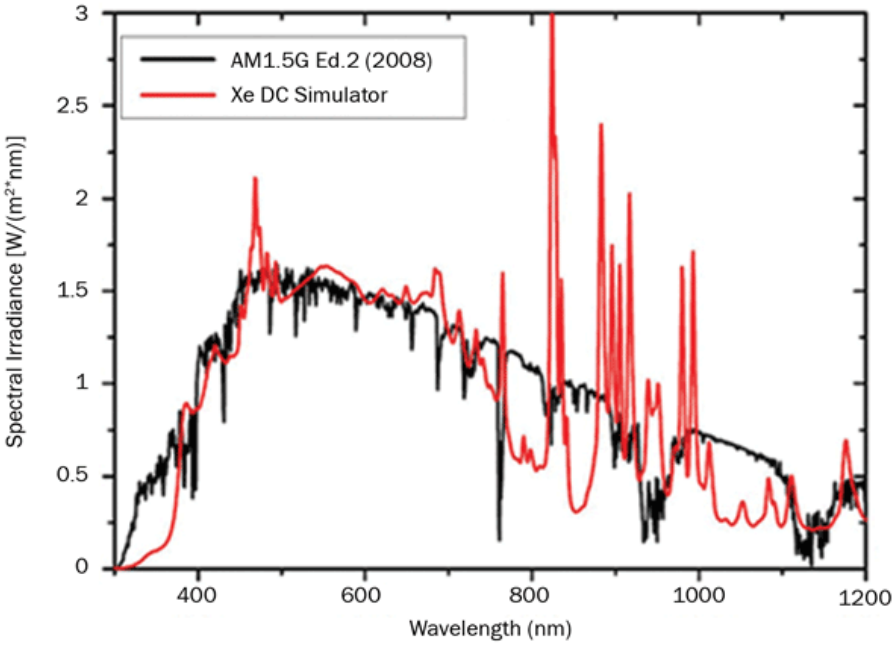
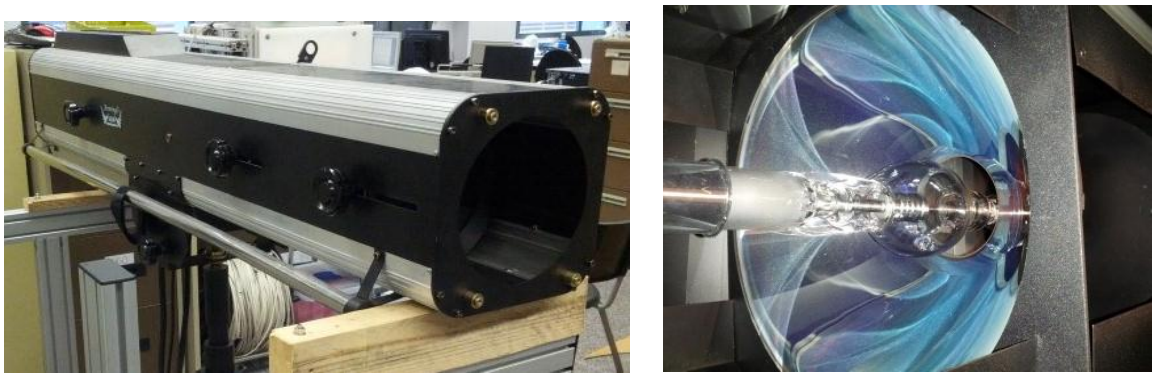


Figure 3.56 – *Xenon arc lamp typical spectrum compared with AM1.5 reference*

An example can be found at the Aerospace Engineering School of the University of Cataluña [98], where it was characterized one of these simulators of the brand **Anmingli**, shown in figure 3.57a, able to deliver up to 4 kW. Figure 3.57b shows the detail of the xenon arc lamp.



(a) *General view* **(b)** *Xenon lamp detail*

Figure 3.57 – *Xenon arc lamp Sun simulator used [98]*

Using a silicon-based pyranometer, they characterized lightning distribution of the simulator. Based on that data the plot has been rebuilt and it is shown in Figure 3.58. Despite the fact that the mentioned work only considered interpolated plot, original measurements have also been represented using the data available, i.e., while

Figure 3.58 shows discrete measurements *as are*, being difficult to interpret the distribution, Figure 3.59 shows an interpolated version of the data, which helps to understand in a more meaningful way how the simulator beam behaves. Indeed, interpolated plot Figure 3.59 allows determining that the beam is off-centred to the left, reaching a maximum irradiation on it of 1500 W/m^2 at a distance of 2.7 m. This is a usual problem with these kind of low-cost simulators, designed to work at larger distances, normally related to suboptimal lens design. This fact should be taken into account when used to characterize solar cells, trying to get maximum irradiation from the simulator. In the same way, it is usually a good practice to work with an irradiation value as close as possible to the expected in a real scenario; assuming AM1.5 reference again, as stated before, a reasonable value to be expected at the surface is 1120 W/m^2. In order to get that irradiation value, the distance must be increased in a factor given by Equation 3.4.15:

$$\frac{I_1}{I_2} = \frac{d_2^2}{d_1^2} \rightarrow d_2 = \sqrt{d_1 \frac{I_1}{I_2}} \tag{3.4.15}$$

Effectively, recalling that the intensity of light is inversely proportional to the square of the distance, standard AM1.5 irradiation value can be reached at **3.13 m**.

In addition, in the same work [98] spectral response was measured and compared with both, Sun's and a high quality simulator of the brand **MKS**, see [31]. It allows them to conclude that their low-cost simulator matches close enough not only the more expensive one, but also Sun's spectrum. More information on this can be found on the cited work.

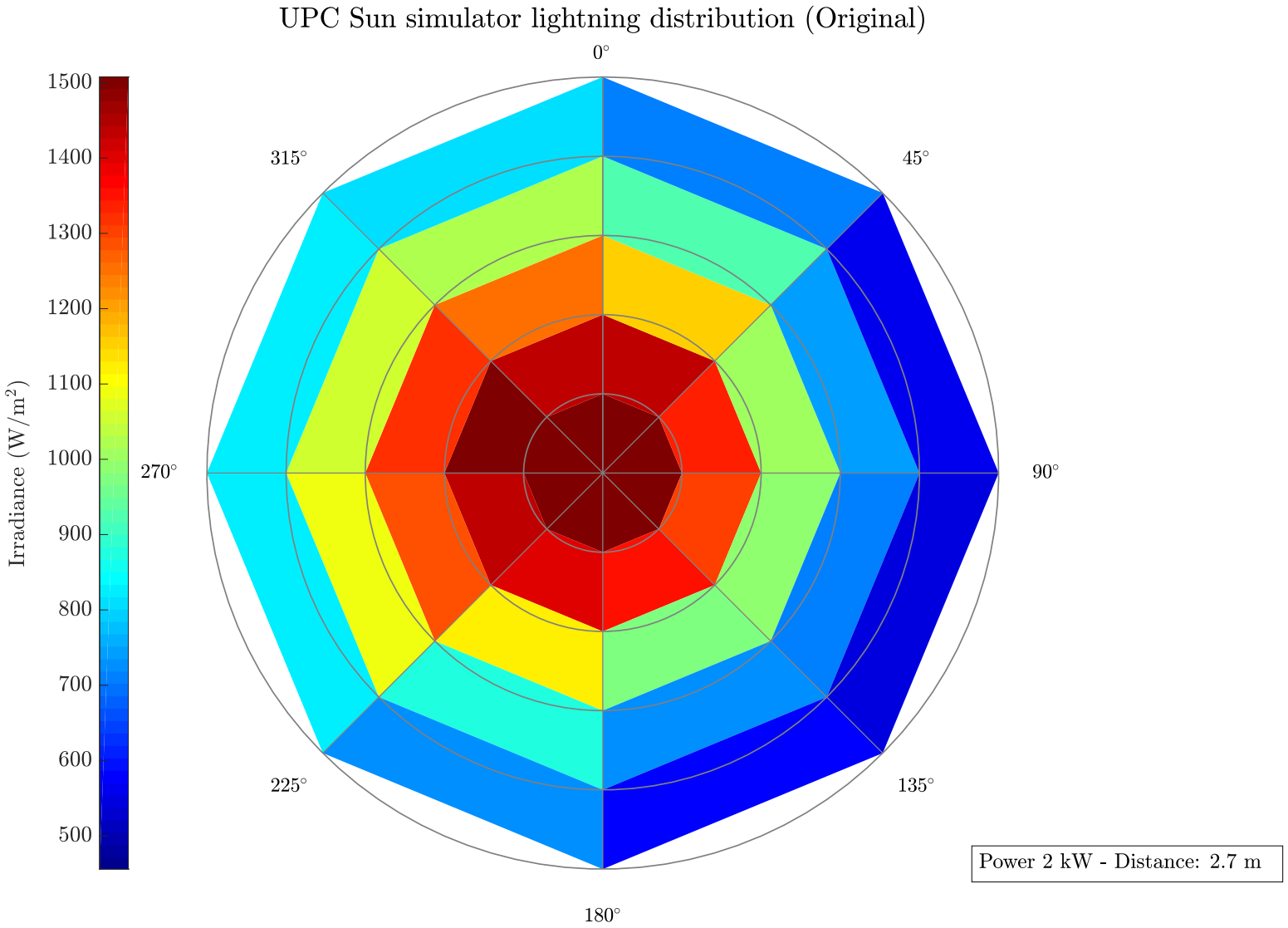


Figure 3.58 – *Original Lightning distribution. Rebuilt from [98]*

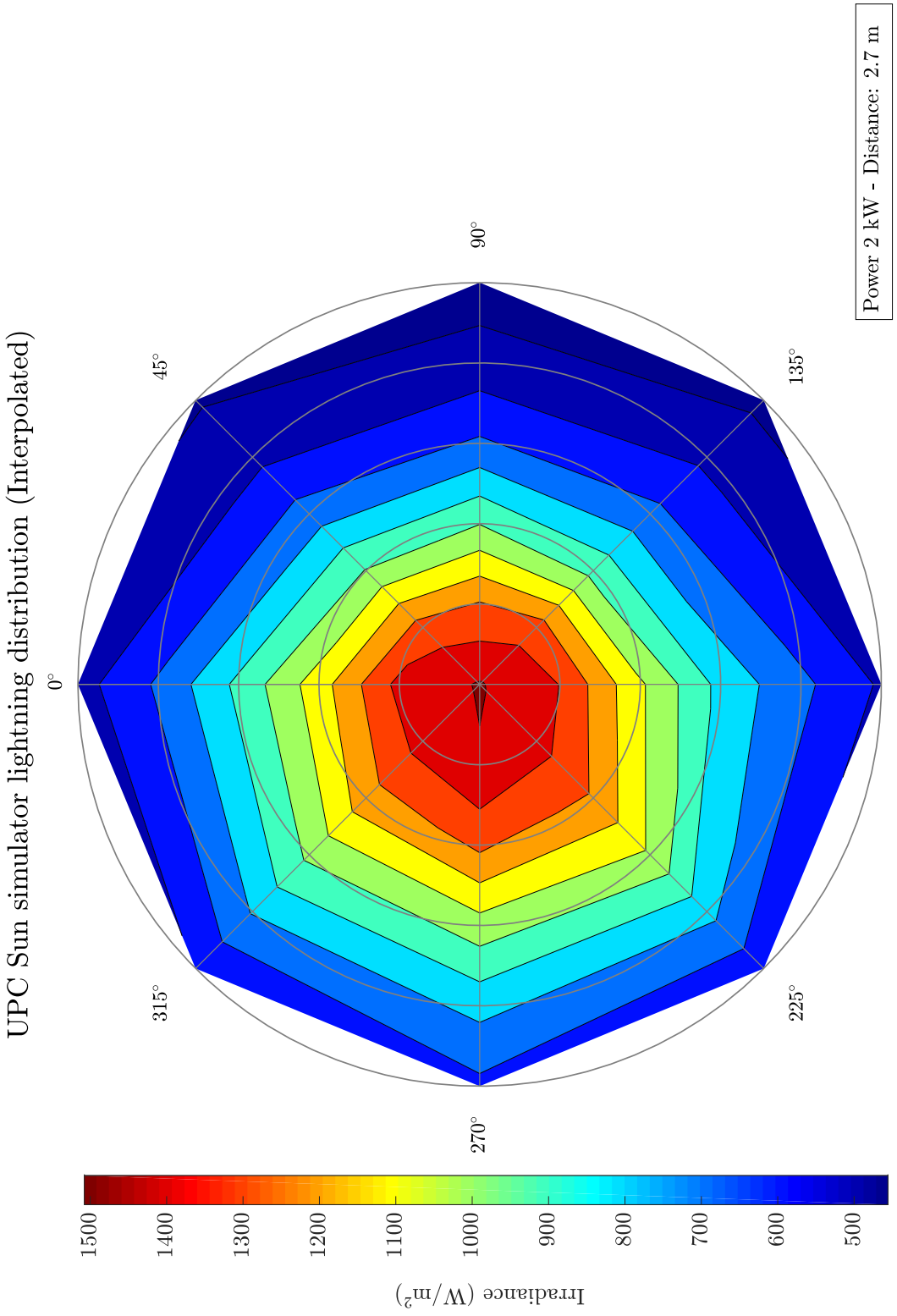
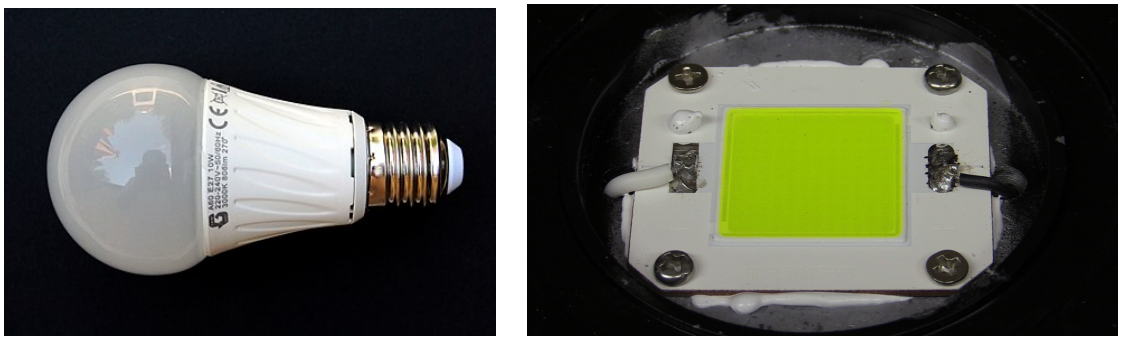


Figure 3.59 – *Interpolated Lightning distribution. Rebuilt from [98]*

- **LED Sun Simulator**

LED-based lamps are another possibility to simulate the Sun. They are usually less similar to the Sun's spectrum but they are more power-efficient, which may make them appropriate to be used in a simulation platform like the one under development in this project. Additionally they are easier to buy and are available in smaller packages than xenon. Figure 3.60a shows an example of a classic LED bulb while 3.60b depicts a COB LED thermally bonded. LED lamps are based on **light-emitting diodes**. LED lamps run on DC with voltages way lower than usual AC which entails the need of some electronics to convert usual AC supply into an appropriate one for the LED; these circuits are usually known as **LED drivers**. Another important issue related to this technology is thermal management: high temperatures can lead to failure and output light reduction, so typically this kind of lamps also include heat dissipation elements or cooling fans [59].



(a) *Classic bulb* **(b)** *Thermally-bonded*

Figure 3.60 – *LED [59]*

Regarding spectral behaviour and AM1.5 comparison, Figure 3.61 shows both as well as another example of xenon arc lamp spectral response, which makes it possible to compare them at the same time.

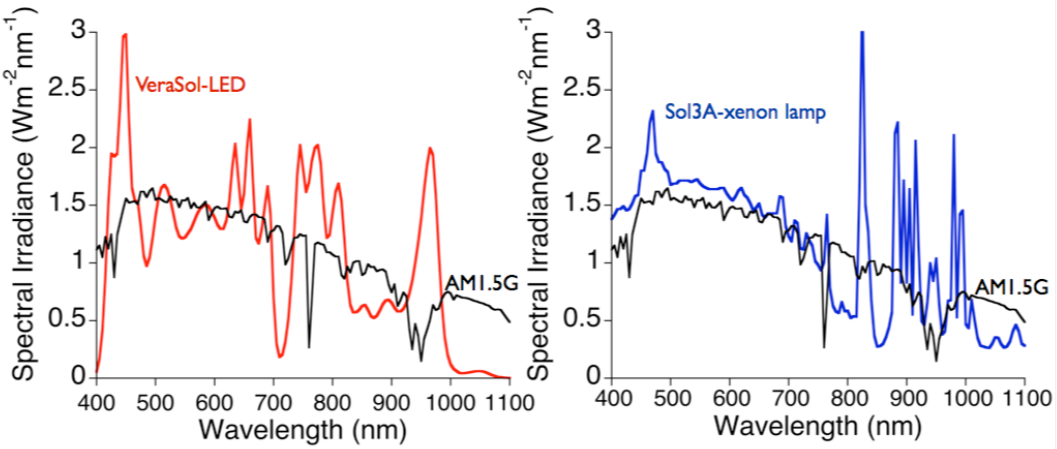


Figure 3.61 – *LED and xenon spectrum compared with AM1.5 reference [96]*

As previously mentioned, LED spectral match differs from Sun's in a more significative way than xenon does. However, once more, this point will vary depending on each lamp and its conditions and should be adequately characterized before going into production stage.

Some research has been previously performed, studying the possibility to use a LED solar simulator to characterized solar cells, for example [61] and [96]. The first of them concludes that their LED-based simulator spectrum matches closely AM1.5 for the visible spectral range. On the other hand, the second one characterizes I-V curves of certain solar cells using both LED and xenon simulators; although spectral match was not perfect in any case (see Figure 3.61), the author concludes that it did not affect the gross properties of the curve. Actually, he finishes featuring that LED simulator behaves better than xenon, given that there were no discernable differences in the I-V response from a number of solar cell, maybe due to the extra radiation generated by the latter and therefore additional heat which, as seen in 3.4.3.1.4, shifts the open circuit voltage of the I-V curve. Nevertheless, energy losses at certain parts of the spectrum are also mentioned (around 700 nm).

3.4.3.2 Power Regulation and Control

As deducible from past sections, spacecrafts usually carry an array of photovoltaic cells (i.e., a solar panel) which powers the load and supplies the whole system. Given the volatility of the power source in a real scenario (e.g., an eclipse) it is not hard to understand that energy must be somewhat stored so the system can keep working properly when the power source is not available. Regarding the simulation platform also under development, this point should also be considered to keep its accuracy. This need will be deeply analyzed in 3.4.3.3 and in few words, it is satisfied using **batteries**, as shown in Figure 3.34 too.

Therefore, during eclipse the battery will be in charge of powering the load. If the solar panel, the battery and the load were operated at the same constant voltage, there would be no need for any kind of power regulation. Nevertheless, as detailed in 3.4.3.1.4.1, the solar panel output voltage is higher at the beginning of life (see Figure 3.50), and when the array is cold for several minutes after each eclipse (see Figure 3.49). Also the battery voltage changes. Moreover, typically the spacecraft will be composed of a variety of components with different voltage needs. Since the system is required to provide power to the load at a voltage regulated within specified limits, see 3.1, a power regulation is always needed to match voltages of various power components during the entire operation time [103].

This power regulation is accomplished by battery charge and discharge converters, a shunt dissipator to control the bus voltage during sunlight and a controller in charge

of adequately cope with the bus voltage error signal. The solar panel as power source, and the battery as load, have both their respective I-V curves, shown in Figure 3.62. The system can work at either of the two intersection points A or B. However, point A is inherently unstable because the load slope is lower than the source slope. Point B, on the other hand, is inherently stable. Without a shunt control, the system would operate at point B, producing a **lower power**. Nevertheless, with adequate shunts regulating the sunlight voltage, the system will pull back from point B to point C, **shunting** the excess current I_{shunt} (difference between the source current at D and the load current at C) to ground, and producing more power.

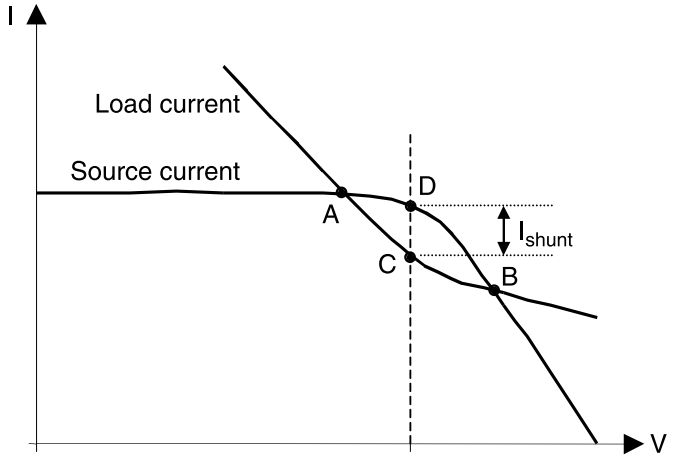


Figure 3.62 – *Stability of operating point and shunt control during sunlight [103]*

As for the controller, depending on the voltage error signal (the difference between actual bus voltage and desired), sends a control signal to one of the regulators in order to keep the bus voltage within the specified limits, allowing the battery to be charged (sunlight) or to provide energy (eclipse), with no damage for the system.

Although the latter is mainly a safety issue, power regulation is also closely related to performance. Indeed, as exposed in 3.4.3.1.4.1, V_{out} and I_{out} will constantly vary, probably mismatching the *MPP*. As Figure 3.47 shows, for the majority of its useful curve, solar cells act as a constant current source, but when reaching *MPP* boundaries (*knee* of the curve), the curve has an approximately inverse exponential relationship between current and voltage. At every moment (depending on striking radiation, temperature etc.) these pair of values will imply a different load $R = V_{\text{out}}/I_{\text{out}}$ given by Ohm's law; for an external device to draw maximum power from the solar cell, it should *see* a load with such a resistance equal to the inverse of this value; other way, and as Equation 3.4.14 states, **efficiency will decrease** and could eventually reach one of the *No power* points. Thus by **varying the impedance seen by the panel**, the operating point can be moved towards the *MPP*.

In sum, the desirable operation point varies considerably and this is not only an efficiency problem by itself, but it is also a problem for the following element in the power chain: **the batteries**. The following example will illustrate the problem: let us assume a set-up where the solar panel is directly connected to a standard 3.7 V Li-on battery. According to the manufacturer, the solar cell is able to provide a power of 1 W at the *MPP* (5 V, 0.2 A); however, given that it is directly connected to a 3.7 V source (the battery, whose voltage is taken as constant) the output voltage of the solar cell is fixed to that value, therefore operating at (3.7 V, 0.2 A) and producing 0.74 W in the best scenario, implying an **unexpected loss of 26 %** (actually, it would be even worse given that the battery voltage is not constant but it will decrease along with its available energy, see section 3.4.3.3). This is another reason why an adequate power regulation is highly desirable.

The photovoltaic power system, therefore, primarily consists of a solar array, a rechargeable battery, and a power regulator which regulates power flow between various components to control the bus voltage. This section deals with the power regulation stage, particularly the different architectures normally used and its efficiency.

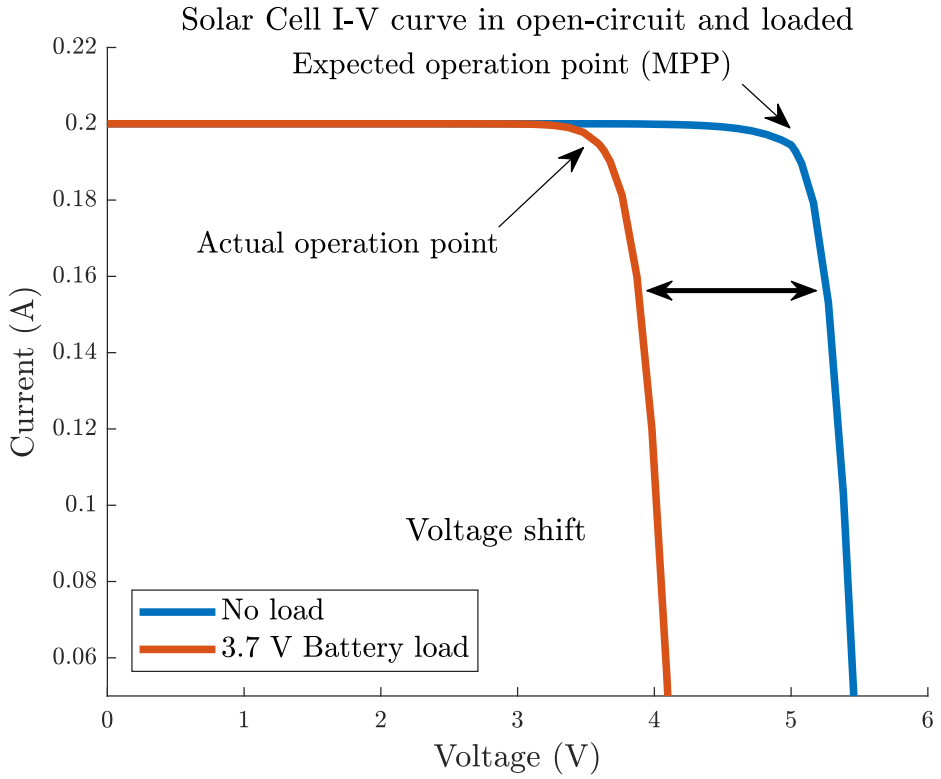
3.4.3.2.1 Direct Energy Transfer

The set-up described before, in which the solar panel is directly connected to a battery, is known as **Direct Energy Transfer** (DET) architecture. In this case, the power bus is said to be *dominated* by the battery voltage, which implies that the solar panel must operate at the same voltage of the power bus, potentially not delivering the full power it is capable of at all times. Since the solar array is designed to never exceed a voltage past the *MPP*, it will reach its full power power producing capability only when the battery is at its highest voltage, which occurs when the battery is completely charged. However, when the battery is at its minimum voltage, at the beginning of its charge cycle, the solar array will operate well below the *MPP*. This phenomena is graphically illustrated in 3.63a which shows the voltage shift produced when the *MPP* is displaced due to mismatch load imposed by the battery. Figure 3.63b depicts the power losses due to that issue.

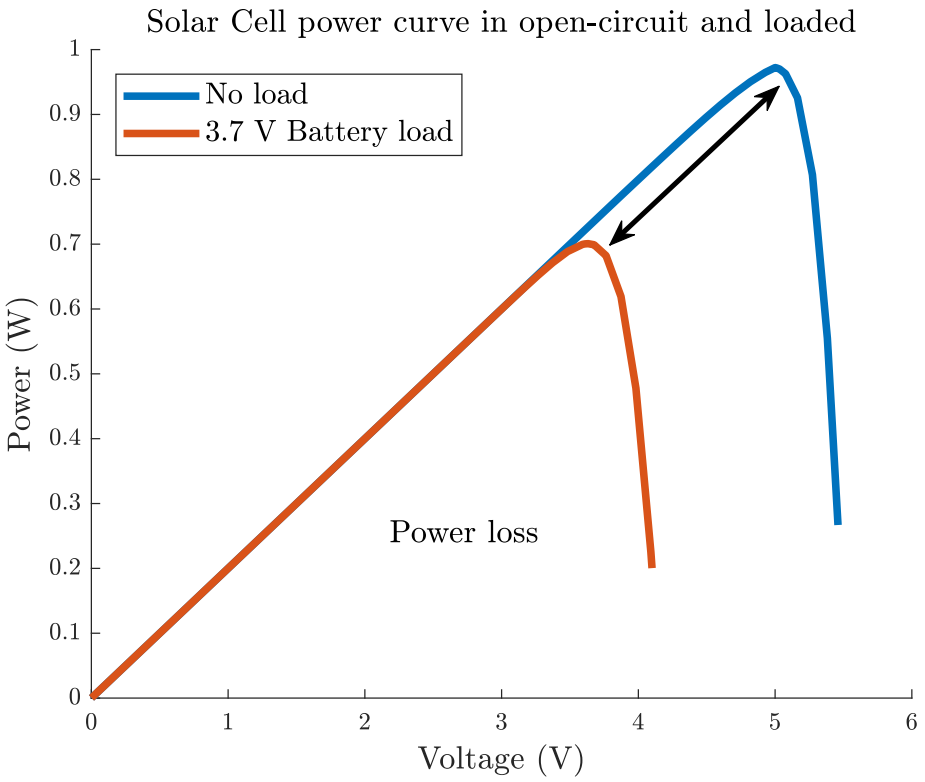
This architecture can be subdivided into two classes which are briefly analyzed next.

3.4.3.2.1.1 Sunlight regulated bus

This sub-architecture is normally used when the objective is to minimize complexity; to achieve that power from both sources available — the solar panel output and the battery — directly to the load. An in-depth analysis of this architecture goes beyond the scope of this project, however typical sun regulated bus architecture is depicted in Figure 3.64 so minimal references can be made.



(a) *I-V curves in open-circuit and loaded*



(b) *Power curves in open-circuit and loaded*

Figure 3.63 – *Solar panel performance directly connected to a battery*

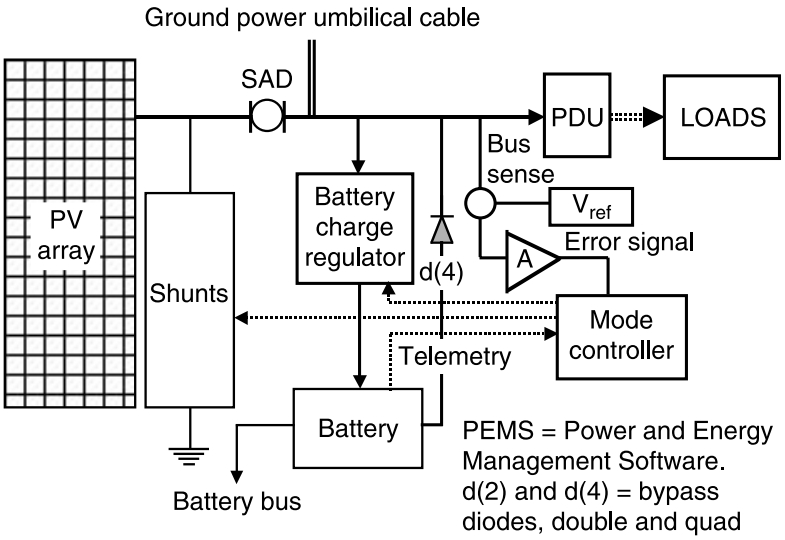


Figure 3.64 – *Sun regulated DET architecture [103]*

In this architecture, the bus is said to be 'regulated' by **shunt control** during sunlight which is accomplished by a battery charge regulator to control charge rate. On the other hand, during eclipse, the battery charges directly to the bus through the *battery discharge diode*, *d* in Figure 3.64; this diode only allows discharge from the battery, blocking any uncontrolled charge current coming to the battery, leaving this task to the charge regulator, and disconnecting the battery from the bus during sunlight. Without this kind of regulation, the solar panel output voltage would settle at operating point B in Figure 3.62, which would be unsafe at some points of operation, for example at BOL and after eclipse (the solar panel would be cold and produce a higher output voltage, as seen in Figure 3.49).

It is simple and reliable but implies variations in bus voltage up to ±25% around nominal value [103]. This architecture finds application mostly in relatively low power needs, such as CubeSats, for example.

3.4.3.2.1.2 Fully regulated bus

This sub-architecture is commonly known as *regulated bus* and it is mainly characterized by a controlled bus voltage within a few percent during orbit period. Contrary to sunlight regulated, it requires a battery discharge converter which is expensive. On the other hand, it allows great flexibility in battery choice.

It is typically used when the load has high power requirements or the spacecraft loads require a lower variation around bus voltage.

3.4.3.2.2 Maximum Power Point Tracking

Section 3.4.3.1.4.1 clearly exposed the advantages of a mechanism able to get the system to the optimum performance point continuously. It can be accomplished with the **Maximum Power Point Tracking** (MPPT) architecture, which makes use of switching regulation between the solar panel and the load, as Figure 3.65 shows.

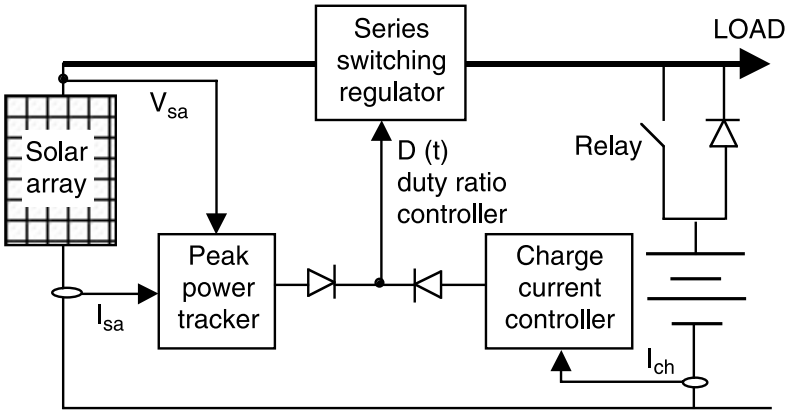


Figure 3.65 – *Maximum Power Point Tracking architecture [103]*

The peak power tracker senses the *MPP* and is in charge of keeping the series regulator input voltage at the maximum power producing level; then the output voltage is stepped down to the load voltage, i.e. the battery, by varying the duty cycle as needed.

Noteworthy advantages of this architecture are that it maximizes the solar panel output power at every moment without neither shunt nor battery charge regulator. As main drawback, it may result in low efficiency due to power losses in the tracking process itself.

Figure 3.66 summarizes pros and cons of the architectures analyzed.

3.4.3.3 Energy storage. Batteries.

In an energy-limited scenario as in orbit, energy storage is a vital issue. For example, when in eclipse, as there is no energy input available, in order for the system to keep working, there must be some kind of reserve; also when the demand exceeds the power generation at any time. This storage is usually accomplished by using **batteries** which store energy in an electrochemical form.

As in previous topics, some battery basic concepts are introduced before going into detail.

System	Sun regulated	Fully regulated	Peak power tracking
Pros	High power transfer efficiency from solar array and battery to load. Fewer power system components.	Well regulated input voltage to all loads. Simpler, lighter and more efficiency load converters.	No need for shunt regulator and battery charge regulator.[a] Makes the maximum use of the incident solar energy.
Cons	More complex load converters. Battery latch-up concern. Larger solar array.	Needs more power converters. Series power loss between battery and load.	Lower efficiency than DET at EOL in many cases. More heat dissipation inside the spacecraft body.
Best application in missions with these features	Small load variations. Small variations in illumination for most of the sun period.	Loads requiring close regulation. Large solar array output voltage variations.	Large variation in solar array input energy (illumination) throughout the mission.

Figure 3.66 – *Pros and cons of the analyzed architectures [103]*

- **Energy density:** electrical energy per unit mass (Wh/kg) available when fully discharged from a fully charged state at a given rate and temperature.
- **Roundtrip efficiency:** also called 'Wh efficiency', is the ratio of energy provided between full charge and the following full discharge at a given rate and temperature.
- **Depth of discharge (DoD):** amount of energy taken out of the battery per cycle. Looking in the opposite direction, the SoC (state of charge) is 100 - %DoD.
- **Nominal capacity (Ah):** typically stated by the manufacturer. This capacity may or may not represent the amount of Ah available upon discharge down to a technology dependent EOL voltage.
- **Dimensional abuse:** it is defined as the charge the cell can deliver at room temperature until it reaches a cut-off voltage of about 2/3 of the fully charged voltage. Charge and discharge currents (A) are expressed as a multiple or fraction of the nominal capacity which is called 'C' and depends on physical size. (e.g. 2C, C/2, C/100...). It is widely used and its practical meaning is the possibility to deliver C amperes for 1 h or C/n amperes for n hours.
- **Cycle life:** defined as the number of charge/discharge (C/D) cycles the battery can deliver while maintaining the cut-off voltage.

Batteries are made of numerous electrochemical cells assembled in a series–parallel circuit combination to obtain the required voltage and current. It has positive and negative electrode plates with insulating separator and a chemical electrolyte in-between. The two electrode plates are connected to two external terminals mounted on the casing. The cell stores electrochemical energy at a low electrical potential. The cell voltage depends solely on the electrochemistry, and not on the physical size. Figure 3.67 shows the typical electrochemical cell construction [103]

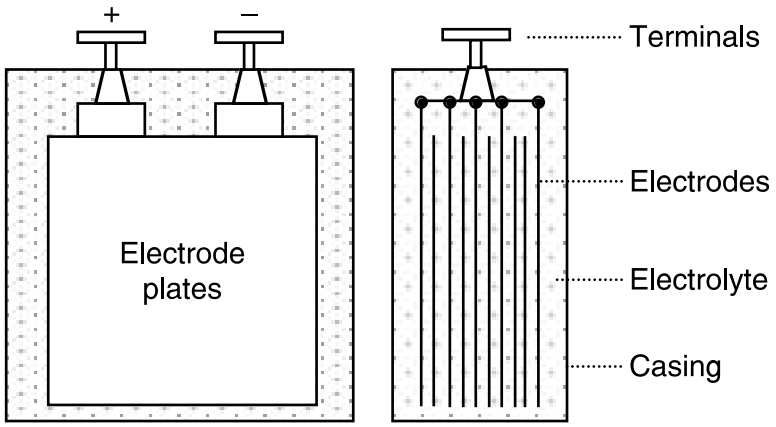


Figure 3.67 – *Typical electrochemical cell construction [103]*

Batteries cells can be thought of as voltage sources with small internal resistance which respectively decreases and increases linearly with the nominal capacity discharge. Ignoring parasitics, it is as a DC source representing the **electromotive force** (EMF); this is the typical term used when a voltage is generated by a battery and represents the **energy** per unit charge which has been made available by the generating mechanism, in this case, the electrochemical cell. EMF depends on the SoC and ageing. On the other hand, the cell resistance in series can be compared with the traditional output series resistance in power sources; besides SoC and ageing, it also depends on temperature. Next, the most used batteries technologies are briefly introduced.

3.4.3.3.1 Nickel Hydrogen

NiH_2 batteries have been the most widely used in space sector during the last 25 years, which makes them highly reliable. They combine some of the best characteristics from other technologies, such as NiCd or the fuel cell (see 3.4.3.1.3) and can tolerate some over-charge or over-discharge without damage; it also has a greater charge/discharge cycle life and low internal resistances than older technologies such as NiCd and do not exhibit a noticeable Memory effect. However, it has a high self-discharge rate, around 0.5 %, and also high loss of capacity. In the typical configuration, each cell develops 1.25 V. Figure 3.68 shows a common NiH_2 batteries stack.

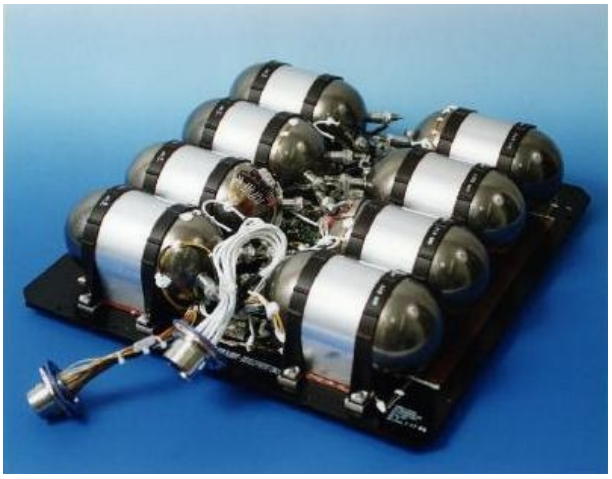


Figure 3.68 – *NiH_2 batteries stack [55]*

3.4.3.3.2 Lithium-Ion

With a great expansion throughout the last years, Li-ion technology exhibits pretty higher energy density than NiH_2. It is corrosion-free but lithium is highly reactive and must be stabilized. Li-ion has a cut-off voltage of 2.7 V, average discharge voltages of 3.5 V and end-of-charge voltage of 4.2 V. The average discharge voltage of 3.5 V contrasts with the 1.25 V for NiCd and NiH_2 requires about one third of the cells in series for a given battery voltage, allowing a smaller assembly and test costs [103]. Another significant advantage is its capacity to deliver peak power without negative consequences on life duration.

As for the negative side, Li-ion cells are sensitive to over-charge and over-discharge and exhibit low performance at low temperature, due to associated high internal resistance under those conditions. Figure 3.69 depicts a common Li-ion battery.

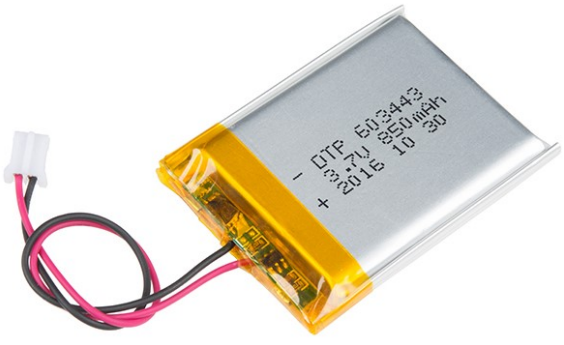


Figure 3.69 – *Li-ion battery [49]*

3.4.3.3.3 Lithium Polymer

This short comparison finalizes with lithium polymer batteries. These batteries are increasingly common these days, because of their wide use in drones and radio-controlled devices. Their electrochemical basis are similar to that of Li-ion but they are more delicate and have the same general problems of Li-ion e.g., over-charge, over-discharge, over-temperature, short circuit, crush and nail penetration may all result in a catastrophic failure. 3.70a shows a standard LiPo battery while 3.70b illustrates the consequences of an incorrect usage.

(a) *Standard LiPo battery [20]* **(b)** *Expanded Lithium-based battery [59]*

Figure 3.70 – *LiPo batteries*

3.4.3.4 Formal Requirements Definition

Ref.	Formal Requirements
EPS.FoR.1	Solar cells shall produce at least 1 W of power.
EPS.FoR.2	Solar cells shall be silicon monocrystalline.
EPS.FoR.3	Batteries shall be Li-ion and have a capacity of at least 1500 mAh.
EPS.FoR.4	The EPS shall have a contingency system to provide 3.3 V in absence of the main regulation subsystem.
EPS.FoR.5	The EPS shall have USB connectors which allow charging batteries from external supply.

Table 3.5 – *EPS - Formal Requirements*

3

Chapter 4

System Design

After the different stages of analysis and requirements definition performed in former chapters, this forth chapter will address the **system design**. It will be structured as Chapter 3, so the flow design coincides as much as possible with the analysis flow.

Firstly, the mechanical platform is designed and characterized using professional tools such as SolidWorks®. Also it is manufactured *in-house* using aluminum milling machine. Regarding the Ground Station, it is developed using an extended framework in space field, even used by NASA. Finally, some of the most important subsystems of CubeSat are developed, concluding with a complete Engineering model, expandable and which can be used as a solid base to keep working on participating in the *Fly your Satellite!* [13] program.

4.1 Inertial 2D Orbit Simulator (I2DOS)

Among the different platforms analyzed in Section 3.2, this project will include an **inertial** one, which has been called Inertial 2D Orbit Simulator (I2DOS). Particularly, the designed platform will be **rotational** along Z axis. I2DOS will be composed of two parts, on the one hand, the rotational platform itself, and on the other hand, the base which will support the platform and will allow including external elements such as irradiance sources, according to the **formal requirements**.

4.1.1 Inertial Platform

4.1.1.1 Design and mechanical characterization

The inertial platform is designed using SolidWorks® and sequentially improved through different redesigns. It is fully designed from scratch, with a physical background behind, reasoned later. It must have some kind of support which hosts the CubeSat during simulation. Given the circular shape of the platform itself, that support will also be circular. Besides, it must have adequate attachment for a 1U CubeSat. Figure 4.1 depicts a high quality render of that piece, including attachments.

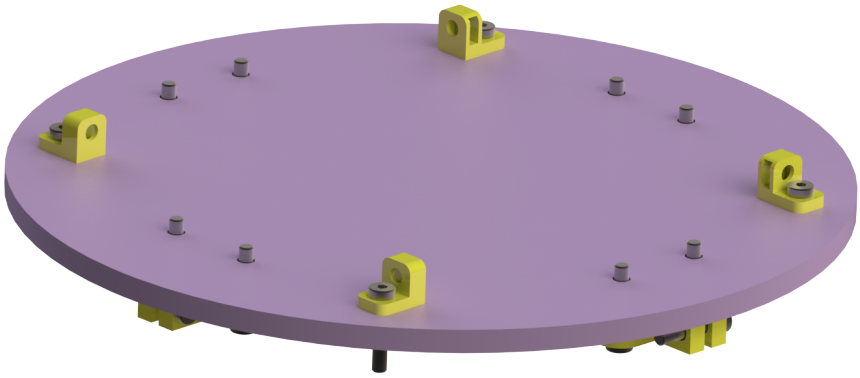
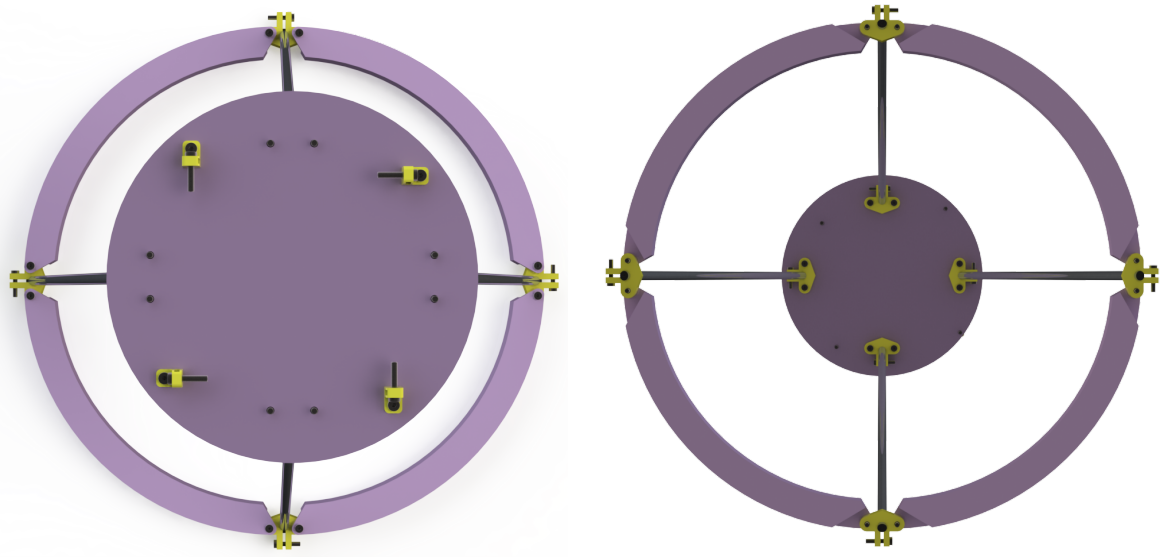


Figure 4.1 – *1U CubeSat support*

Top view and bottom view are depicted in Figure 4.2a and Figure 4.2b while another 3D render of the final design is shown in Figure 4.3.



(a) *Top view* **(b)** *Bottom view*

Figure 4.2 – *Inertial platform*

Figure 4.3 – *Inertial platform 3D render*

Regarding the **mechanical characterization**, simplifying, the *functioning* is based on the inertia moment stored on the bigger ring at the bottom area (I_1) plus the one stored on the CubeSat support (I_2). The first is considered a thick-walled cylindrical tube with open ends of inner radius r_1, outer radius r_2, length h, mass m and a density ρ whose moment of inertia is given by the first term in Equation 4.1.1; the second can be approximated by a solid cylinder of radius r, height h and mass m and its moment of inertia is given by the second term in Equation 4.1.1 [59]. The inertia moment stored at the rods is considered negligible as well as the components related to X and Y axis.

$$I_z = I_{z1} + I_{z2} = \frac{\pi\rho h}{2}(r_2^4 - r_1^4) + \frac{1}{2}mr^2 \tag{4.1.1}$$

It has been considered a density ρ of $1.24\,g/cm^3$ corresponding to PLA material and a mass m given by SolidWorks® of 453 g. Substituting with the rest of design parameters, the approximated total moment of inertia stored when rotating around Z axis is $94.6\,kg/cm^2$. SolidWorks® also calculates the inertia moment exactly, featuring a value of $108\,kg/cm^2$. It makes a difference of about 13 % which is coherent with the approximations taken and verifies the result of Equation 4.1.1.

4

I2DOS, and particularly this inertial platform, simulates the moment of inertia that the CubeSat will face when deployed, in Tumbling. It will have to use different detumbling mechanisms to counter that uncontrolled movement.

4.1.1.2 Manufacturing

The inertial platform is 3D-printed using PLA and aluminum rods as the ones depicted in Figure 4.3. The final result is shown in Figure 4.4.

Figure 4.4 – *Manufactured Inertial platform*

4.1.2 Base

4.1.2.1 Design

In order for the inertial platform to rotate, it is necessary a base which supports it. It is also designed using SolidWorks® according to the requirements (see section 3.1); it must allow including external components such as sun simulators. The high quality 3D render of the base is depicted in Figure 4.5.

Figure 4.5 – *Base 3D render*

The center hole holds the inertial base while the external ones allows adding external components; the attachments of the inertial platform are reused. It is designed to be wooden manufactured and the additional components can be attached using 4 cm aluminum rods. Figure 4.6 depicts the base with the different rods.

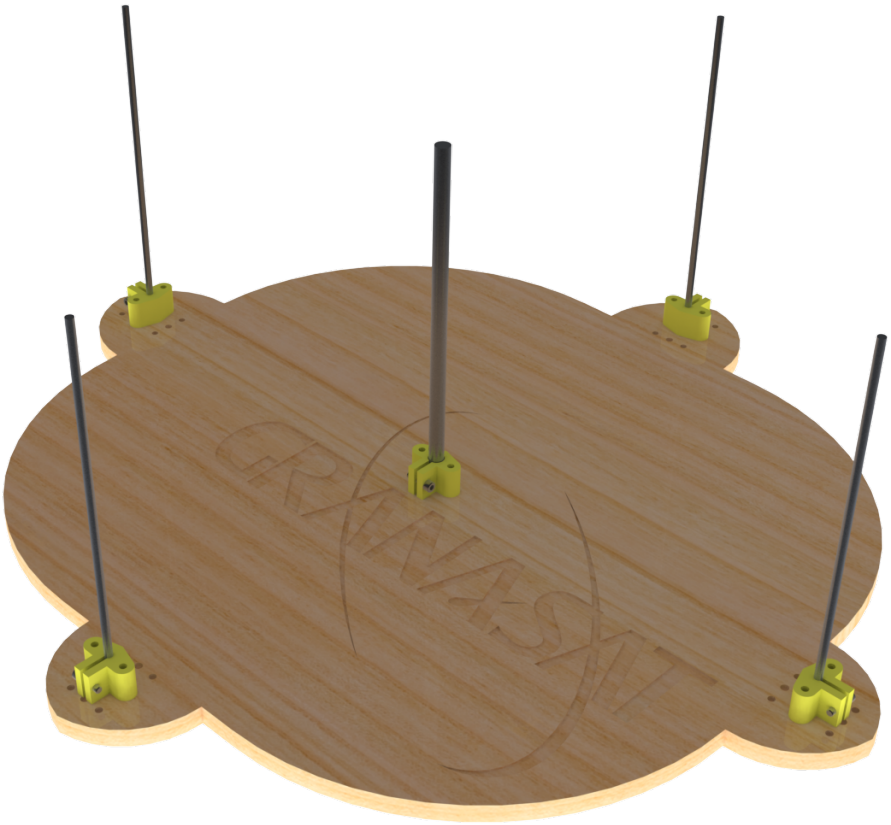
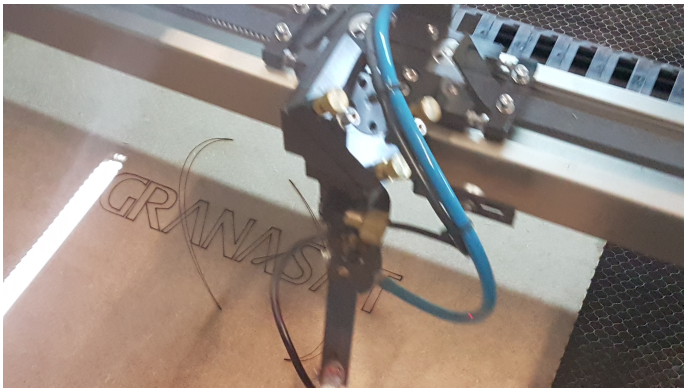


Figure 4.6 – *3D render of the base with the aluminum rods*

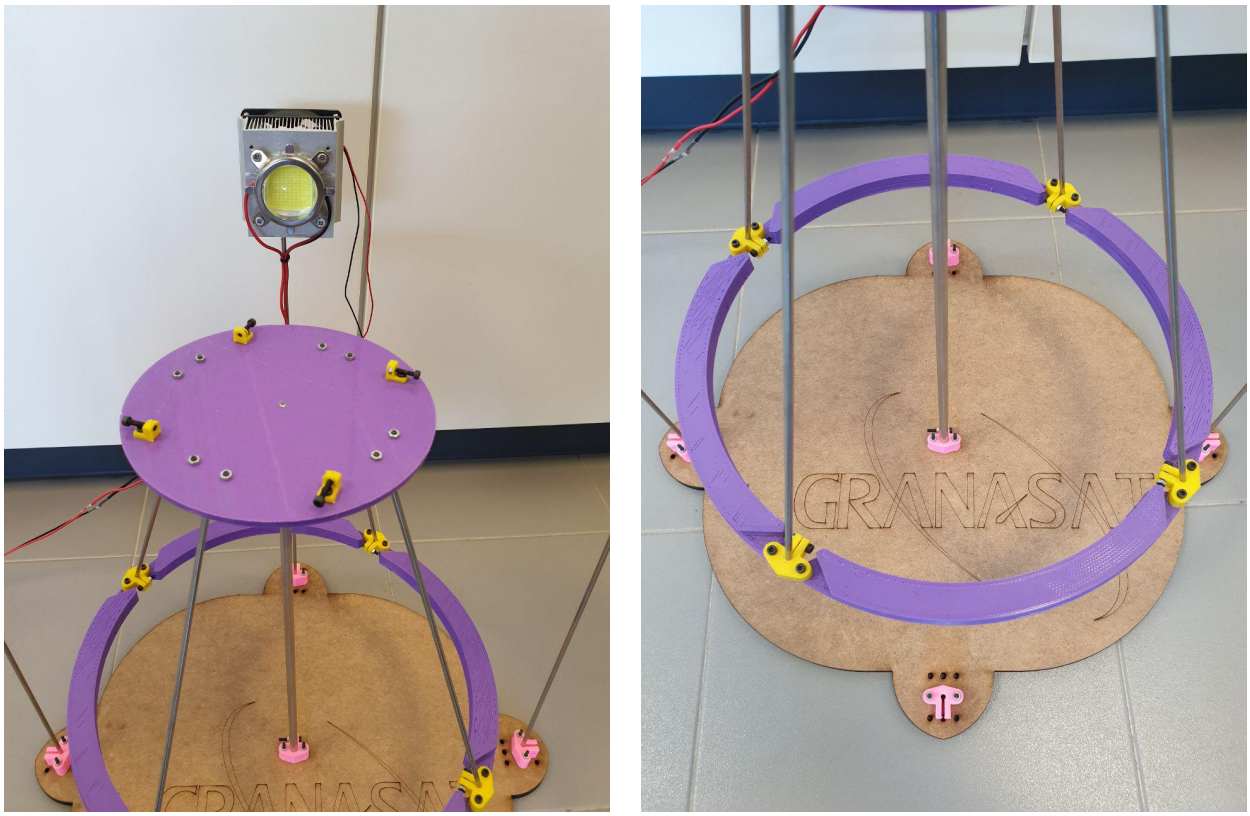
4.1.2.2 Manufacturing

The base was manufactured at one of the *makerspace* of the University of Granada in which the base could be cut. Video 4.1 shows part of that process (Adobe Reader needed).



Video 4.1 – *Wooden base manufacturing (double click)*

Figure 4.7a and Figure 4.7b depict the final result, with an external irradiance source attached, analyzed next in Subsection 4.1.3.



(a) *Front* **(b)** *Detail*

Figure 4.7 – *Inertial 2D Orbit Simulator (I2DOS)*

This completes the Inertial 2D Orbit Simulator (I2DOS).

4.1.3 Irradiance sources characterization

In order to have a simulation platform as realistic as possible, it is needed an irradiance source which emulates the Sun. In this subsection, different lightning options are compared, following the analysis performed in section 3.4.3.1.4.2. As studied before, it is desirable a spectrum as similar as possible to the Sun's, so the spectral response of the solar cells is such that the output power is maximum.

4.1.3.1 Xenon Arc Sun Simulator

Firstly, the Xenon Arc Sun Simulator of the GranaSAT laboratory is characterized following the procedures detailed in Chapter 3. Although it is not usable in the proper

mechanical platform designed in Section 4.1, it is useful as a reference and can be eventually used in order to perform real missions simulations. This simulator features six different power modes, in increasing order. Figure 4.8a and Figure 4.8b show the simulator to be characterized.

(a) *Side view*

(b) *Front detail*

Figure 4.8 – *Xenon Arc Sun Simulator at GranaSAT laboratory*

4.1.3.1.1 Spectral response

To characterize the spectral response of the simulator, we had the collaboration of the Department of Optics from the University of Granada. To perform the measurement, it was used an spectrometer THORLABS CCS200/M [52] as the one depicted in Figure 4.9.

Figure 4.9 – *Spectrometer THORLABS CCS200/M [52]*

The Sun simulator was measured with the different selectable power levels, getting the plot shown in Figure 4.10.

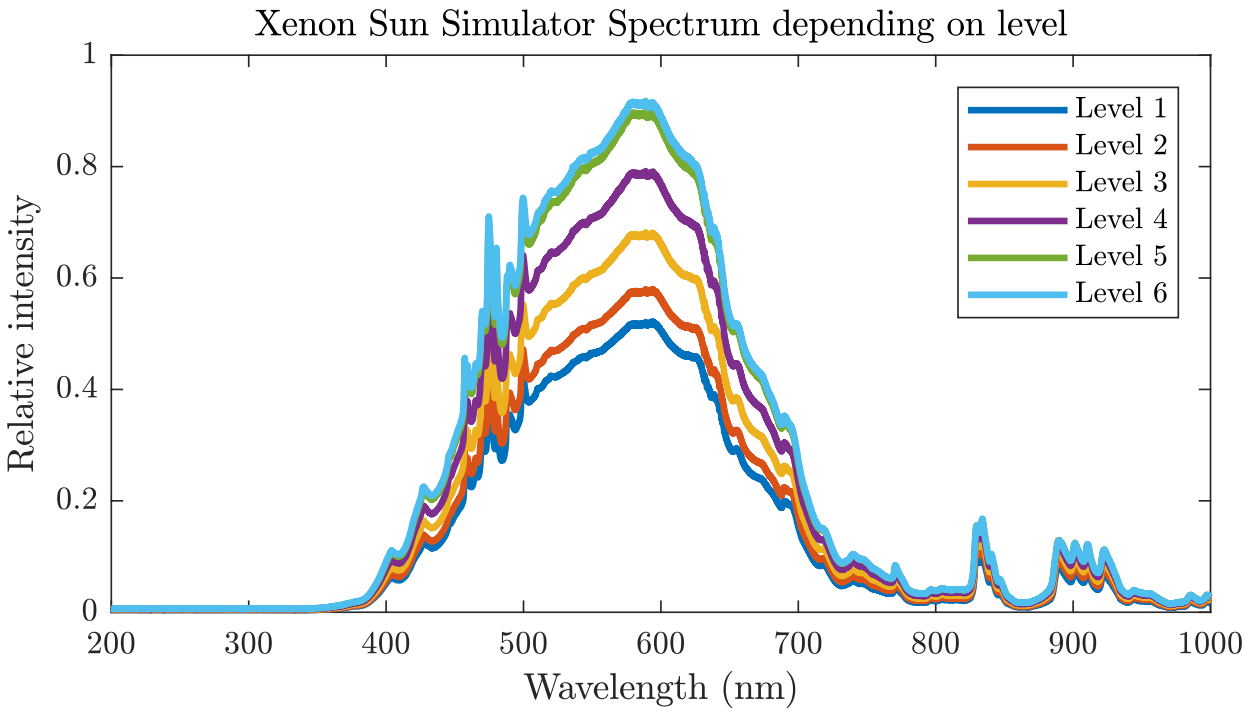


Figure 4.10 – *Spectral response of the different power levels*

Figure 4.10 illustrates a relative intensity plot, as the used spectrometer cannot measure absolute values, which will be determined later. It is easy to see that the spectrum remains the same regardless of the chosen level, as expected, featuring an spectral response in the range between 400 and 700 nm. It is worth mentioning that power does not increase uniformly with levels, for instance, while the increase is greater between levels 3 and 4, the difference between levels 5 and 6 is almost imperceptible.

Recalling that the CubeSat will count with solar panels as power source, it is interesting to see how the spectral response of the simulator matches the Sun's. It is depicted in Figure 4.11, using AM1.5 again.

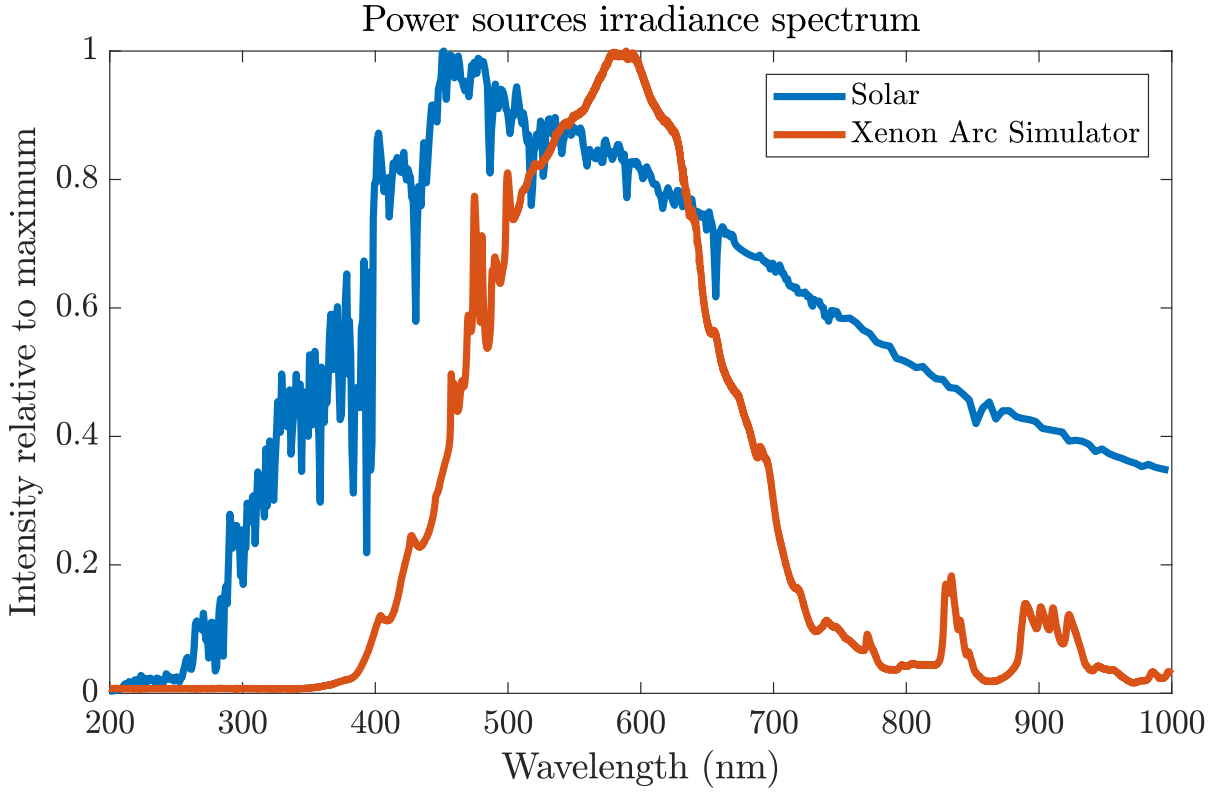
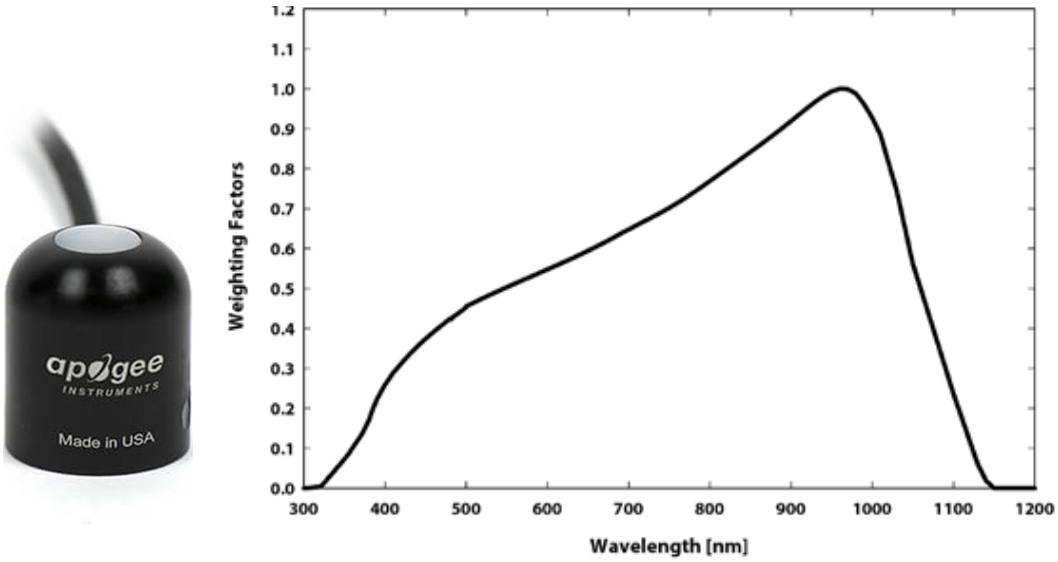


Figure 4.11 – *Spectrum comparison*

The simulator exhibits a narrower wavelength range in comparison with Sun's, which features a wide spectral response between 300 nm up to 1000 nm; however, the area with a greater intensity is similar to the simulator's one, between 450 and 600 nm. It must be recalled from Figure 3.53 that a standard Si solar cell gets excited in the wavelength range 400-1100 nm, so it is reasonable to expect a considerably worse performance when this Xenon Arc Sun Simulator is used.

4.1.3.1.2 Irradiance

After characterizing the xenon arc simulator in terms of spectral response, this section addresses the irradiance power and its variation with distance. In order to perform that measurement, it is used the Silicon-base pyranometer **Apogee SP-110-SS** [3]. Figure 4.12a depicts it while Figure 4.12b shows its spectral response.



(a) *Detail* **(b)** *Spectral response*

Figure 4.12 – *Apogee SP-110-SS [3]*

Using the pyranometer, several measurements are taken at different distances and plotted in Figure 4.13.

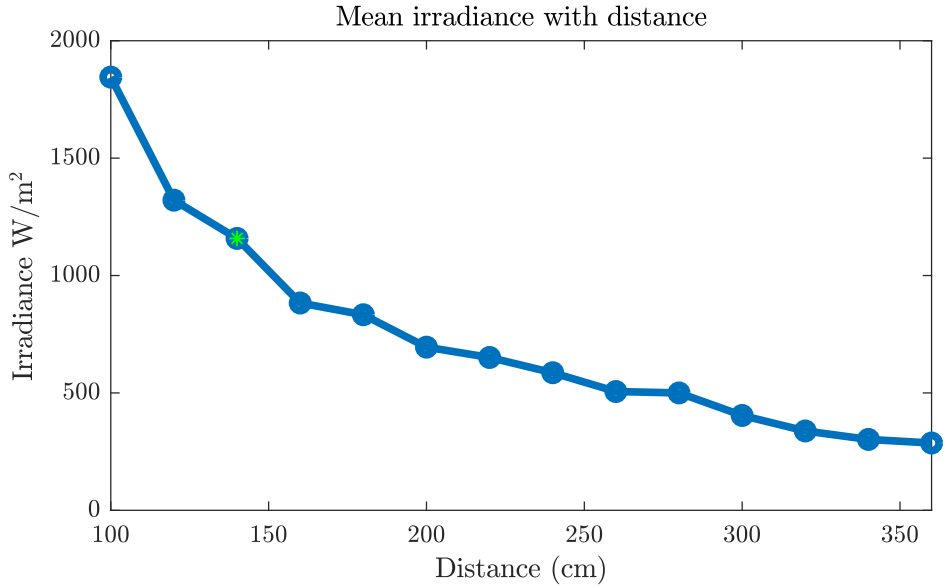


Figure 4.13 – *Irradiance decay with distance*

As Figure 4.13 shows, the irradiance decays with distance approximately in a slightly exponential or power tendency. Marked in green, it is the point in which irradiance matches the amount received at the Earth's surface (1120 W/m^2), approximately at a distance of 140 cm. This shall be the point to perform simulations when emulating the Sun.

4.1.3.1.3 Lightning distribution

As performed in Chapter 3 with the measurements from [98], it is analyzed the lightning distribution of this Sun simulator, in order to detect off-centered beam or any other inconsistencies. This is a complex procedure which may be performed with different instruments. In this case, the previously mentioned pyranometer is used along with a designed **test template** to be lighted up with the xenon arc simulator. Figure 4.14 shows that template, designed with AutoCAD.

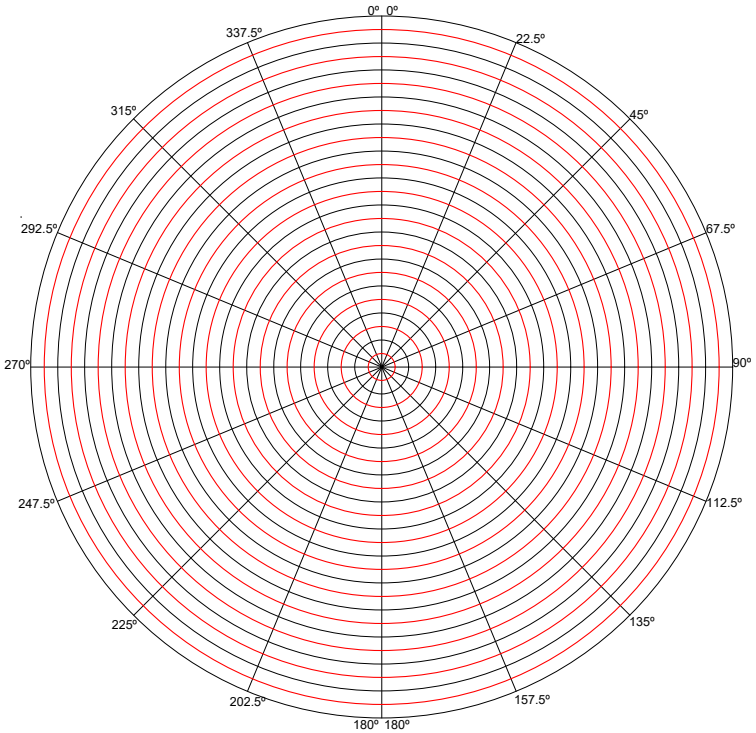


Figure 4.14 – *Lightning distribution test template*

The procedure consists in performing measurements with the pyranometer at the different angles depicted in the test template, until completing the whole circumference. Besides, these measurements must be taken at several distances, marked in red. When completed, it is possible to plot the lightning distribution, just as performed in Figure 3.58 and Figure 3.59. The results are shown in Figure 4.15 and Figure 4.16.

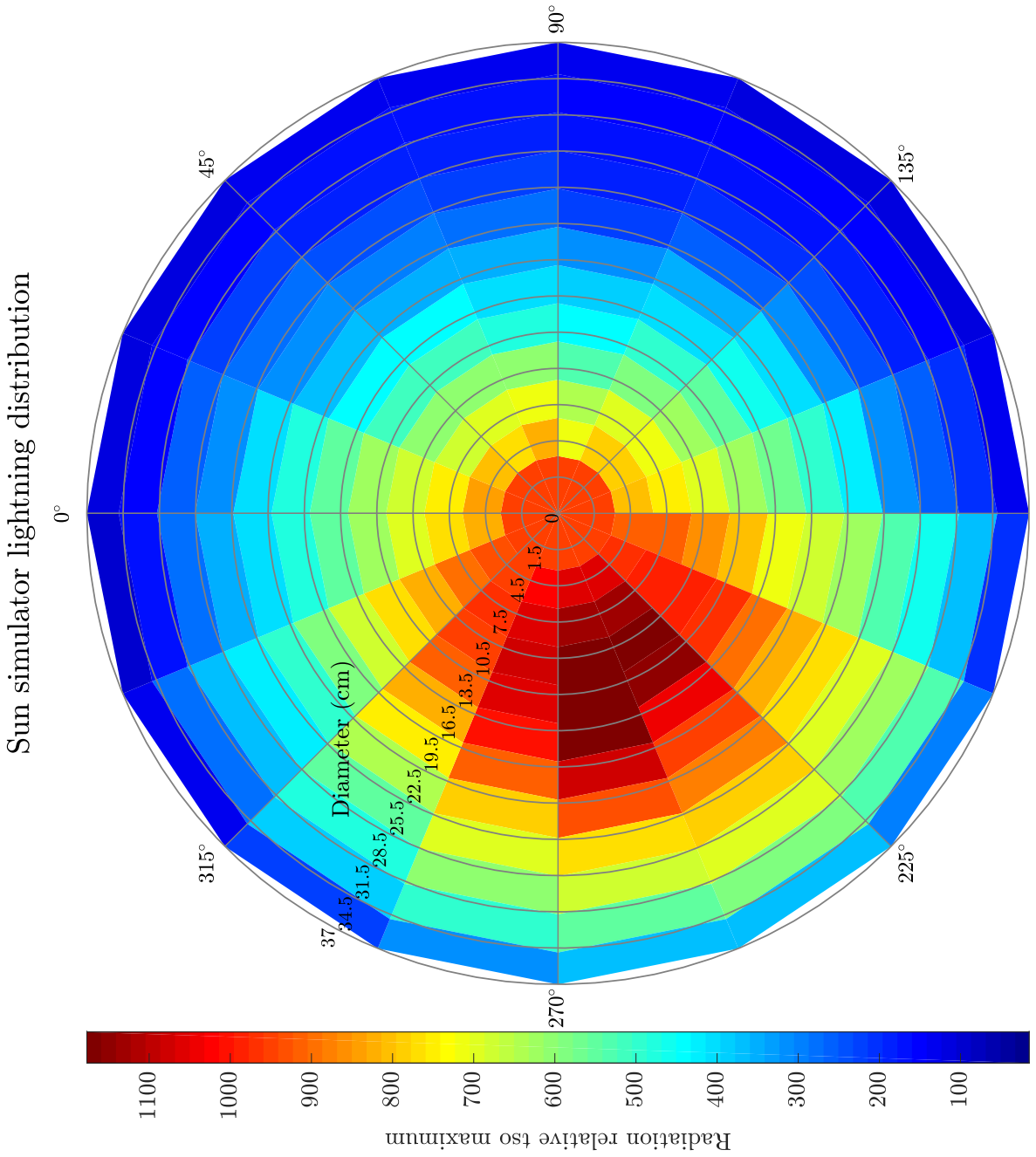


Figure 4.15 – *Lightning distribution (raw data)*

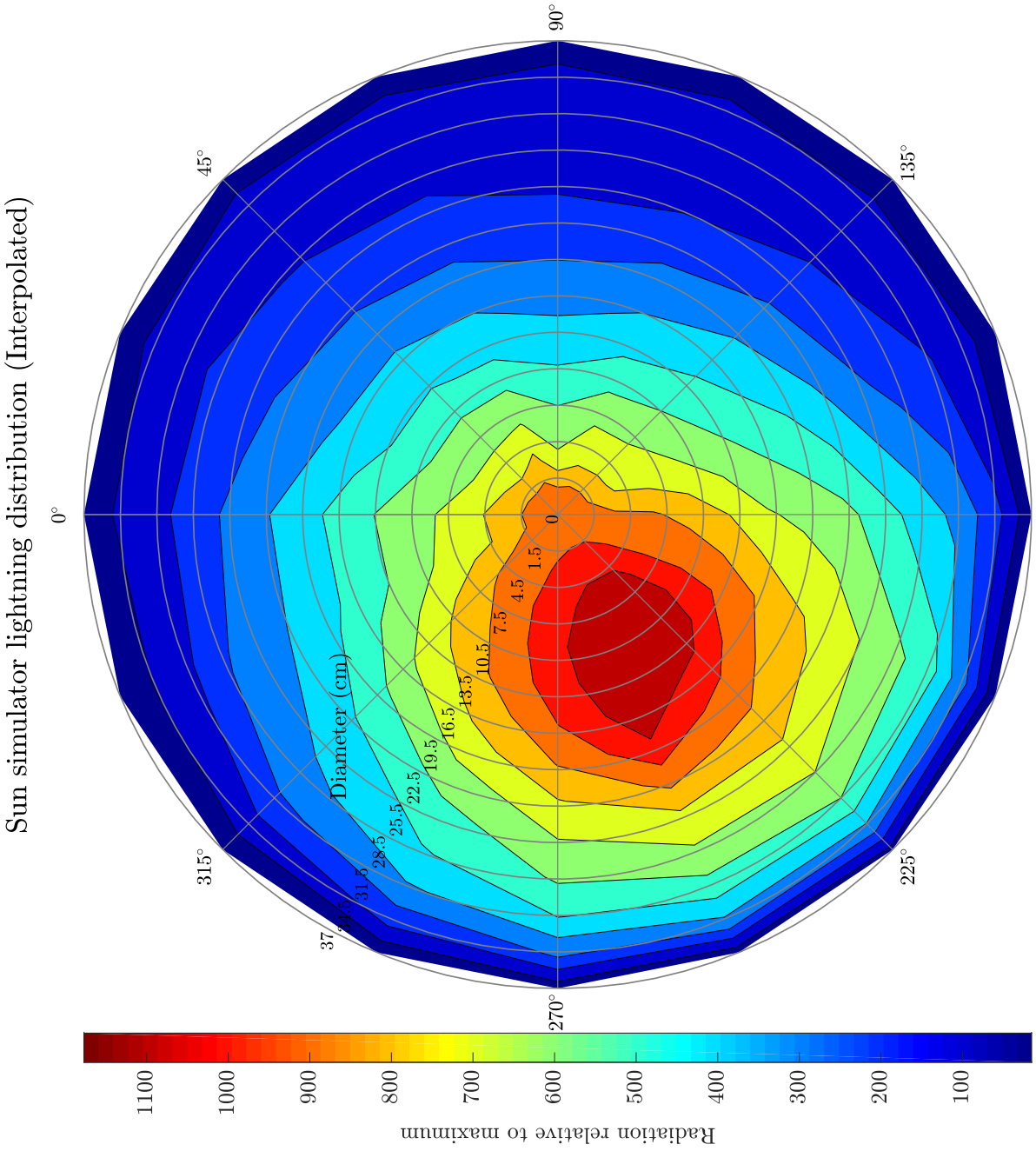


Figure 4.16 – *Interpolated Lightning distribution*

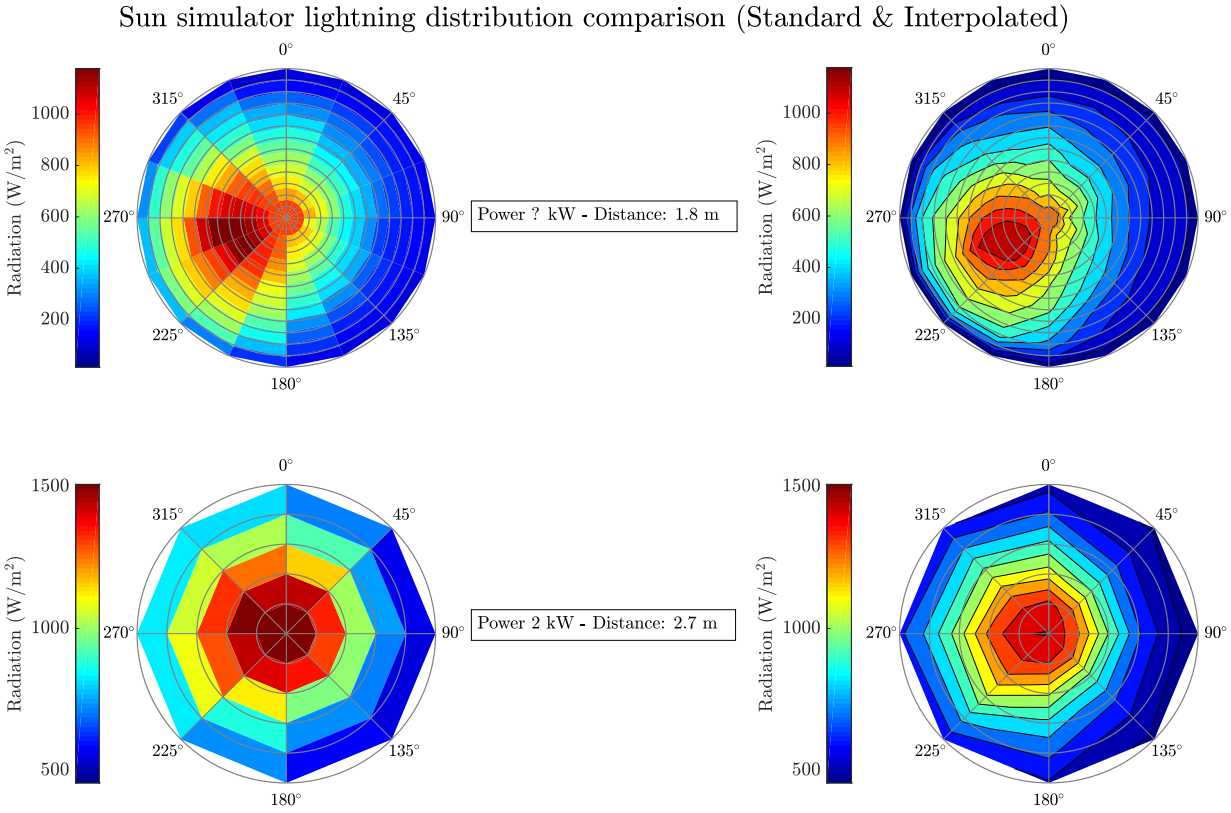


Figure 4.17 – *Lightning distribution comparison*

As in Chapter 3, the measurements are plotted twice, one with the *raw* data in Figure 4.15 and another interpolating the data, allowing a more understandable graphic in Figure 4.16. In order to ease the comparison, they are both plotted together in Figure 4.17. The ones above correspond to the simulator at GranaSAT laboratory and the ones below to the one used in [98].

The first conclusion which can be extracted in comparison with the simulator used in [98] is a noticeably less powerful xenon bulb in my case, even at a shorter distance (1.8 m vs. 2.7 m) the maximum irradiance is 33 % lower than his. By inspection at Figure 4.13, the equivalent irradiance at 2.8 m is about 500 W/m^2, once again one third lower. Therefore, although the power of the xenon bulb used is unknown, assuming similar losses, it can be estimated around **650 W**. Another possibility is that the bulb has reached its end-of-life and power gets lower.

As stated in 3.4.3.1.4.2, the closer the simulated is used, the greater the beam gets off-centered; indeed, as seen in Figure 4.17, the beam of the simulator in our case is clearly off-centered, even more so than the other, at a distance of about 5 cm from the center, within the range between 180° and 290°. Therefore, there are two possibilities in order to use this Sun simulator:

- Use it at greater distances, so the beam gets centered, sacrificing power, which will get below the 1100 W/m^2. Allows characterizing the solar panel in the center of the test template (see Figure 4.14) but it is not useful for real mission simulations.

- Use it at closer distances, with an off-centered beam as depicted in previous plots, but with realistic irradiance. Allows emulating Sun's irradiance, needed in order to check functioning in a real mission scenario, but needs to place the solar panel in the area where the beam focus, stated before.

The choice will depend on the purpose of the simulation.

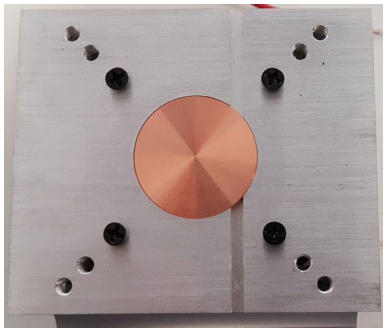
4.1.3.2 LED Sun Simulator

Because of its size, the xenon arc simulator from 4.1.3.1 cannot be used along with the inertial platform designed. Therefore, in this section it will be characterized another technology previously analyzed in Chapter 3, LED. According to the requirements, it has to be usable with the platform, so it must be small-sized. It is proposed using a low-cost thermally-bonded LED lamp, such as the one shown in Figure 4.18.

Figure 4.18 – *LED proposed*

It is a 100 W lamp to be supplied between 20 V and 32 V, with a maximum consumption of 3 A. It may reach high temperatures so it needs cooling; besides, as it is desired a lightning beam as concentrated as possible, it raises the need for a **collimator**. To satisfy these needs, it is used a kit designed for this kind of lamps, which includes both, a collimator and a 12 V fan. It is shown in Figure 4.19. When it is completely assembled, the result is the one which could be seen in Figure 4.7.

(a) *Collimator and body*



(b) *Copper base to cool the LED*

(c) *Fan*

Figure 4.19 – *Kit used for the LED lamp*

4.1.3.2.1 Spectral response

The spectral response of the LED is measured using the same procedure detailed in 4.1.3.1.1. The results are plotted in Figure 4.20, along with the Sun's.

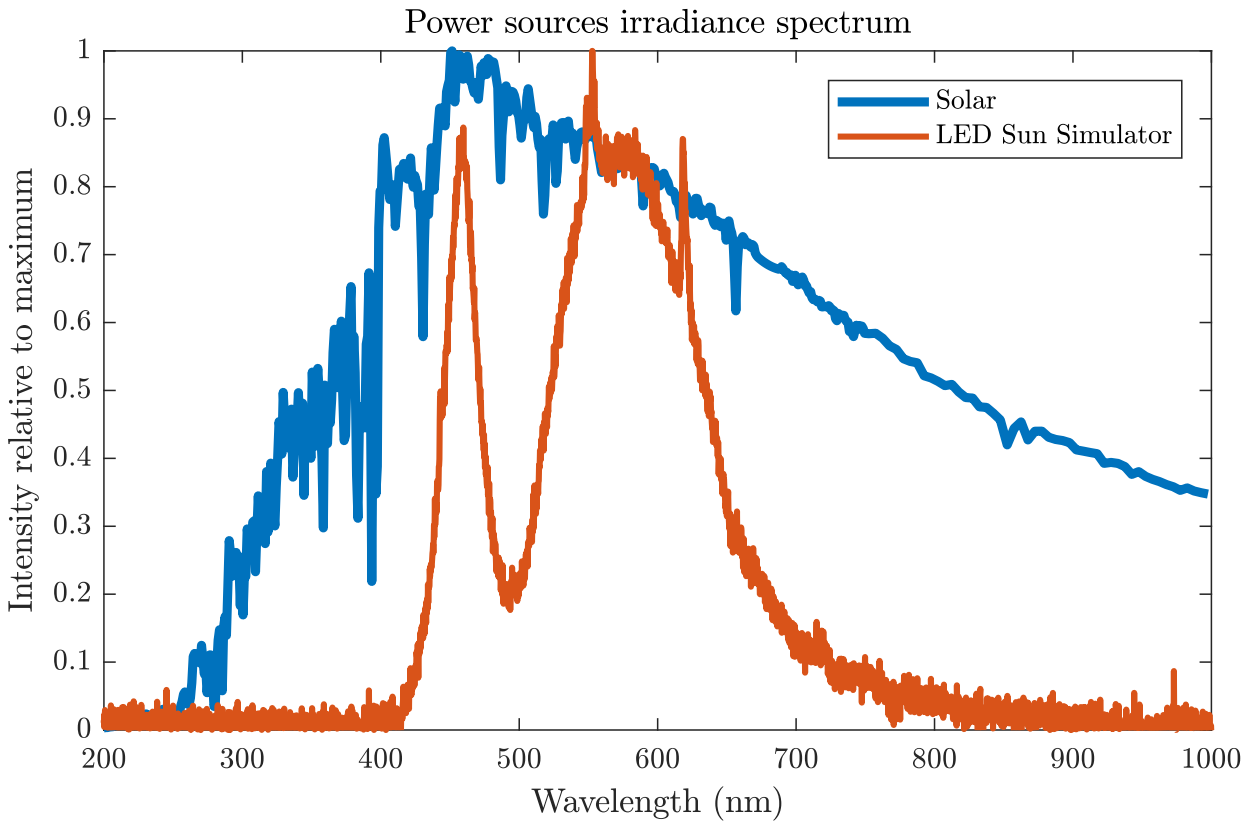


Figure 4.20 – *Spectrum comparison*

In this case, the LED spectrum covers the range 400 nm to 650 nm, with a pronounced peak at 500 nm in which almost no intensity is irradiated. Therefore, the range of a typical solar panel is slightly covered and in order to get at least an acceptable efficiency, a really high power would be needed.

4.1.3.2.2 Irradiance and consumption

Irradiance is measured again using the same pyranometer. As this lamp is intended to be used with the inertial platform at an approximately fixed distance, it will be characterized at that single point, **20 cm far**. Figure 4.21 shows part of the assembly needed to perform the measurement.

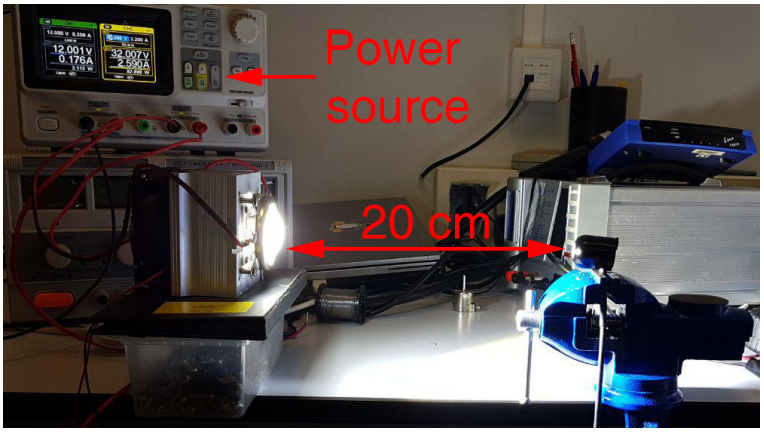


Figure 4.21 – *LED irradiance measurement*

The measurement needs to control simultaneously two devices: on the one hand, the power source, which must sweep between 20 V and 32 V and on the other hand, the oscilloscope to check for signal integrity first and perform the measurements. Specifically, the power source used is a **Siglent SPD3303X** while the oscilloscope is an **Agilent MXO-X-4104A**, both are shown in Figure 4.22.

(a) *Siglent SPD3303X [47]*

(b) *Agilent MXO-X-4104A [25]*

Figure 4.22 – *Measurement devices*

Firstly, the irradiance is measured along with the LED current consumption. In order to get it done synchronically, the following Python code is developed.

```
import visa
from visa import constants
import vxi11
import csv
import pandas as pd
import time
import math
```

```
import os
import numpy as np

# GPIB INIT
# visa.log_to_screen()
SG =  vxi11.Instrument("192.168.1.119")
OSC = visa.ResourceManager('@py').get_instrument('TCPIPo
 ::192.168.1.121::insto::INSTR')
OSC.timeout=2500000

# IDENTIFYING
print("SG found: " + SG.ask("*IDN?").strip())
print("OSC found: " + OSC.query('*IDN?').strip())

#OSC Setting-up

OSC.write(':CHANnel1:DISPlay ON')
OSC.write(':DISPlay:SIDebar MEASurements')
OSC.write(':MEASure:VPP CHANnel1')

#SG Setting-up

SG.ask("CH1:VOLT 12")
SG.ask("CH2:VOLT 20")

SG.ask("CH1:CURRent 0.3")
SG.ask("CH2:CURRent 3.2")

SG.ask("OUTPut CH1,ON")
SG.ask("OUTPut CH2,ON")

volt_Sweep=np.arange(20,32.5,0.5)
current_Sweep=[]
measured_Osc=[]

#Allow time for the measurement to stabilize at the final
  values

for V in volt_Sweep:

  SG.ask("CH2:VOLT %f" %V)
```

```
    if V==31 or V==31.5 or V==32:
49
        time.sleep(60)
    else:
52
        time.sleep(1)
55
    current_Sweep.append(SG.ask("MEASure:CURRent? CH2"))
    measured_Osc.append(OSC.query(':MEASure:VRMS? CHANnel1'
  ))
58

voltage_Data=pd.DataFrame(volt_Sweep)
61 current_Data=pd.DataFrame(current_Sweep)
measured_Voltage_Data=pd.DataFrame(measured_Osc)

64 pd.concat([voltage_Data,current_Data,
 measured_Voltage_Data],axis=1).to_csv("data5.csv")

print("Data written to CSV")
67
SG.ask("OUTPut CH1,OFF")
SG.ask("OUTPut CH2,OFF")
```

Code 4.1 – *Polling based measurements script*

The results are plotted in Figure 4.23. The tendency is clearly lineal, as expected; the higher the current, the greater the irradiance, until reaching the maximum current allowed by the LED. The highest irradiance accomplished by the simulator at that distance (20 cm) is 235 W/m^2. It is about five times lower than the one received at Earth's surface, therefore, it can be expected a poor performance from the solar panels with this irradiance. Of course it is also way lower than the value produced by the xenon arc lamp at that distance. However, that comparison is not realistic; while this LED is about 80 W (see Figure 4.24), the xenon arc simulator has an estimated power of about 600 W and is not intended to work at such a short distance. **Their simulation purposes are different.**

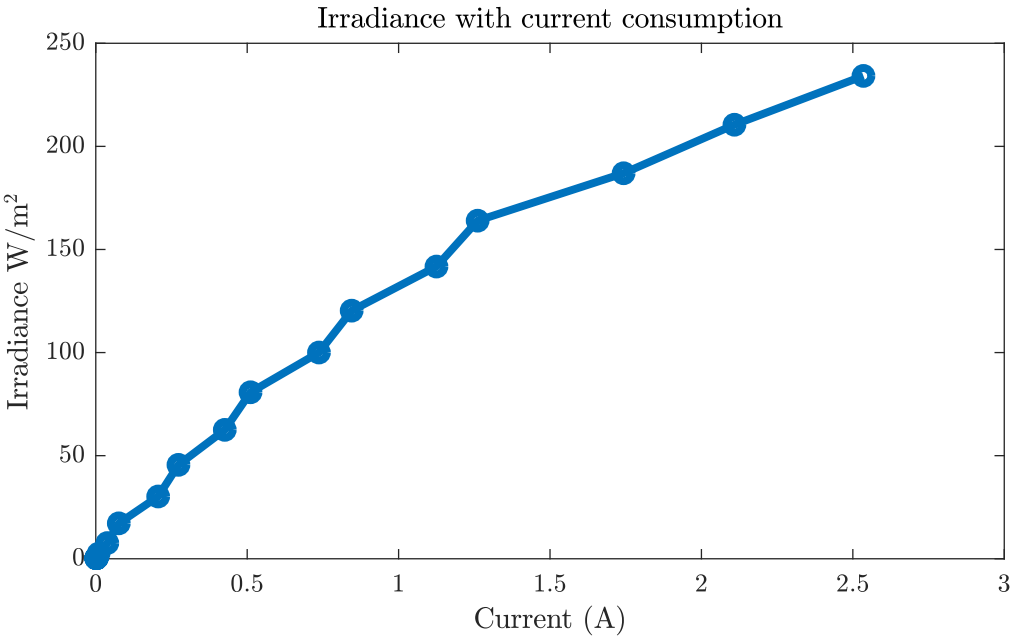


Figure 4.23 – *Irradiance with LED current consumption*

However, the irradiance is not linear with **voltage supply**, because of the exponential relationship between voltage and current in a LED; indeed, neither the functioning range given by the manufacturer nor the power drawn are completely correct. This can be concluded from the plot in Figure 4.24, which relates I-V curve of the LED with the power drawn.

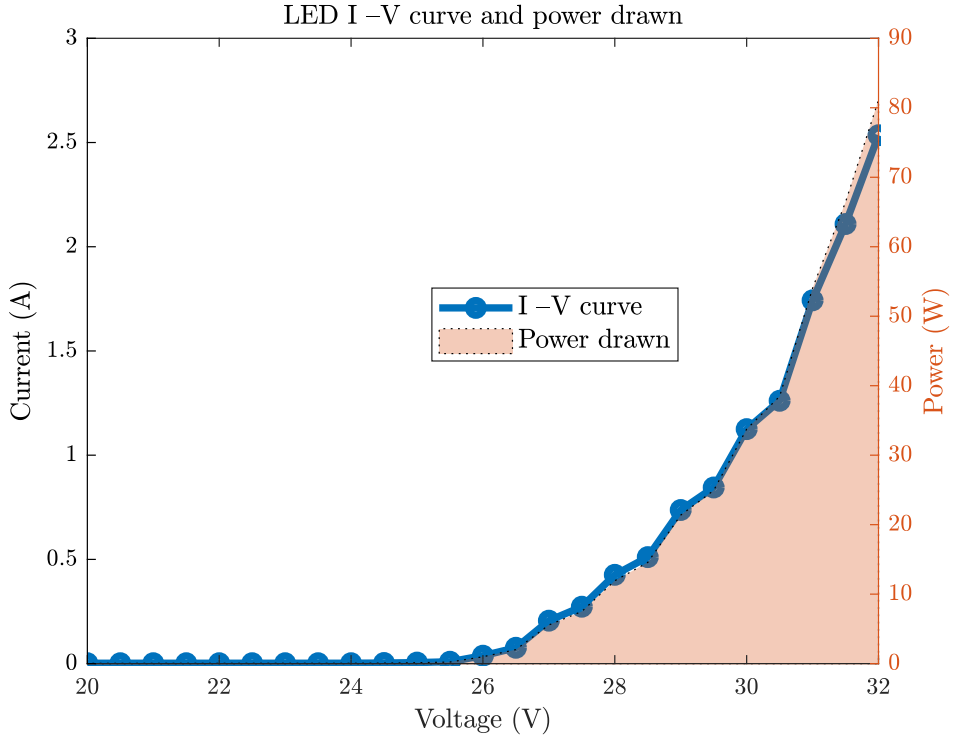


Figure 4.24 – *I-V curve and power drawn*

That plot is important because it allows checking the real requirements of the lamp and therefore, of the LED driver. It can be concluded that the LED does not start lightning until supplied with 26 V, and the maximum power drawn is about 80 W, unlike the range stated by the manufacturer 20-32 V and 100 W power, respectively. The power losses were obviously expected, but an under consumption of 20 % is worth noting. Although the expected maximum is not reached, the lamp heats considerably, which makes it necessary to install the fan mentioned before. Even when the fan is on, the temperature borders on 100 °C, as shown in Figure 4.25.

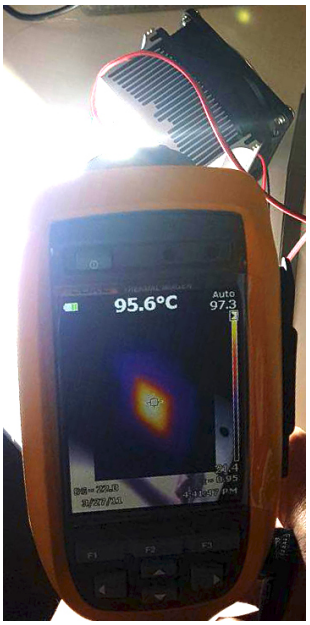


Figure 4.25 – *Measured temperature on the LED*

In sum, in order to supply the LED Sun simulator, it is necessary a **driver** with an output voltage about 32 V and 100 W of power capability. Figure 4.26 shows the chosen one.

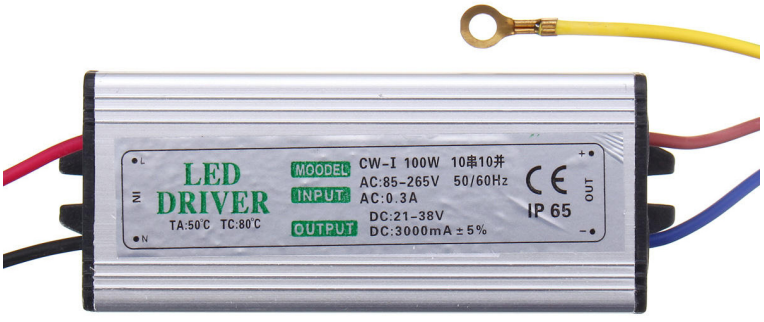


Figure 4.26 – *Low cost 100 W LED driver*

4.1.3.3 Comparative

The irradiance sources characterization finalizes comparing the spectrums of the proposed technologies with the Sun's. Figure 4.27 plots this comparison with all the data normalized to the Sun's spectrum irradiance, in order to ease the spectral visualization.

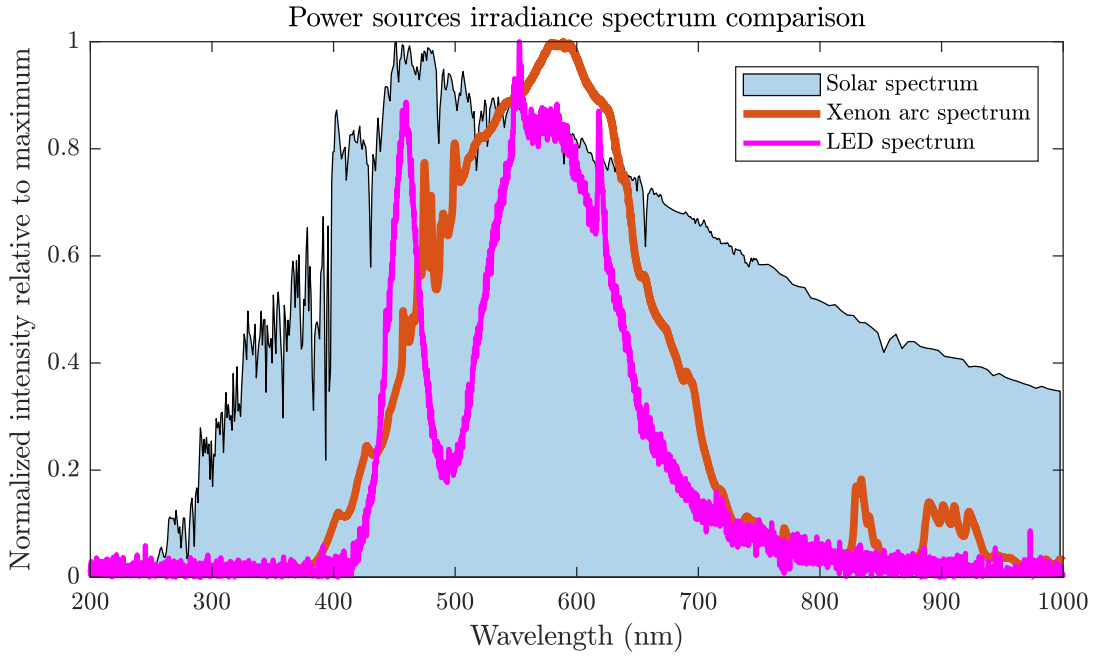


Figure 4.27 – *Spectral comparison of the different irradiance sources*

Xenon arc and LED spectrums are actually pretty similar, except for the 500 nm area in the latter. However, from Figure 4.27 and previous plots it is easy to conclude that **none of them matches the Sun's**, neither spectrum nor power. That is not necessarily a problem, though; the LED is intended to be used along with I2DOS so its final purposes are not strictly real simulation. When that is required however, it will be needed to use a different xenon arc simulator. Even if spectral matching is enough, low power and off-centered beam makes it unreliable for real missions simulations.

4.2 Ground Station

Regarding the Ground Station, this Master's Thesis will address the design and launch of the central controller. It will be based on the environment **Ball Aerospace COSMOS** and will count with different user interfaces and a basic telemetry database. It is intended to function as the main point of control and

communications with the CubeSat.

The designed Ground Station software will be expandable so it can be used as it would in a real mission.

4.2.1 Ball Aerospace COSMOS

COSMOS is a complete environment designed by Ball Aerospace which is composed of some sample applications and allows defining new interfaces and functionalities, covering the **full lifecycle** of the mission, as the same interface can be used from test to operation. It is written in **Ruby** and provides with a command and control system to interact with a variety of embedded systems such as a CubeSat but also electronics equipment such as a radio transmitter. It is also compatible with international standards analyzed before like CCSDS or XTCE and is **open source**.

Using COSMOS allows adding a new abstraction layer to develop the tools needed by the mission focusing on the required functionalities, instead of rewriting existing tools. This makes it a really powerful and productive tool, which has been widely used by international institutions including NASA, particularly the Goddard Space Flight Center. Implementing it in GranaSAT laboratory gives future students the valuable opportunity to work and train themselves using real cutting-edge technology, which satisfies the academic perspective of this Master's Thesis.

4.2.1.1 Design and configuration

Once installed, COSMOS design and configuration is based on plain text files which can be either modified directly or using a dedicated tool included with COSMOS, as it is proposed. Figure 4.28 displays that configuration editor.

Figure 4.28 – *Configuration editor*

Next, some of the most important concepts about COSMOS are briefly introduced and configured.

- **Target**

Device or spacecraft intended to establish a communication. As for this project, there will be only one target, the designed CubeSat. Firstly, they have to be **declared**; this can be done in the **system.txt** file, under the following path.

```
/GranaSAT-GroundStation
└─ +config
   └─ system
      └─ system.txt
```

Besides, it is necessary to create a folder with the name of the target and replicate the following folder structure:

```
/GranaSAT-GroundStation
└─ +config
   └─ +targets
      └─ GRANASAT-I
         └─ +cmd-tlm
            └─ target.txt
```


The file *target.txt* must contain the name of the database with the telemetry and telecommands information.

- **Interface**

In COSMOS, interfaces refer to the protocol used to communicate with a certain target, i.e., TCP/IP, serial or any other defined. Particularly, TCP/IP is already defined and is very convenient to use in a local environment such as the one used. They are defined in the file *cmd-tlm-server.txt* under the following path.

```
/GranaSAT-GroundStation
└─ +config
   └─ +tools
      └─ +cmd-tlm-server
         └─ cmd-tlm-server.txt
```

Figure 4.29 illustrates the information needed in that file. Further information on the different options can be found on the official website [62].

```
TITLE 'GRANASAT-I Command and Telemetry Server'

#AUTO_INTERFACE_TARGETS # Use cmd_tlm_server.txt from all targets declared in system.txt

INTERFACE TCPIP_8445 tcpip_server_interface.rb 8445 8445 10.0 nil BURST
  TARGET GRANASAT_I



ROUTER TCPIP_8445_ROUTER tcpip_server_interface.rb 8505 8505 10.0 nil BURST
          ROUTE TCPIP_8445
```

Figure 4.29 – *Interface configuration screenshot*

4.2.1.2 Telecommand & Telemetry database

The whole framework revolves around the database with the telemetry and telecommand information. Once again, it can be defined using plain text; however, COSMOS is also compatible with XTCE format, with an XML syntax. Once the implementation is decided, the data structure can be standardized using CCSDS. Out of simplicity, in this Master's Thesis the database will be implemented using XTCE and CCSDS, but this one only partially. Expanding the database using completely that format is proposed as a future line of work, specially when working on a real mission.

There are countless XML editors, any of which can be used to create the needed database. It is proposed **Altova XMLSpy** because of the variety of different representations it allows. Code 4.2 shows an example of telemetry packets defined using XTCE.

```
<xtce:FloatParameterType name="OBC_Bosch_Temperature"
 shortDescription="OBC_Bosch_Temperature" signed="true">
        <xtce:UnitSet>
          <xtce:Unit description="deg">C</xtce:Unit>
        </xtce:UnitSet>
        <xtce:FloatDataEncoding sizeInBits="32" encoding=
 "signed" bitOrder="LITTLE_ENDIAN">
          </xtce:FloatDataEncoding>
      </xtce:FloatParameterType>
      <xtce:FloatParameterType name="ESP_Data_Rate"
 shortDescription="ESP_Data_Rate" signed="true">
        <xtce:UnitSet>
          <xtce:Unit description="Mb/s">Mb/s</xtce:Unit>
        </xtce:UnitSet>
        <xtce:FloatDataEncoding sizeInBits="32" encoding=
```

```
"signed" bitOrder="LITTLE_ENDIAN">
            </xtce:FloatDataEncoding>
        </xtce:FloatParameterType>
        <xtce:FloatParameterType name="ESP_Signal_Level"
shortDescription="ESP_Signal_Level" signed="true">
            <xtce:UnitSet>
              <xtce:Unit description="dBm">dBm</xtce:Unit>
            </xtce:UnitSet>
            <xtce:FloatDataEncoding sizeInBits="32" encoding=
"signed" bitOrder="LITTLE_ENDIAN">
            </xtce:FloatDataEncoding>
```

Code 4.2 – *Example of telemetry implemented using XTCE*

Figure 4.30 shows part of the implemented database with one of the visual representations provided by Altova.

xtce:TelemetryMetaData

xtce:ParameterTypeSet

xtce:IntegerParameterType (2)

	name	shortDescription	signed	sizeInBits	xtce:IntegerDataEncoding
1	PACKET_DATA_LENGTH	CCSDS packet data length	false	8	xtce:IntegerDataEncoding sizeInBits=8 encoding=unsigned bitOrder=LITTLE_ENDIAN
2	PACKET_ID_Data_Type	Packet subsytem identification	false	8	xtce:IntegerDataEncoding sizeInBits=8 encoding=unsigned bitOrder=LITTLE_ENDIAN

xtce:FloatParameterType (21)

	name	shortDescription	signed	xtce:UnitSet	xtce:FloatDataEncoding
1	wX	Angular speed of the x axis	true	xtce:UnitSet	xtce:FloatDataEncoding sizeInBits=32 encoding=signed bitOrder=LITTLE_ENDIAN
2	wY	Angular speed of the y axis	true	xtce:UnitSet	xtce:FloatDataEncoding sizeInBits=32 encoding=signed bitOrder=LITTLE_ENDIAN
3	wZ	Angular speed of the z axis	true	xtce:UnitSet	xtce:FloatDataEncoding sizeInBits=32 encoding=signed bitOrder=LITTLE_ENDIAN
4	bdotX	Intensity of the magnetic field of the x axis	true	xtce:UnitSet	xtce:FloatDataEncoding sizeInBits=32 encoding=signed bitOrder=LITTLE_ENDIAN
5	bdotY	Intensity of the magnetic field of the y axis	true	xtce:UnitSet	xtce:FloatDataEncoding sizeInBits=32 encoding=signed bitOrder=LITTLE_ENDIAN
6	bdotZ	Intensity of the magnetic field of the z axis	true	xtce:UnitSet	xtce:FloatDataEncoding sizeInBits=32 encoding=signed bitOrder=LITTLE_ENDIAN
7	aX	Acc of the x axis	true	xtce:UnitSet	xtce:FloatDataEncoding sizeInBits=32 encoding=signed bitOrder=LITTLE_ENDIAN
8	aY	Acc of the y axis	true	xtce:UnitSet	xtce:FloatDataEncoding sizeInBits=32 encoding=signed bitOrder=LITTLE_ENDIAN
9	aZ	Acc of the z axis	true	xtce:UnitSet	xtce:FloatDataEncoding sizeInBits=32 encoding=signed bitOrder=LITTLE_ENDIAN
10	ADC1	OBC_ADC_1	true	xtce:UnitSet	xtce:FloatDataEncoding sizeInBits=32 encoding=signed bitOrder=LITTLE_ENDIAN
11	ADC2	OBC_ADC_2	true	xtce:UnitSet	xtce:FloatDataEncoding sizeInBits=32 encoding=signed bitOrder=LITTLE_ENDIAN
12	ADC3	OBC_ADC_3	true	xtce:UnitSet	xtce:FloatDataEncoding sizeInBits=32 encoding=signed bitOrder=LITTLE_ENDIAN
13	ADC4	OBC_ADC_4	true	xtce:UnitSet	xtce:FloatDataEncoding sizeInBits=32 encoding=signed bitOrder=LITTLE_ENDIAN
14	ADC5	OBC_ADC_5	true	xtce:UnitSet	xtce:FloatDataEncoding sizeInBits=32 encoding=signed bitOrder=LITTLE_ENDIAN
15	ADC6	OBC_ADC_6	true	xtce:UnitSet	xtce:FloatDataEncoding sizeInBits=32 encoding=signed bitOrder=LITTLE_ENDIAN
16	OBC_RPI_Temperature	OBC_RPI_Temperature	true	xtce:UnitSet	xtce:FloatDataEncoding sizeInBits=32 encoding=signed bitOrder=LITTLE_ENDIAN
17	OBC_Bosch_Temperature	OBC_Bosch_Temperature	true	xtce:UnitSet	xtce:FloatDataEncoding sizeInBits=32 encoding=signed bitOrder=LITTLE_ENDIAN
18	OBC_Bosch_Press	OBC_Bosch_Pressure	true	xtce:UnitSet	xtce:FloatDataEncoding sizeInBits=32 encoding=signed bitOrder=LITTLE_ENDIAN
19	ESP_Data_Rate	ESP_Data_Rate	true	xtce:UnitSet	xtce:FloatDataEncoding sizeInBits=32 encoding=signed bitOrder=LITTLE_ENDIAN
20	ESP_TX_Power	ESP_TX_Power	true	xtce:UnitSet	xtce:FloatDataEncoding sizeInBits=32 encoding=signed bitOrder=LITTLE_ENDIAN
21	ESP_Signal_Level	ESP_Signal_Level	true	xtce:UnitSet	xtce:FloatDataEncoding sizeInBits=32 encoding=signed bitOrder=LITTLE_ENDIAN

Figure 4.30 – *Implemented telemetry and telecommand database using XTCE format*

4.2.1.3 Interface

Once the configuration parameters are set, COSMOS allows defining different user interfaces following the same procedure. For example, Figure 4.31 displays the main interface of the developed Ground Station software.

It shows the different tools developed in this Master's Thesis. It is divided into three differentiated areas: the first one, which initiates the controller, the second one 'Commanding and Scripting´, with the tools necessary to send commands to the CubeSat and the last one 'Telemetry´, with tools to visualize and post-process the telemetry. Some of them are briefly described next.

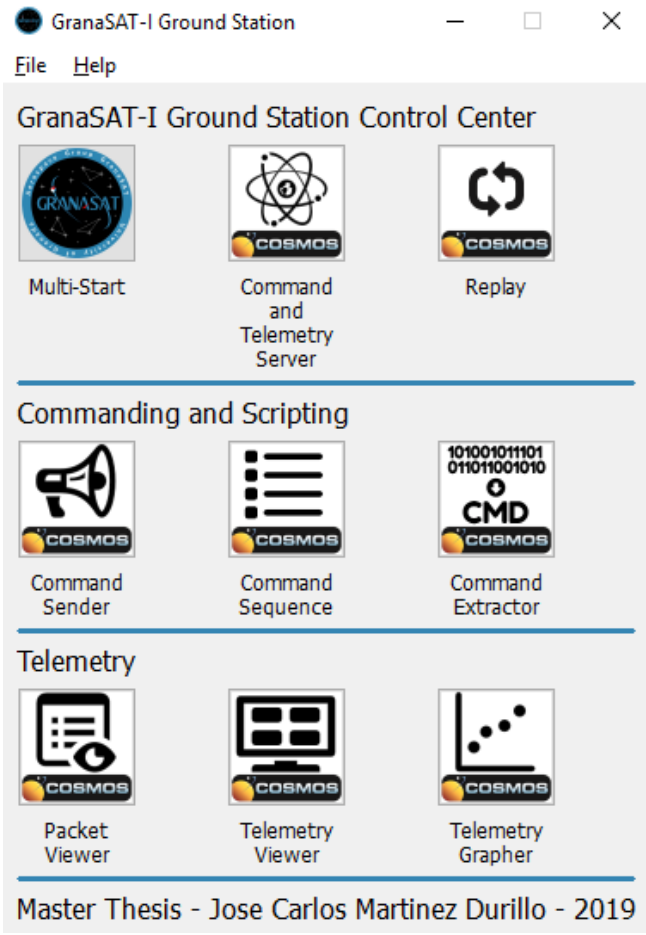


Figure 4.31 – *Main interface of the GranaSAT Ground Station controller*

- **Command and Telemetry Server:** Used to connect to the different targets. It shows information regarding the number of packages received, errors detected and so forth.

- **Replay:** Used to *repeat* a past mission. By loading the log of a certain session, it allows repeating the processing performed in real time, plot the information again with a temporal reference, etc. Figure 4.32 shows the interface of this tool.

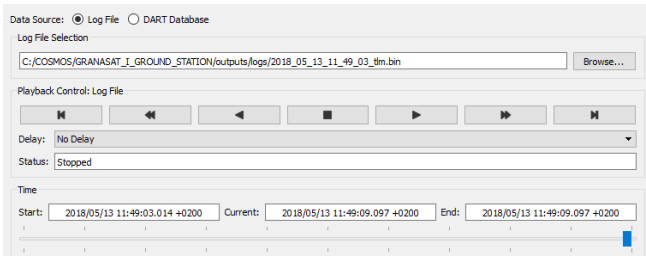
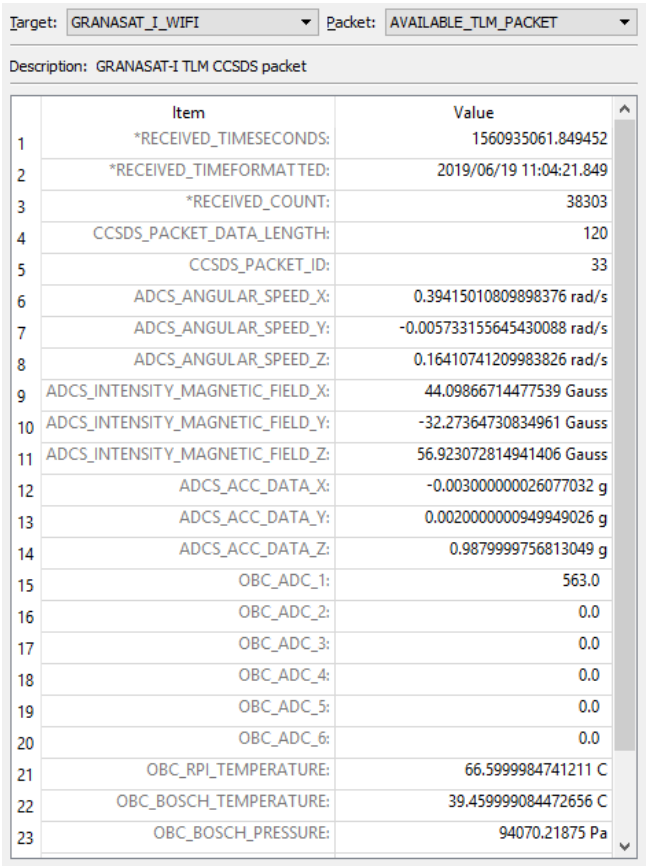


Figure 4.32 – *Replay tool*

- **Packet viewer:** Used to visualize the telemetry data as received, either in real time or using the **Replay** tool. Figure 4.33 shows an example of received packet in this tool.


Target: GRANASAT_I_WIFI Packet: AVAILABLE_TLM_PACKET

Description: GRANASAT-I TLM CCSDS packet

	Item	Value
1	*RECEIVED_TIMESECONDS:	1560935061.849452
2	*RECEIVED_TIMEFORMATTED:	2019/06/19 11:04:21.849
3	*RECEIVED_COUNT:	38303
4	CCSDS_PACKET_DATA_LENGTH:	120
5	CCSDS_PACKET_ID:	33
6	ADCS_ANGULAR_SPEED_X:	0.39415010809898376 rad/s
7	ADCS_ANGULAR_SPEED_Y:	-0.005733155645430088 rad/s
8	ADCS_ANGULAR_SPEED_Z:	0.16410741209983826 rad/s
9	ADCS_INTENSITY_MAGNETIC_FIELD_X:	44.09866714477539 Gauss
10	ADCS_INTENSITY_MAGNETIC_FIELD_Y:	-32.27364730834961 Gauss
11	ADCS_INTENSITY_MAGNETIC_FIELD_Z:	56.923072814941406 Gauss
12	ADCS_ACC_DATA_X:	-0.003000000026077032 g
13	ADCS_ACC_DATA_Y:	0.0020000000949949026 g
14	ADCS_ACC_DATA_Z:	0.9879999756813049 g
15	OBC_ADC_1:	563.0
16	OBC_ADC_2:	0.0
17	OBC_ADC_3:	0.0
18	OBC_ADC_4:	0.0
19	OBC_ADC_5:	0.0
20	OBC_ADC_6:	0.0
21	OBC_RPI_TEMPERATURE:	66.5999984741211 C
22	OBC_BOSCH_TEMPERATURE:	39.459999084472656 C
23	OBC_BOSCH_PRESSURE:	94070.21875 Pa

Figure 4.33 – *Packet viewer*

Telemetry Grapher is not described because it will be widely seen in Chapter 5 to verify the functioning of the system.

4.2.2 Real time 3D viewer

Besides the COSMOS environment developed and explained in the latter section, it has been developed a web-based real time 3D viewer. Once again, it is intended to be in a local environment; in this case the reason is obvious: the communication delay existing in a real in orbit communication makes it impossible a real time display. However, it is an attractive tool for students learning about CubeSats with this platform.

The diagram of Figure 4.34 depicts the functioning and technologies used in the viewer.

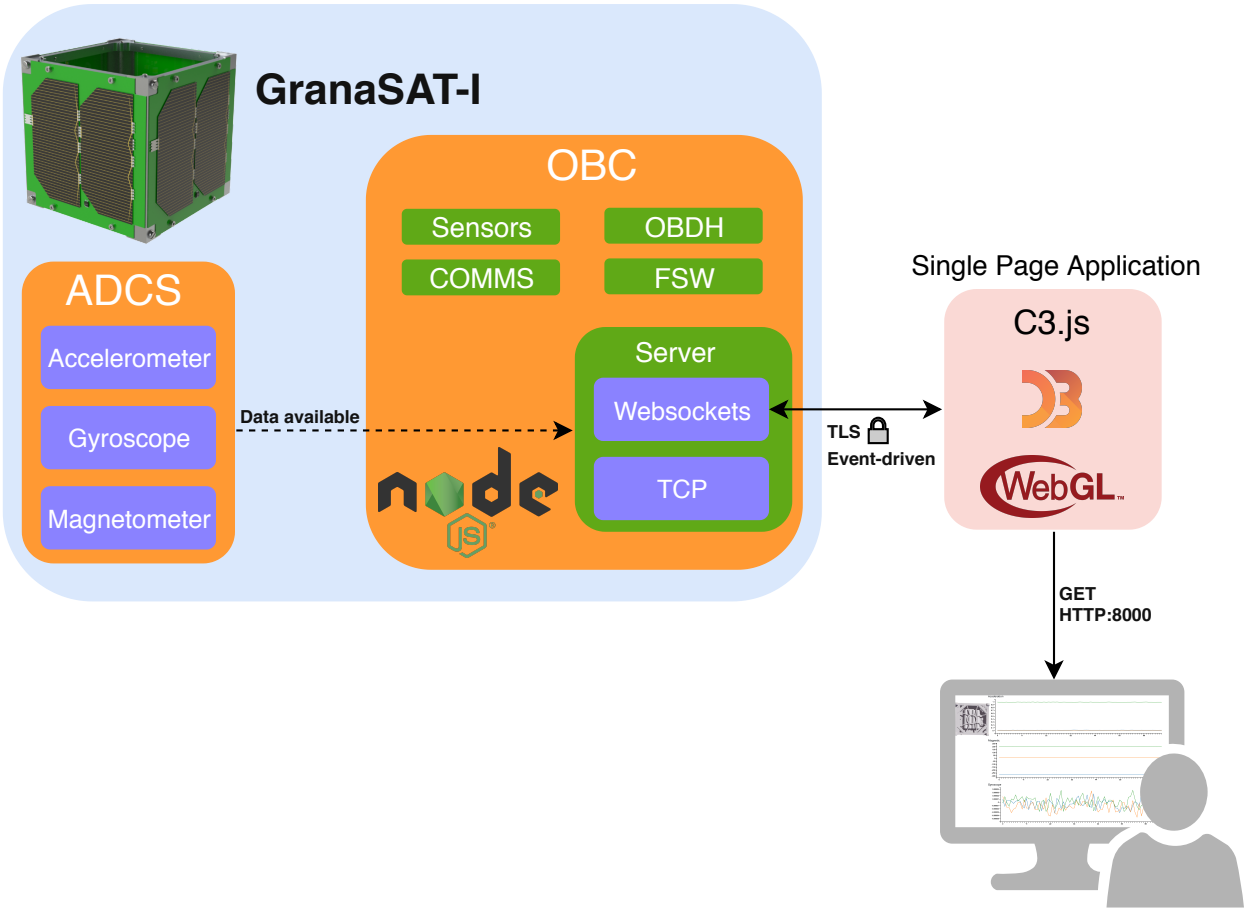


Figure 4.34 – *3D viewer architecture*

The proper CubeSat hosts in the OBC, along with the rest of components (OBDH, Flight Software...), a **Node.js** [34] server. When the attitude-related sensors (accelerometer, gyroscope and magnetometer) have data available, they pass it directly to the web server. Once the server has the attitude information, it communicates with the user interface using secure Websockets. The user interface is designed as a Single Page Application (SPA) so it is fully downloaded just once. On the other hand, the server is listening at port 8000, waiting for connections.

The designed SPA uses three technologies to plot a real time 3D view of the CubeSat. The one in the bottom is **WebGL** [58], graphic libraries optimized for web applications, in charge of rendering the 3D visualization. Right above it, **D3.js** [9] which adds an abstraction layer over WebGL, easing the process by managing STL models, controlling light, cameras, background, etc; in sum, it is a 3D graphic library. Finally, on the top of the stack, the library **C3.js** [4] is used to generate real time 2D plots.

The website can be seen in Figure 4.35.

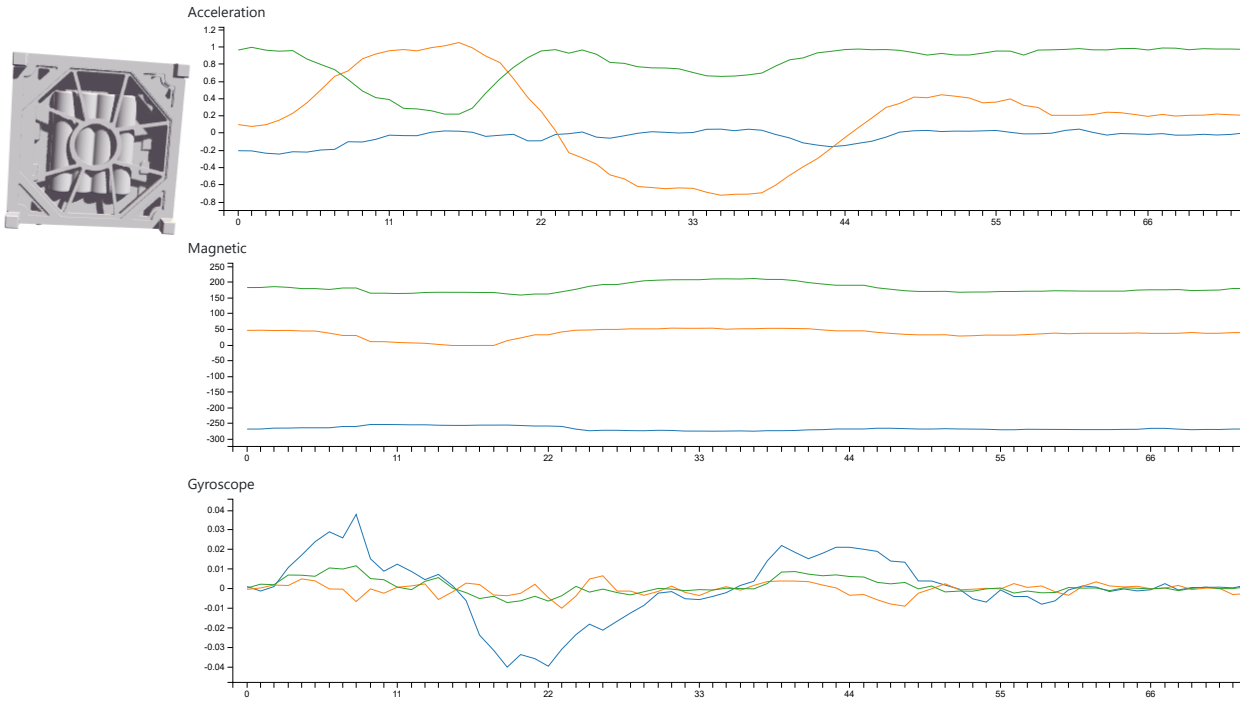


Figure 4.35 – *3D viewer web application*

4.3 Simulation CubeSat

The design phase finalizes with the simulation CubeSat, which will lay the foundations to complete a mission-ready device in the future. This section addresses all the aspects around CubeSat design, from the mechanical design of the structure to the electronics, which makes it **the most complex part of the whole platform**.

4.3.1 Mechanical structure

As described in Subsection 1.2.1, one of the most extended CubeSat standards was defined by the California Politechnical State University [5], depicted in Figure 1.7. Therefore, the design performed in this Master's Thesis will stick as much as possible to it.

4.3.1.1 Design

Once again, the mechanical structure is designed using SolidWorks®. Figure 4.37 shows the drawings of the design. On the other hand, Figure 4.36 shows another high quality 3D render.

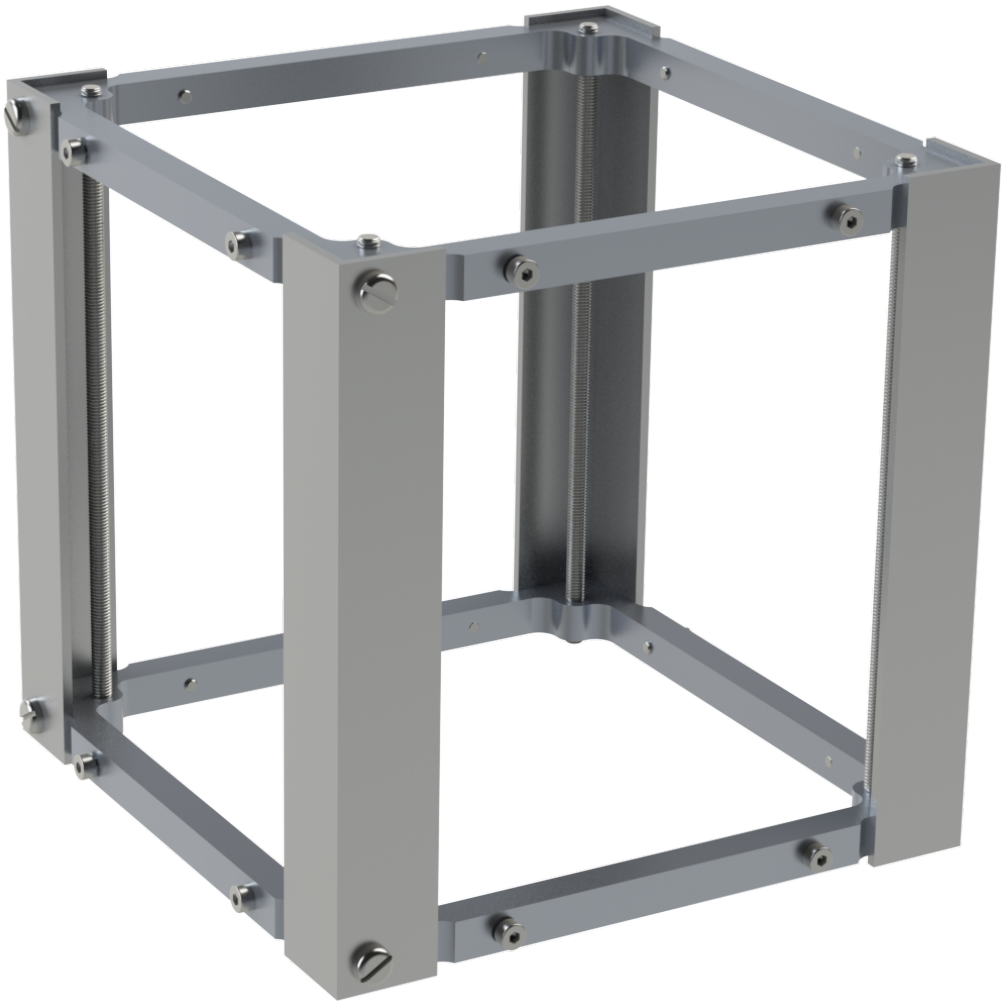


Figure 4.36 – *Mechanical structure 3D render*

Besides the outer edge design, the non-standard inner is important because it will constrain the PCB edge.

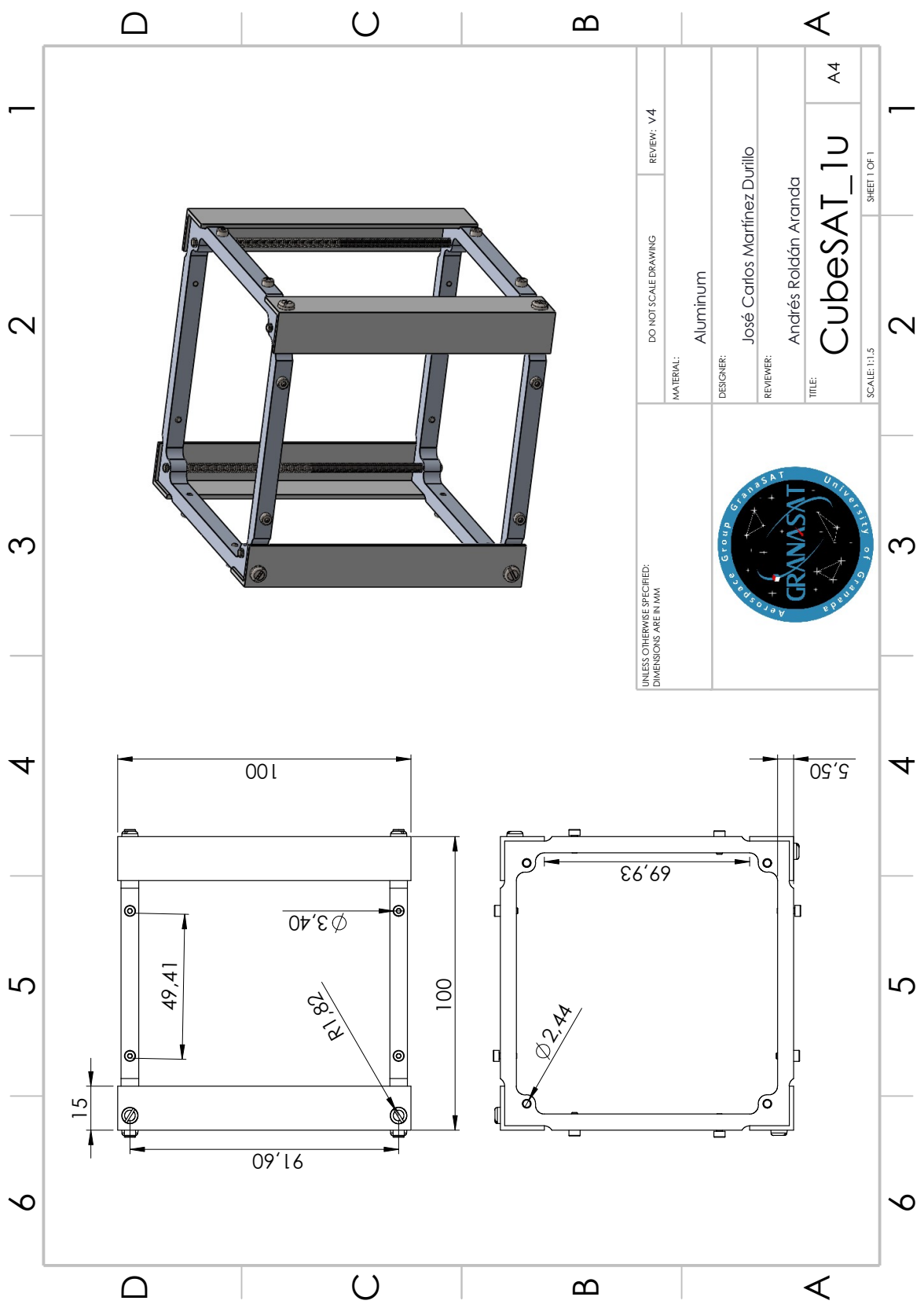


Figure 4.37 – *Mechanical structure design*

4.3.1.2 Manufacturing

The structure is made of aluminum and manufactured using a milling machine (see Figure 4.38) in our own laboratory.

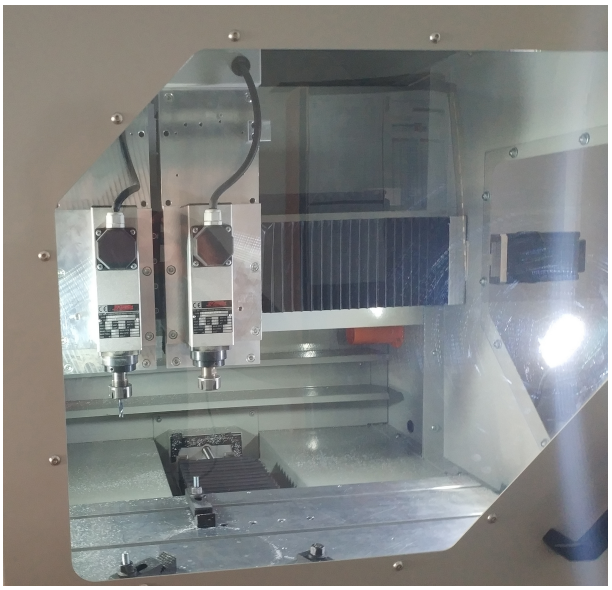
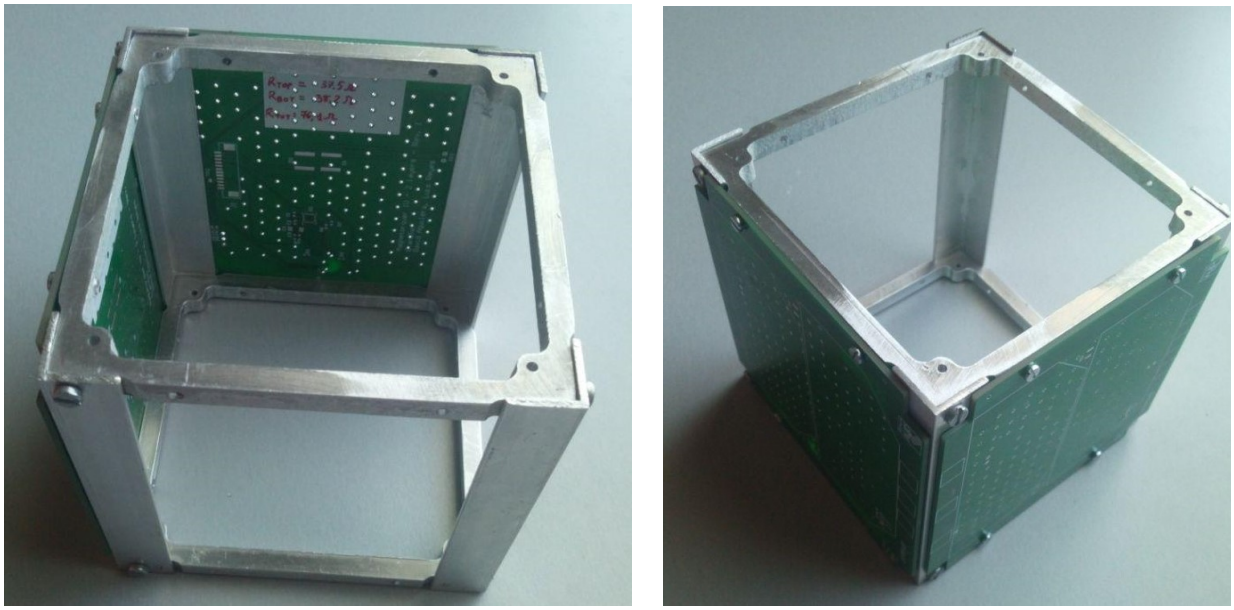


Figure 4.38 – *Aluminum milling machine*

After an iterative process, an optimized structure is accomplished, displayed in Figure 4.39. Some testing solar panels PCB have been attached in order to check the correctness of the design.



(a) *1* **(b)** *2*

Figure 4.39 – *Mechanical structure manufactured*

4.3.2 On-board computer (OBC)

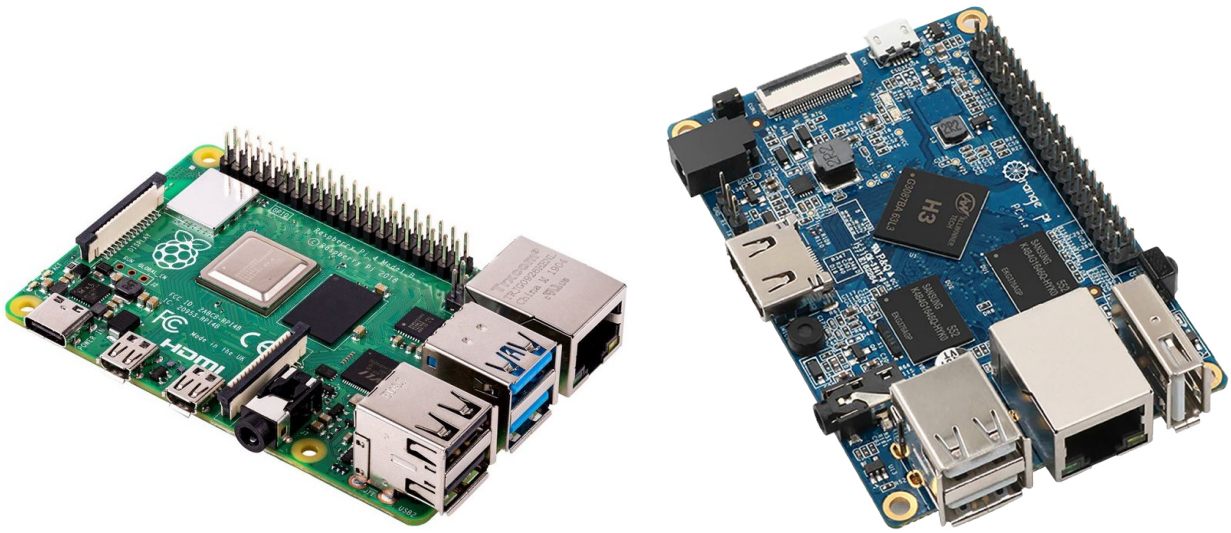
As stated in Chapter 3, the OBC is probably the most complex part of the CubeSat. It is in charge of managing and synchronizing the rest of the system **in-time** and acting consequently. This section is divided into different subsections according to the main functioning modules of the OBC.

4.3.2.1 Central Processing Unit

The Central Processing Unit of a CubeSat, typically abbreviated as CPU, can indeed be understood in the same manner of a standard computer. It is some kind of circuitry, typically a microprocessor, which performs certain instructions consisting in arithmetic, I/O operations, logic, etc.

This task can be addressed by a variety of architectures. One of the most common options is using a **microcontroller**, for example the MSP430 by Texas Instruments [54]. However, in this project the design will go one step beyond using a so-called micro-computer or more specifically an embedded **Single-Board Computer** (SBC). They are a complete computer built on a single PCB, including not only microprocessor, but also memory, RAM, or I/O capabilities. Unlike microcontrollers, SBC allows storage memory and can run an operating system completely. On the other hand, they can be more difficult to implement.

SBC can be found in a variety of formats; for instance, Figure 4.59 shows a couple of examples in *standalone* format.



(a) *Raspberry Pi 4 B+* **(b)** *Orange Pi*

Figure 4.40 – *Different SBC available on the market*

Standalone SBC just need a loaded operating system and a power supply. Although that format is convenient for hobby or *domestic* projects, as for a CubeSat project is too heavy and furthermore really difficult to fit into the structure. For these reason, this design will make use of **industrial** SBC, characterized by a more flexible form factor with standard DDR2 SODIMM connector. It increases substantially the complexity of the design but also allows a much more professional result and gets the project closer to the ticket-to-fly in the *Fly Your Satellite!* program, described in 1.2.2.

4.3.2.1.1 Comparative

In order to choose the SBC which fits best the requirements needed, it is performed a comparative between different options available in the market. The comparative is also needed because of the cost of this part, which is usually the highest of the whole CubeSat. Normally, this is also the component with the greatest power consumption, therefore an extensive analysis is performed on the option chosen.

4.3.2.1.1.1 Apalis TK1

The Apalis TK1 by Toradex [2] is based on the NVIDIA Tegra K1 SoC. It counts with a Cortex A15 quad-core CPU up to 2.1 GHz and a powerful GPI Geforce Kepler, also from NVIDIA, as well as 2 GB RAM.

Additionally, the Apalis TK1 features a low power ARM Cortex M4 up to 100 MHz which extends the ADC, GPIO and several other interfaces. It fits into a standard 200 pins SODIMM connection. Figure 4.41 shows this SBC.

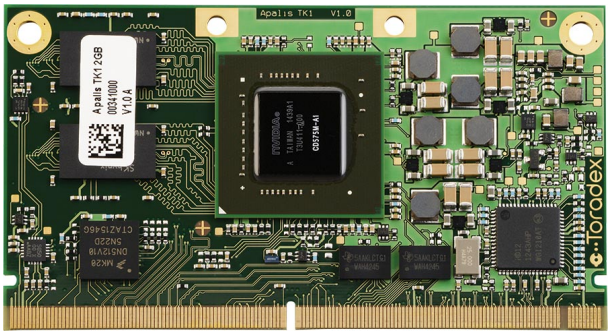


Figure 4.41 – *Apalis TK1*

4.3.2.1.1.2 Raspberry Pi 3 Compute Module

The second option is one of the most known in the SBC field: the Raspberry Pi 3 Compute Module. It is the industrial equivalent to the standard Raspberry Pi displayed

in Figure 4.59; therefore, it is based on the same SoC, the Broadcom BCM2837 up to 1.2 GHz and features 1 GB RAM, all integrated on a 67.6 mm x 31 mm board with a SODIMM connector. As for storage capabilities, it allows both 4 GB integrated flash memory and SD card interface.

One of the greatest advantages of this solution is the amount of drivers already available when Linux is used as OS, because of the broad usage of this board. Figure 4.42 shows the PCB.

Figure 4.42 – *Raspberry Pi 3 Compute Module [41]*

4.3.2.1.1.3 Colibri VF50

This solution is provided by Toradex [7] and is based on the NXP Freescale Vybrid SoC, featuring a microprocessor Cortex A5 at 400 MHz which delivers cost effective processing and graphic performance. It counts with 128 MB RAM and 128 MB of storage.

Its major advantage is its low power consumption and value for money, however its storage capabilities are not enough for this design. Figure 4.43 shows it.

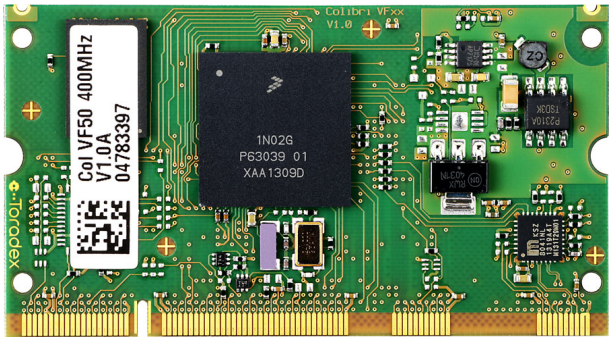


Figure 4.43 – *Colibri VF50 [7]*

4.3.2.1.1.4 CL-SOM-iMX8X

Based on the i.MX8X processor family by NXP, features a quad-core ARM Cortex-A35 up to 1.2 GHz and an integrated GPU. It also counts with 4 GB RAM and WiFi 802.11ac capabilities. Another interesting point is its wide temperature range, -40 °C to 85 °C.

Regarding connectivity, it counts with 96 GPIO, 4 UART and is USB 3.0 ready. It is shown in Figure 4.44.

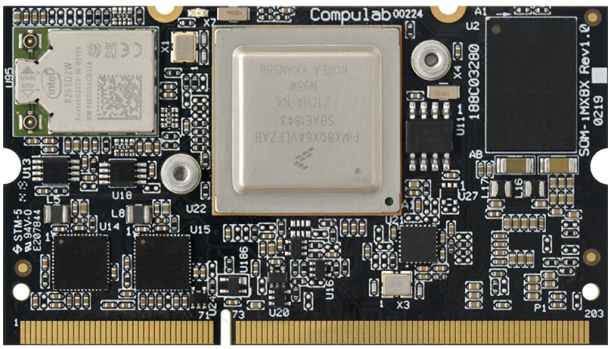


Figure 4.44 – *CL-SOM-iMX8X*

Table 4.1 sums up the main features of each solution.

SBC	System-On-Chip	RAM (GB)	Flash (GB)	GPIO	Price (€)
Apalis TK1	NVIDIA Tegra K1	2	16	87	211.25
Raspberry Pi 3 CM	Broadcom BCM2837	1	4	48	30.30
Colibri VF50	NXP Freescale Vybrid	0.128	0.128	103	40.40
CL-SOM-iMX8X	ARM Cortex A-35	4	4	96	75

Table 4.1 – *SBC comparative*

The first option, Apalis TK1 has a powerful GPU which is not likely to be used for the CubeSat purposes; therefore, taking into account its high cost, it is not a viable solution. On the hand, the Colibri VF50 is not powerful enough, although it counts with the highest number of GPIO it lacks RAM and storage; the Raspberry Pi 3 CM exhibits better features at a lower price. Among the two possibilities left, besides a better value for money, the Raspberry solution, as mentioned, counts with one of the greatest communities because of its extended usage. For all these reasons, it is the option chosen.

4.3.2.1.2 Raspberry Pi 3 CM consumption analysis

As described before, the CPU is one of the most power consuming components of the CubeSat. Therefore, it is necessary to perform a consumption analysis in different situations which allows characterizing the device completely, so the EPS can be adequately designed, increasing reliability.

To perform this analysis, different performance tests will be executed using the Raspberry Pi 3 CM. To ease these tests, the operating system **DietPi** [10] is used; it is a lightweight system which already includes different stress tests. By executing those tests, the device will increase its consumption until different ranges, including maximum before powering itself off.

Regarding the data gathering, it is used the DC Power Analyzer Keysight N6705B, shown in Figure 4.45, by establishin a TCP connection over a LAN network, using MATLAB®.

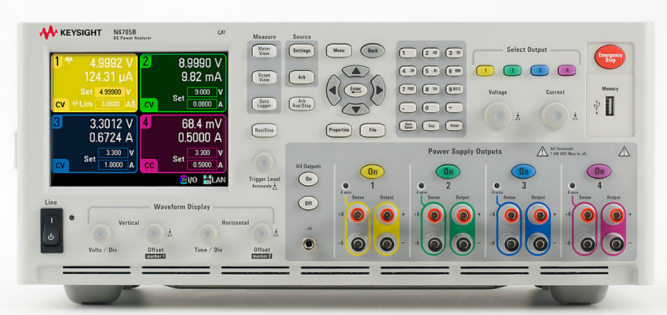


Figure 4.45 – *Keysight N6705B [25]*

The structure of the tests will be the following: as the CM allows being supplied with two voltages (3.3 V and 5 V), both must be tested in order to decide. On the other hand, DietPi allows choosing the number of cores of the microprocessor used (1 to 4) as well as the CPU **governor** (*ondemand, conservative, powersave or perfomance*). With all these parameters, the CM will be analyzed under three situations: at the boot (power peaks are common), in stationary state and when dealing with high CPU loads (maximum consumption).

Of course, this schema has produced a large number of plots; out of simplicity, only a selection of the most representative ones are plotted. Particularly, it is compared the power consumption for each voltage supply in every three situations. Additionally, they are compared twice: with the *Powersave* governor and 1 core enabled (minimum consumption) and with *Perfomance* governor and 4 core enabled (maximum consumption). This setting-up will allow determining the whole consumption range, from the minimum to the maximum.

4.3.2.1.2.1 *Powersave* governor - 1 core

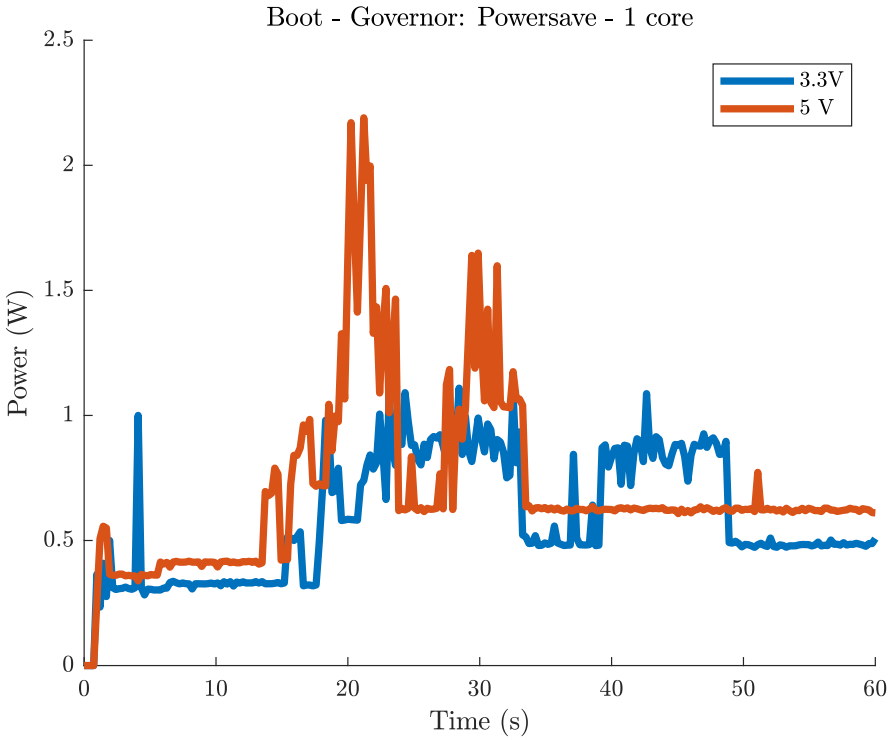


Figure 4.46 – *Power Consumption at* ***Boot***

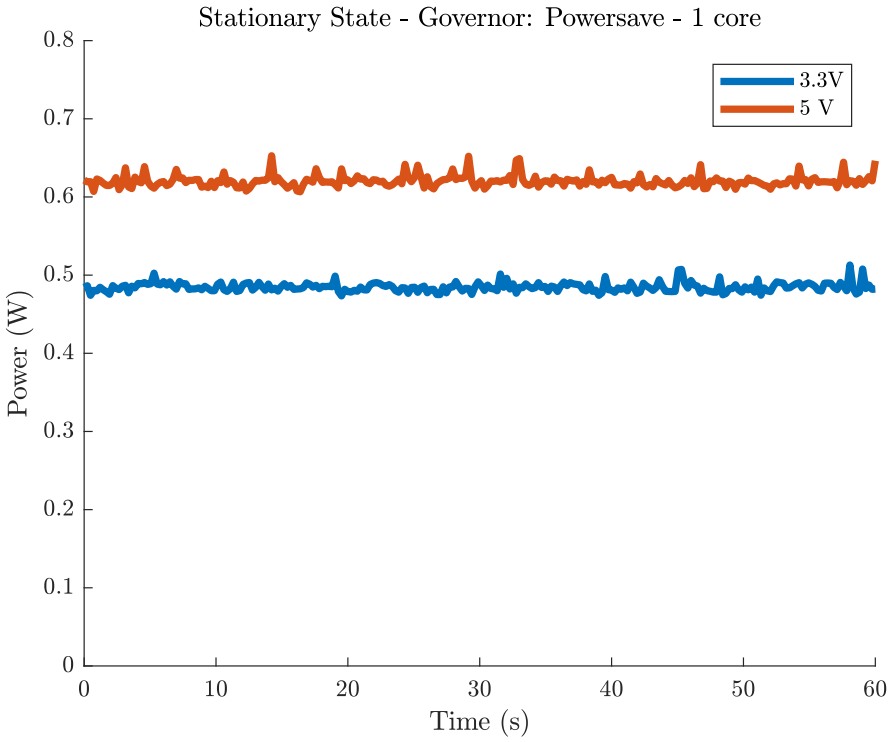


Figure 4.47 – *Power Consumption at* ***Stationary State***

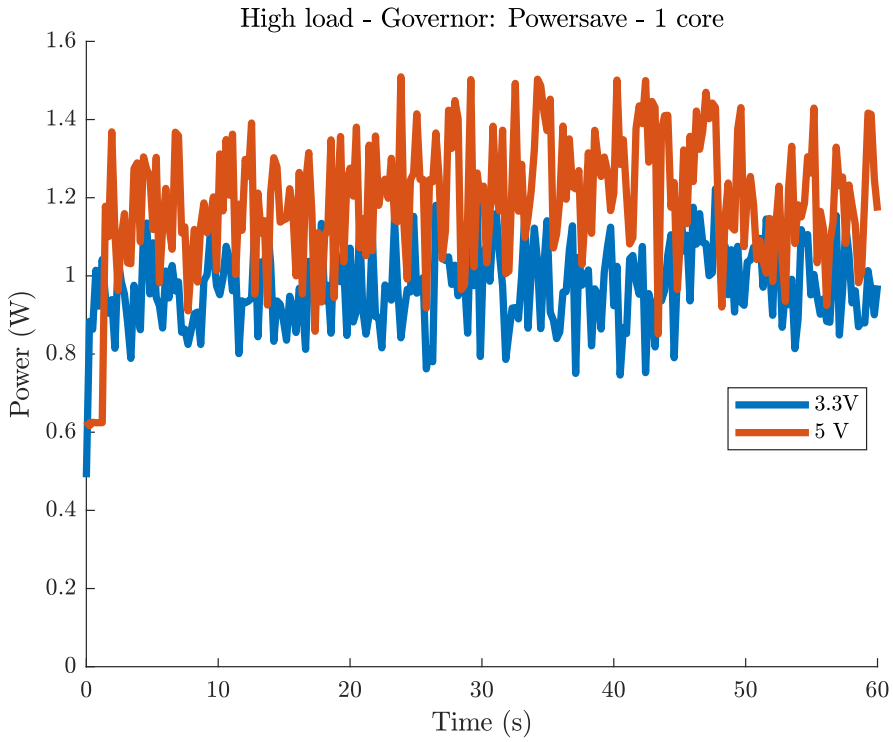


Figure 4.48 – *Power Consumption at **High load***

4.3.2.1.2.2 *Performance* governor - 4 cores

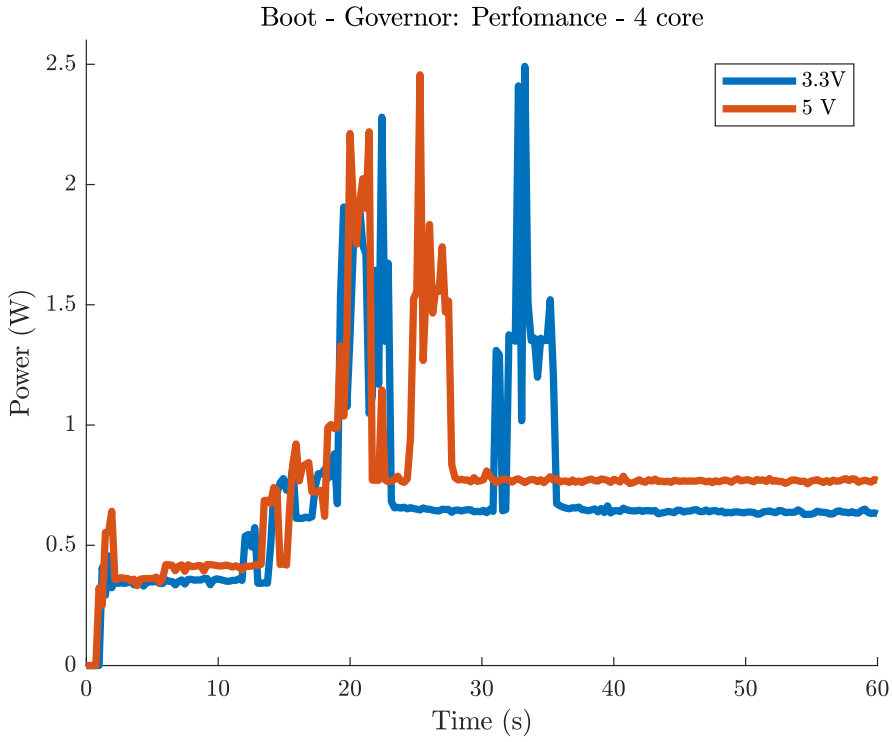


Figure 4.49 – *Power Consumption at **Boot***

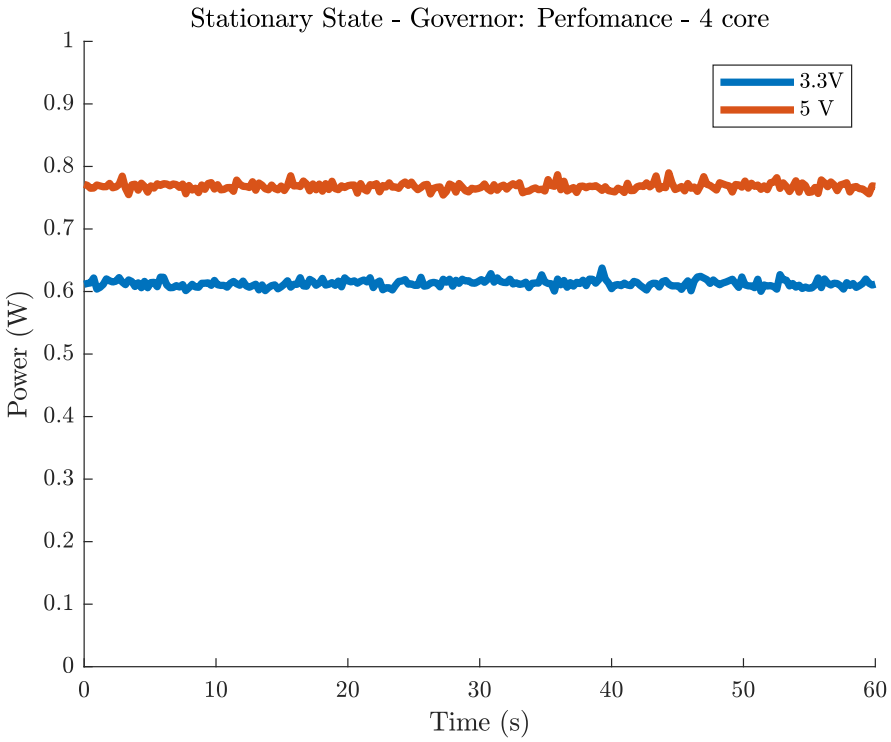


Figure 4.50 – *Power Consumption at **Stationary State***

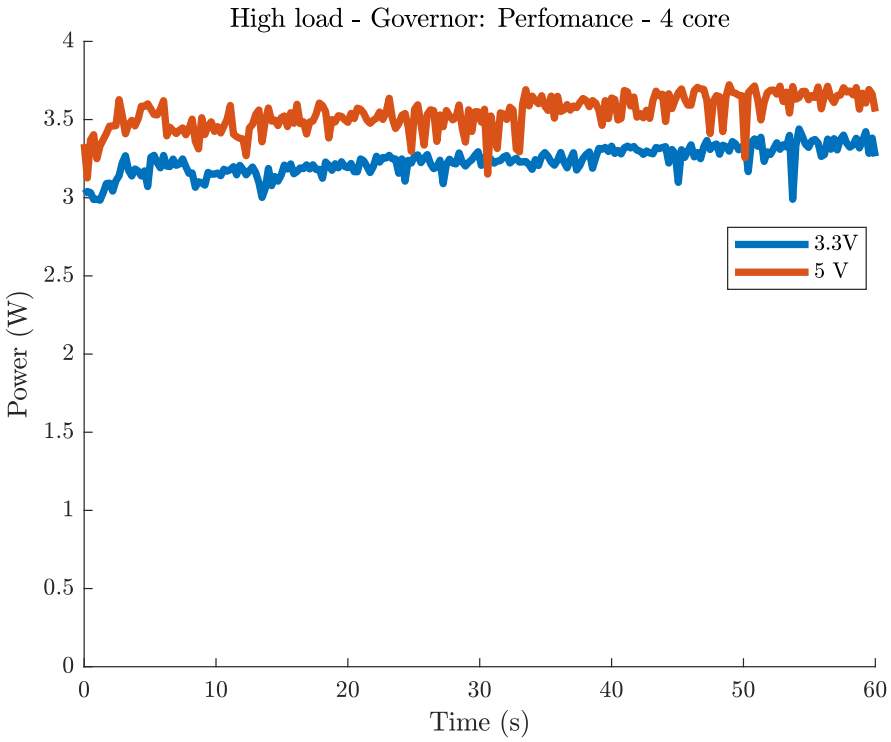


Figure 4.51 – *Power Consumption at **High load***

From the previous plots, some conclusions can be extracted:

- At **boot**, the **minimum** power consumption peak (with *Powersave* governor and 1 core enabled) is **2.25 W, with 5 V supply**.
- At **boot**, the **maximum** power consumption peak (with *Performance* governor and 4 core enabled) is **2.5 W**, regardless of the voltage supply; 11.1 % higher than the minimum.
- At **stationary state**, power consumption is higher in the *Performance* configuration, regardless of the voltage supply (0.49 W vs 0.61 W with 3.3 V supply and 0.62 W vs 0.78 W with 5 V supply).
- When dealing with a **high work load**, the power consumption is again higher with 5 V supply (1.5 W and 3.5 W with *Powersave* and *Performance* configuration, respectively) than with 3.3 V supply (1.2 W and 3.25 W with *Powersave* and *Performance* configuration, respectively). This case features the greatest increase in power consumption between the voltages supply: **170.83 %** and **133.33 %** increase with 3.3 V and 5 V respectively.
- Considering equal configuration, **power consumption with 3.3 V supply is always lower** than the equivalent with 5 V supply.

Therefore, in order to save the most energy possible (given that the system will be supplied by batteries) 3.3 V supply is chosen. All this information will be used in 4.3.4.2 to estimate the operating time of the OBC depending on the battery used.

4.3.2.2 Communications

As described in previous sections, the Communications subsystem is included within the OBC. As for this part of the CubeSat, only the simulation purposes are considered. In a real mission, COMMS area will count with different radio-frequency systems, antennas and the rest of elements mentioned in section 3.4.1.3.2.1. Once again, that task can suppose a complete project by itself, so in this section, the focus will be on communications system which allows simulation-related tasks such as programming the device, downloading data, connecting to the Internet and so on.

4.3.2.2.1 Wired

Wired connections will be the easiest way of communication when working with the platform. Although Wireless communications are intented to be used under operation, some tasks need to be performed over a wire, for example, the first programming.

4.3.2.2.1.1 USB & Ethernet

USB is used to first program the device, perform low-level tasks and **charge** the batteries. This is important considering the poor performance expected from the solar panels, as described in subsubsection 4.1.3.2; although that charging procedure can be used to test the EPS, when the simulations are related to another subsystems there must be some quick charge method, in this case provided through USB. Several connections will be disposed, allowing not only the mentioned tasks but also connecting external devices for prototyping, such as cameras.

Both USB and Ethernet connections are complex to design as they are **differential transmission lines**. Differential signaling is a technique to transmit information using complementary signals (positive and negative) using different wires, in contrast to the more common **single-ended signaling**. Figure 4.52 shows a couple of differential lines with a representation of each signal when traveling across each of the lines.

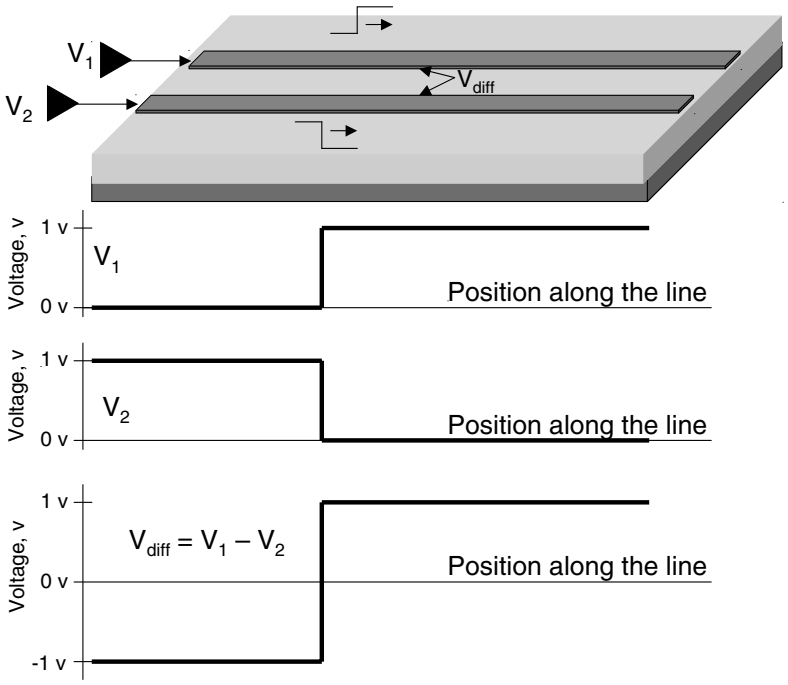


Figure 4.52 – *A differential pair with a signal propagating on each line [65]*

In an impedance-matched circuit, external EMI affects both wires equally. Since the receiver only detects the difference between the conductors, this technique resists electromagnetic noise better than a single-ended conductor, as well as improving SNR and minimizing crosstalk [59]. Figure 4.53 depicts the advantages of this method.

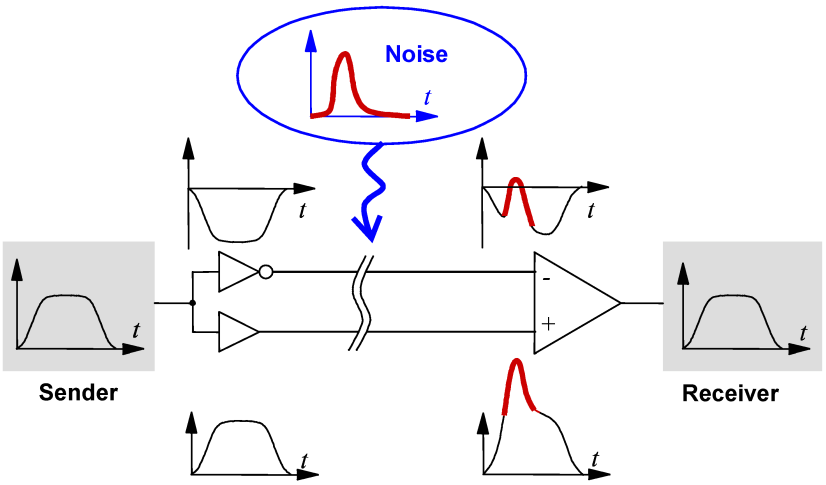


Figure 4.53 – *Differential signaling functioning [59]*

In order to design a USB line, it is necessary a **Host Controller Interface** HCI or simply a controller. It is an integrated circuit which allows the computer to communicate with the USB device. Indeed, Ethernet connections need another too. That is the reason for this common section: it will be used a common controller for both, particularly the **Microchip LAN9514** [29]. It is a dual USB 2.0 and 10/100 Ethernet (full-duplex) Controller which features four USB downstream ports and one upstream. Figure 4.54 depicts the internal block diagram of the chip.

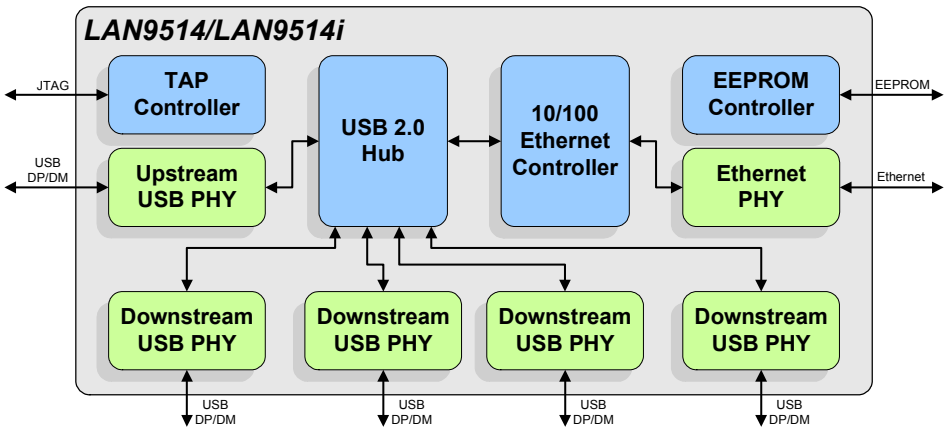


Figure 4.54 – *LAN9514 internal block diagram [29]*

This common usage has drawbacks though. First, the Raspberry Pi 3 Compute Module has a single connection point (which is actually composed of two differential pins) for both, Ethernet and USB connections. However, if the controller is directly

connected to the SBC, the Compute Module cannot be turned into **slave mode**, which means that **it cannot be programmed**. In this context, *programmed* means mounting the memory storage of the Compute Module as USB slave and then **flash** it with an operating system image. Therefore, it is necessary a **USB switch** that allows both modes depending on the connector used and the configuration selected. For this purpose, it is chosen the integrated circuit **ON Semiconductor FSUSB42** [36]; it is a bi-directional two USB 2.0 ports switch in charge of turning the Compute Module into slave mode when a USB is connected to the programming port. Figure 4.55 depicts the functioning of this circuitry.

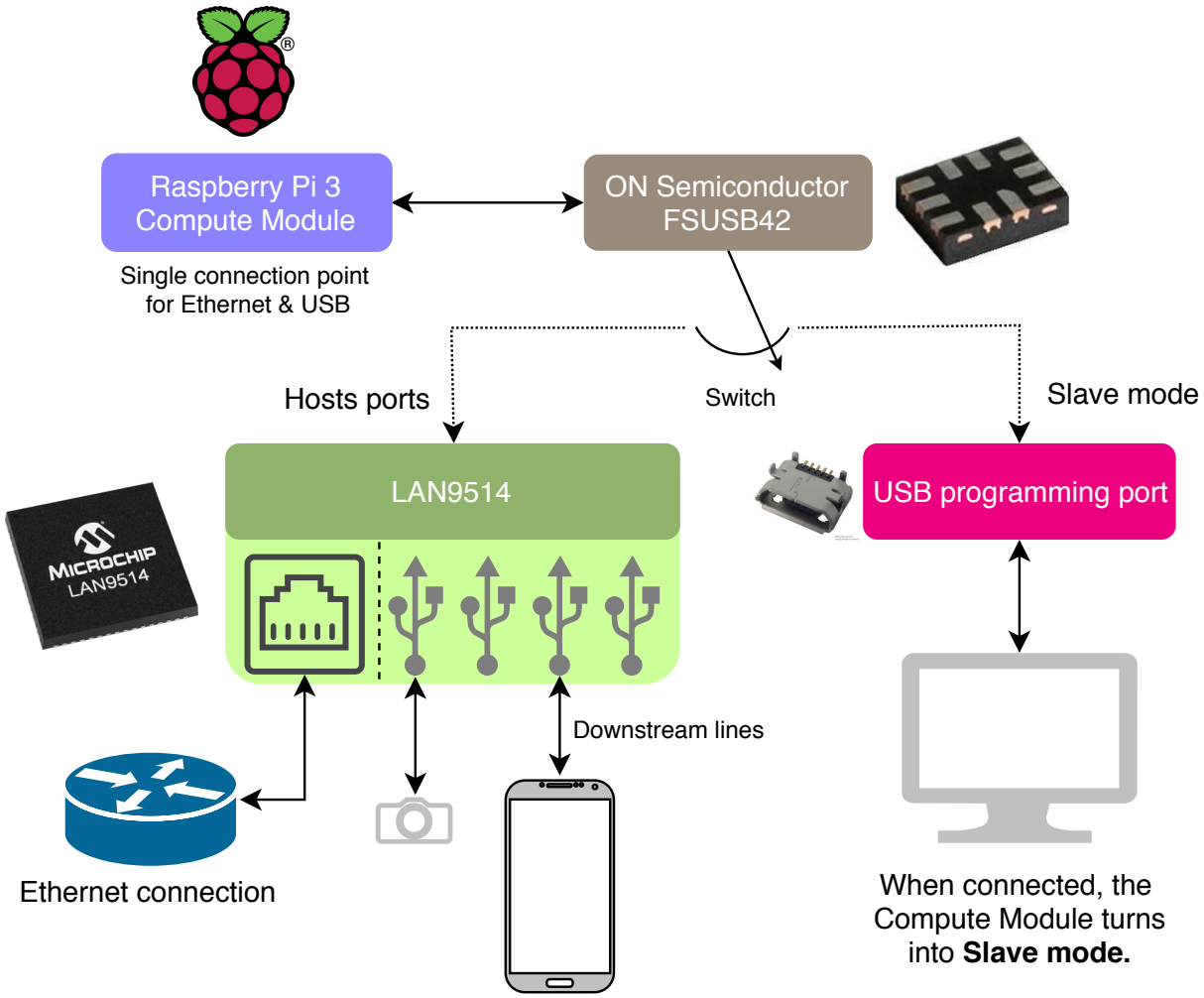


Figure 4.55 – *LAN9514 and USB switch mechanism*

Ethernet connections are designed for a differential impedance of 100 Ω. Differential impedance depends on a variety of factors, featuring: track width, differential tracks separation, substrate used, substrate height and frequency. This kind of design is a

complex issue which usually needs to assume different tradeoffs. There are a variety of methods to compute differential impedance; in short, every couple of differential impedance exhibits two different impedances: on the one hand, the differential one and, on the other hand, the single-ended corresponding to each of the lines (i.e., the **characteristic impedance of the line**). The equations under differential impedance are experimental, as there is no exact solution. Some of the most typically used are shown in Equation 4.3.1, where ϵ_r is the Relative Permittivity, h is the substrate's height and w and t the width and thickness of the line, respectively. It has been used the values given by the manufacturer which will be in charge to produce the PCB.

$$\begin{aligned} Z_o &= \frac{87}{\sqrt{\epsilon_r + 1.41}} \ln \frac{5.98h}{0.8w + t} \\ Z_{diff} &= 2 \times Z_o(1 - 0.48e^{-0.96\frac{d}{h}}) \end{aligned} \tag{4.3.1}$$

To illustrate the complexity under this operation, the differential impedance corresponding to an Ethernet line is graphically calculated, using the equations above. Particularly, Figure 4.56 depicts the differential impedance depending on the trace width, for three different space lengths between the lines. The first noticeable thing is that differential impedance decreases as trace width increases. The horizontal discontinuous lines depicts the target impedance for both, differential and single-ended, in the Ethernet case **100** Ω and **50** Ω, respectively.

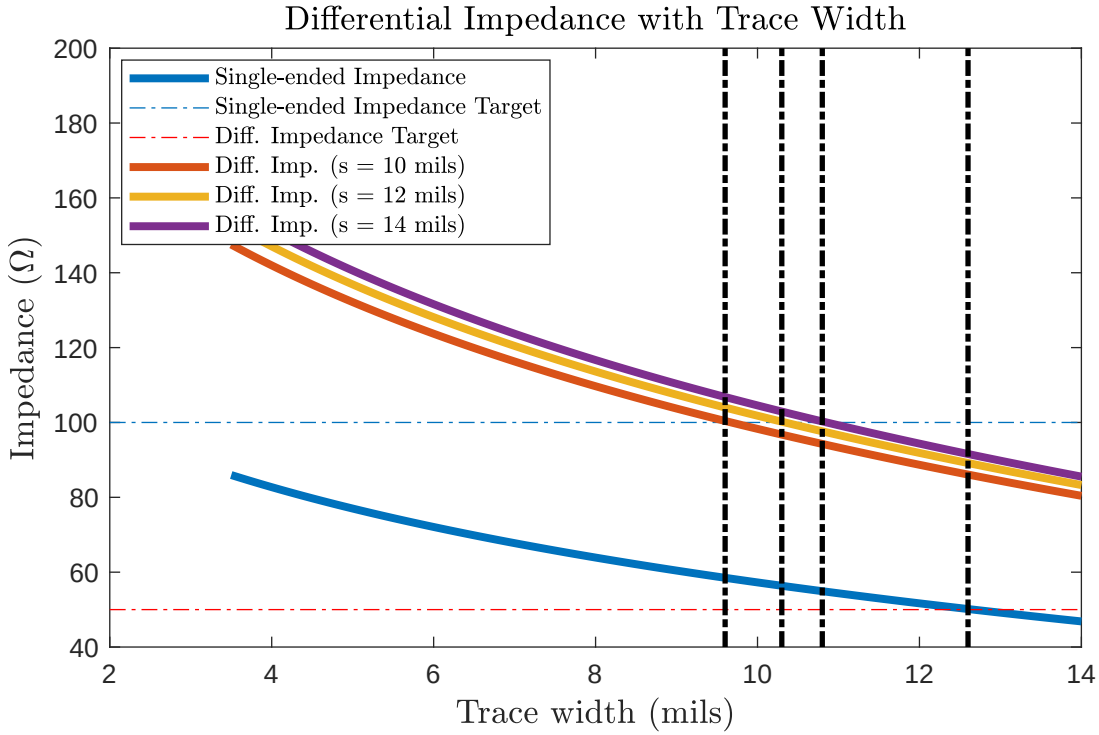


Figure 4.56 – *Graphical calculation of differential impedance*

The differential impedance varies with the spacing between lines, while the single-ended is obviously independent of this parameter. By plotting vertical lines (in discontinuous black) at the point in which each series matches its target is easy to see that **it is not possible to simultaneously have 100 Ω as differential impedance and 50 Ω as single-ended**. A certain **tradeoff** has to be assumed. In this case, the closest option is using traces of **10.8 mils** width with a separation of **14 mils** (purple series). With that design, single-ended impedance would be about 55 Ω while differential would be about 100 Ω.

While the latter example helps to understand the complexity under this kind of designs, the calculation is usually performed using dedicated and more accurate tools such as **Saturn PCB Design Toolkit** [43]. Figure 4.57 shows the calculation using that tool. It allows checking that the result accomplished by analytical calculation before is really close to the one estimated by Saturn, **105 Ω/59 Ω**, which falls into a 10 % tolerance, enough for the design to work properly.

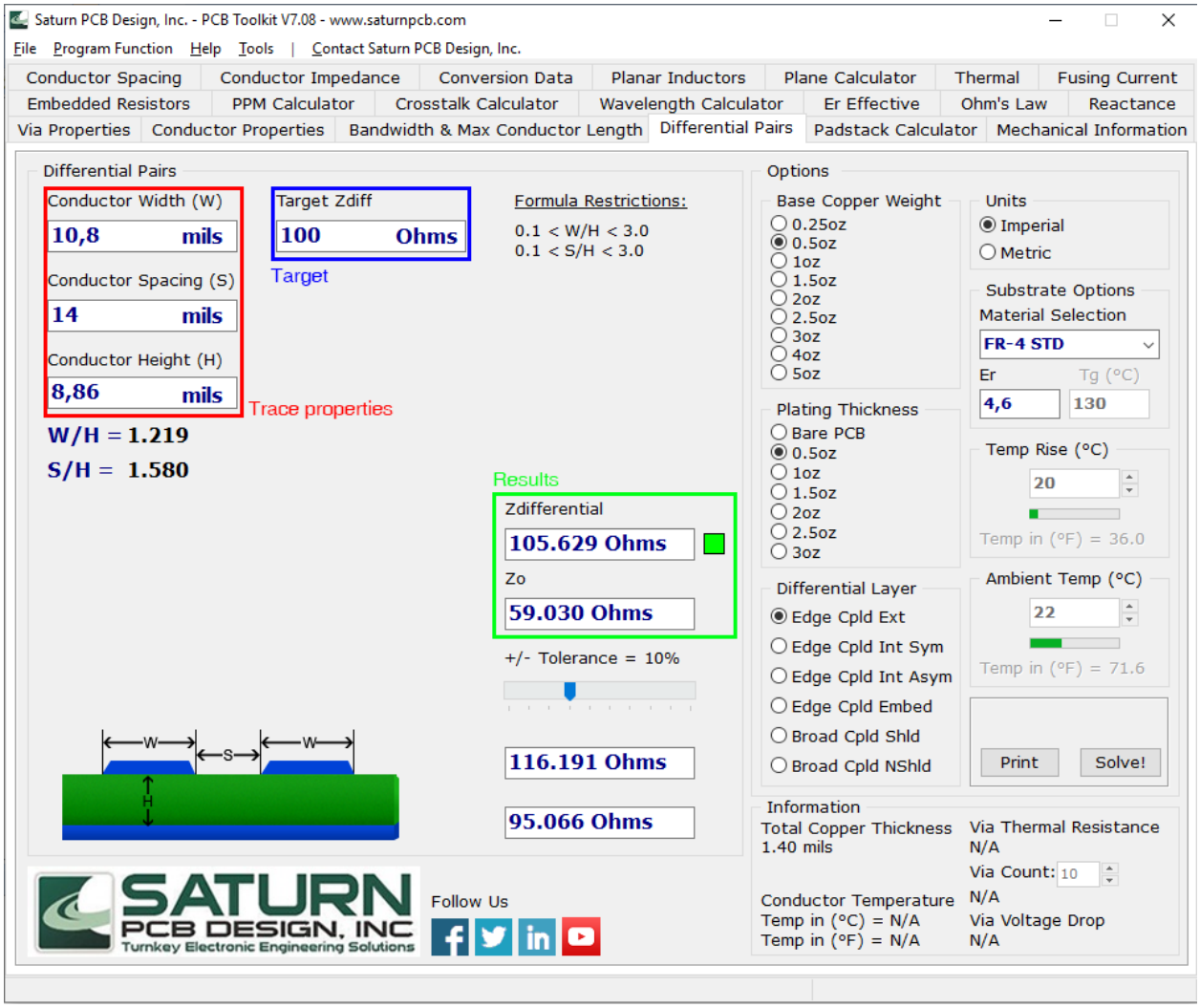


Figure 4.57 – *Calculation of differential impedance using Saturn PCB Design Toolkit*

In order for the signals to reach the end of the line at the same time, it is important for the lines to be **length matched**. It is accomplished by tweaking the lines as shown in Figure 4.58.

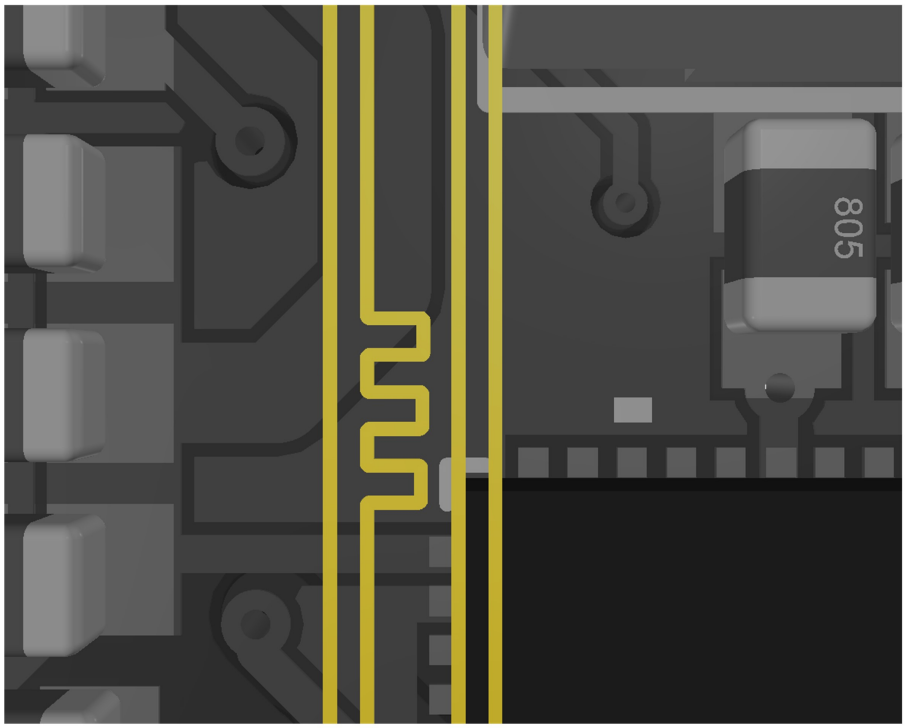
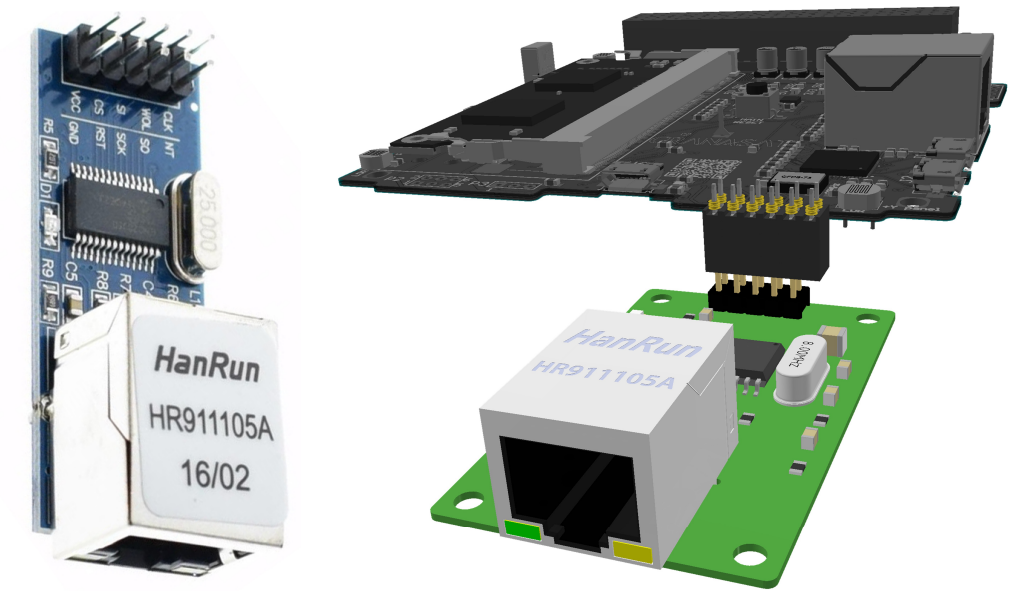


Figure 4.58 – *Lines tweaking to achieve length matching*

Just like Ethernet lines, USB designs must be based on differential signaling, particularly with a 90 Ω differential impedance and 45 Ω single-ended, which is accomplished using the same procedure.

Besides the embedded Ethernet design described above, in the OBC design it is included a ENC28J60 [28] Ethernet breakout which provides with direct Ethernet connection by SPI, shown in Figure 4.59a. Its obvious drawback is its size, as seen in Figure 4.59b, with the breakout on, the PCB cannot be fitted into the CubeSat structure; also the data rate accomplished with it is significantly inferior. However, in case the embedded design fails, this could be used as **contingency option**.



(a) *Breakout* **(b)** *Placed in the OBC, impeding its placement in the CubeSat*

Figure 4.59 – *ENC28J60*

4

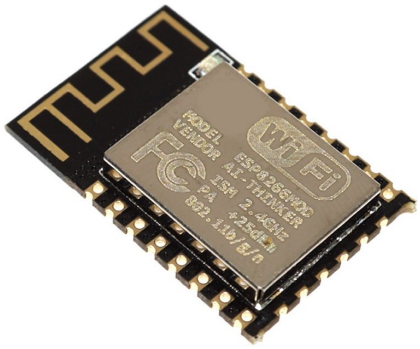
4.3.2.2.2 Wireless

The formal requirement **OBC.FoR.15** described in Table 3.3, states the need for a wireless communication system, provided by an integrated device fully compatible with IEEE 802.11g standard. After designing the wired communications subsystem, this section addresses the wireless ones, specially important in order to allow a realistic simulation using the I2DOS: wired communications allow neither a free rotation nor a convenient telemetry sending. Therefore, it will be crucial to accomplish a reliable communication over the air.

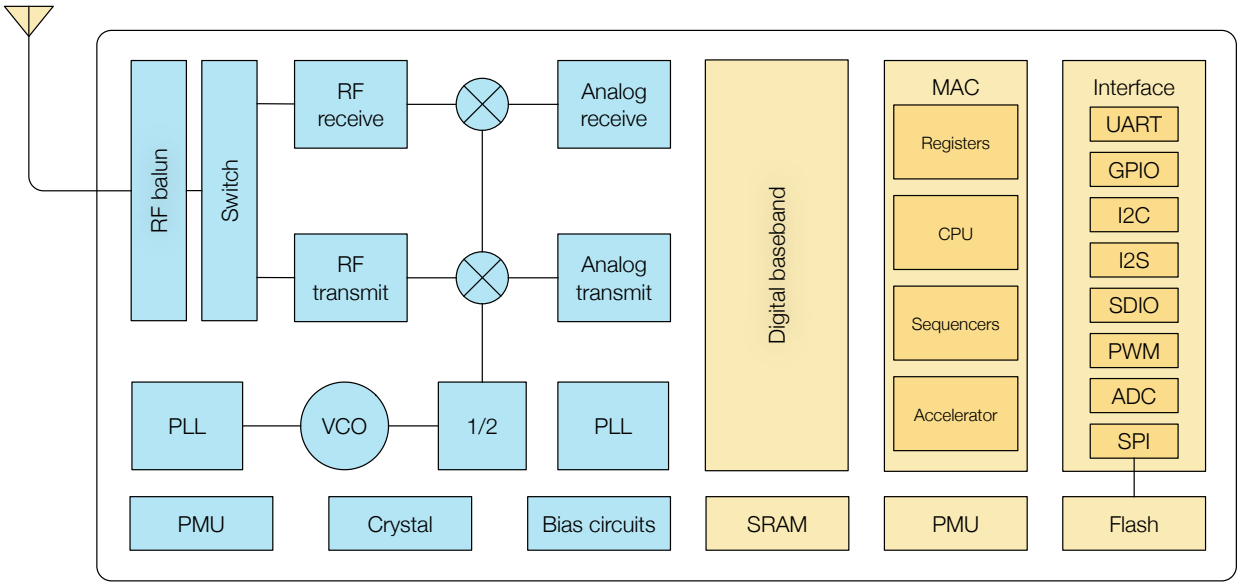
4.3.2.2.2.1 WiFi

Another communications related requirement was wireless communications. It will be provided with the typical technology family **WiFi**, based on the IEEE 802.11 standards.

In order to include this feature, it is used the **ESP8266** microcontroller by Espressif [15]. It is a highly integrated Wi-Fi programmable SoC solution, really power-efficient with an output power of up to 17.5 dBm. It will be used as slave to the Raspberry Pi 3 Compute Module with which will be completely integrated as a network interface. Figure 4.60a shows the module while Figure 4.60b depicts the functional block diagram.



(a) *External aspect*



(b) *Functional block diagram*

Figure 4.60 – *ESP8266 [15]*

4.3.2.3 Payload. Sensors

As defined in previous sections, the payload is the **useful equipment** carried by the CubeSat, able to provide with valuable information. Besides dedicated payloads such as cameras, also sensors and equipment necessary for the correct functioning of the CubeSat itself can be considered as a payload, from Star tracker to magnetometers or similar. This Master's Thesis provides with the basis for a simulation CubeSat which includes the majority of the sensors needed to successfully perform a mission, allowing adding more elements as payload in the future. In this section, the design of the different sensors implemented is outlined. Although most of them are used by the ADCS in its functioning, they can be considered part of the OBC and will be placed on its board.

4.3.2.3.1 Inertial Measurement Unit

IMU have been previously defined in this project. It is in charge of measuring different parameters such as gravitational acceleration, angular rate and orientation of the device, by using accelerometers, gyroscopes and magnetometers. This design will use the **MPU9250** by Invensense [93].

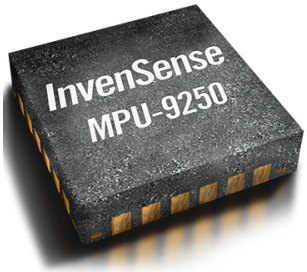


Figure 4.61 – *MPU9250*

MPU9250 is a multi-chip module consisting of two dies integrated into a single package, displayed in Figure 4.61; while the first hosts the 3-Axis gyroscope and the 3-Axis accelerometer, the second houses the 3-Axis magnetometer, resulting in a 9-axis MotionTracking device that combines a 3-axis gyroscope, 3-axis accelerometer, 3-axis magnetometer in a small 3x3x1 mm package, challenging to solder. It allows communicating using both I2C and SPI and features nine 16-bit ADC to digitize the output of each sensor.

In order to gather the data from the sensors, it is implemented a Python library in which the different registers addressed are assigned a variable name, as well as the default I2C slave addresses. Also, the different methods to read from the sensors are coded. For example, Code 4.3 shows the method to read the accelerometer data, using pseudocode.

```
    def readAccelerometer(self):

        x = round(data[0]*resolution, 3)
        y = round(data[1]*resolution, 3)
        z = round(data[2]*resolution, 3)

        return {x,y,z}
```

Code 4.3 – *Method to read from the accelerometer sensors*

A similar procedure is followed for the rest of the sensors, which eases accessing to the data from the main script.

4.3.2.3.2 Barometer & Thermometer

Specially useful for local simulation purposes, implementing a sensor easy to understand such as this gets students closer to the device and increases its interest. In this case, the integrated circuit used is the tiny Bosch BMP280 [66], shown in Figure 4.62.

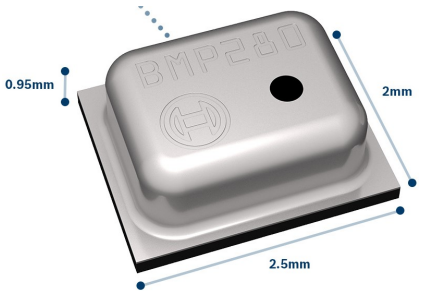


Figure 4.62 – *Bosch BMP280 [66]*

Bosch BMP280 exhibits high accuracy and linearity, as well as an almost negligible power consumption (with an average current of 2.74 µA. Besides, it allows a wide range of voltage supply, between 1.71 V and 3.6 V. Its technical parameters comply with the requirements defined in Chapter 3.

Again, it is developed a Python library to read from the integrated sensors, thermometer and barometer. This chip is also able to estimate the altitude above sea level from the pressure detected. Also, it is able to be programmed with different over-sampling settings for both barometer and temperature sensor; each oversampling step reduces noise and increases the output resolution by one bit. Figure 4.63 shows the BMP280 placed in the OBC PCB.

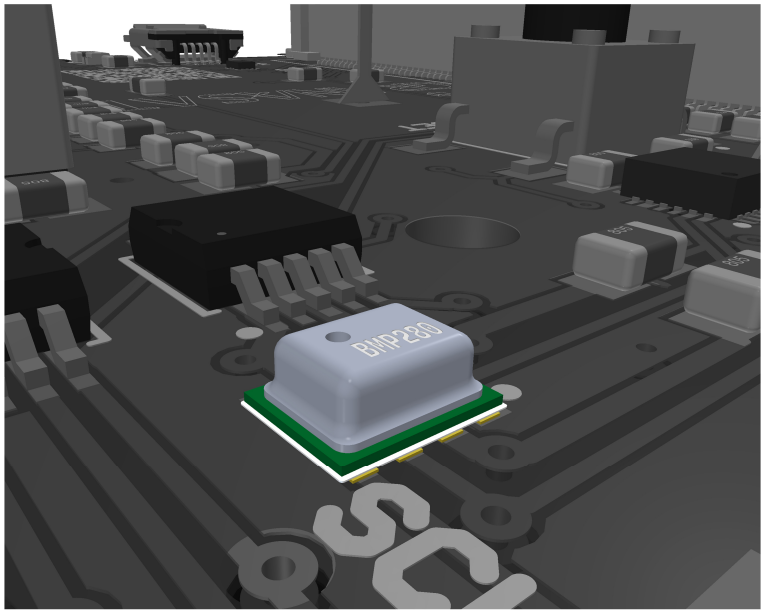


Figure 4.63 – *Bosch BMP280 placed in the OBC PCB*

4.3.2.3.3 Sun sensors. ADC

In order to determine the position of the real or simulated Sun, it is necessary to include the so-called **Sun sensors**. As for every component implied in a space-oriented design, there are qualified Sun sensors, intended to be used in a real mission, Figure 4.64 shows an example from [8].

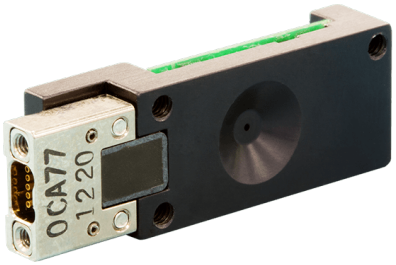


Figure 4.64 – *NCSS-SA05 Sun Sensor*

As usual, the *problem* with space qualified components is the cost: the NCSS-SA05 costs around 3300 €. That value makes it unfeasible for this project so, instead, the Sun sensors used will be based on LDR.

LDR will make it possible, depending on the intensity measured, to determine with enough accuracy the position of the light. Given that LDR are ultimately resistances, it is necessary to arrange some circuitry to determine the Sun's position. The easiest way to accomplish that point is designing a voltage divider between the LDR itself and another resistor. When supplied, a certain voltage falls in the measurement point, which is gathered using an ADC. Figure 4.65 depicts the circuit described.

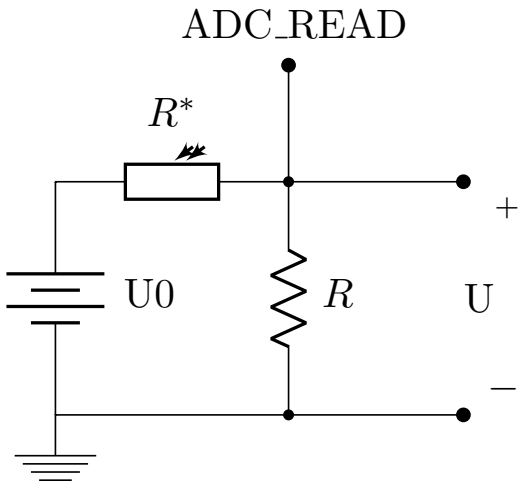


Figure 4.65 – *Sun sensor based on LDR circuit*

As seen in Figure 4.65, it is needed another resistor to complete the voltage divider. Given that a certain LDR will exhibit a fixed resistance range, the **excursion output range** of the voltage divider (i.e., the difference between the minimum and the maximum voltage measured at the point ADC_READ) will depend on the resistor magnitude chosen. Therefore, it is necessary to choose that resistor in order to **optimize** the excursion output range and get the most resolution possible.

In the Figure 4.65 above, LDR is denoted as R^* and R is another resistance completing the voltage divider. Depending on the light falling upon the LDR, R^* varies, so does the voltage V. That voltage is directly connected to the ADC so it is converted into a discrete value. With this circuit, V will increase when light level is high (facing Sun) whereas it will decrease when light reduces (eclipse).

Ideally, U voltage should go through the range $[0\ \text{V}, U_0]$ to maximize distinguishable values. Considering that the only degree of freedom is given by R value, it must be the point of the optimization. On the other hand, extreme resistance values of the LDR must be known so the whole resistance of the voltage divider can be calculated. Therefore, for a certain resistance $R^* \in [R^*_{\text{min}}, R^*_{\text{max}}]$:

$$U(R) = U_0 \frac{R}{R^* + R}$$

Since the aim is maximizing the voltage margin for U, it will be denoted as $\Delta U(R)$, which is the electric potential differential across R^*:

$$\begin{aligned}
\Delta U(R) &= U_{\text{max}}(R) - U_{\text{min}}(R) \\
&= U_0 \left[\frac{R}{R^*_{\text{min}} + R} - \frac{R}{R^*_{\text{max}} + R} \right] \\
&= U_0 \left[\frac{R(R^*_{\text{max}} - R^*_{\text{min}})}{(R^*_{\text{min}} + R)(R^*_{\text{max}} + R)} \right]
\end{aligned}$$

In order to maximize the function $\Delta U(R)$, the derivative $\frac{\partial}{\partial R}\Delta U$ is calculated:

$$\frac{\partial}{\partial R}\Delta U = U_0 \left[\frac{(R^*_{\text{min}} - R^*_{\text{max}})\left[R^2 - R^*_{\text{min}} R^*_{\text{max}}\right]}{(R^*_{\text{min}} + R)^2 (R^*_{\text{max}} + R)^2} \right]$$

Then, it is solved $\frac{\partial}{\partial R}\Delta U = 0$ to find extreme point or **extrema**:

$$U_0\left[\frac{(R^*_{\text{min}} - R^*_{\text{max}})\left[R^2 - R^*_{\text{min}}R^*_{\text{max}}\right]}{(R^*_{\text{min}} + R)^2(R^*_{\text{max}} + R)^2}\right] = 0 \rightarrow (R^*_{\text{min}} - R^*_{\text{max}})\left[R^2 - R^*_{\text{min}}R^*_{\text{max}}\right] = 0$$

Assuming $R^*_{\text{max}} > R^*_{\text{min}}$, solving the last equation yields to:

$$R = \sqrt{R^*_{\text{min}}R^*_{\text{max}}}$$

Therefore, the optimum resistance in order to get the greatest excursion in the output voltage is given by the square root of the product of the LDR extreme values, which means that the optimum resistor depends on the chosen LDR. In order to make that decision, Figure 4.66 shows the behavior of the function $\Delta U(R)$ depending on the extreme resistance values of different photoresistor. Because of availability reasons, the LDR corresponding to the **blue** series is chosen which, with an associated optimum resistance $R = 1\,\text{k}\Omega$, reaches around 80 % of the possible excursion output.

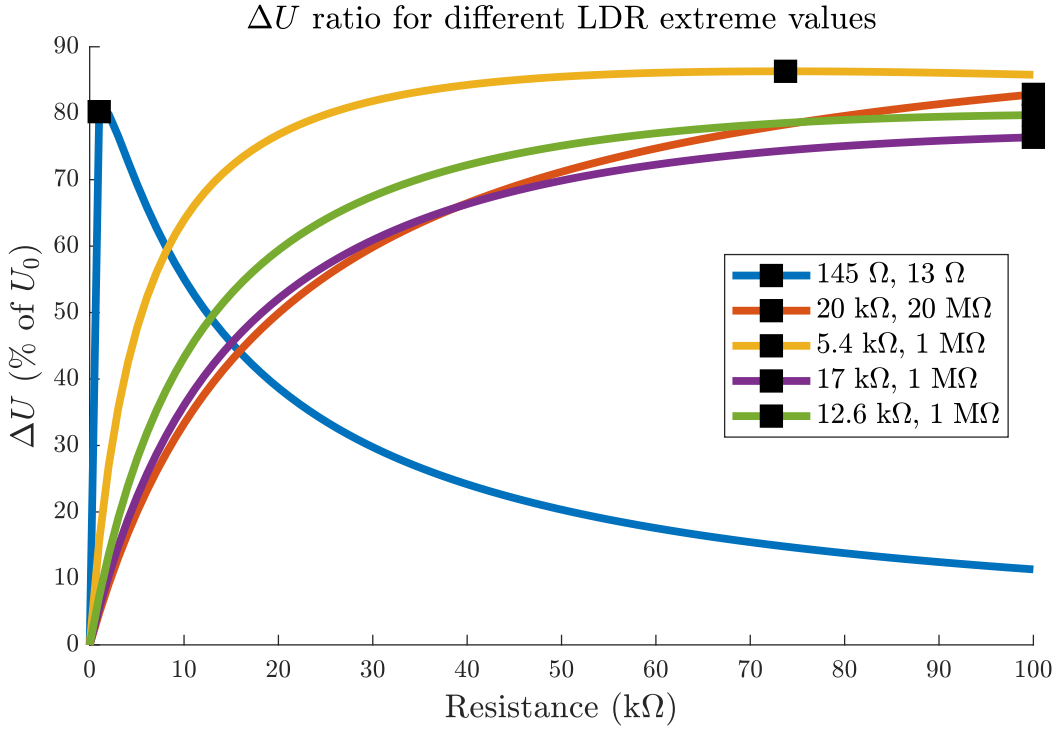


Figure 4.66 – *ΔU ratio for different LDR extreme values*

Once it has been accomplished an optimum excursion output, that value has to be read using an ADC. As for this design, the one chosen is the **Texas Instruments ADS1015** [53]. It is a low-power I2C compatible ADC with a maximum resolution of 12 bits and up to four measurements; therefore, considering that there is one Sun Sensor per face of the CubeSat, two ADC are needed. They must be assigned two different I2C addresses in order to allow communication with both.

Following the same procedure, a simple I2C communication library has been developed, allowing data polling. This ADC also features the *ALERT* function, which uses a digital comparator that can issue an alert on the corresponding pin when conversion data exceeds the limits set in threshold registers.

4.3.2.3.4 Real-Time Clock

A Real-Time Clock or RTC is an integrated circuit which keeps track of the time. They must be distinguished from the typical hardware clocks, which are simple signals not related to time in typical units.

RTC are based on crystal oscillators, typically at a frequency of **32.768 kHz**, because of its convenience with binary counter circuits. In this case, it is used in conjunction with the CPU, so it is able to be aware of the current time; this is not needed when it is connected to the Internet, as it gets time by that via, but in any other case the system will start in **Unix Time**, the first of January, 1970.

As for this design, the RTC chosen is the PCF8523 by NXP [35], along with a CR2032 battery which ensures it keeps tracking time when the central system is unpowered. Figure 4.67 shows the placement of the ADS1015 with the crystal oscillator in the PCB.

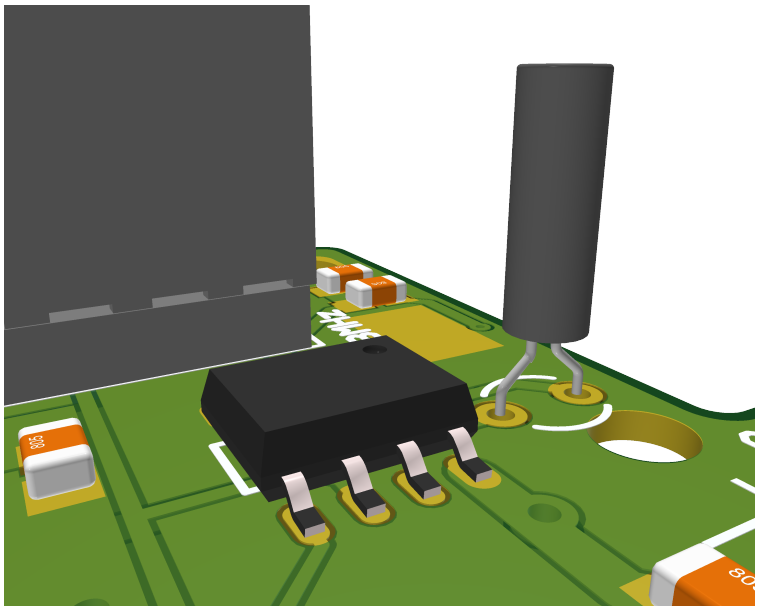


Figure 4.67 – *RTC and crystal oscillator in the OBC PCB*

4.3.2.4 Flight Software. On-Board Data Handling

As stated in section 3.4.1.5, Flight Software is in charge of controlling the whole system and dealing with all the possible situations during a mission. In this section, it is addressed the software under the whole CubeSat as well as implemented a basic OBDH.

4.3.2.4.1 Operating System

OS were divided into Real-time and Non-real-time. The solution taken in this project is somewhere in the middle of both: it is used an initially Non-RTOS such as Linux, which is patched to be turned into a RTOS. This way, it is possible to maintain the great drivers compatibility of Linux but adding the deterministic nature of the RTOS.

In order to perform this change, it is necessary to **recompile the Linux kernel**. By default, Linux kernel is non-preemptive. To turn it into a preemptive scheduling, it is applied the **PREEMPT_RT patch** [39], which provides with faster response times and removes unbounded latencies.

The first task is downloading the desired version of the kernel. Then, the corresponding version of the patch is also downloaded and unpacked. It can be patched following the instructions in [45]. After patching, the kernel is rebuilt and configured as **preemptive**.

4.3.2.4.2 Telemetry System

As for this first prototype, it is designed a simple telemetry sending system using **Python**, particularly the **socket** library. It connects to a certain IP address and port with the instructions shown in Code 4.4.

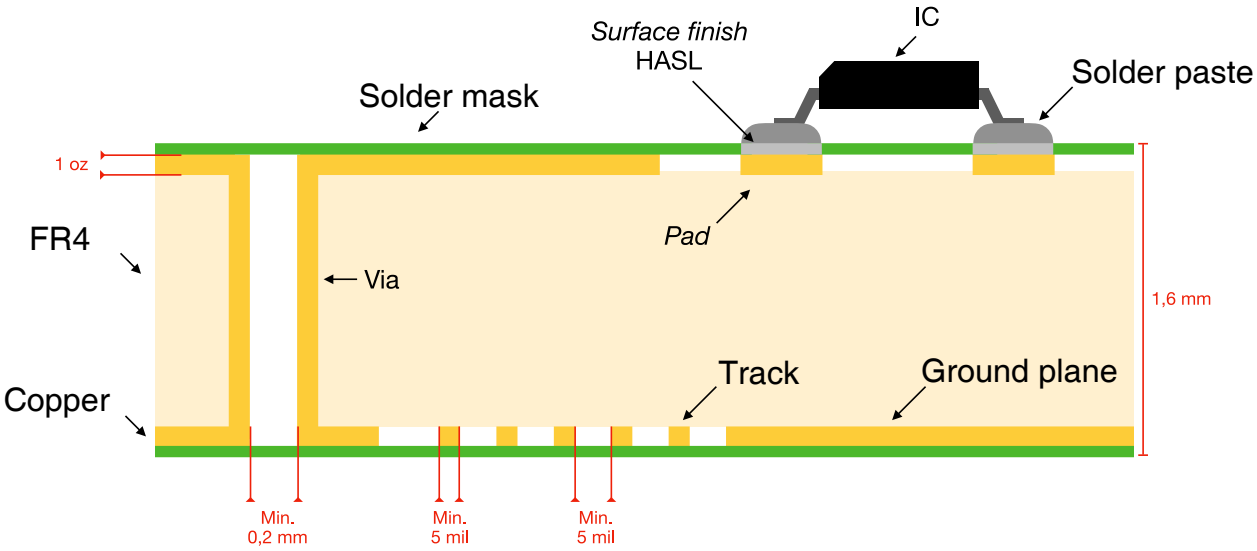
```
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
host = '192.168.1.118'
port = 8445
s.connect((host, port))
```

Code 4.4 – *Connection to server using the **socket** library*

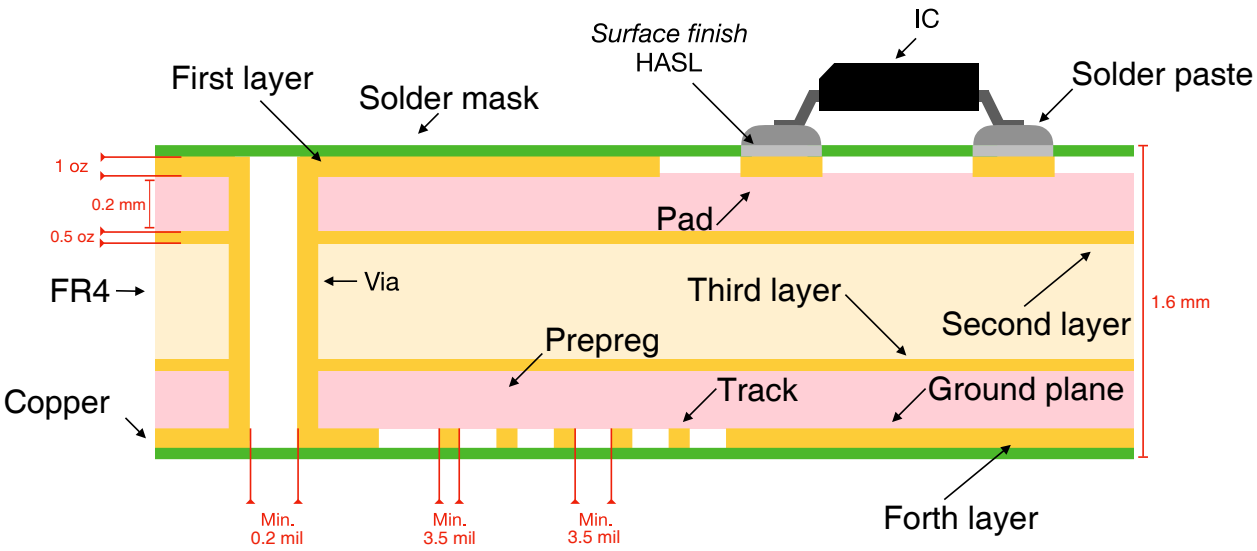
Particularly, the tuple IP:Port must match the existing at the other end of the communication, in which **the COSMOS Ground Station is listening**, as seen in Subsection 4.2.1. Once the connection is established, the flow of communication with telemetry and telecommands can start.

4.3.2.5 PCB design

This section includes the different schematics and information related to the PCB designed for the OBC. In order to allow its production, this design must comply with the constraints given by the manufacturer; these depends on the number of layers of the PCB, while the OBC is a 4-layers board, the ADCS PCB (addressed in section 4.3.3.5) is a 2-layers board and the constraints **change**. Particularly, Figure 4.68 depicts the different design requirements expected by the manufacturer: the first one for **2-layers** and the second one for **4-layers**.



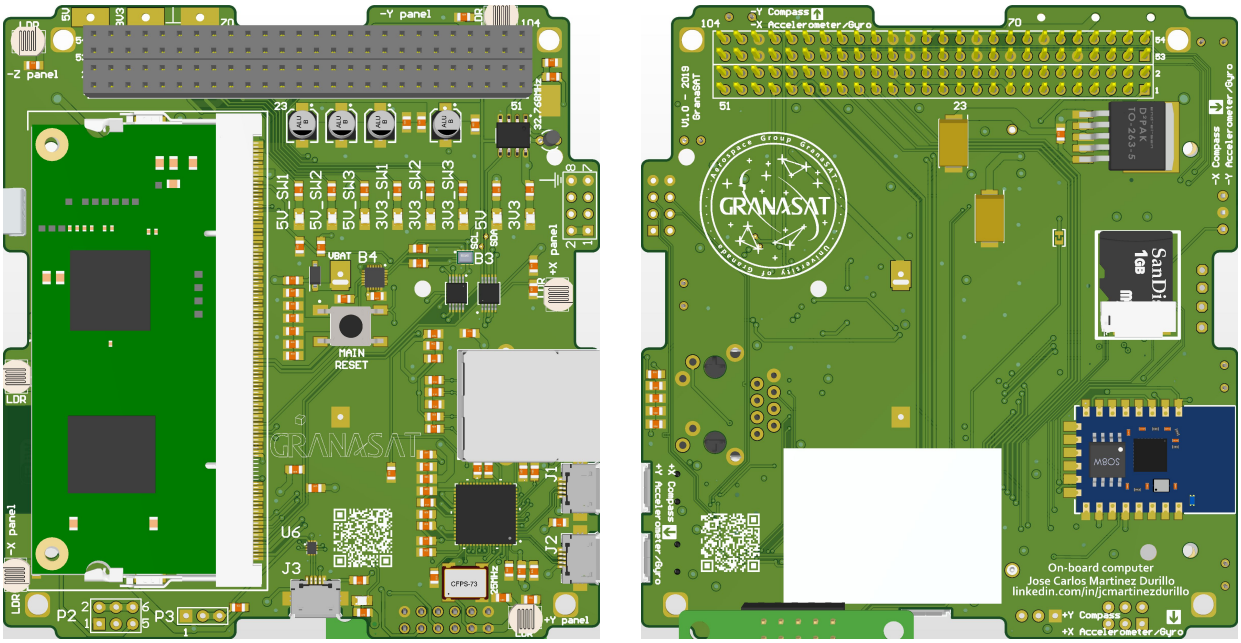
(a) *2-layers*



(b) *4-layers*

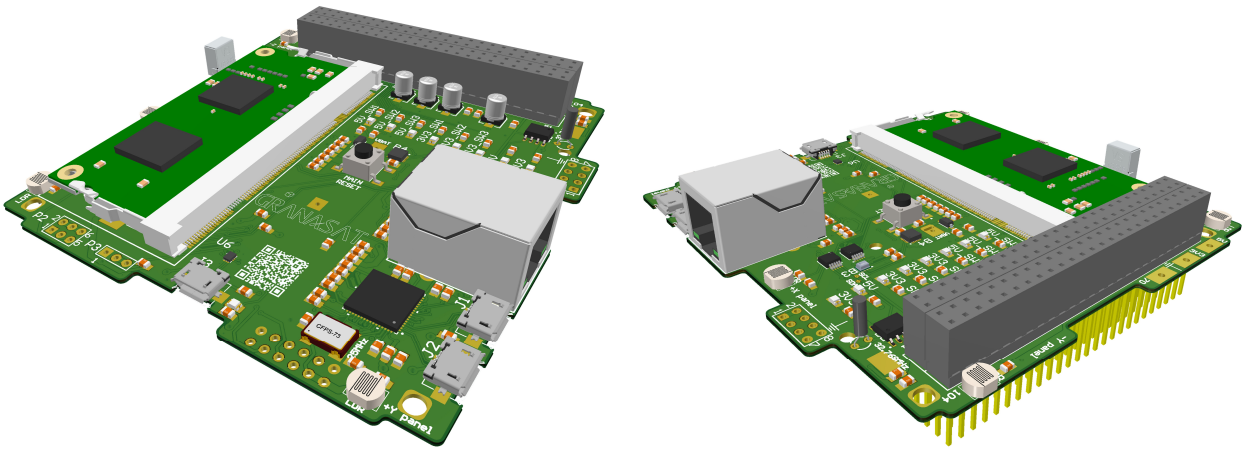
Figure 4.68 – *PCB Manufacturing constraints [105]*

Figure 4.69 shows a 3D render of the top and bottom layer in the final design, while Figure 4.70 shows a perspective view from different angles.



(a) *Top* **(b)** *Bottom*

Figure 4.69 – *OBC PCB final design*

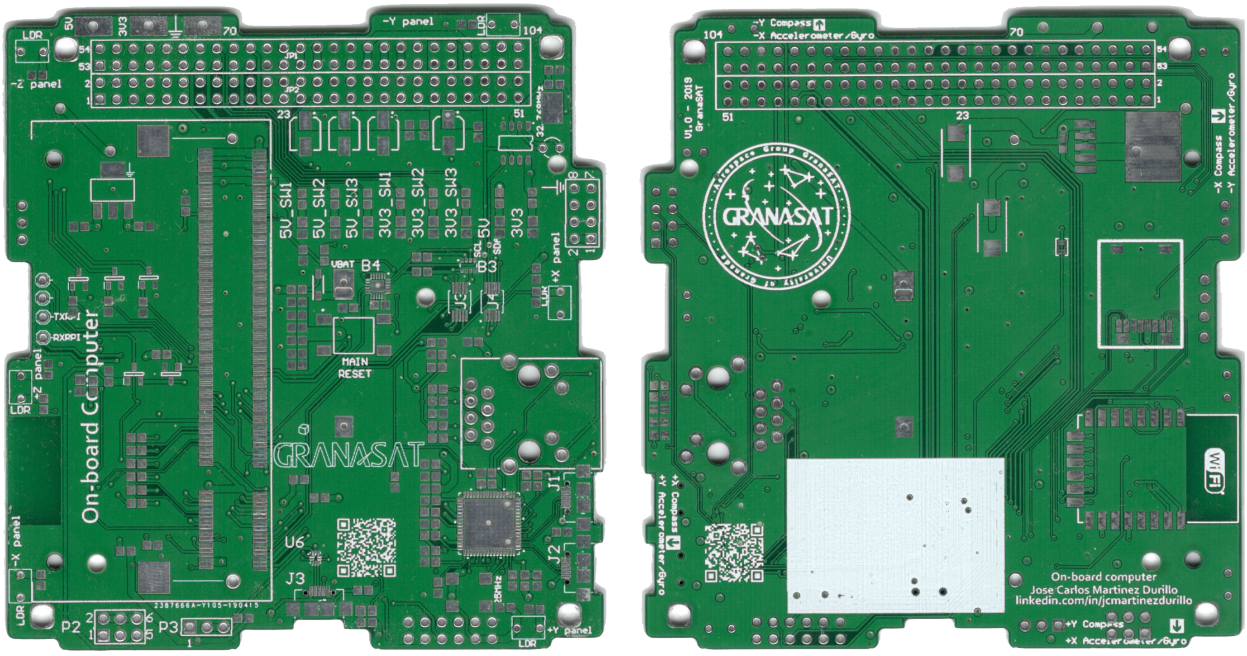


(a) **(b)**

Figure 4.70 – *OBC PCB 3D perspective*

4.3.2.6 Manufacturing and soldering

Once the PCB are designed, they are sent to a professional manufacturer. The result is shown in Figure 4.71.

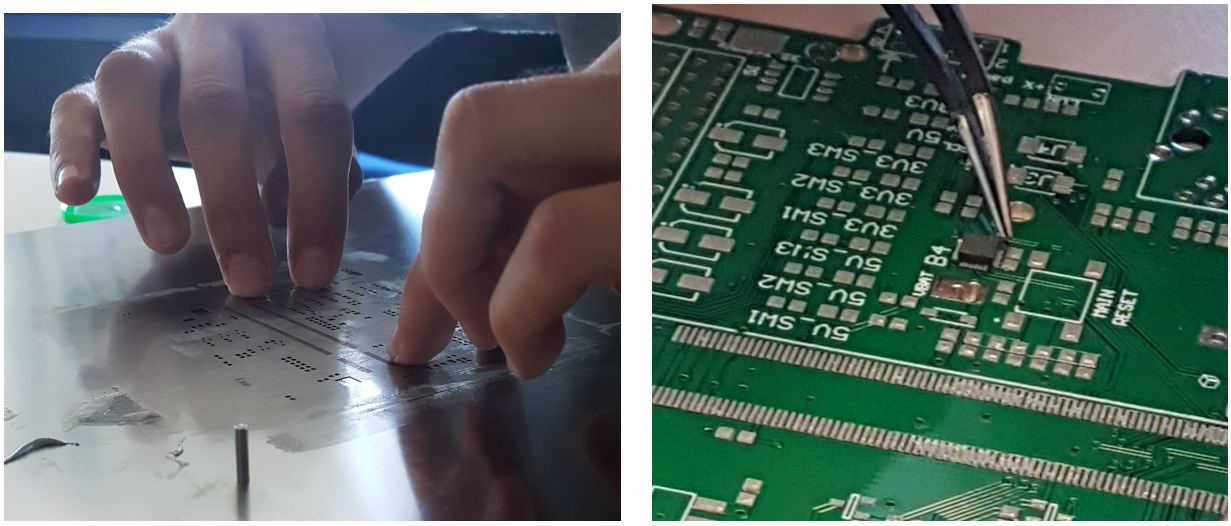


(a) *Top* **(b)** *Bottom*

Figure 4.71 – *OBC PCB manufactured*

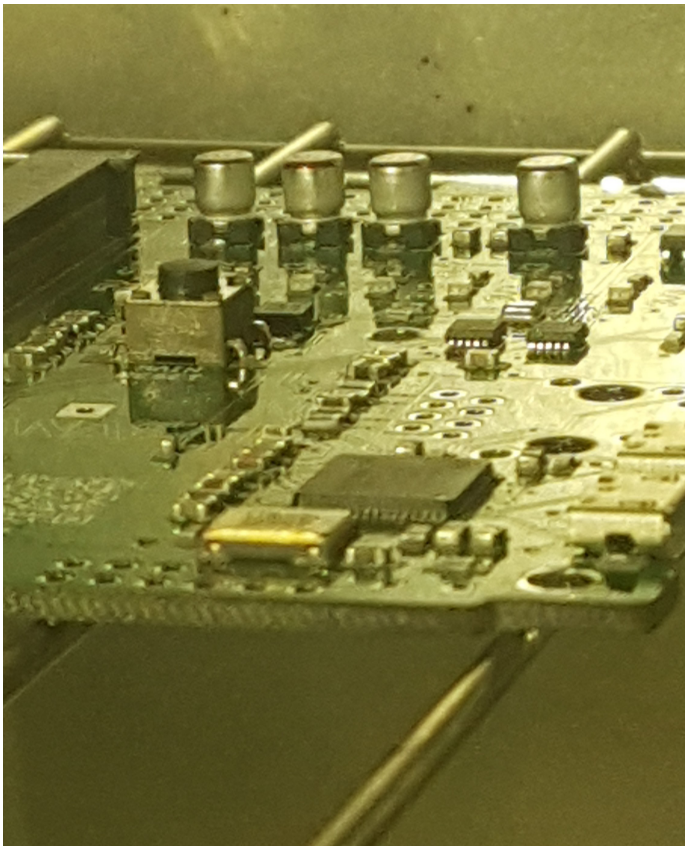
Because of certain components packages (for instance, the LAN9514 or the Compute Module), this PCB cannot be soldered by hand. It is needed a technique called **oven reflow**. It is a procedure in which the first step consists in covering the PCB with **solder paste** using a Stencil and then the components are placed onto the board. Figure 4.72 shows different moments of this process.

Once the PCB is fully assembled and all the components are in place, it undergoes the reflow process, which in this case is performed with a domestic oven which had been previously characterized with a trial-and-error approach. Video 4.2 shows the PCB undergoing the process (video available with Adobe Acrobat Reader).



(a) *Solder paste transfer using a Stencil* **(b)** *QFN package placement*

Figure 4.72 – *Soldering process*



Video 4.2 – *Oven reflow (video available with Adobe Acrobat Reader)*

Figure 4.73 and Figure 4.74 show the final product after completing soldering.

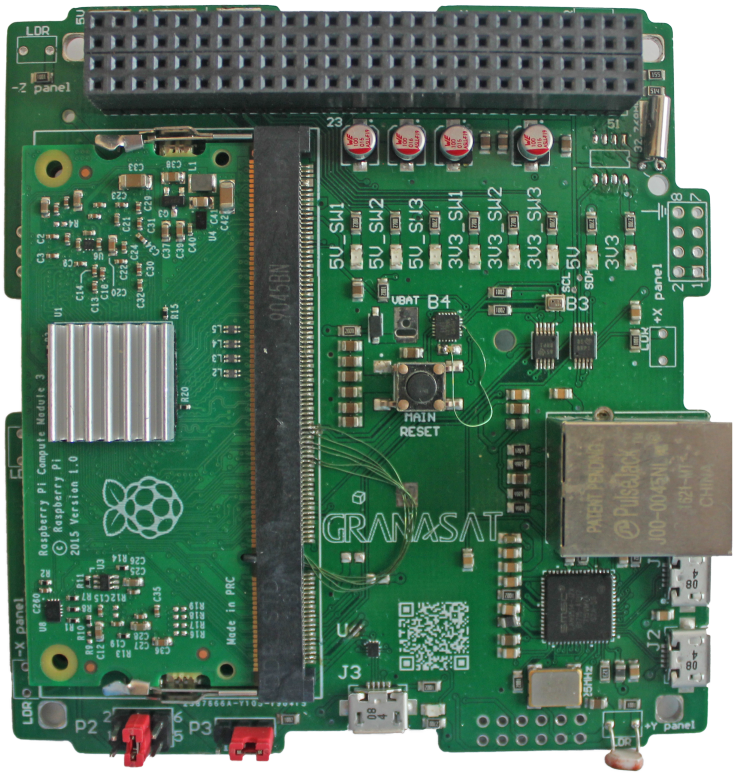


Figure 4.73 – *Top layer of the OBC*

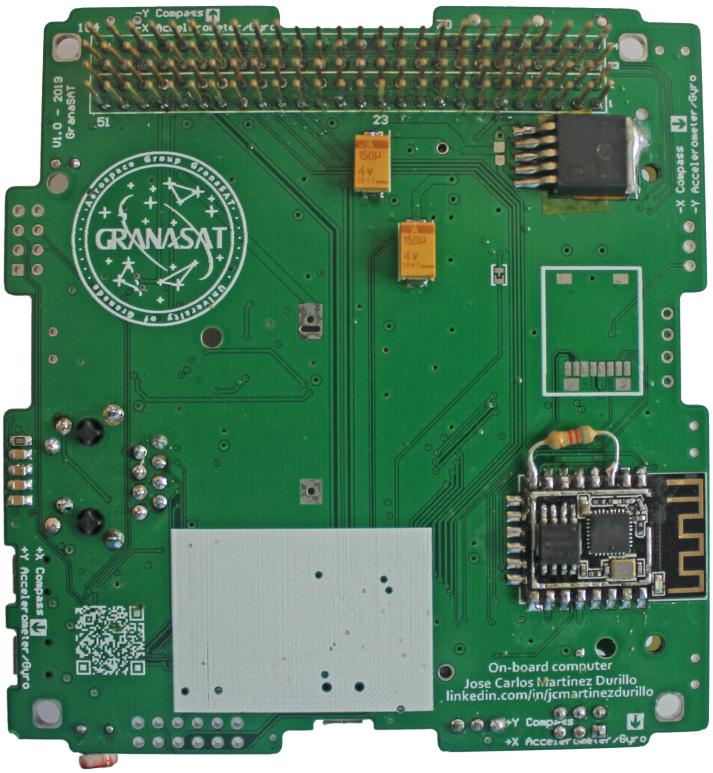


Figure 4.74 – *Bottom layer of the OBC*

4.3.3 Attitude Determination and Control System (ADCS)

This section addresses the design of the ADCS, in charge of keeping the CubeSat correctly oriented and managing external torques properly. At this point, some clarifications must be made with respect to the ADCS analysis in Chapter 3. For instance, this subsection will include the **Co-processing Programmable Core** initially analyzed in the OBC section; this is due to the need of an independent processing unit which frees the CPU of this tasks and allows having a real ADCS managing the issues regarding this matter. On the other hand, the **tachometer** is the only sensor which is included in this section, as it is placed in the ADCS board and is explicitly necessary for its operation. The **actuators** included in the design are also considered here.

4.3.3.1 Co-processing Programmable Core

As described in section 3.4.1.2, this architectures are usually implemented using FPGA. Using an FPGA provides with **time-critical** capabilities which goes in the same line of the RTOS introduced in paragraph 4.3.2.4.1: while with the operating system that capability is provided inherently by software, an FPGA allows expanding that concept to the hardware parcel. Time accuracy and management is a vital issue in space field, which justifies the different designs addressed in this project on said line.

Therefore, this Master's Thesis will make use of an FPGA, particularly the ICE40UP5K by Lattice [26]. It counts with 5000 LUT and is provided in a variety of packages, in this case, **QFN** is chosen. Typically, FPGA are programmed from an external memory in a **master-slave** configuration in which the FPGA acts as master. In this project, however, given the presence of a central processing unit (the Raspberry Pi 3 Compute Module), it is designed the other way around: it acts as master while the FPGA does as **slave**. This way, each time the SBC boots, it sends (as master) a **bitstream** through an SPI channel which contains the programming code. Additionally, the same channel is used to transfer the information during the FPGA operation, detailed next.

4.3.3.1.1 PWM generation

In this project, the FPGA main functionality will be generating different PWM signals to control the actuators, given that the CPU cannot produce the amount needed; however, it could be used for a variety of purposes, which is the idea under the Co-processing Programmable Core. As described before, the same SPI channel is used to configure each of the PWM. Figure 4.75 depicts the basis of this operation.

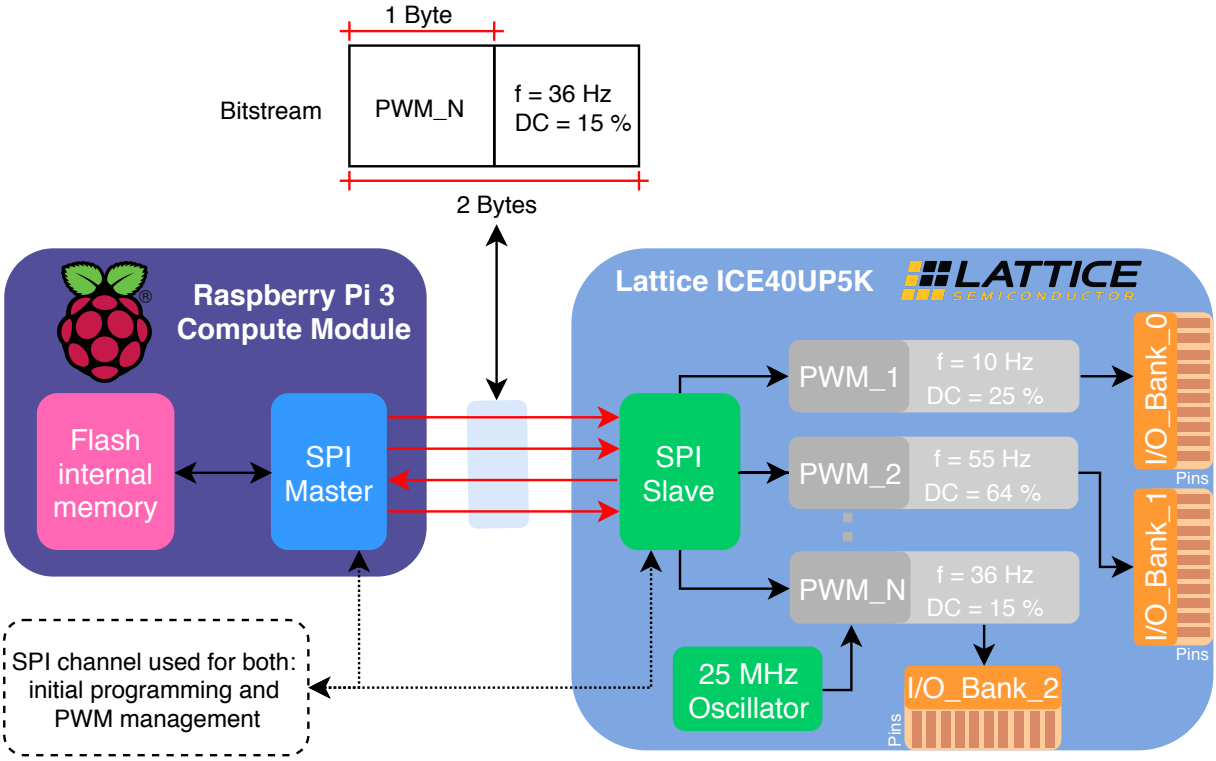


Figure 4.75 – *FPGA functioning as PWM generator*

The **bitstream** is composed of two bytes; while the first of them chooses the particular PWM to be set up, the second one states the configuration of it, in terms of **frequency** and **duty cycle**. The PWM are generated by using a 25 MHz oscillator, which is divided as needed to accomplish the desired frequency. Finally, the generated PWM outputs in **one of the pins** of the GPIO from the banks. As **Hardware Description Language** it has been used **Verilog**.

4.3.3.2 Actuator: reaction wheel

Actuators are used to change or control the attitude of the CubeSat. As for this prototype, it has been implemented a **reaction wheel**. However, the ADCS is capable of managing up to three Magnetorquers which can be added in the future, using also the different PWM signals available.

As reaction wheel, it has been reused the one available at GranaSAT laboratory, shown in Figure 3.32. In order to propel the wheel, it is necessary a DC motor. As usual, it is not possible to supply a DC motor directly; in this case, for example, it can provide with just a few mA, so an **H-bridge** is needed. It is a device composed of four switches that control the flow of large currents to a load, which is precisely its usefulness: allows, with

a low-power control signal (in this case, the PWM) to drive high loads, such as a motor. It is composed of a simple circuit, depicted in Figure 4.76.

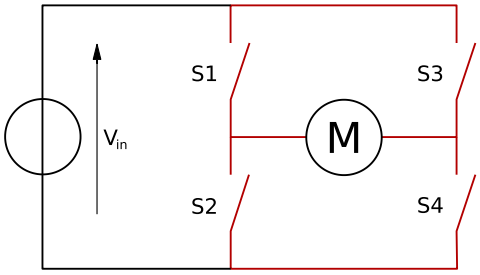


Figure 4.76 – *H-bridge basic circuit [59]*

In sum, the PWM generated by the FPGA attack the H-bridge (control signals) and it supplies the load with the needed current (larger than the one the FPGA can provide). The H-bridge could attack the motor directly and supply it with all the current demanded, however, this implies two drawbacks: on the one hand, the motor would be driven at the maximum speed continuously, which reduces its control dramatically; on the other hand, it would be propelled in the same fixed direction. An H-bridge allows controlling both parameters.

As for this design, several H-bridges are used, listed next:

- **ST L298N**

The L298N by ST [50] is an integrated high voltage, high current Dual Full-Bridge Driver. It allows an operating supply voltage of up to 46 V @ 4 A. Additionally, it features two *enable* inputs that allow the L298 to be enabled or disabled independently of the input signals. Because of its high power capabilities, it is used to control the motor driving the reaction wheel. It will be used a single unit, which provides with **four outputs**; as the motor only needs a couple, the other two are used for the X-axis magnetorquer.

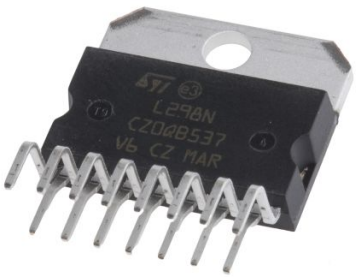


Figure 4.77 – *ST L298N [50]*

- **ROHM BD6211F-E2**

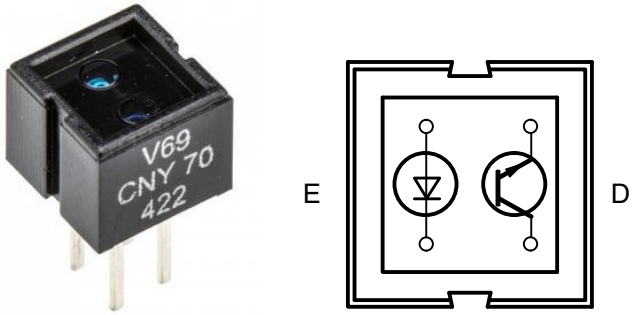
As stated before, the ADCS will count with the capability to use Magnetorquers in the future. They must be supplied with an H-bridge, because of the same reasons of a motor, essentially. However, the current needed for Magnetorquers is much lower, so a smaller and cheaper bridge is used, particularly the BD6211F-E2 by ROHM [42]. It is able to supply with up to 1 A with 5.5 V.

Figure 4.78 – *ROHM BD6211F-E2 [42]*

Each BD6211F-E2 exhibits two outputs; considering that the X-axis magnetorquer is supplied by the L928N, each of the axis left needs another couple of outputs so they are necessary two BD6211F-E2.

4.3.3.3 Sensor: Tachometer

In order to adequately control the reaction wheel, it is implemented a **tachometer**, able to measure the rotation speed of the wheel. To design the tachometer, it is used a **Reflective Optical Sensor**. These devices produce an infrared light onto a certain surface and the reflection is measured; when there is no reflection the sensor will show a **0** at the output, while a certain reflection will produce a **1**. Particularly, the sensor used is the CNY70 by Vishay [56], shown in Figure 4.79a.



(a) *Package* **(b)** *Top view*

Figure 4.79 – *Vishay CNY70 [56]*

The maximum distance at which the CNY70 is able to measure reflection depends on the **forward current** of the transistor, which in turn is subject to the emitter diode current. Therefore, the resistors needed to bias the device have to be optimized. The maximum measurement distance is accomplished when the forward current is about **20 mA**, so regarding the **diode**, the following equations can be considered:

$$I_{\text{forward}} = \frac{V_{\text{cc}} - V_{\text{forward}}}{R_{\text{diode}}} \tag{4.3.2}$$

Considering a supply of 3.3 V and taking into account that, according to the datasheet [56], the typical forward voltage is about **1.25 V**, it yields to:

$$20\ \text{mA} = \frac{5\ \text{V} - 1.25\ \text{V}}{R_{\text{D}}} \rightarrow R_{\text{D}} = 102.5\,\Omega \tag{4.3.3}$$

As for the phototransistor, for a typical collector current of 0.1 mA, at a distance of 0.3 mm, the maximum V_{CE} under saturation is 0.3 V. So, substituting in Equation 4.3.4:

$$I_{\text{C}} = \frac{V_{\text{cc}} - V_{\text{CE}}}{R_{\text{collector}}} \rightarrow R_{\text{collector}} = 30\,\text{k}\Omega \tag{4.3.4}$$

In order to test the accuracy of the system, the CNY70 is used with the configuration determined to measure the rotation rate of a reaction wheel. At the same time, it is measured using a professional tachometer. The results are plot together at Figure 4.80.

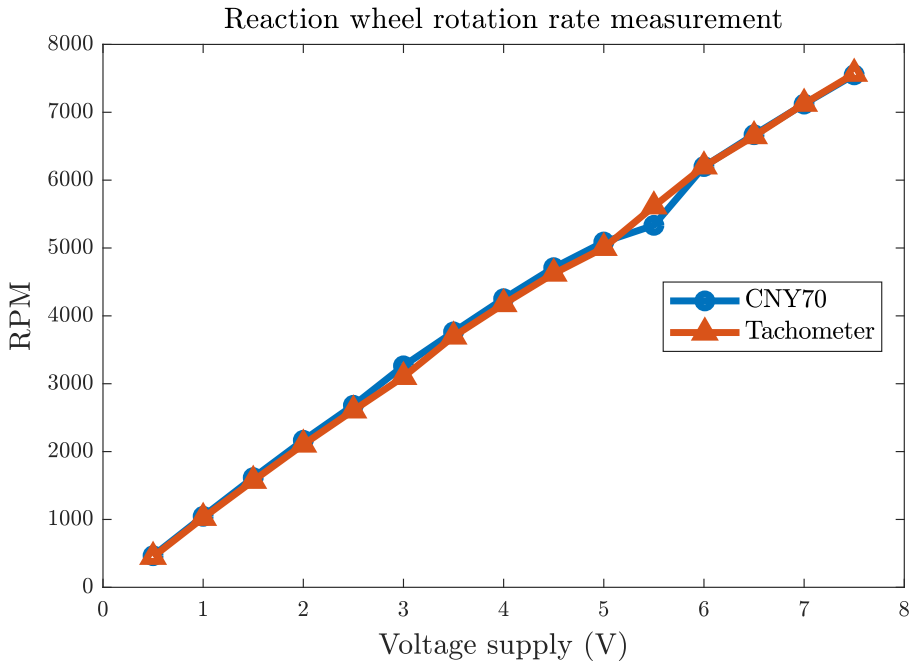
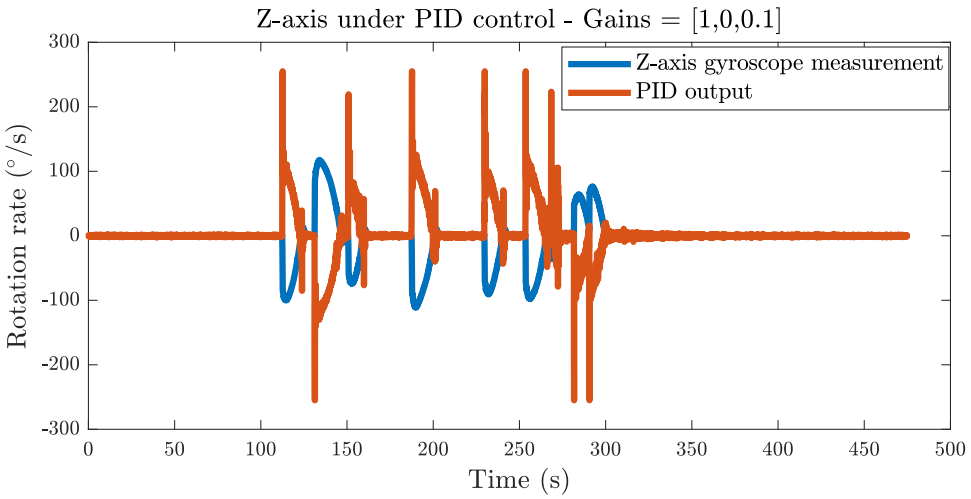


Figure 4.80 – *Rotation rate measured with a professional tachometer and with the CNY70 circuit designed*

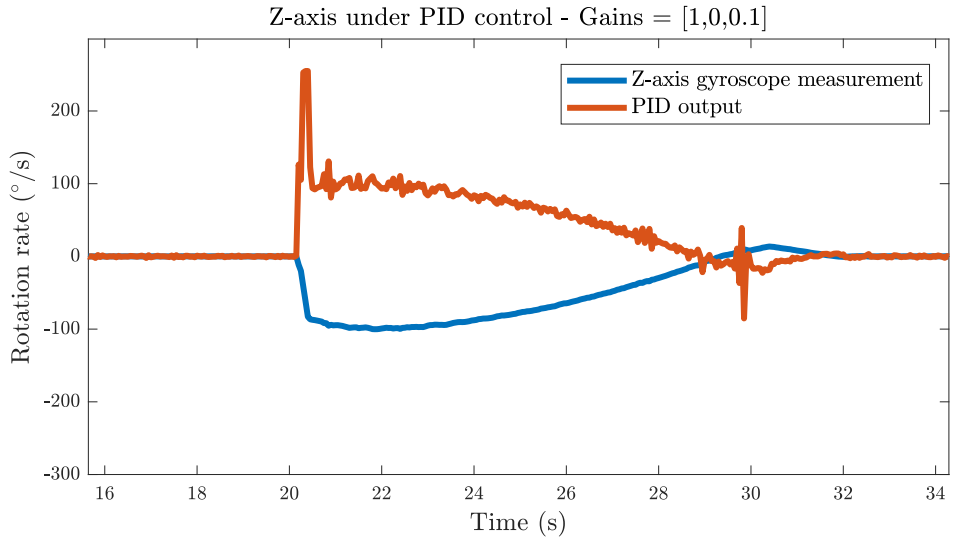
As expected, the rotation rate is linear with the voltage supply. On the other hand, the error made is completely negligible, with an almost exact coincidence between both measurements. This validates the design of the tachometer based on the CNY70.

4.3.3.4 Control law

The control law is the software implementation governing the hardware above. It deals with the variety of situations along a mission and acts consequently. As starting point for this prototype, it is implemented a control law based on a PID, described in section 3.4.2.3.1. Particularly, it has been used the library **simple_pid** in Python [48]. Figure 4.81 shows an example of the PID counteracting the movement of the CubeSat.



(a) *Long operation*

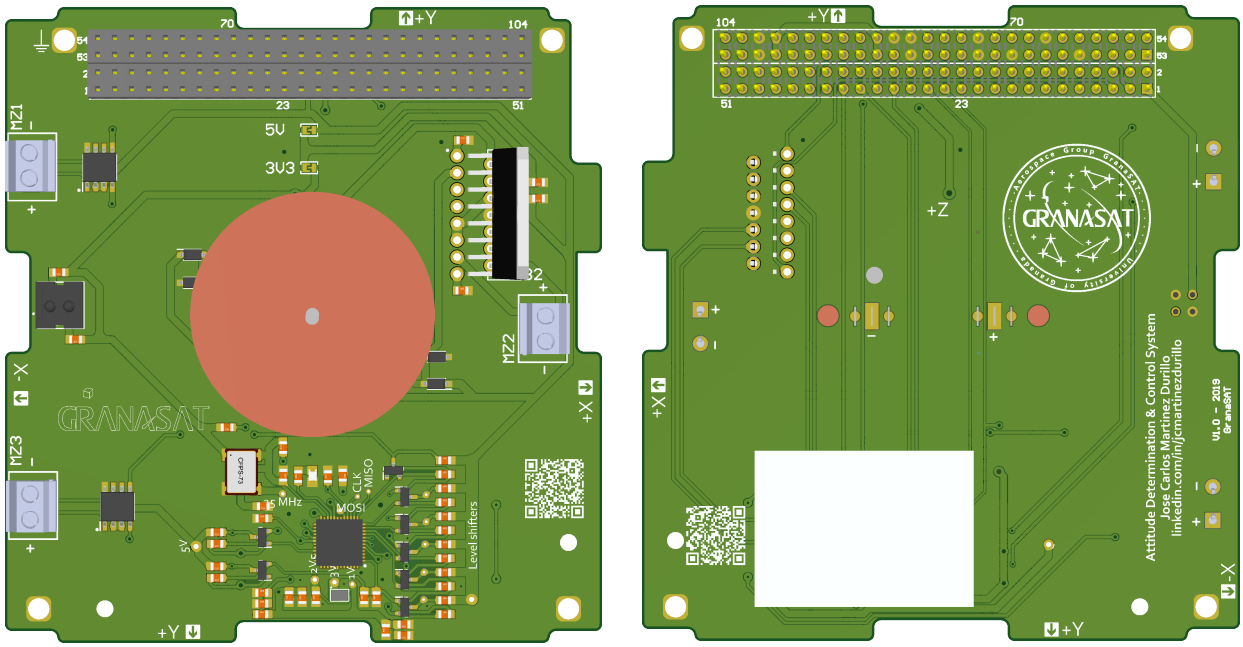


(b) *Detail*

Figure 4.81 – *PID counteracting Z-axis rotation*

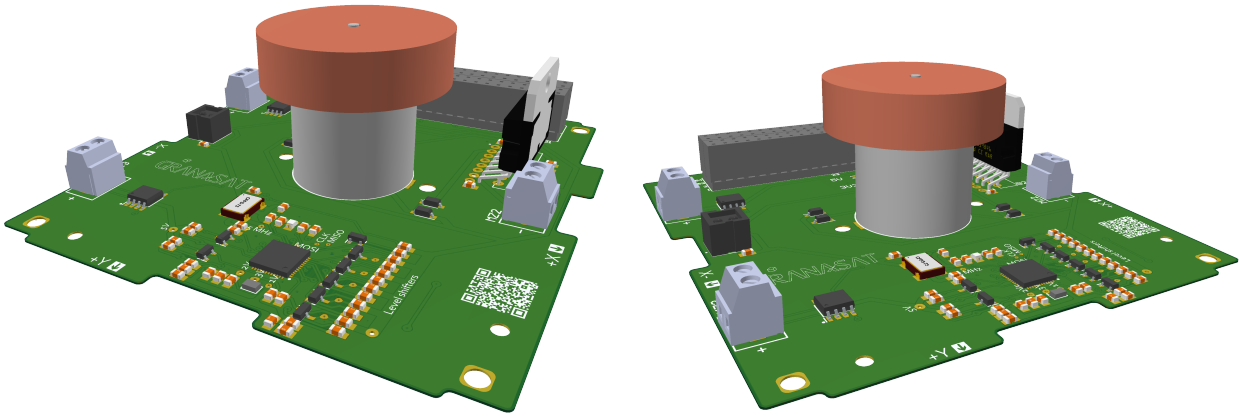
4.3.3.5 PCB design

Figure 4.82 shows a 3D render of the top and bottom layer in the final design, while Figure 4.83 shows a perspective view from different angles.



(a) *Top* **(b)** *Bottom*

Figure 4.82 – *ADCS PCB final design*

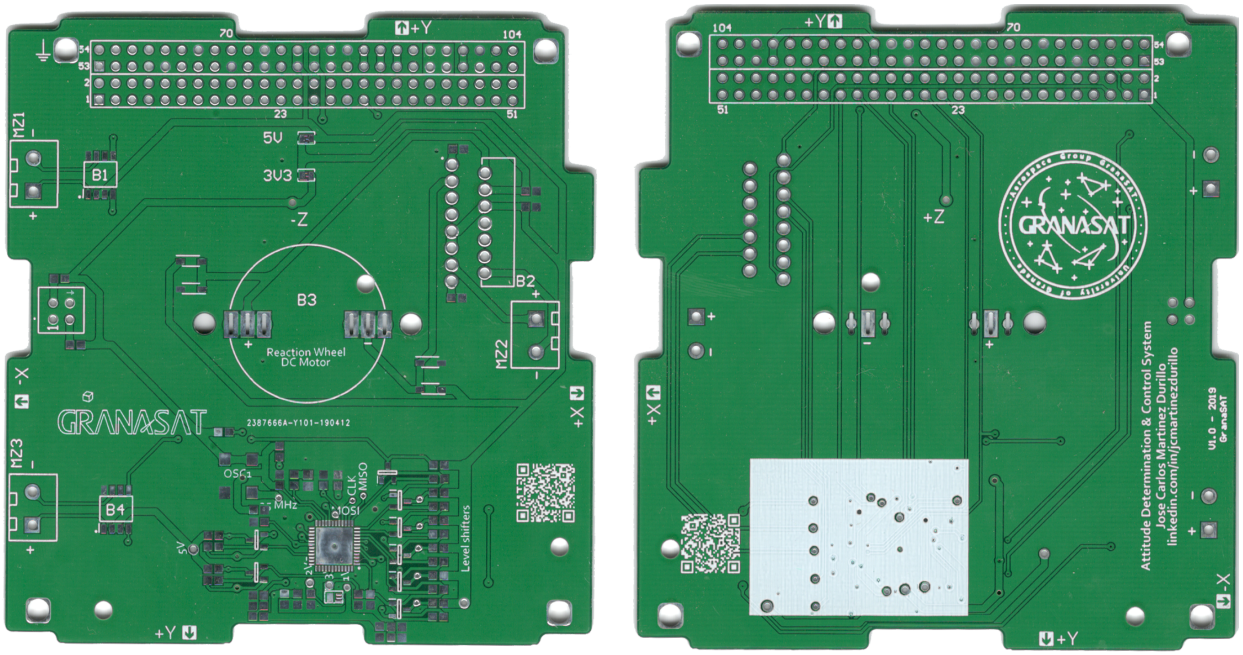


(a) **(b)**

Figure 4.83 – *ADCS PCB 3D perspective*

4.3.3.6 Manufacturing and soldering

In this case, the PCB is composed of 2 layers. The design is sent to a professional manufacturer too, resulting in the PCB shown in Figure 4.84.



(a) *Top* **(b)** *Bottom*

Figure 4.84 – *ADCS PCB manufactured*

As for the soldering process, the procedure is the same as the one described in section 4.3.2.6; the result is shown in Figure 4.85.

Figure 4.85 – *ADCS PCB after assembling and soldering*

4.3.4 Electrical Power System (EPS)

Regarding the EPS, this Master's Thesis addresses some of the crucial choices to be made when facing the design of this kind of system. While the electronics design is left for future projects, this one will lay the foundations for the photovoltaic subsystem of the **GranaSAT-I**, by analyzing and characterizing different solar panels in order to determine the optimum one according to the rest of the CubeSat. On the other hand, a set of batteries are characterized at different charge/discharge rates, solving part of the decisions regarding the energy storage system.

4.3.4.1 Solar panels

Solar cells were profusely analyzed in Chapter 3 so in this section, the **testbench** designed for its characterization is presented first and then different solar panels are characterized.

4.3.4.1.1 Measurements testbench

Characterizing solar panels is a complex matter. As described in previous sections, generally speaking, solar cells behavior is described by its I-V curve. However, obtaining said curve is not as simple as it could seem: if the output of a solar panel is measured when it is under light, a certain voltage is obtained, the V_{OC}, which is useless as it cannot deliver any power at that point (see section 3.4.3.1.4). So in order to get the complete curve it is necessary to sweep a voltage range when the solar panel is **loaded**, this is, emulating a real operation environment. It is easy to understand under the basis stated in previous explanations: if the solar panel is forced to work at a certain voltage, against a certain load, it will deliver a given power; if that voltage varies, so does the rest, getting the complete I-V curve. Particularly, the voltage sweep must cover the whole expected functioning range.

Simulating all these variables requires an adequate **testbench** composed of different equipment, depicted in Figure 4.86. The testbench has been designed in a way which allows measuring in both, **outdoor** (solar light) and **indoor** (simulators) environments. Firstly, the solar panel is connected to the **load**, which is composed of two devices: the **HP 6063B** [23] and the **HP E3631A** [22]. While the first is an electronic load, used to perform the **voltage sweep**, the second is a simple DC power source used to impose a certain voltage offset to the HP 6063B output, needed for correct functioning. Both, acting as load, forces the solar panel output a certain voltage ($V_{\mathbf{sweep}}$); as it is exposed to an irradiance source (whether actual Sun light or simulated), the solar panel provides with the **current** corresponding to the impedance seen from the panel, which varies with the voltage sweep. As the sweep is performed, both the voltage imposed and the current

provided is measured using the multimeter SIGLENT SDM3065X [46]. Additionally, the testbench counts with a pyranometer to be placed along with the solar panel, so the irradiance present at the test can be taken into account; it is also connected to the multimeter, and the voltage measured is converted into an irradiance measurement.

The whole equipment is controlled using a MATLAB® library developed progressively throughout the years at the GranaSAT laboratory, which also allows downloading the data. Finally, the I-V curve can be plotted.

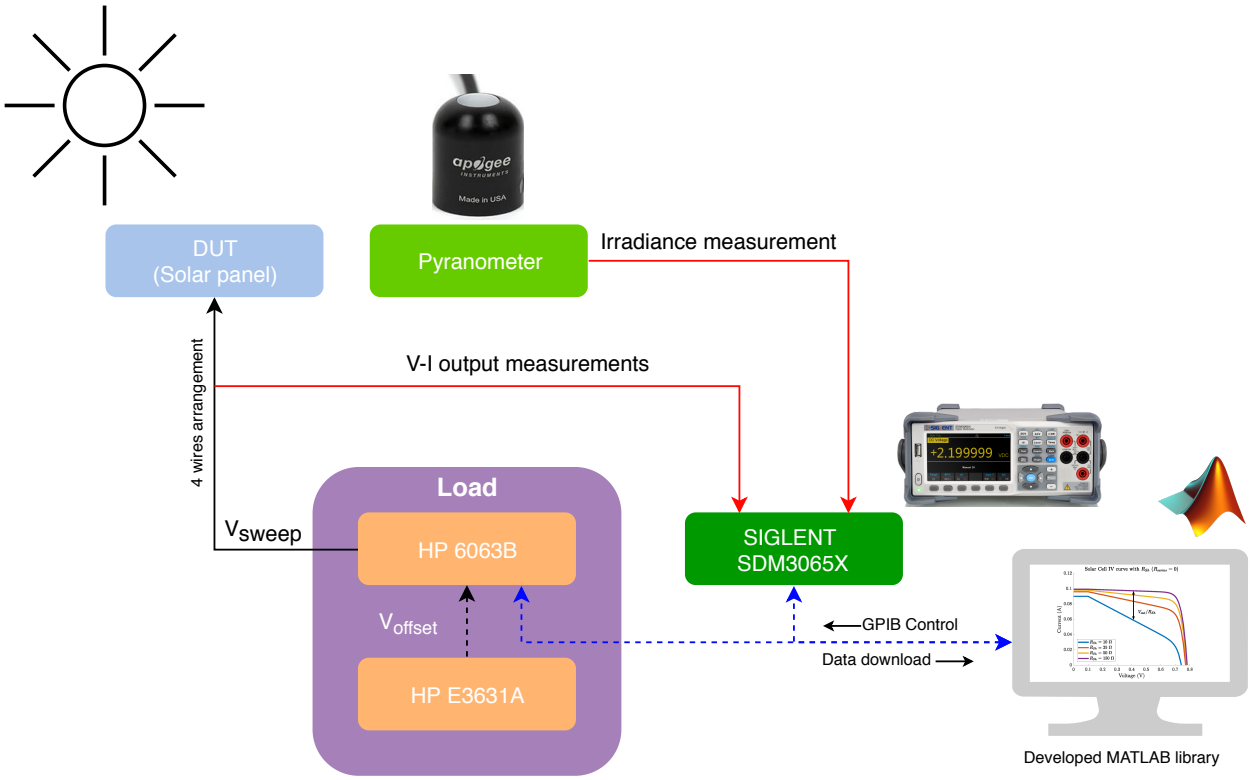


Figure 4.86 – *Solar panels characterization testbench*

4.3.4.1.2 Characterizations

In this section, different solar panels are characterized using the testbench just described. Firstly, in order to assure the correct functioning of the characterization bench, it is performed a test under Sun's illumination, on a high quality solar panel with a known IV curve. It is a 5 W polycristalline solar panel with an open-circuit voltage of 22 V [1], shown in Figure 4.87.

The irradiance measured outdoors is 1000 W/m^2 and it slightly varied for all the characterizations with solar lightning performed.

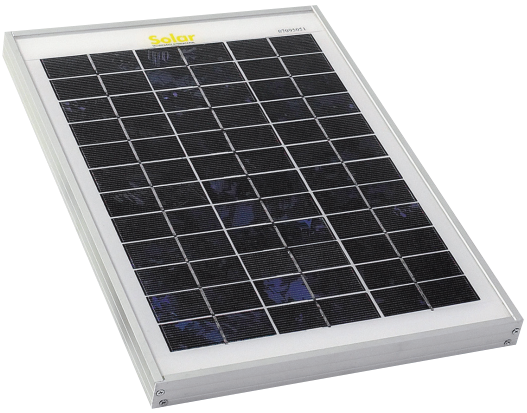


Figure 4.87 – *5 W solar panel [1]*

Figure 4.88 shows the IV curve of the panel, along with the power delivered by in those conditions.

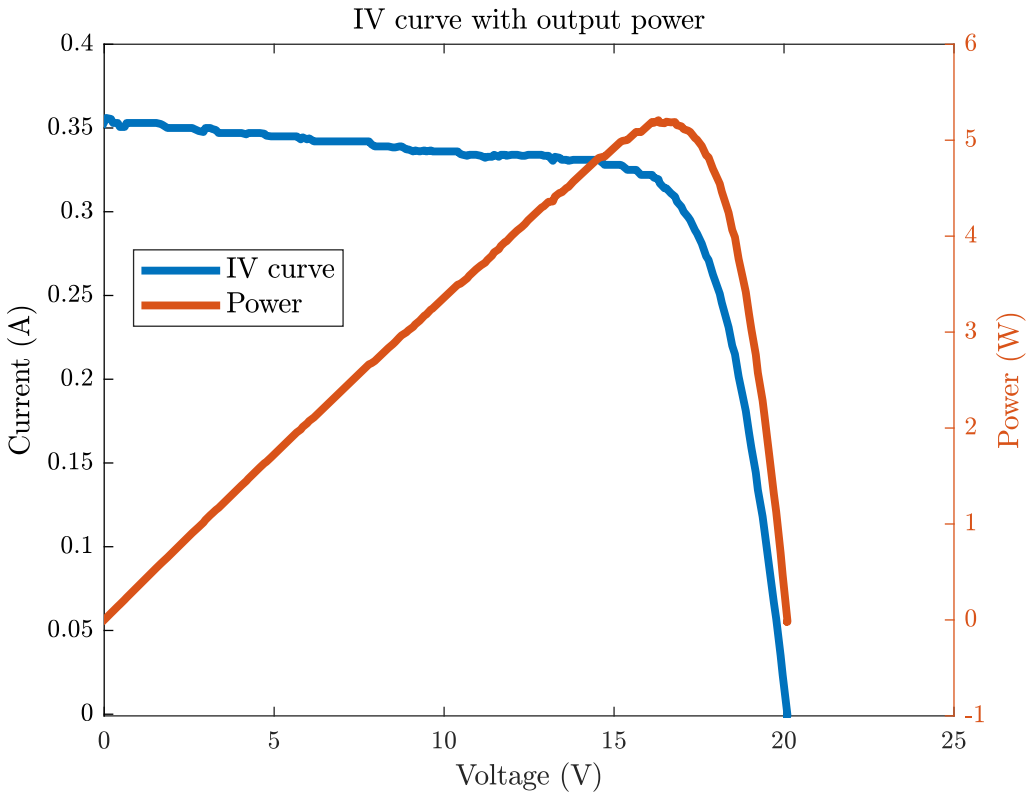
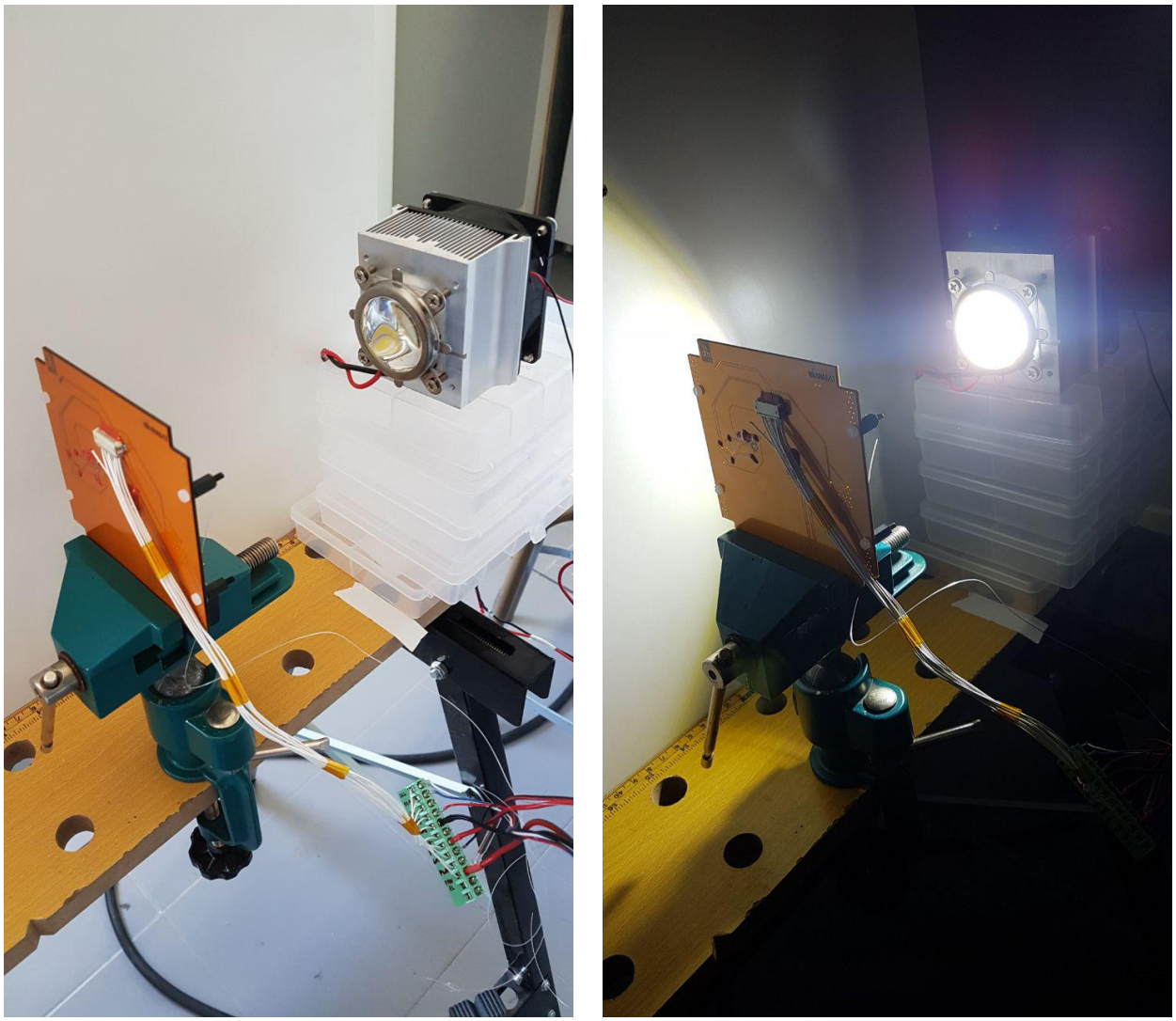


Figure 4.88 – *Characterization of the RS solar panel*

The panel behaves exceptionally well, with an output even **above 5 W** and an **MPP = [17 V, 0.31 A]**. This **validates the functioning of the testbench**, so in the next pages different small-sized solar panels are also characterized.

The solar panels candidates to be used in the CubeSat will be characterized facing both Sun's and LED lightning. Figure 4.89 shows the assembly for the measurement using the LED simulator.



(a) **(b)**

Figure 4.89 – *Assembly to characterize solar panels with LED lightning*

The results are divided into the three solar panels characterized, each one with two plots depending on the illumination. For each panel, it is indicated the theoretical performance according to the manufacturer. Figure 4.90 shows the panels under test.

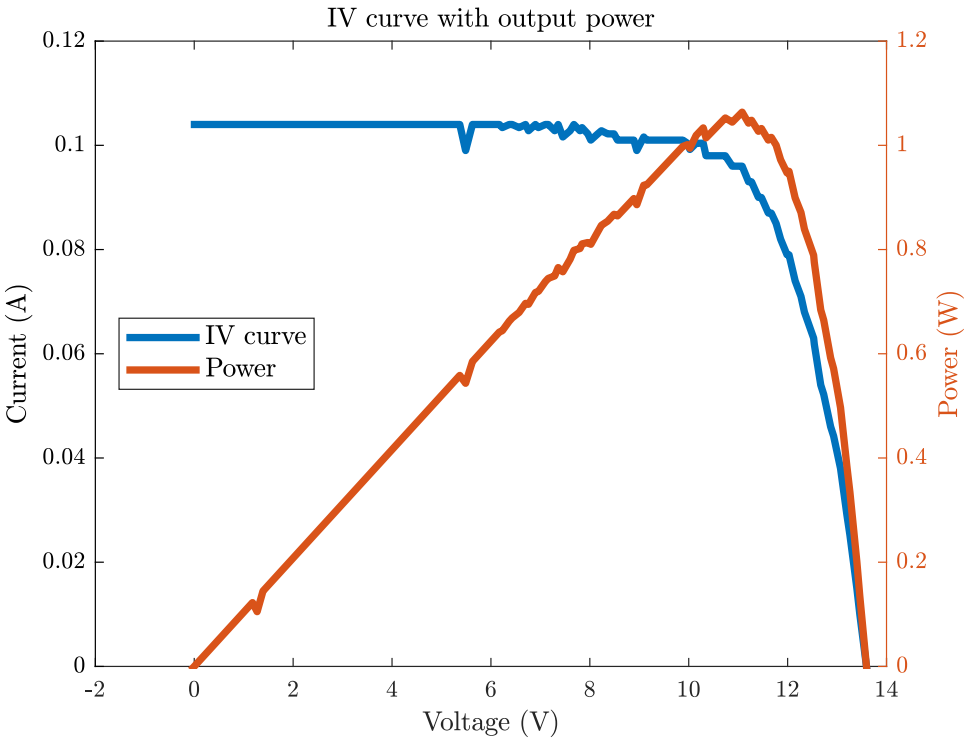
(c) *3: 5.5 V @ 1 W*

(b) *2: 12 V @ 1.5 W*

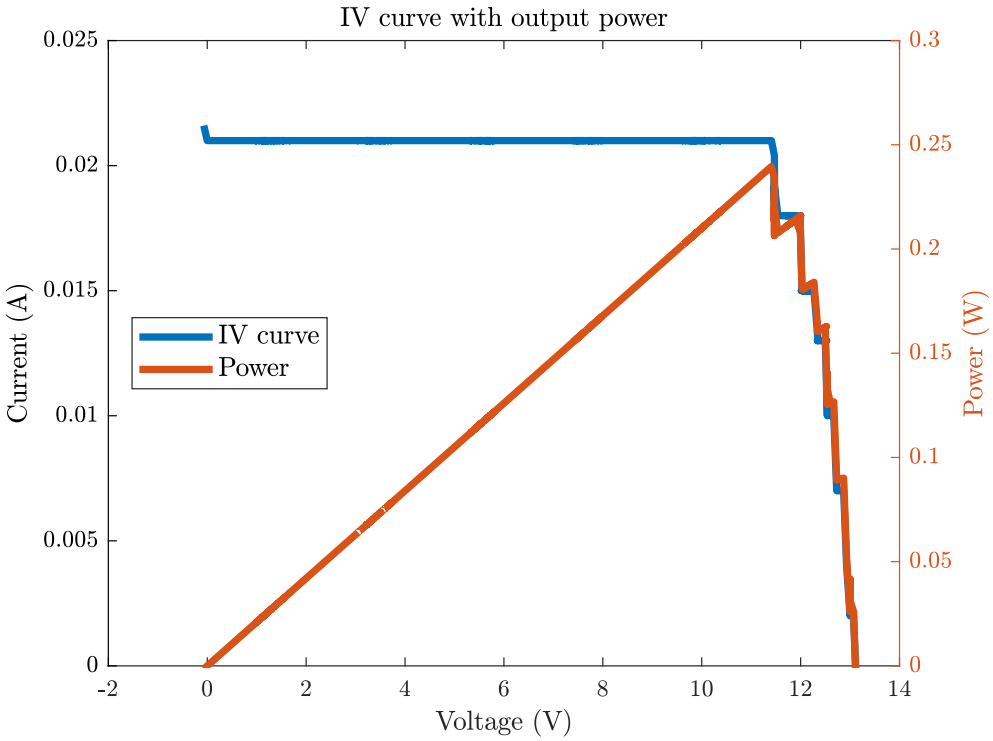
(a) *1: 5 V @ 1.3 W*

Figure 4.90 – *Solar panels characterized*

- **Solar panel 1: 5 V @ 1.3 W**



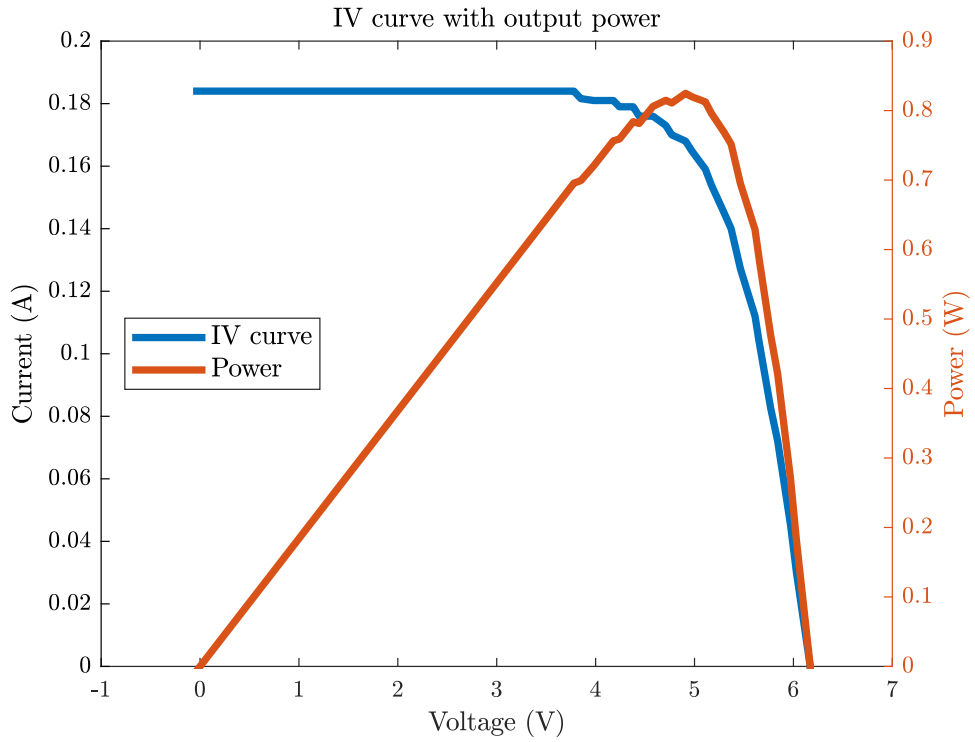
(a) *Solar lightning - MPP = (11 V, 0.097 A)*



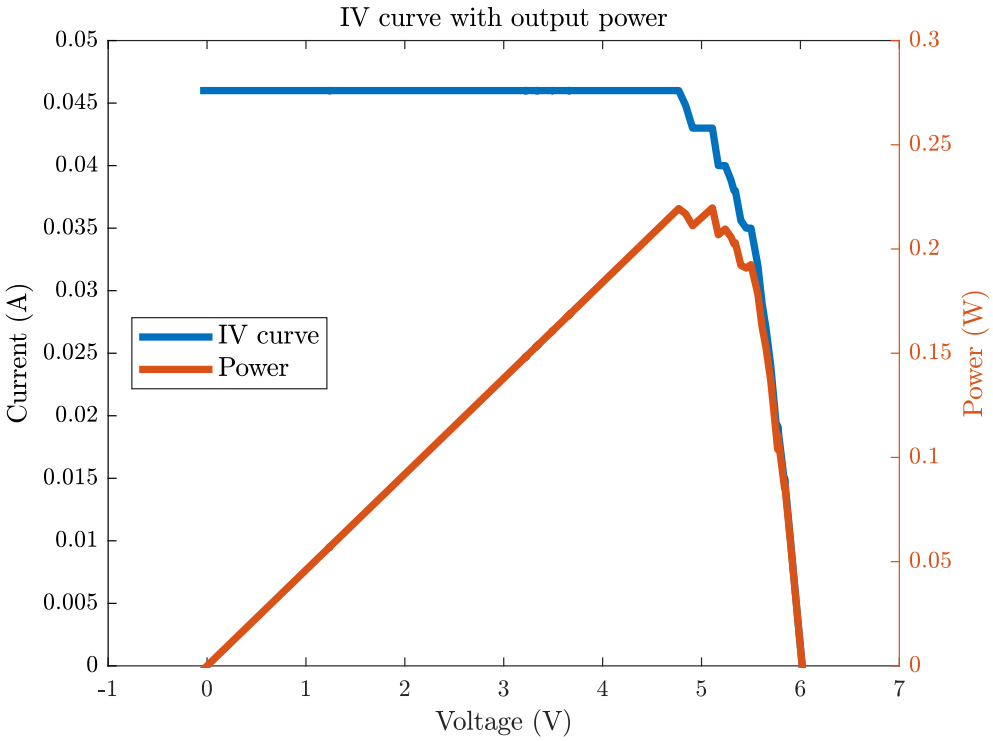
(b) *LED lightning - MPP = (11.5 V, 0.0197 A)*

Figure 4.91 – *Solar panel 1. Characterization results*

- **Solar panel 2: 12 V @ 1.5 W**



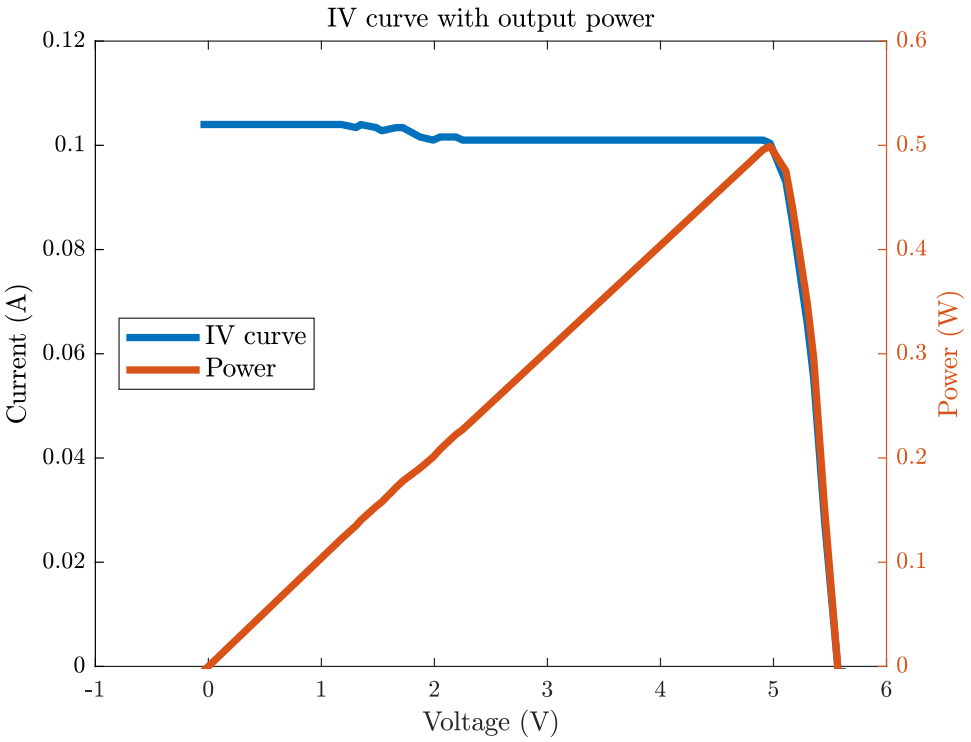
(a) *Solar lightning - MPP = (5 V, 0.165 A)*



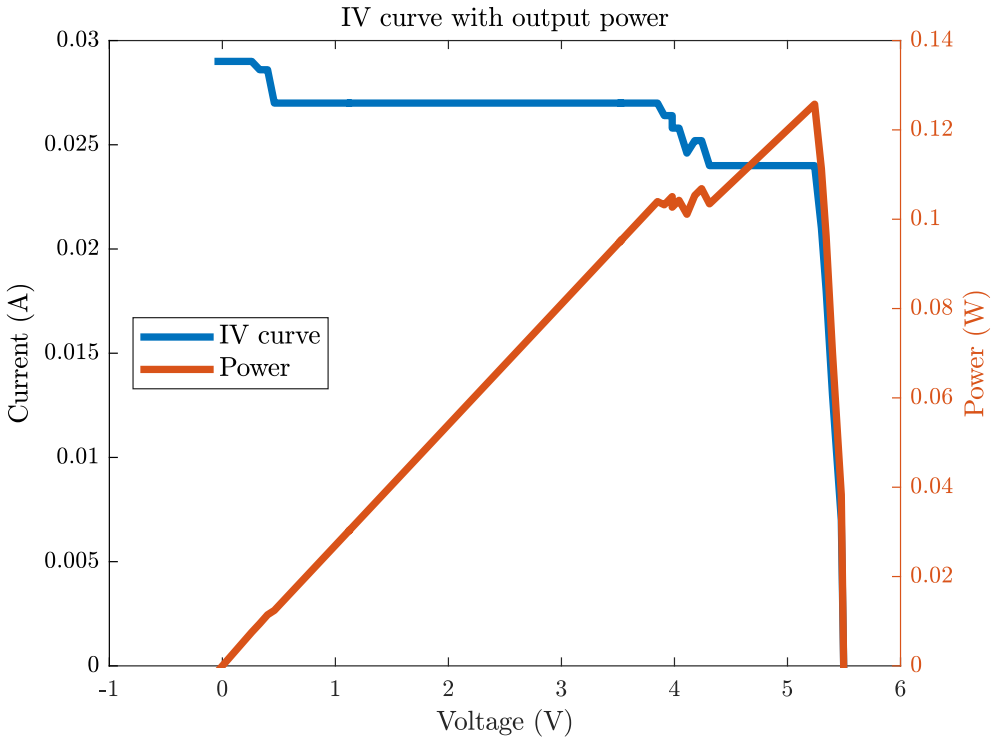
(b) *LED lightning - MPP = (5 V, 0.035 A)*

Figure 4.92 – *Solar panel 2. Characterization results*

- **Solar panel 3: 5.5 V @ 1 W**



(a) *Solar lightning - MPP = (5 V, 0.1 A)*



(b) *LED lightning - MPP = (5.3 V, 0.024 A)*

Figure 4.93 – *Solar panel 3. Characterization results*

Panel	Lightning	V_{MP} (V)	I_{MP} (A)	P_{MP} (W)	Fill Factor	Incident power (W)	η
1							
	Sun	11	0.097	1.067	0.70	8.8	12 %
	LED	11.5	0.0197	0.23	0.82	2.07	11 %
2							
	Sun	5	0.165	0.825	0.75	9.78	8 %
	LED	5	0.035	0.175	0.64	2.30	8 %
3							
	Sun	5	0.1	0.5	0.81	9.03	6 %
	LED	5.3	0.024	0.13	0.80	2.12	6 %

Table 4.2 – *Results of the solar panels characterized*

From the previous plots, several parameters can be obtained; they are arranged in Table 4.2 along with the figures of merit of each panel. Definitions and formulas for the parameters exposed can be found in section 3.4.3.1.4.

Additionally, several conclusions can be extracted from the previous characterization, summarized next.

- In order to get realistic results, the incident power has been calculated for each panel, i.e., the power received by the solar panel is not the one irradiated by the source (1000 W/m^2 for the Sun or 235 W/m^2 for the LED), but the **proportional part** considering the panel's size. Indeed, the **packing density** also has to be taken into account, with an estimated value of **0.8**.

- Although, at first sight, the panels seem to exhibit a significantly worse performance when the LED simulator is used, actually **the efficiency is almost the same**. Particularly, panels 2 and 3 features the same **power conversion efficiency** regardless of the lightning.

- On the other hand, the power delivered is about a fifth when the LED simulator is lightning, but that is logical, considering that the incident power is also about **20 %** of the one provided by the Sun.

- The V_{MP} is also similar regardless of the irradiance source used, which means that, for an equivalent load, **the spectrum of both power sources is reasonably similar too**. The decrease in power is then a consequence of the decrease in the **output current**, I_{MP}, directly related to the incident power.

- In the same way, the **Fill Factor** slightly differs with the kind of lightning, indeed, the panel 3 features almost an identical factor. Furthermore, **in all cases the factor is pretty acceptable**, with a minimum value of 0.64 when the panel 2 is lighted with the LED simulator. With a mean value of about 0.75, this implies that about that percentage of the power in shortcircuit is provided as **real power**.

- Regarding the **power conversion efficency** or η is really poor in all cases; this issue was expectable considering that these are low-cost solar panels. However, while the efficiency exhibited by panels 2 and 3 is even lower than typical values, the efficiency associated to the panel 1 (12 %) **can be deemed acceptable.**
- In sum, this testbench has allowed, on the one hand: characterizing the behavior of different solar panels in terms of **power and figures of merit**; on the other hand, **validating** the LED simulator, as the results are coherent with the ones obtained using solar lightning, considering the **difference in incident power**.

The final choice will depend on the overall design of the EPS as well as on the battery used.

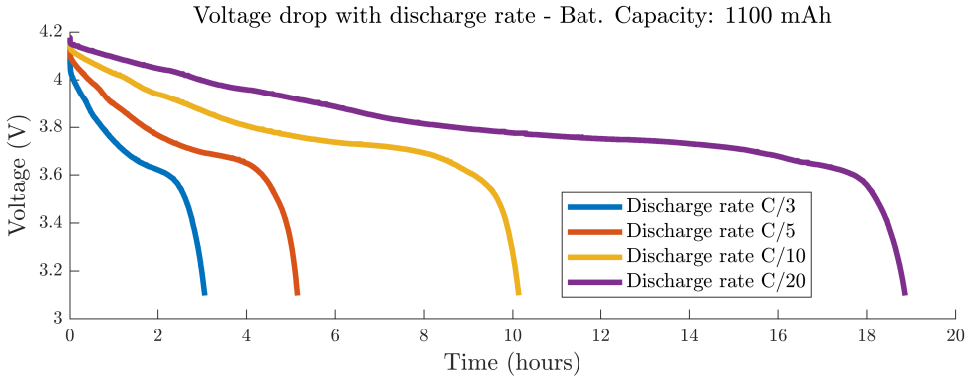
4.3.4.2 Batteries characterization

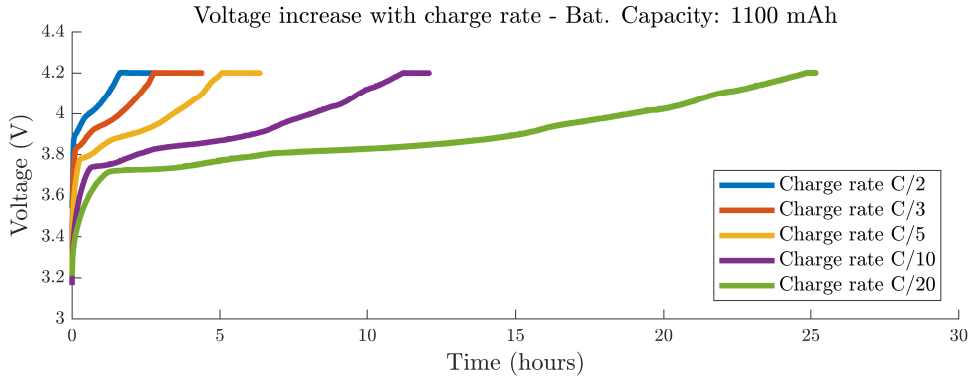
As for the energy storage subsystem, because this Master's Thesis has addressed the power consumption of some elements, such as the **Central Processing Unit** (see section 4.3.2.1.2), some Li-ion batteries are characterized, so it can be used as a reference for the future EPS. The definitions of the most relevant **figures of merit** related to batteries, mentioned next, can be found in section 3.4.3.3.

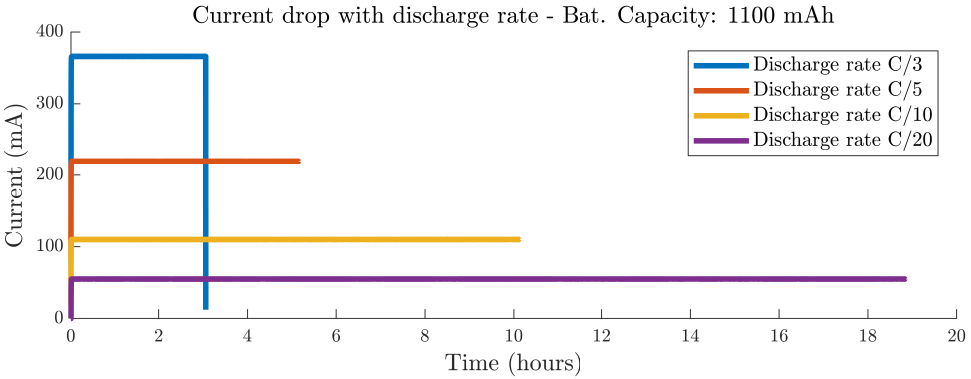
Particularly, several batteries with different **capacities** (1100 and 1800 mAh) are tested using a wide range of values for the **dimensional abuse**. This test is performed using the battery charging centre **VOLCRAFT ALC-8500**, shown in Figure 4.94.

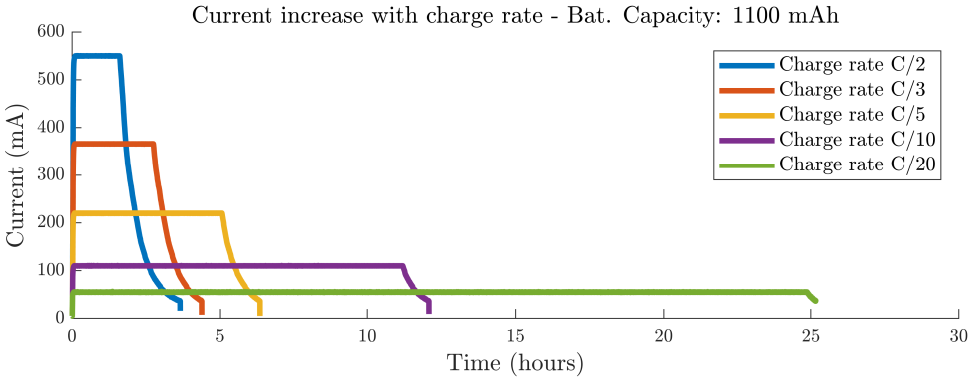
Figure 4.94 – *VOLCRAFT ALC-8500*

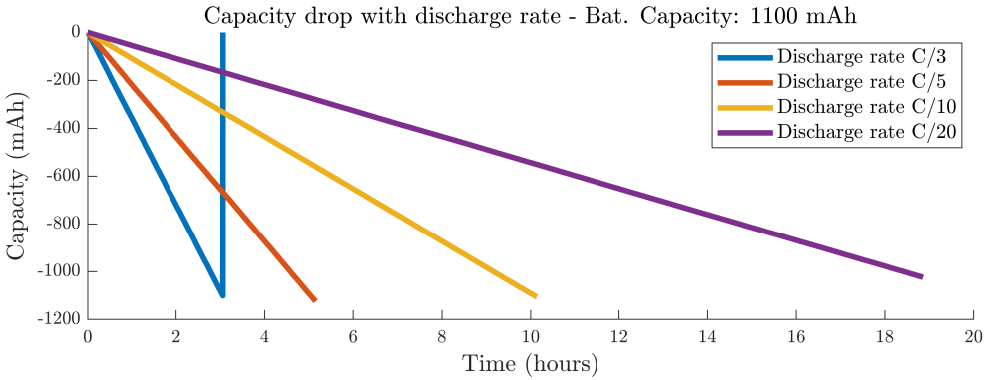
Once again, the results allow both, characterizing the batteries and making a first approximation to the energy needs of the system and the battery **run-time** when supplying it. The results are shown in the following pages.

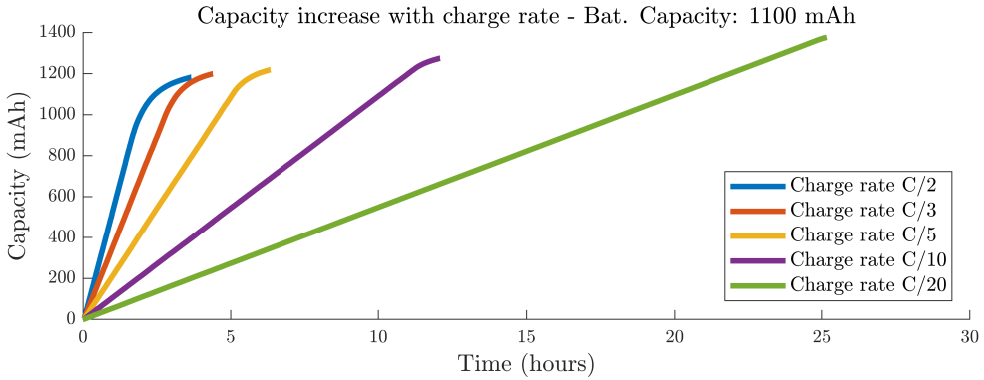

Voltage drop with discharge rate - Bat. Capacity: 1100 mAh
Voltage (V)
Time (hours)
Discharge rate C/3
Discharge rate C/5
Discharge rate C/10
Discharge rate C/20

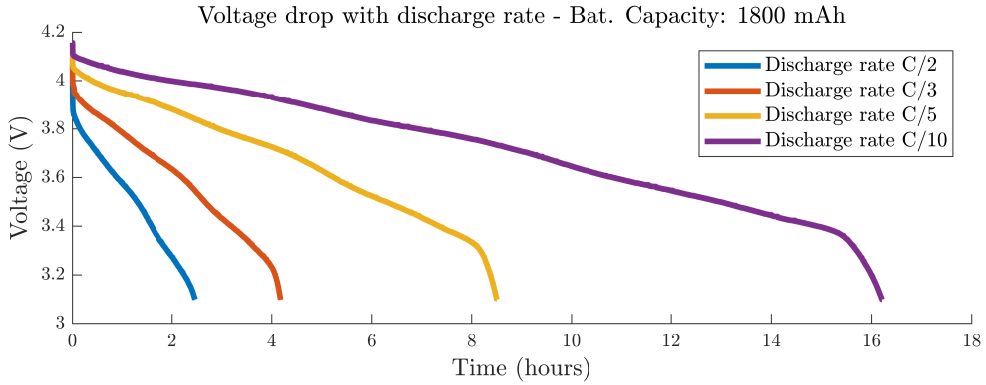

Voltage increase with charge rate - Bat. Capacity: 1100 mAh
Voltage (V)
Time (hours)
Charge rate C/2
Charge rate C/3
Charge rate C/5
Charge rate C/10
Charge rate C/20

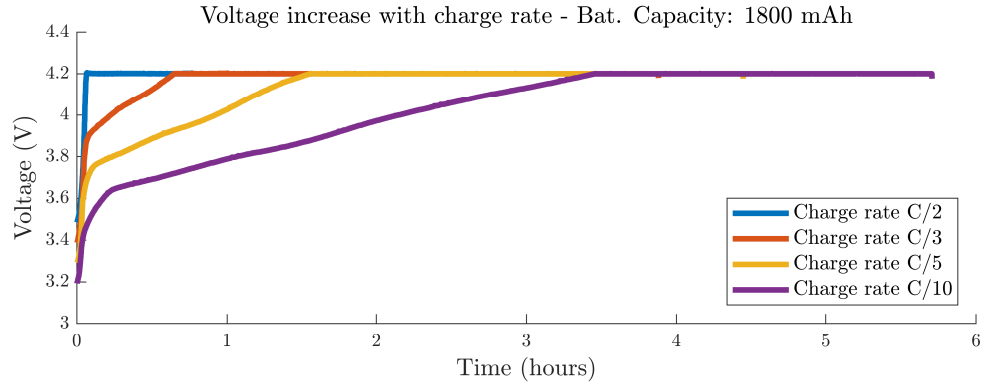

Current drop with discharge rate - Bat. Capacity: 1100 mAh
Current (mA)
Time (hours)
Discharge rate C/3
Discharge rate C/5
Discharge rate C/10
Discharge rate C/20

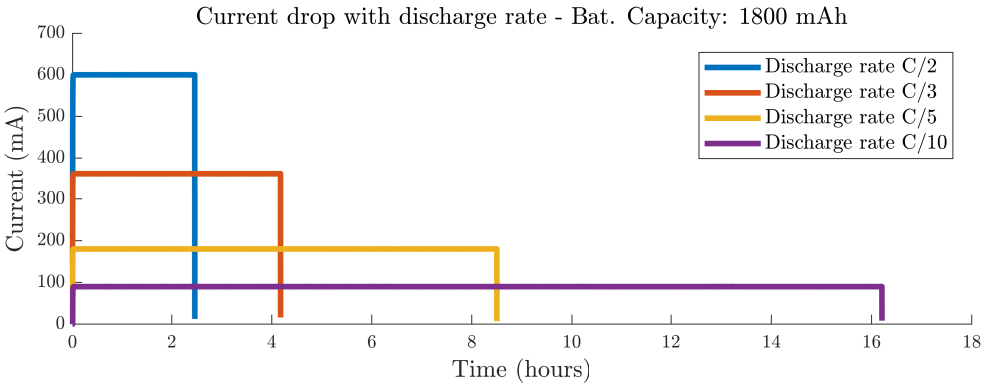

Current increase with charge rate - Bat. Capacity: 1100 mAh
Current (mA)
Time (hours)
Charge rate C/2
Charge rate C/3
Charge rate C/5
Charge rate C/10
Charge rate C/20

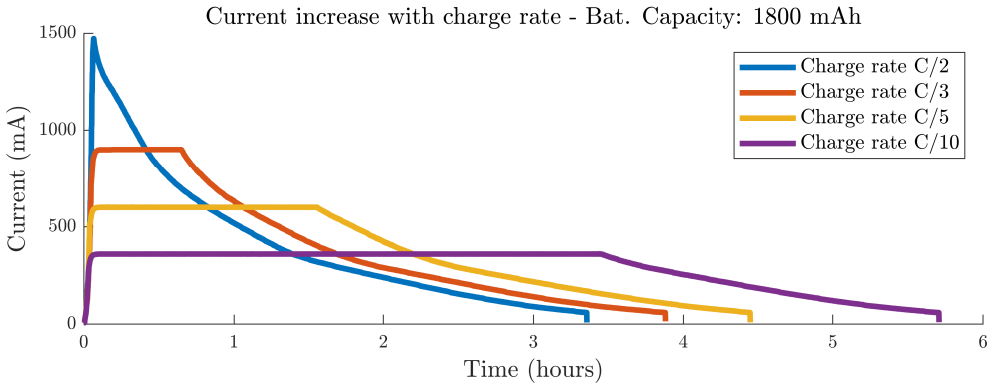

Capacity drop with discharge rate - Bat. Capacity: 1100 mAh
Capacity (mAh)
Time (hours)
Discharge rate C/3
Discharge rate C/5
Discharge rate C/10
Discharge rate C/20

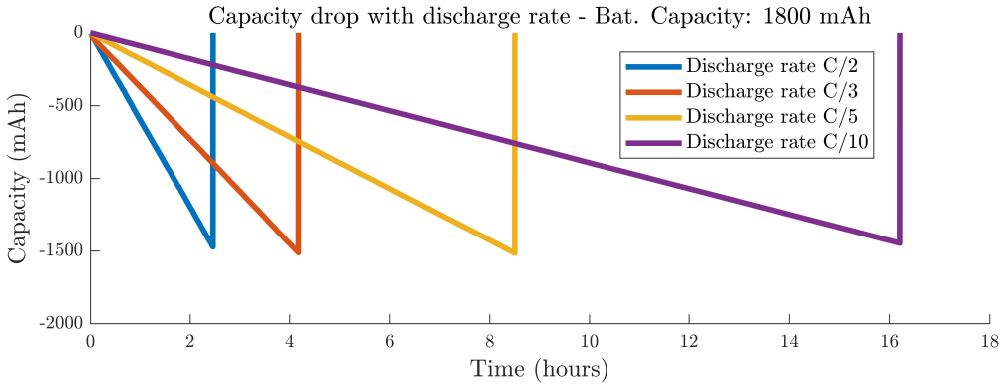

Capacity increase with charge rate - Bat. Capacity: 1100 mAh
Capacity (mAh)
Time (hours)
Charge rate C/2
Charge rate C/3
Charge rate C/5
Charge rate C/10
Charge rate C/20

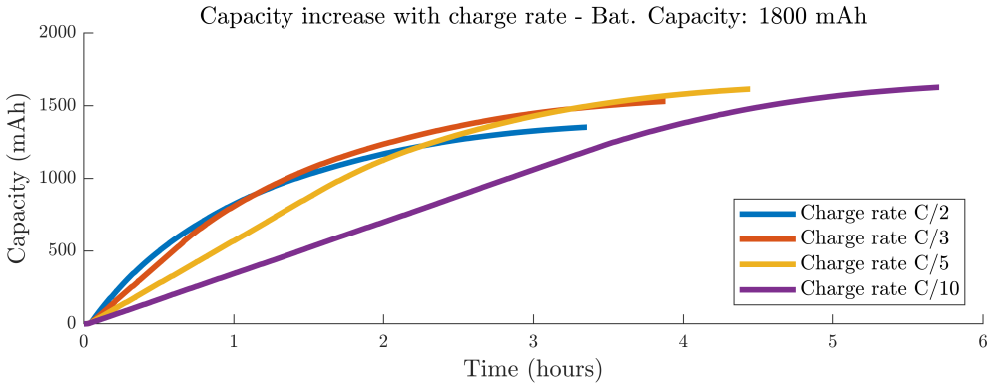

Voltage drop with discharge rate - Bat. Capacity: 1800 mAh
Voltage (V)
Time (hours)
Discharge rate C/2
Discharge rate C/3
Discharge rate C/5
Discharge rate C/10


Voltage increase with charge rate - Bat. Capacity: 1800 mAh
Voltage (V)
Time (hours)
Charge rate C/2
Charge rate C/3
Charge rate C/5
Charge rate C/10


Current drop with discharge rate - Bat. Capacity: 1800 mAh
Current (mA)
Time (hours)
Discharge rate C/2
Discharge rate C/3
Discharge rate C/5
Discharge rate C/10


Current increase with charge rate - Bat. Capacity: 1800 mAh
Current (mA)
Time (hours)
Charge rate C/2
Charge rate C/3
Charge rate C/5
Charge rate C/10


Capacity drop with discharge rate - Bat. Capacity: 1800 mAh
Capacity (mAh)
Time (hours)
Discharge rate C/2
Discharge rate C/3
Discharge rate C/5
Discharge rate C/10


Capacity increase with charge rate - Bat. Capacity: 1800 mAh
Capacity (mAh)
Time (hours)
Charge rate C/2
Charge rate C/3
Charge rate C/5
Charge rate C/10

Some conclusions can be extracted from the previous results:

- **Neither voltage drop nor increase is linear** with discharge or charge respectively. Therefore, neither battery duration can be expected to behave linearly, e.g., with a **discharge rate** C/10, the run-time of the 1100 mAh battery is about 10 hours, and with C/20, instead of 20 hours, is slightly 19 hours.

- As expected, the battery behaves as a **constant current source**, whose voltage varies. This can be seen clearly in the plot analyzing the current drop when discharging. However, when charging the behavior is different, although current is constant for a while, at the end of the charge, **it decreases almost exponentially**.

- Because current drop is constant, capacity drop is too, shown at the third plot on the left. It is interesting to point out another phenomena: **the batteries do not completely discharge**. For instance, see the capacity increase in the 1100 mAh battery; it clearly **surpasses its theoretical capacity**, reaching more than 1200 mAh. However, the capacity drop reaches the 1100 mAh stated by the manufacturer. This is actually a **protection mechanism** of the charging centre, which prevents the battery voltage from dropping below a certain value (about 3.1 V), as this would permanently **damage the battery.**

- It is worth noting that, when charging the 1800 mAh battery, the capacity slightly surpasses 1500 mAh, but it should reach almost 1900 mAh, according to the behavior seen before. Of course, this supposes that the capacity drop is about 1500 mAh too, instead of 1800 mAh. In this case, the manufacturer has provided a **fake total capacity.**

- Another interesting point is that batteries **cannot work at every dimensional abuse**; it depends on the specific battery. If a certain battery is forced to work at an inadequate rate, its behavior will be somewhat erratic. An example of this can be seen when the 1800 mAh battery is charged at a C/2 rate. In fact, the current increase with charge is not constant at any point, after reaching the maximum it immediately start falling exponentially. The voltage increase with this rate differs from the others too. Although not strange behavior is noticed when discharged at this rate, **it is inadvisable to use the battery with this configuration**, to prevent any damage.

In order to get a rough approximation of the battery run-time when supplying the CubeSat, it can be applied the **Pareto principle**: the 80 % of the whole power consumption is produced by only a 20 % of the components, in this case, the Raspberry Pi 3 Compute Module. Therefore, it is estimated the battery run-time assuming that it supplies only that component, using a discharge **dimensional abuse**

coherent with the results gathered in section 4.3.2.1.2. The results are summarized in Table 4.3 and Table 4.4, for the 1800 mAh battery.

3.3 V supply - 1800 mAh battery run-time estimation - Stationary state (C/10, 5.51 Wh)

Cores	Mode	Current (A)			Power (W)			Time (h)		
		Mean	RMS	Peak	Mean	RMS	Peak	Mean	RMS	Peak
1	Powersave	0.147	0.147	0.156	0.484	0.484	0.514	11.38	11.38	10.72
	Ondemand	0.147	0.147	0.155	0.485	0.485	0.511	11.36	11.36	10.77
	Performance	0.188	0.188	0.213	0.620	0.620	0.702	8.88	8.89	7.85
	Conservative	0.146	0.146	0.155	0.482	0.482	0.510	11.43	11.43	10.80
4	Powersave	0.145	0.145	0.149	0.477	0.477	0.491	11.54	11.54	11.23
	Ondemand	0.148	0.147	0.168	0.487	0.486	0.554	11.32	11.34	9.94
	Performance	0.186	0.186	0.197	0.612	0.612	0.651	9.00	9.00	8.46
	Conservative	0.146	0.145	0.150	0.481	0.479	0.495	11.46	11.50	11.13

Table 4.3 – *1800 mAh battery run-time estimation in stationary state, with a dimensional abuse C/10*

As seen in the battery analysis performed, the capacity can vary depending on the dimensional abuse used to charge it; in this case, the battery exhibits a stored energy of 5.51 Wh. Recalling the results in section 4.3.2.1.2, the CPU current drain in stationary state can be approximated by C/10 for this battery. With these considerations, the battery-runtime is estimated taking three values: mean, RMS and Peak. The results are really satisfactory as, for the maximum performance configuration, the battery is estimated to supply enough power for more than 8 hours. Table 4.4 summarizes the results of this test under a high workload.

3.3 V supply - 1800 mAh battery run-time estimation - High load (C/5, 5.25 Wh)

Cores	Mode	Current (A)			Power (W)			Time (h)		
		Mean	RMS	Peak	Mean	RMS	Peak	Mean	RMS	Peak
1	Powersave	0.294	0.296	0.370	0.971	0.976	1.221	5.41	5.38	4.30
	Ondemand	0.461	0.459	0.568	1.521	1.515	1.874	3.45	3.47	2.80
	Performance	0.461	0.465	0.567	1.523	1.533	1.872	3.45	3.42	2.80
	Conservative	0.456	0.459	0.563	1.504	1.515	1.858	3.49	3.47	2.83
4	Powersave	0.524	0.524	0.598	1.729	1.731	1.974	3.04	3.03	2.66
	Ondemand	1.016	0.999	1.082	3.353	3.298	3.569	1.57	1.59	1.47
	Performance	0.986	0.978	1.042	3.254	3.229	3.437	1.61	1.63	1.53
	Conservative	0.979	0.981	1.055	3.231	3.238	3.483	1.63	1.62	1.51

Table 4.4 – *1800 mAh battery run-time estimation under high load stress, with a dimensional abuse C/5*

When the CPU is dealing with a high load stress, the battery run-time falls drastically, as expected. However, it reaches an hour and a half, which is a great value taking into account that this workload will not be usual.

4

Chapter 5

Validation

This fifth chapter is used to validate the solutions proposed in this Master's Thesis and check the accomplishment of the goals of the project. This verification is performed using two resources: on the one hand, different photos of the developed products, which complement the ones introduced in each section of the Chapter 4; on the other hand, the Ground Station software, COSMOS, is used to verify the functioning of the different subsystems, but simulating several operations using the I2DOS and the GranaSat-I CubeSat prototype.

5.1 Final product

In this section, different photos of the design simulation platform are shown. For example, Figure 5.1 shows the I2DOS with the developed 1U CubeSat, while using the LED simulator.

The CubeSat is powered with an external power bank, which complies with the the EPS requirement **EPS.FoR.4**. The solar panel test board has been kept in order to check the behaviour of the platform when the **mass distribution** is different to the expected; it is able to deal with the imbalance correctly. Figure 5.2 shows a detail of the CubeSat electronics.

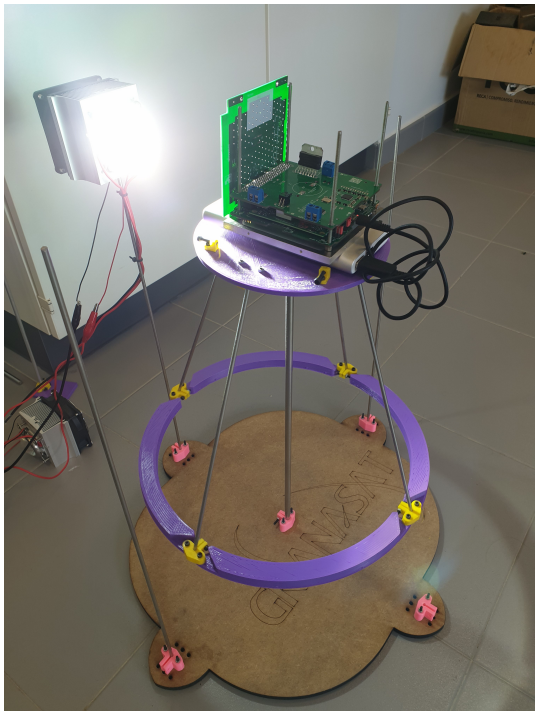


Figure 5.1 – *I2DOS and GranaSat-I prototype*

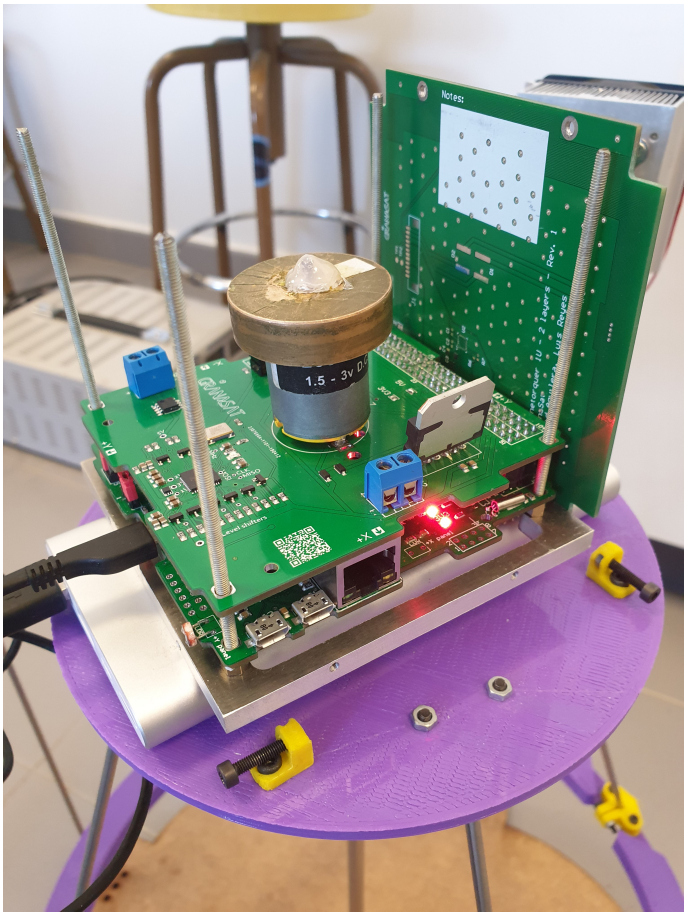
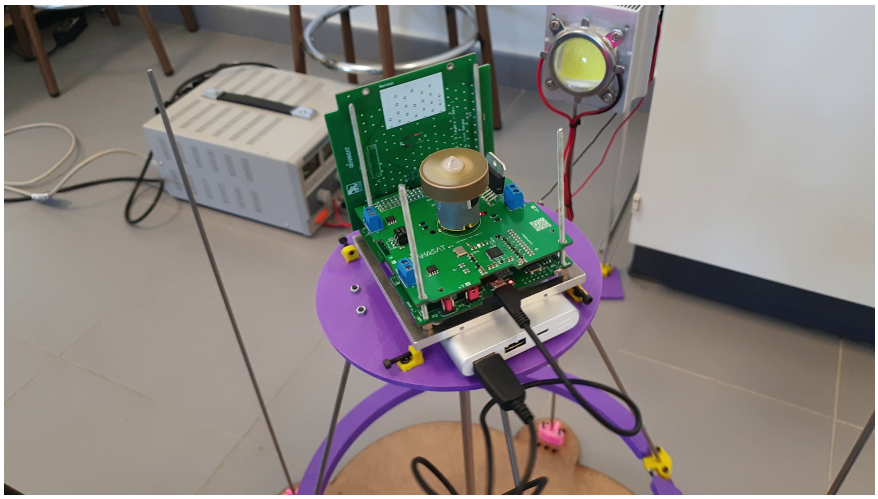


Figure 5.2 – *GranaSat-I CubeSat detail*

Video 5.1 shows a slow motion clip (Adobe Reader or another compatible PDF reader needed) of the platform moving.



Video 5.1 – *I2DOS platform in movement (Adobe Reader needed)*

5.2 OBC

As for OBC, once verified electrical connectivity and supply this section validates the functioning of the sensors performing different test and checking the results with COSMOS.

5.2.1 Sensors

As validation for the OBC sensors, two screenshots of the Ground Station software are included. Figure 5.4 illustrates the state of the sensors in a static position, as inferable from the static illumination received (see ADC_Measurements plot); the previous oscillations imply facing the *Sun* repeatedly when rotating. Also, pressure is stable so there is no variation in height and it confirms that the CubeSat **is stopped**. The plot **OBC_Main_Temperatures** shows the temperatures measured by two different sensors: the Bosch BMP280 and the internal sensor of the CPU. It helps to see the **gradient** of temperature across the OBC.

On the other hand, Figure 5.5 shows the result sent by the sensors when the CubeSat has gone through a complete **start-and-stop** cycle, i.e., it has started tumbling and the ADCS has counteracted. Therefore, there is appreciable variations in pressure (height) and in illumination (the Sun sensors have faced it multiple times and finally has left in eclipse position). Also, because of the higher workload and the operation of the actuators the overall temperature of the system have increased (especially in the CPU, with a gradient of 10 °C).

5.2.2 Communications

The whole telemetry data for this validation testbench is being sent over the air, using the wireless communication system of the CubeSat. The OBC sends several parameters regarding this subsystem, and they are plotted in one of the section designed in COSMOS, see Figure 5.6. Particularly, this data corresponds to the same start-and-stop cycle mentioned before; it is specially interesting the **signal level**, which varies according to the position of the antenna with respect to the receiver.

Although this validation is being performed using wireless communications, the **Ethernet** interface has been also tested, with successful results. In fact, the wired connection is necessary to configure the system the first time. In conjunction with the Ground Station management software, COSMOS, the communications subsystem is able to interact in a swarm of CubeSats, which allows the use of this platform in environments such as classrooms or training sessions. Figure 5.3 depicts this possible utilization.

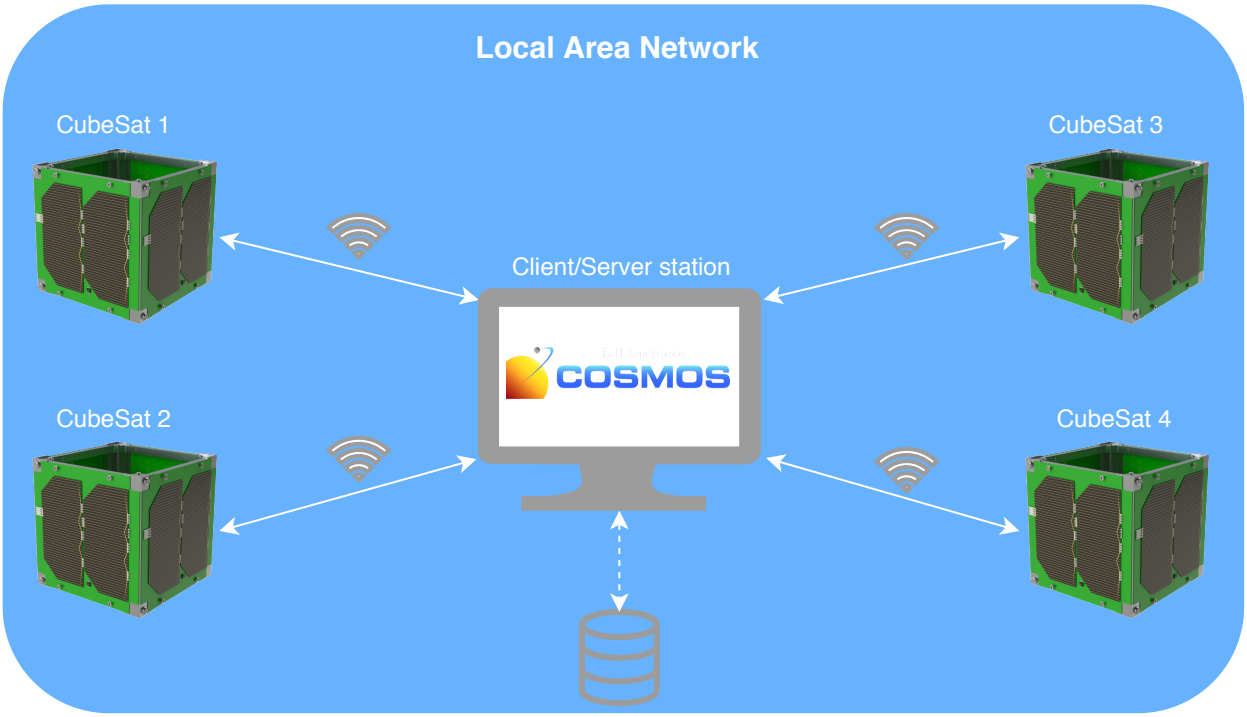


Figure 5.3 – *Communication subsystem allows being used in a LAN with multiple CubeSats*

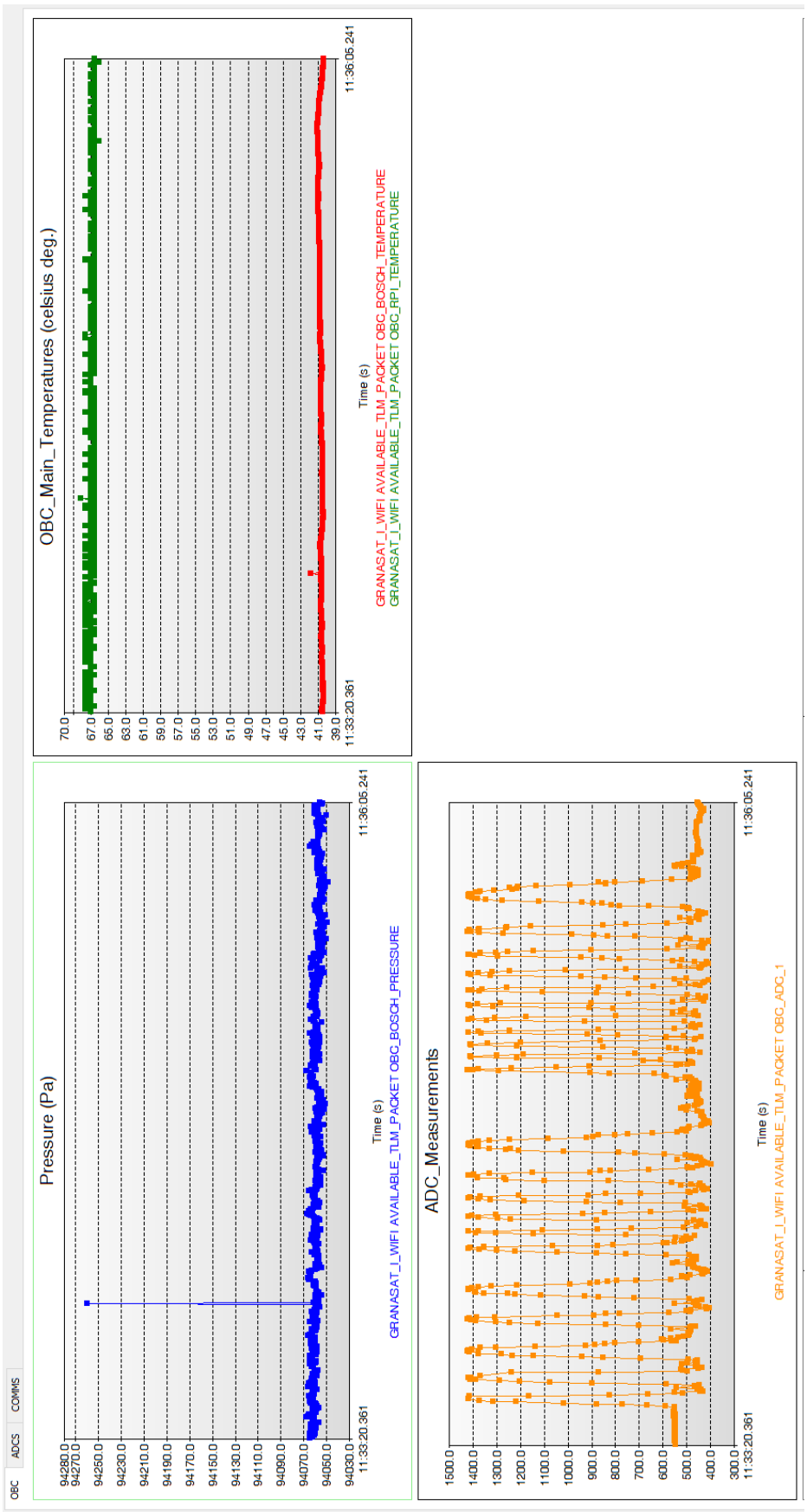


Figure 5.4 – *OBC measurements with static CubeSat*

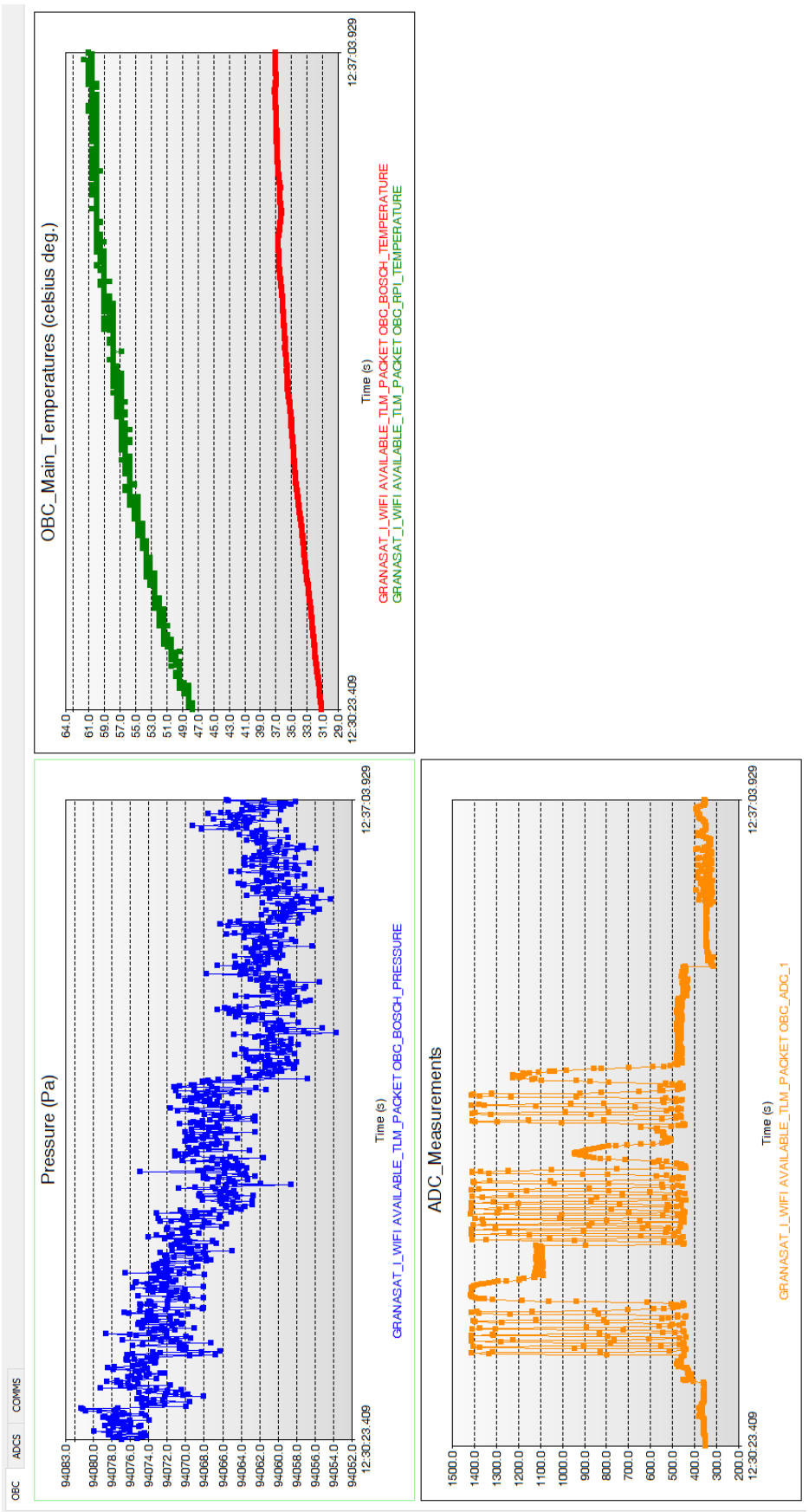


Figure 5.5 – *OBC measurements with static CubeSat*

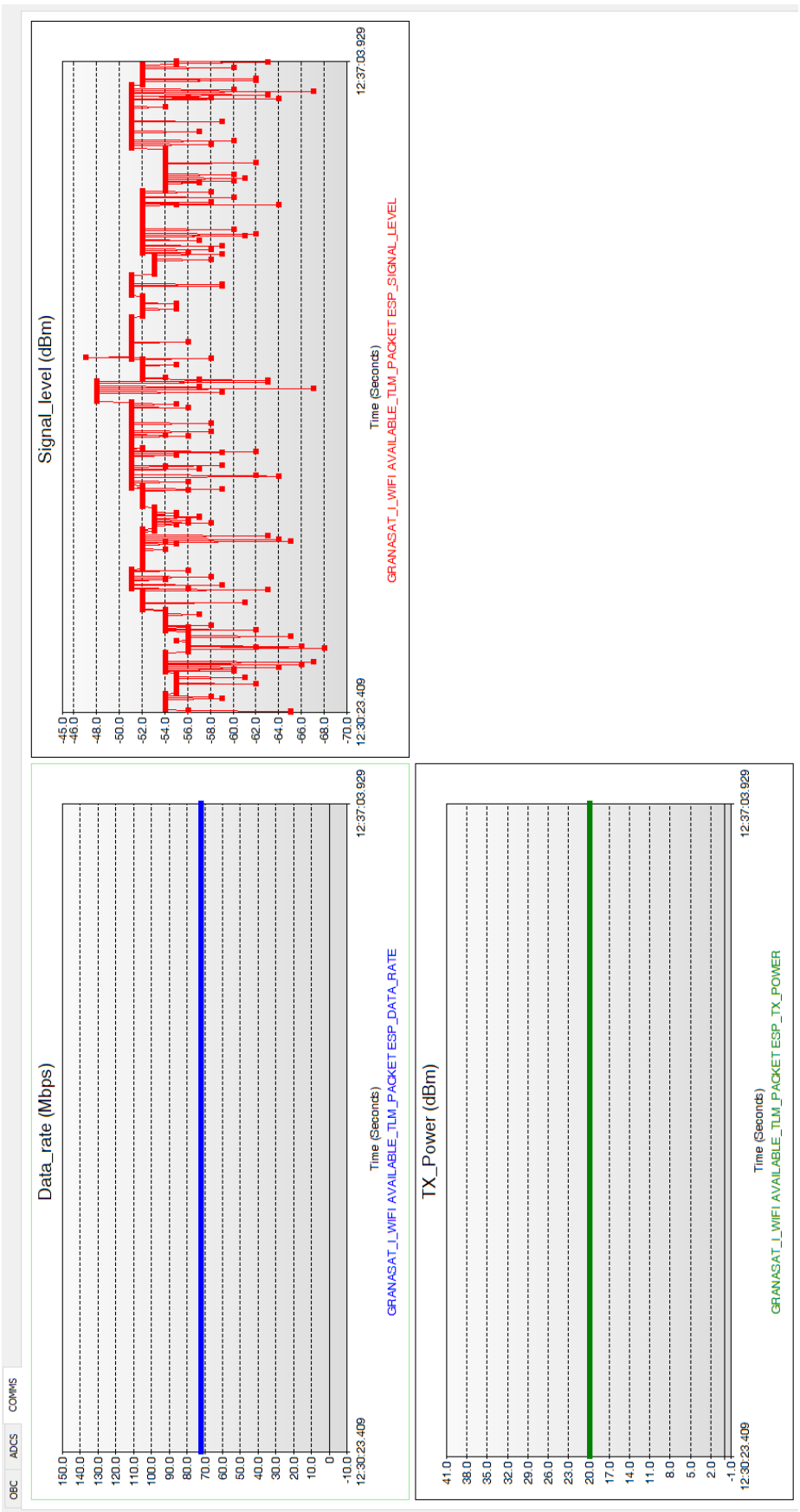


Figure 5.6 – *COMMS subsystem telemetry during rotation*

5.3 ADCS

This section addresses the validation of the ADCS subsystem. Firstly, the magnetometers are characterized using the **Helmholtz Cage** available at GranaSAT laboratory. Then, just as with the OBC, the correct functioning of the attitude-related sensors is verified with COSMOS.

5.3.1 Magnetometer characterization

Figure 5.7 shows the testbench used to perform this characterization, with the GranaSat-I in the center of the cage.

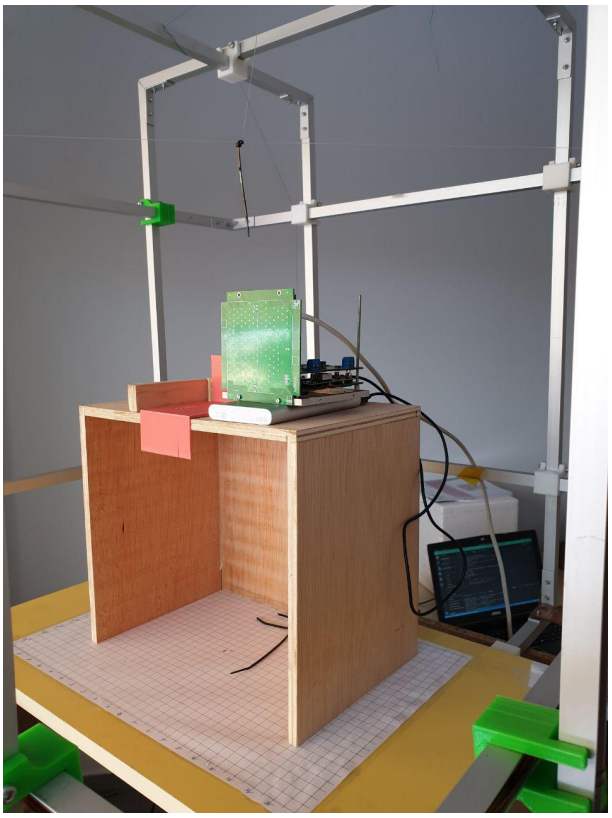


Figure 5.7 – *Magnetometer characterization testbench*

It is performed a simple characterization test, to assure the correct functioning of the sensor. In order to get the CubeSat ready for a real mission, it is necessary to go through the procedure described in section 3.1.1.4.2, standardized in [77]. Figure 5.8 shows the results; because timing is not important in this test, X-axis is shown in samples.

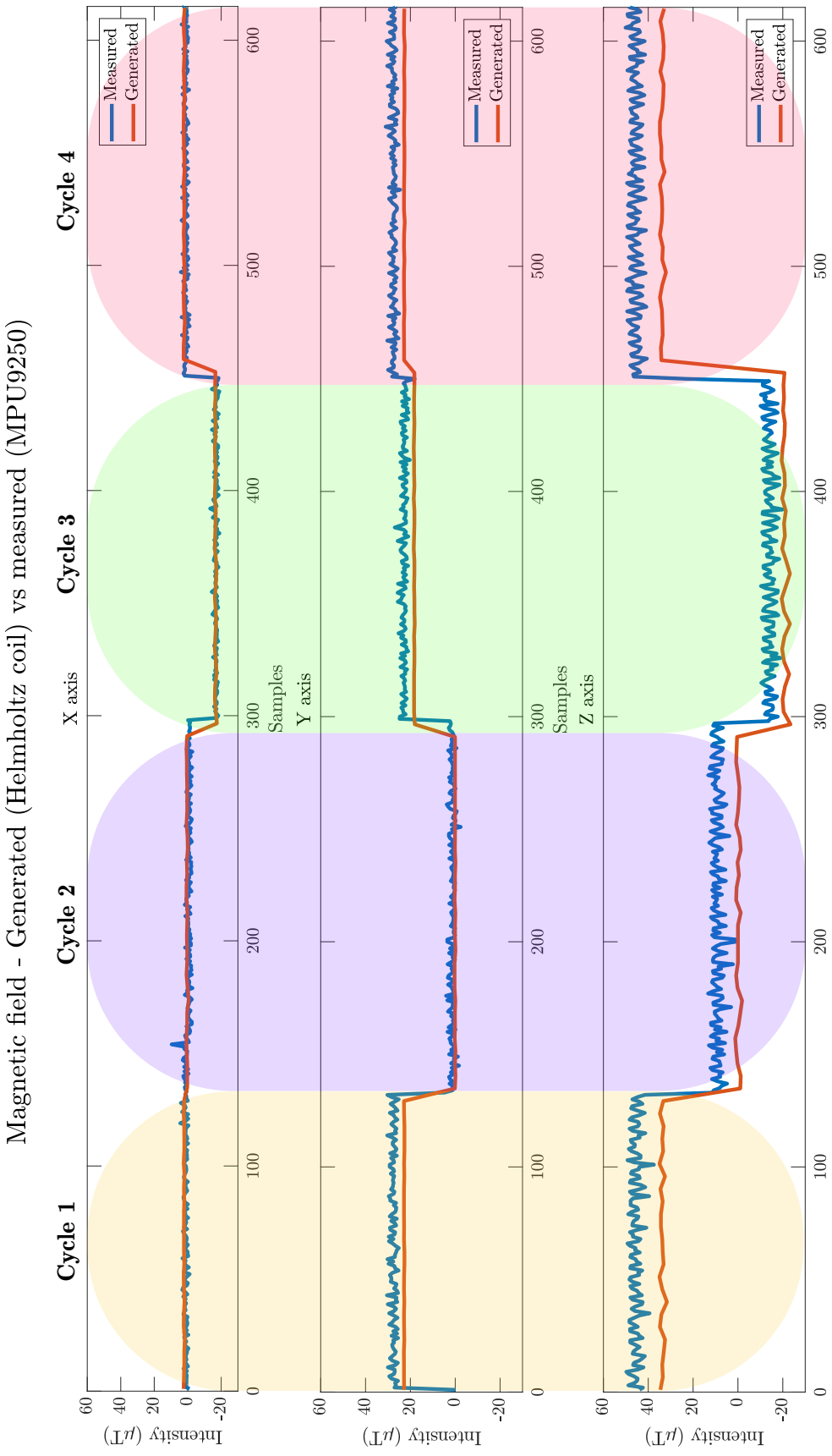


Figure 5.8 – *Magnetic field measurement test*

The procedure to characterize the magnetometer is composed of 4 cycles, clearly differentiated in Figure 5.8 with colours, described next.

- The first cycle, in orange, is a measurement of the **Earth's magnetic field**, i.e., the Helmholtz Cage is not generating any field, but measuring the field of the Earth. The coincidence between both measurements is already a good sign in itself.
- The second cycle, in purple, depicts the **cancellation of the Earth's magnetic field**. Once again, the measurements taken by MPU-9250 magnetometers are pretty accurate, and it is able to detect a null field in the three axes.
- The third cycle, in green, shows the measurements of a certain magnetic field generated by the cage. Particularly, with a magnitude of 35 µT, which is the module of the components measured $[x_{\text{field}}, y_{\text{field}}, z_{\text{field}}] = [-10\,\mu\text{T}, 27\,\mu\text{T}, -18\,\mu\text{T}]$.
- The fourth cycle, in red, measures Earth's magnetic field again.
- The test of the magnetometers can clearly be considered a success, taking into account the high degree of accuracy exhibited. However, it is worth noting the amount of **noise** present in the measurements, specially in the Z-axis. This spatial component is particularly affected by this phenomena probably because it is in the same plane of the DC-motor and the rest of the electronic components, which produce a non-negligible amount of noise in its plane. It would be recommendable to implement some **filtering** to improve the results.

5

5.3.2 Sensors

Once again, to validate the functioning of the sensors, they are included the results received and plotted in COSMOS during operation, with special interest in the maneuver of **detumbling**; Figure 5.9 shows this operation artificially repeated. It can be seen that the system has to deal with different rotational uncontrolled movements, in different senses of rotation; it is easy to see this situation in the ***Angular_Speed*** plot. In all cases, it is able to counteract that movement and get the system **to a stable position**.

Regarding the magnetic field, it also varies in the Z-axis (plane of **rotation**) and X-axis (because the platform is not completely **perpendicular** to the Z-axis) but remains approximately **constant in the Y-axis**, as the CubeSat is at a constant **elevation**. Finally, the **Accelerometer** measurements, as expected are approximately null in X and Y axis, and **1 g in the Z-axis** ($1\text{ g} = 9.8\,\text{m/s}^2$).

On the other hand, Figure 5.10 shows a maneuver inverting its sense of rotation. By looking at the *Angular_Speed* plot, the CubeSat is rotating counterclockwise, and the ADCS instead of stabilizing it, inverts its rotation; although this is not a realistic maneuver, it is useful to check the capabilities and functioning of the control system. The inversion is also easy to see in the *Magnetic_Field* plot.

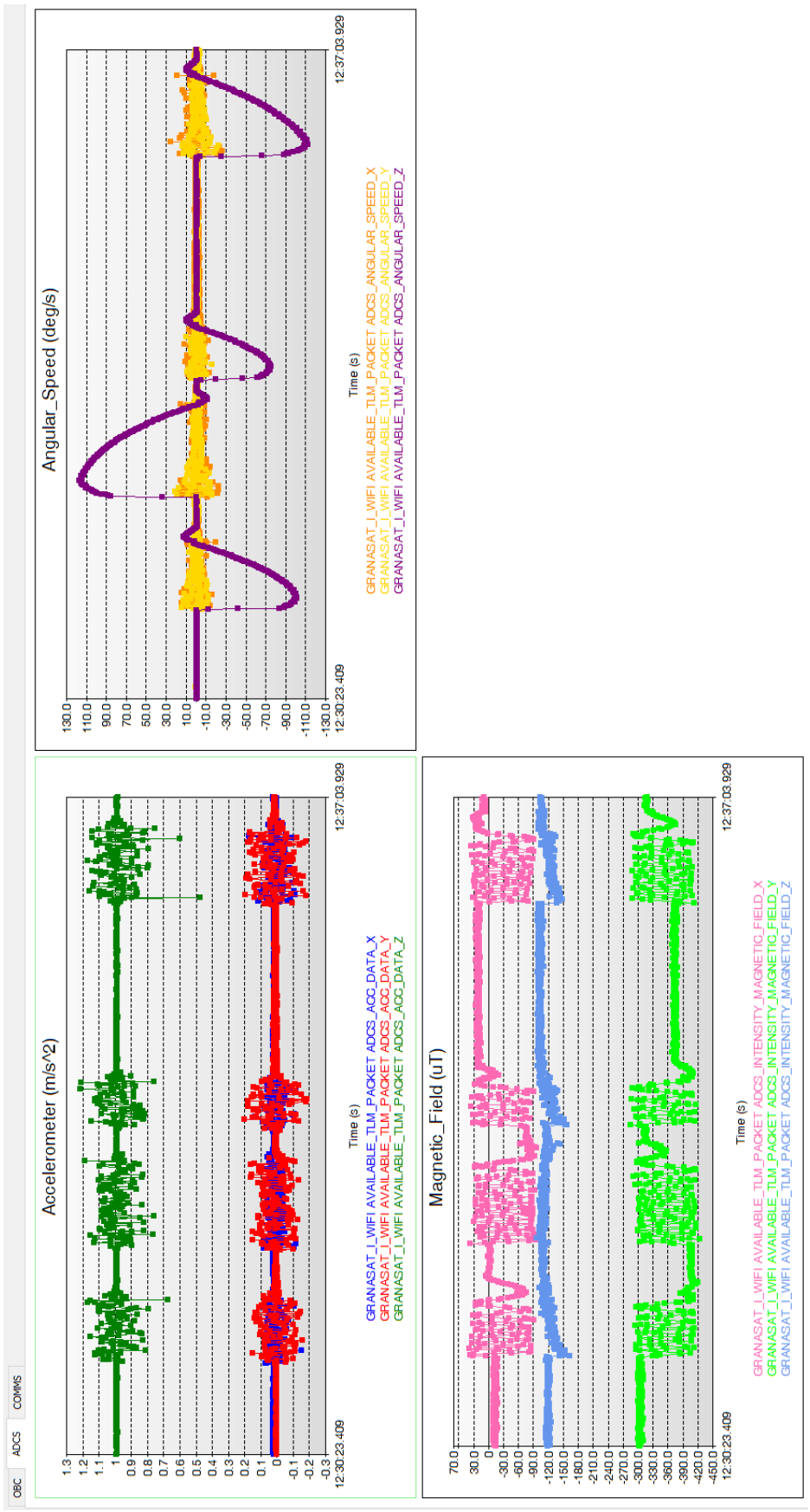


Figure 5.9 – *Detumbling operation*

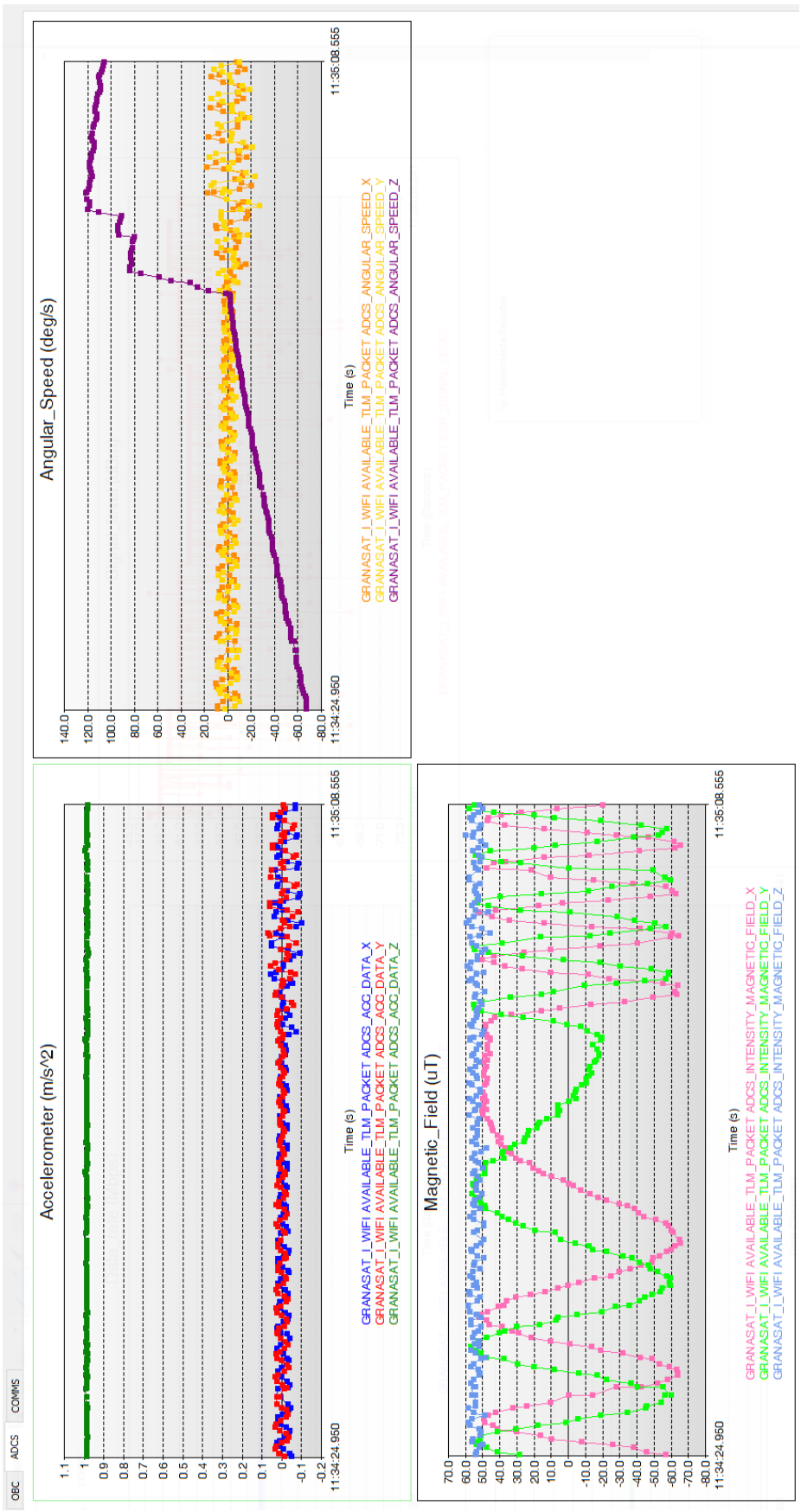


Figure 5.10 – *Rotation sense inversion*

Chapter 6

Conclusions and Future Lines

This document has shown the tough process under space-related projects by developing an innovative and functional CubeSat Simulation Platform. This Master's Thesis has been undertaken following several standards intensively used in space industry, using professional equipment and applying complex design techniques which go beyond the theory typically learnt during this tuition period.

This project provides a complete integrated environment in relation with CubeSats. As stated all along this document, two perspectives have been addressed: **mission-oriented** and **academic approach**. Particularly, the main contributions of this Master's Thesis are:

- The I2DOS Platform, which allows simulating a vacuum environment by using a low-friction rotating table, as well as exposing the CubeSat to external lightning (simulating the Sun) or magnetic fields (simulating Earth's field).
- An expandable Ground Station software, based on the Ball Aerospace COSMOS framework, previously used by NASA or **Lockheed Martin**; therefore, with this environment, the GranaSAT laboratory counts now with a professional mission management tool which can be used not only on the basis of a simulation, but also in a real mission.
- A simulation CubeSat prototype, which includes an aluminum mechanical structure, and counts with the main subsystems such as OBC, ADCS, local COMMS as well as a basic OBDH. It is ready to send telemetry information to COSMOS and allows defining new telemetry packages or telecommands. The ADCS includes a PID-based control law, which can be configured to detumble the device or follow a certain target. Additionally, this Master's Thesis includes an intensive characterization of different solar panels and batteries to be used with a CubeSat, which will ease design decisions in the future EPS.

Following the System Engineering methodology proposed by Wertz and Everett [94] has allowed minimizing risks and costs in the project and furthermore, accomplishing the goals defined.

This variety has supposed an increasing complexity which has given realism to this project, but also demanded an incredible effort. As for me, it has supposed an unprecedented challenge not only because of the inherent difficulty associated to this kind of project, but also because of its **extraordinarily wide scope**. Indeed, this project has made me learn things from a variety of fields of the engineering, which undoubtedly will be priceless in my professional future.

In a project of these characteristics, the end is never reached. Therefore, during the development of this Master's Thesis, some improvements and or future lines of work have naturally arisen, featuring the following ones:

- Diminishing the **moment of inertia** of the inertial platform by reducing its diameter. This would ease the rotation control and detumbling of the CubeSat and prevent the motor of the reaction wheel from saturation.
- Defining **new telemetry packages**, including all the information provided by the sensors, and implementing automatic *keep alives* or Housekeeping.
- Developing **new COSMOS interfaces**, which cover all the aspects of a mission and take full advantage of the capabilities of the GranaSAT-I, defining new telecommands and reports.
- Keeping with the development of an adequate EPS, based on the analysis and characterizations addressed in this project, which provides with power lines of 5 V and 3.3 V.
- Implementing printed Magnetorquers, which increases the **control capabilities** of the CubeSat. The designed ADCS already takes this possibility into account and includes terminals for three Magnetorquers.
- Defining new ADCS-related functions for the co-processing programmable cores (FPGA) which frees the OBC of tasks and enables a **time-critical** management when needed.
- Generating documentation regarding the academic orientation of the platform, with examples, case of use and guided procedures.

All in all, this Master's Thesis supposes a finalizing touch to my academic period but it does not end here; it is a live project which is now another part of GranaSAT, and I hope it is useful and brings success to its members henceforth.

Once again, the future is exciting.

References

[1] 5 W solar panel by RS. https://bit.ly/31wCIGb.

[2] Apalis tk1. https://www.toradex.com/computer-on-modules/apalis-arm-family/nvidia-tegra-k1.

[3] Apogee sp-110-ss. https://www.apogeeinstruments.com/sp-110-ss-self-powered-pyranometer/.

[4] C3.js graphic library. SB Admin Webpage. http://c3js.org/.

[5] California Polytechnic State University. https://www.calpoly.edu/.

[6] Colegio Oficial De Ingenieros Tecnicos De Telecomunicacion. https://telecos.zone/.

[7] Colibri vf50. https://www.toradex.com/computer-on-modules/colibri-arm-family/nxp-freescale-vybrid-vf5xx.

[8] CubeSat shop. https://www.cubesatshop.com/.

[9] D3.js. https://d3js.org/.

[10] Dietpi. https://dietpi.com/.

[11] Direct Industry. http://www.directindustry.com/prod/weiss-klimatechnik/product-13648-2029925.html.

[12] ECSS Document Tree. https://ecss.nl/ecss-architecture/ecss-document-tree-for-download/.

[13] Esa fly your satellite programme. https://www.esa.int/Education/CubeSats_-_Fly_Your_Satellite.

[14] Escuela Tecnica Superior de Ingenierias Informatica y de Telecomunicacion. UGR. https://etsiit.ugr.es/.

[15] Espressif. https://www.espressif.com/.

[16] Euclidean Space. http://www.euclideanspace.com.

[17] European Space Agency. https://www.esa.int/ESA.

[18] FPGA Embedded Processors. https://pdfs.semanticscholar.org/dcae/a58c93cab2bc068aba771b4e6705344a2942.pdf.

[19] GranaSAT website - Bexus experience. https://granasat.ugr.es/category/rexus/.

[20] HobbyKing. https://hobbyking.com.

[21] How a Solar Cell Works. https://www.acs.org/content/acs/en/education/resources/highschool/chemmatters/past-issues/archive-2013-2014/how-a-solar-cell-works.html.

[22] HP E3631A. https://www.keysight.com/main/techSupport.jspx?pid=836433&pageMode=FQ&lc=eng&cc=US.

[23] HP E6063B. https://www.keysight.com/en/pd-1000001520%3Aepsg%3Apro-pn-6063B/250-watt-dc-electronic-load?nid=-536902299.536880313&cc=ES&lc=spa.

[24] ISIS Space. https://www.isispace.nl/.

[25] Keysight. https://www.keysight.com/.

[26] Lattice ICE40UP5K. http://www.latticesemi.com/en/Products/FPGAandCPLD/iCE40UltraPlus.

[27] Linksys. https://www.linksys.com.

[28] Microchip ENC28J60. https://www.microchip.com/wwwproducts/en/en022889.

[29] Microchip lan9514. https://www.microchip.com/wwwproducts/en/LAN9514.

[30] MIT Open Courseware. https://ocw.mit.edu.

[31] MKS Sun simulators. https://www.newport.com/c/solar-simulators.

[32] Monocrystalline and Polycrystalline Solar Cells. https://www.greenmatch.co.uk/blog/2015/09/types-of-solar-panels.

[33] National Aeronautics and Space Administration (NASA). https://nasa.gov/.

[34] Node js. https://nodejs.org/es/.

[35] NXP PCF8523. https://www.nxp.com/products/analog/interfaces/ic-bus/ic-real-time-clocks-rtc/real-time-clock-rtc-and-calendar:PCF8523.

[36] On semiconductor fsusb42. https://www.onsemi.com/pub/Collateral/FSUSB42-D.PDF.

[37] Photoelectric Effect. https://www.khanacademy.org/science/physics/quantum-physics/photons/a/photoelectric-effect.

[38] PicoSoC. https://github.com/cliffordwolf/picorv32/tree/master/picosoc.

[39] PREEMPT_RT patch. https://mirrors.edge.kernel.org/pub/linux/kernel/projects/rt/.

[40] PV Lighthouse. https://www2.pvlighthouse.com.au/resources/courses/altermatt/The%20Solar%20Spectrum/The%20air%20mass%20(AM).aspx.

[41] Raspberry pi compute module 3. https://www.raspberrypi.org/products/compute-module-3/.

[42] ROHM BD6211F-E2. http://rohmfs.rohm.com/en/products/databook/datasheet/ic/motor/dc/bd621x-e.pdf.

[43] Saturn PCB Design Toolkit. http://www.saturnpcb.com/pcb_toolkit/.

[44] Sentiance. IMU on mobile phones. https://www.sentiance.com.

[45] Setup Linux with PREEMPTRT . https://wiki.linuxfoundation.org/realtime/documentation/howto/applications/preemptrt_setup.

[46] SIGLENT SDM3065X. https://www.siglent.eu/siglent-sdm3065x.html.

[47] Siglent spd3303x. Siglent Website. https://www.siglent.eu/spd3303x-e-power-supply.html.

[48] Simple_pid library. https://pypi.org/project/simple-pid/.

[49] Sparkfun. https://www.sparkfun.com/.

[50] ST L298N. https://www.st.com/en/motor-drivers/l298.html.

[51] Sun Power UK. https://www.sunpower-uk.com.

[52] Thorlabs. https://www.thorlabs.com/.

[53] TI ADS1015. http://www.ti.com/lit/ds/symlink/ads1015.pdf.

[54] Ti msp430. http://www.ti.com/microcontrollers/msp430-ultra-low-power-mcus/overview.html.

[55] UPS Battery Center. http://www.upsbatterycenter.com/.

[56] Vishay CNY70. https://www.vishay.com/docs/83751/cny70.pdf.

[57] VxWorks by Wind River. https://www.windriver.com/products/vxworks/.

[58] Webgl. https://get.webgl.org/.

[59] Wikipedia. https://en.wikipedia.org.

[60] *Achieving Science with CubeSats*. National Academies of Sciences, Engineering, and Medicine, 2016.

[61] Abhay Mohan M.V, Jishnu Pavithran, e. a. LED Based Solar Simulator. http://www.standardsuniversity.org/wp-content/uploads/led_based_solar_simulator_mohan_final_paper.pdf.

[62] Aerospace, B. Cosmos. https://cosmosrb.com/.

[63] Antonio Luque, S. H. *Handbook of Photovoltaic Science and Engineering*. John Wiley and Sons, 2011.

[64] AzurSpace. Triple-Junction GaAs Solar Cells datasheet. http://www.azurspace.com/images/0003429-01-01_DB_3G30C-Advanced.pdf.

[65] Bogatin, E. *Signal and Power Integrity*. Prentice Hall, 2010.

[66] Bosch. BMP280. https://www.bosch-sensortec.com/bst/products/all_products/bmp280.

[67] Brian Gasberg Thomsen, J. N. Cubesat sliding mode attitude control. Master's thesis, Aalborg University, 2014.

[68] CCSDS. Packet Telemetry. https://public.ccsds.org/Pubs/102x0b5s.pdf.

[69] CCSDS. Space Packet Protocol. https://public.ccsds.org/Pubs/133x0b1c2.pdf.

[70] CCSDS. Standard Formatted Data Units. https://public.ccsds.org/Pubs/621x0g1.pdf.

[71] Durillo, J. C. M. DATV Receiver System Design and Development. Master's thesis, University of Granada, Granada, September 2017.

[72] ESA. A Concept for a Radioisotope Powered Lunar CubeSat. https://www.hou.usra.edu/meetings/leag2016/pdf/5081.pdf.

[73] ESA. ECSS-A Single Set of European Space Standards. https://ecss.nl/home/ecss-a-single-set-of-european-space-standards/.

[74] ESA. ECSS-E-HB-31-03A - Thermal analysis handbook. https://ecss.nl/home/ecss-e-hb-31-03a-15november2016/.

[75] ESA. ECSS-E-ST-10-03C - Testing. https://ecss.nl/standard/ecss-e-st-10-03c-testing/.

[76] ESA. ECSS-E-ST-10C - System engineering general requirements. https://ecss.nl/standard/ecss-e-st-10c-rev-1-system-engineering-general-requirements-15-february-2017/.

[77] ESA. ECSS-E-ST-20-07C Rev.1 - Electromagnetic compatibility. https://ecss.nl/standard/ecss-e-st-20-07c-rev-1-electromagnetic-compatibility/.

[78] ESA. ECSS-E-ST-60-30C - Space engineering Satellite attitude and orbit control system (AOCS) requirements. https://ecss.nl/standard/ecss-e-st-60-30c-satellite-attitude-and-orbit-control-system-aocs-requirements/.

[79] ESA. ECSS-E-ST-70-41C - Telemetry and telecommand packet utilization . https://ecss.nl/standard/ecss-e-st-70-41c-space-engineering-telemetry-and-telecommand-packet-utilization-

[80] ESA. ECSS-Q-ST-70-01C - Cleanliness and contamination control. https://ecss.nl/standard/ecss-q-st-70-01c-cleanliness-and-contamination-control/.

[81] ESA. ECSS-Q-ST-70-04C - Thermal testing for the evaluation of space materials, processes, mechanical parts and assemblies. https://ecss.nl/standard/ecss-q-st-70-04c-thermal-testing-for-the-evaluation-of-space-materials-processes

[82] ESA. European Cooperation for Space Standardization. https://ecss.nl/.

[83] ET AL., J. L. S. Historical Review of Air-Bearing Spacecraft Simulators.

[84] ET AL., R. F. H. Dynamic Simulation Platform Design for Cubesat with Reaction Wheel Control System.

[85] EXTREMERA, R. L. Simulador de campo magnetico terrestre. Master thesis, University of Granada, Granada, September 2017.

[86] FAROOQ, S. Aligarh Muslim University. http://www.geol-amu.org/notes/m1r-1-6.htm.

[87] GONZALEZ, V. B. Testbed for a 1u cubesat. Bachelor thesis, University of Granada, Granada, September 2016.

[88] GRAN, A. S. R. F. A. F. M. Investigation of the Acquisition Problem in Satellite Attitude Control. http://contrails.iit.edu/files/original/AFFDLTR65-115.pdf.

[89] Group, O. M. XTCE Standard. https://www.omg.org/xtce/index.htm.

[90] Hamadani, B. H., and Dougherty, B. Solar cell characterization. *National Institute of Standards and Technology, Springer, New York)* (2015).

[91] IEEE. IEEE802.11 Standard. https://ieeexplore.ieee.org/document/7786995.

[92] IEEE. IEEE802.3 Standard. https://ieeexplore.ieee.org/document/8457469.

[93] Invensense. MPU9250. https://www.invensense.com/products/motion-tracking/9-axis/mpu-9250/.

[94] James R. Wertz, David F. Everett, J. J. P. *Space Mission Engineering:The New SMAD*. Space Technology Library, 2018.

[95] Laboratory, N. R. E. Best Research-Cell Efficiencies. April 2019. https://www.nrel.gov/pv/assets/pdfs/best-research-cell-efficiencies-190416.pdf.

[96] Leary, G. P. Comparison of xenon lamp-based and led-based Solar simulators. https://scholarworks.montana.edu/xmlui/bitstream/handle/1/9837/LearyG0516.pdf?sequence=1&isAllowed=y.

[97] Li, X. Science results from colorado student space weather experiment (csswe): Energetic particle distribution in near earth environment.

[98] Lin, M. Caracterizacion e implementacion de un simulador solar. https://upcommons.upc.edu/bitstream/handle/2099.1/16083/memoria.pdf?sequence=4&isAllowed=y.

[99] Mcevoy, A. *Practical Handbook of Photovoltaics : Fundamentals and Applications*. Elsevier Science & Technology, 2011.

[100] Miotla, D. Assessment of Plutonium-238 Production Alternatives. https://www.energy.gov/sites/prod/files/NEGTN0NEAC_PU-238_042108.pdf.

[101] of Bristol, U. PocketRTG.

[102] of Colorado, U. Satellite Testbed for Attitude Response. https://www.colorado.edu/aerospace/sites/default/files/attached-files/star_aiaa.pdf.

[103] Patel, M. R. *Spacecraft Power Systems*. CRC Press, 2005.

[104] Patrick Stakem, Jose Carlos Martinez, Vishnu Chandrasenan et al. A Cubesat Swarm Approach for the Exploration of the Asteroid Belt. *The Planetary CubeSats Symposium* (2018).

[105] Sanchez, P. G. Diseno e implementacion de un limitador electroacustico mediante procesadores de audio de altas prestaciones. PhD. Thesis, University of Granada, Granada, September 2019.

[106] the corner, A. Gimbal lock illustration. https://around-the-corner.typepad.com/.

[107] Torralba, J. M. L. Characterization equipment of aerospace photovoltaic panels. M.s thesis (dissertation), University of Granada, Granada, September 2018. This M.S Thesis is a contribution to the GranaSAT project.

[108] USB.org. USB 2.0 Specification. https://www.usb.org/document-library/usb-20-specification.

[109] Wertz, J. R. *Spacecraft Attitude Determination and Control*. Springer Science, 1978.

Appendix A

Project Budget

A.1 Materials and hardware

In this section, project cost regarding materials and hardware sections will be detailed. Each one of the different hardware subsections is differentiated. Human resources area also included.

A.1.1 Inertial 2D Orbit Simulator

According to the System Design, Chapter 4, the materials related with the I2DOS are mainly plywood for the base and PLA for the 3D printed inertial platform. The plywood board has a cost of **11.30 €**, while the amount necessary of PLA costs **4.50 €**. That makes **15.80 €** without manufacturing costs, detailed in Section A.2.

A.1.2 Simulation CubeSat

A.1.2.1 Mechanical structure

The mechanical structure of the CubeSat is made of aluminum EN AW 5754, also sold in boards. Because of the extremely light weight of a CubeSat, the attributable cost for the manufacturing of the GranaSAT-I is only **2.50 €**.

A.1.2.2 On-board Computer

The cost of the OBC is breakdown into the electronic components and the manufacturing, as follows.

A.1.2.2.1 Bill of Materials

Item	Units	Cost/u. (€)	Cost (€)
LDR	6	0.15	0.9
Bosch BMP280	1	2.68	2.68
MPU-9250	1	5.1	5.1
CR2032 Battery	1	0.35	0.35
Electrolitic cap.	4	0.1	0.4
Tantalum cap	2	0.8	1.6
Ceramic cap.	36	0.012	0.432
LED SMD	10	0.05	0.5
1N4148W	1	0.05	0.05
ENC28J60 breakout	1	3	3
ESP8266	1	2	2
Micro USB 2.0 B	3	3.5	10.5
J00-0045L	1	4.5	4.5
ESQ-126-39-G-D	2	7	14
Canned Oscillator 25 MHz	1	3.5	3.5
Transistors 2N7002P	5	0.3	1.5
Resistors	51	0.01	0.51
Raspberry Pi 3 CM	1	32	32
PCF8523	1	1.13	1.13
ADS1015	2	2.84	5.68
LAN9514	1	5.13	5.13
FSUSB42	1	0.5	0.5
Subtotal (before VAT):			**79.31**
Total (VAT included):			**95.96**

Table A.1 – *OBC PCB components*

A.1.2.2.2 Manufacturing

The manufacturing of a 4-layers PCB like the OBC, including Stencil has a cost of **43.40 €**.

A.1.2.3 Attitude Determination and Control System

A.1.2.3.1 Bill of Materials

Item	Units	Cost/u. (€)	Cost (€)
BD6211F	2	1.9	3.8
L298N	1	5.2	5.2
DC motor	1	7	7
ICE40UP5K	1	7.1	7.1
1N4148W	4	0.05	0.2
Ceramic cap.	16	0.012	0.192
CNY70	1	1.8	1.8
ESQ-126-39-G-D	2	7	14
Canned Oscillator 25 MHz	1	3.5	3.5
Transistors 2N7002P	8	0.3	2.4
Resistors	25	0.01	0.25
TLV7111225	1	0.5	0.5
Subtotal (before VAT):			**37.97**
Total (VAT included):			**45.94**

Table A.2 – *ADCS PCB components*

A.1.2.3.2 Manufacturing

The manufacturing of a 2-layers PCB like the ADCS, including Stencil has a cost of **18.15 €**.

A.1.2.4 Electrical Power system

A.1.2.4.1 Bill of Materials

Item	Units	Cost/u. (€)	Cost (€)
Li-ion battery (1100 mAh)	1	8.60	8.60
Li-ion battery (1800 mAh)	1	11.11	11.11
Solar panel 1 (5 V @ 1.3 W)	1	3.44	3.44
Solar panel 2 (12 V @ 1.5 W)	1	2.83	2.83
Solar panel 3 (5.5 V @ 1 W)	1	2.48	2.48
Subtotal (before VAT):			**23.52**
Total (VAT included):			**28.46**

Table A.3 – *EPS components*

A.2 Human Resources Cost

In the first place, the manufacturing of the base of the I2DOS required the collaboration of the **makerspace** at the Architecture School of the University of Granada. The cost for an hour of machinery and technical assistance was **15 €**.

Additionally, the development of this Master's Thesis has required hiring two people. The first one is a **junior engineer** (10 €/h), hired as a full-time worker during twelve months. Secondly, as Project Supervisor a **senior engineer** is hired (50 €/h), computing 5 hours per week. Then, Human Resources amounts to **31200 €**, as detailed in table A.4.

Post	Time (Hours)	Cost (€)
Junior Engineer	1920	19200
Senior Engineer	240	12000
	TOTAL	**31200 €**

Table A.4 – *Human Resources Cost*