

POLITECNICO DI TORINO

Master's Degree in Aerospace Engineering



Master's Degree Thesis

Development and validation of a test bench for Attitude Determination and Control System for small satellite

Supervisors

Prof. Fabrizio STESINA

Prof. Dario MODENINI

Prof. Sabrina CORPINO

Candidate

Chiara LUGHI

March 2023

Summary

In the last decades, nanosatellites have made space accessible to a wide public thanks to their low cost and fast development time, compared to traditional satellites, profoundly changing both the market and the space industry. Since their inception, they have played an important role in the so-called new space economy. In particular, CubeSats nanosatellites, thanks to their standardization, evolved from being simple educational tools to being used in multiple space applications. Thanks to their use in constellations with increasing success, CubeSats offer a valid alternative to traditional spacecraft in areas of great scientific-application relevance.

One of the most complex and challenging subsystems in CubeSats is the Attitude Determination and Control System (ADCS). This subsystem employs complex components, involving highly integrated hardware and software. Its testing is particularly challenging since it requires a free-rotational motion in a low-torque environment of the satellite and a physical stimulus of the sensors.

This thesis fits into the context of Hardware-In-The-Loop (HIL) ADCS testing. The first objective consisted of the update of the subsystem test facility already implemented at the Microsatellites and Space Microsystems Laboratory of the University of Bologna. The upgrade consisted of an improvement in the facility performance in terms of minimization of the disturbance torques and the introduction of a mechanical interface that allows the hosting of nanosatellites compliant with the CubeSat standard, in the broader perspective of a nationwide distributed laboratory. The first part of the work has been therefore focused on the design and selection of the new components of the facility with the subsequent work of verification, integration and validation. Taking into account that for an air-bearing type facility, the biggest disturbance torque is the one due to gravity and the misalignment between the centre of rotation and the centre of mass. A minimisation of the torque is obtained by minimizing the distance between CM and CR. A balancing procedure employing three sliding masses, moved by three stepper motors, is used to minimize such torque. As a further work, an embedded PCB has been designed

to reduce the presence of moving cables that could affect the disturbance torque acting on the facility. In the second part of the thesis, the magnetic actuators have been dimensioned taking into consideration the residual disturbance torque of the facility. Moreover, a *single axis* control law was implemented in the ADCS board and its effectiveness has been verified through HIL tests. Further work involved the testing of proper torque generation by air-coil magnetic actuators. The purpose of the test is to verify that the control torque is suitable to set back the system from a tilted angle configuration to the equatorial one.

Acknowledgements

Special thanks go to my professors Fabrizio Stesina and Dario Modenini for providing me with the priceless opportunity to carry out an exciting and challenging thesis project, while giving me all the support I needed. Many thanks to Andrea, Francesco and Antonio for being invaluable colleagues, your support and endless hours working together were indispensable through the entire work.

A special thank to my family, who supported and motivated me every moment. Thanks to my lifelong friends and those who have become ones during this extraordinary journey, your affection, countless advice and endless encouragement have been of fundamental importance to me.

Special thanks go to Leo, your infinite affection and your smiles mean the world to me. Thank you for having taught me to see the world with eyes always full of enthusiasm.

“Per aspera ad astra”

Table of Contents

List of Tables	VIII
List of Figures	IX
Acronyms	XIV
1 Introduction	1
1.1 General contest	1
1.1.1 Cubesat Standard and Model	1
1.1.2 Introduction to ADCS	3
1.1.3 Test bench for ADCS state of the Art	4
1.2 Objectives	6
1.3 Outline	8
2 Preliminaries	9
2.1 Reference Frames	9
2.1.1 Spacecraft Body Frame	9
2.1.2 Inertial Reference Frames	10
2.1.3 Earth-Centered/Earth-Fixed Frame	10
2.1.4 Local-Vertical/Local-Horizontal Frame	12
2.2 Attitude Representation	13
2.2.1 Euler Axis/Angle Representation	13
2.2.2 Quaternion Representation	15
2.3 Attitude Dynamics and Kinematics	15
2.3.1 Attitude Kinematics	16
2.3.2 Attitude Dynamics	16
2.3.3 Disturbance Torques	17
2.3.4 Testbed Dynamics	20
3 Facility Description	22
3.1 Test bench description	22

3.1.1	Microsatellites and Space Microsystems Laboratory	22
3.1.2	Space environment reproduction components	24
3.1.3	Testing Components	28
3.2	Design of the interface system	35
3.2.1	CubeSat prototype	35
4	Integration, calibration, and verification	44
4.1	Verification	45
4.1.1	Stepper Motors and Motor Drivers	45
4.1.2	CubeSat boards verification	53
4.2	Integration	53
4.3	Validation	59
5	Balancing Procedure	62
5.1	Manual Balancing	62
5.2	Automatic balancing	63
5.3	Balancing Results	69
5.4	Electrical Board PCB	76
6	Final tests	81
6.1	Magnetic Torquers	81
6.1.1	Introduction to Magnetic Actuation	81
6.1.2	Sizing	84
6.2	Magnetic Actuation Tests	88
6.2.1	Preliminary Set-up	88
6.2.2	ADCS software control software	93
6.3	Verification Program	97
6.3.1	Functional verification of ADCS board, base functional board and connected boards	97
6.3.2	Updated boards configuration for final magnetic tests	103
6.3.3	Magnetic Test	107
7	Conclusions	116
	Bibliography	119

List of Tables

3.1	On-orbit environment aspects that needs to be simulated	23
3.2	Size 28mm DLM characteristics	33
3.3	Distribution of masses of the satellite	36
3.4	Function board components and their functions	38
3.5	Distribution of masses between the components of the mass mock-up	42
4.1	X-sens specifics	57
4.2	Technical requirements specification related with the facility set-up	61
5.1	Gain adopted for x-y balancing	73
6.1	Air core magnetic torquer known characteristics	86
6.2	Sizing Results for Air Core Magnetic actuator	87
6.3	Disturbances estimation for 71°, 700 Km 1U CubeSat application .	87
6.4	Microprocessor ELPA RD129 technical characteristics	90
6.5	MTi specifics	91

List of Figures

1.1	Trend of CubeSat missions [3]	2
1.2	CubeSat standard [8], [9]	3
1.3	ADC functioning loop [10]	4
1.4	Example of full ADCS board for CubeSat applications [12]	5
1.5	Failure rate of CubeSat mission over the year [14]	5
1.6	Failure rate of CubeSat mission divided in subsystem failure [15]	5
1.7	CubeTAS facility [21]	7
1.8	LAICA facility [22]	7
1.9	Facility of Montpellier [23]	7
1.10	$\mu 3S$ Facility	7
2.1	ECI Reference Frame	10
2.2	ECEF reference frame	11
2.3	Local Vertical-Local Horizontal frame	12
2.4	Euler Rotation 3-1-3	14
3.1	Air bearing spherical setups: table-top (a) , umbrella (b), and dumbbell (c) [27]	23
3.2	Magnetic field generated from two co-axial coils	25
3.3	Solar Radiation	26
3.4	Sun simulator	26
3.5	Air Bearing system implemented in $\mu 3S$ facility consisted of an articulated stand (Ferronato® ULTAS-1 from Serviciencia, Toledo, Spain), a spherical air bearing (150mm of diameter) by Physical Instrumentns and tabletop-style platform	27
3.6	Interface plate with electronical components and stepper motors on x,y axes mounted	30
3.7	Single winding functioning for a variable reluctance stepper motor	31
3.8	Bipolar functioning scheme for an External Linear stepper motor	32
3.9	DINGS' Stepper motor implemented in the facility	32
3.10	Blocking with interference solution	33

3.11	Blocking with L-shaped solution	34
3.12	Mass distribution on the satellite	35
3.13	High level functions of the functional board	37
3.14	Functional Board Block Diagram	39
3.15	Functional Board	39
3.16	Initial structural solution for the host CubeSat	40
3.17	Isometric view	41
3.18	Top view	41
3.19	External case	41
3.20	Internal structure	41
3.21	Mass distribution for the mass mock-up	42
3.22	Final mass mock-up	43
3.23	Satellite components of the facility	43
4.1	System Engineering V-model [30]	45
4.2	H-bridge microcontroller scheme	46
4.3	H-bridge functioning sequence	47
4.4	Arduino Uno set up for stepper motors verification	48
4.5	Voltage sequence for DINGS' stepper motors	49
4.6	On the right is the voltage sequence proposed by the MD 2448, on the left that of the MD 8833	50
4.7	Current waves while increasing micro-stepping	50
4.8	PWM Control of Motor Speed	51
4.9	Current paths in different drive and decay modes	51
4.10	Curves representing the change in vibration intensity as the imposed speed and acceleration vary	52
4.11	Electronic scheme	54
4.12	Facility CAD Solidworks global model	58
4.13	Upper view	59
4.14	Bottom view	59
4.15	Lateral view	59
4.16	Frontal view	59
4.17	Main functional tree	60
4.18	Space environment ramification	60
4.19	Set up ramification	61
5.1	Inertial and Body frame representation	64
5.2	Stable Pendulum condition	68
5.3	Inverted pendulum behaviour	69
5.4	Representation of the gravitational vector variation	70
5.5	Representation of the gravitational vector variation	71

5.6	$R_{CR,z}$ offset estimation	71
5.7	Angular velocities and acceleration proceeds from IMU and accelerometers	72
5.8	Shifting masses positions	73
5.9	Oscillations of x,y masses	73
5.10	Non filtered	74
5.11	Filtered	74
5.12	Gravitational vector reduction during time	74
5.13	Torque estimation results	75
5.14	Torque estimation filtered data	75
5.15	Schematic of Motor Driver and Voltage Regulator for the fully integrated board	77
5.16	Schematic of battery pack, Feather M0 and Arduino Protoshield for the fully integrated board	77
5.17	Fully integrated PCB	78
5.18	Printed board	78
5.19	Schematic of Motor Driver and Voltage Regulator for the pin board	79
5.20	Schematic of battery pack, Feather M0 and Arduino Protoshield for the pin board	79
5.21	Functional Board with pin integration PCB	80
5.22	Functional Board with pin integration printed board	80
6.1	MT principle of operation [38]	82
6.2	Air Core MT [39]	84
6.3	PCB MT [40]	84
6.4	Rods MT [39]	84
6.5	ADCS board components	89
6.6	Control System with PD Controller	91
6.7	HC reference frame	92
6.8	Body and Inertial RF	92
6.9	Global ADCS Simulink Model	95
6.10	Attitude model	95
6.11	Control model	96
6.12	Representation of the variation of quaternions during magnetic control	96
6.13	MT air-core magnetic torque actuation variation	97
6.14	Angular velocity variation during magnetic actuation	97
6.15	ADCS pinout	98
6.16	Raspberry pinout [42]	98
6.17	Verification test set up	98
6.18	Set up for RD129 verification test	100
6.19	First verification set-up for communication test	102

6.20	Second verification set-up for communication test	102
6.21	New power circuit	104
6.22	ADCS's IMU connections	104
6.23	Card system updated by applying corrections to bypass RD129 . . .	104
6.24	Serial data reception from IMU test set up	105
6.25	PWM signal output	106
6.26	Set up for Python code function test	106
6.27	Post transfer Functional test set up	108
6.28	High CM configuration	110
6.29	Low CM configuration	110
6.30	Residual Torque estimation	110
6.31	Magnetic Control Test set up	111
6.32	Modified Magnetic Control test set up	113
6.33	MT duty cycle	114
6.34	MT Magnetic dipole	114
6.35	g components	115
6.36	Angular velocities	115

Acronyms

ADCS Attitude Determination and Control System

ADC Attitude Determination and Control

HC Helmholtz Cage

AU Astronomical Unit

CM Center of Mass

CR Center of Rotation

ABS Automatic Balancing System

SC Spacecraft

SRP Solar Radiation Pressure

MSM Microsystem and Space Microsatellites

GSE Ground Support Equipment

TRR Test Readiness Review

AIT Assembly, Integration and Test

COST Commercial Off-The-Shelf

HIL Hardware-In-the-Loop

ECI Earth Centered Inertial frame

GCI GeoCentric Inertial frame

ECEF Earth-Centered/Earth-Fixed frame

GMST Greenwich Mean Sidereal Time

JD Julian Day

LVLH Local-Vertical/Local-Horizontal

LEO Low Earth Orbit

IGRF International Geomagnetic Reference Frame

IAGA International Association of Geomagnetism and Aeronomy

MT Magnetic Torquer

PD Proportional Derivative control law

IMU Inertial Measurement Unit

RF Reference Frame

PID Proportional Integrative Derivative

Chapter 1

Introduction

1.1 General contest

A CubeSat is a type of very small satellite which is based on a standardized unit of mass and volume. The initial basic CubeSat unit measured 10x10x10 centimetres, conforming to specific interfaces for allowing a standardized containerized launch and had a maximum mass of 1 kilogram (the mass was later on increased to 1,33 kilogram) [1]. It was quickly realized that such basic CubeSat units could be combined to form slightly larger spacecraft while mostly adhering to the same requirements and constraints. Multiples of the basic CubeSat unit were combined together to establish larger CubeSats. The CubeSat concept has become very popular, both in university groups, as well as for researchers, space agencies, governments, and companies. CubeSats offer a fast and affordable way for a wide array of stakeholders to be active in space and allow for a fast innovation cycle.

1.1.1 Cubesat Standard and Model

Beginning in 1999 the university professors' purpose was to come up with a new concept of satellite that would not only allow university groups to rapidly implement a small space mission but also to ensure that the chances of being embarked on a space launch as a secondary passenger were maximized [2]. This was enabled by standardizing interfaces and prohibiting or limiting design aspects that could be potentially hazardous and would reduce the chances of being allowed to be launched next to larger, more expensive spacecrafts. The forecast of CubeSat launches until the year 2023 is shown in figure 1.1. The CubeSat reference design was proposed by professors Jordi Puig-Suari of California Polytechnic State University and Bob Twiggs of Stanford University. The CubeSat as initially proposed did not set out to become a standard; rather, it became one over time. Given the standardized size of CubeSats, it has become possible to develop prefixed sized functional subsystems

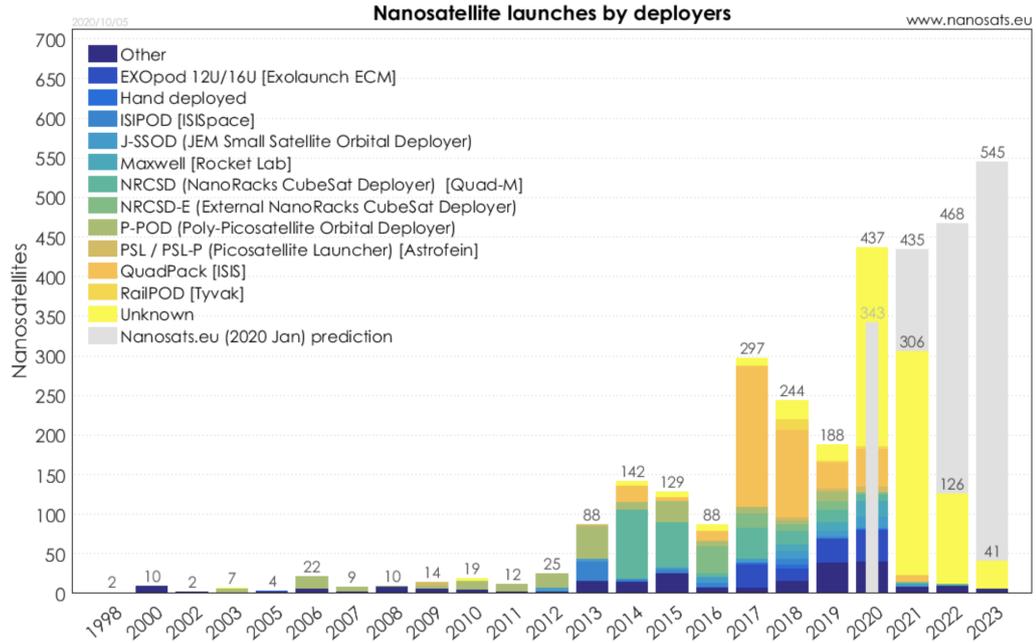


Figure 1.1: Trend of CubeSat missions [3]

such as power, communication, ADCS, etc. The work of this thesis will focus on the **Attitude Determination and Control system**, so this subsystem will be discussed in more detail in the following chapters.

The use of commercial off-the-shelf (COTS) parts is a major reason why CubeSat can be implemented quickly and cost-effectively [4]. CubeSats are typically launched into space as a containerized payload, inside a deployer (P-POD), as a means to reduce launch campaign complexity and associated costs. CubeSats are typically involved in missions that are implemented using a very low budget. Parts Non-space are often used, and accepted, in CubeSat missions. This enables a low-cost, short implementation cycle approach and allows developers to make use of the latest commercial and industrial grade components. The CubeSat approach is typically one with a higher accepted technical risk in exchange for either lower cost, faster implementation, more innovation or a combination of those elements. These elements also allow for different risk mitigation approaches (re-flights, in-orbit spares, etc) than traditional ones. The strength of CubeSats lies not so much in having the best performance in terms of bandwidth or ground sampling resolution, as this is not very compatible with the extreme restrictions that the satellite size imposes. When used in networks or constellation, however, CubeSats can provide much improved temporal resolution at affordable prices. Further information on

this can be found in [5], [6] and [7].

A formal standard for CubeSats is under development at the International Organization for Standardization (ISO). The standardized mass of a CubeSat also turns out to be less of a standard element across all CubeSats but is rather driven by the limitations of the specific deployer used. The applicable requirements and constraints to a particular CubeSat mission is a multi-layered composition of requirements. Depending on the chosen development method and standards, possible additional requirements and constraints may be applied. A number of these requirements cannot be known from the start of the mission development. Therefore, it is common practice to use sets of ‘envelope specifications’ that allow more flexibility during the initial design stages.

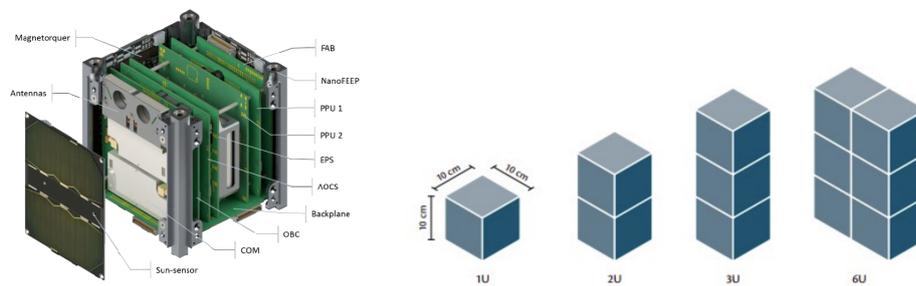


Figure 1.2: CubeSat standard [8], [9]

1.1.2 Introduction to ADACS

This section briefly introduces the pivotal subsystem of the present discussion, the Attitude Determination and Control System. This satellite subsystem deals with the determination and control of the satellite’s attitude. In particular, the control of the satellite can be achieved in a **passive** or **active** manner, and it is on the second category that will focus the present work, as this is the most used one.

The ADC subsystem has been introduced as one of the most critical for mission accomplishment, since mission-critical functions such as detumbling, stabilization, nominal pointing, science mode pointing, emergency orientation, etc. depend on it.

To perform its assigned functions, the system consists of hardware and software. In particular: the sensors, used for attitude determination, and the actuators (generally bulkier and with higher power consumption) used to control the satellite’s attitude, are part of the hardware. As far as the software is concerned, this is divided into two

sub-parts. The attitude determination algorithm is used to determine the current attitude of the satellite (using inputs from the sensors). The current attitude of the satellite is then taken as input and processed by the control algorithm, which, by comparing the current position with the desired one, calculates the actuation to bring the error between the two configurations to zero. These two parts of software are interdependent and constitute the control loop, which example is reported in figure 1.3. The ADCS for a 1U CubeSat is shown in figure 1.4.

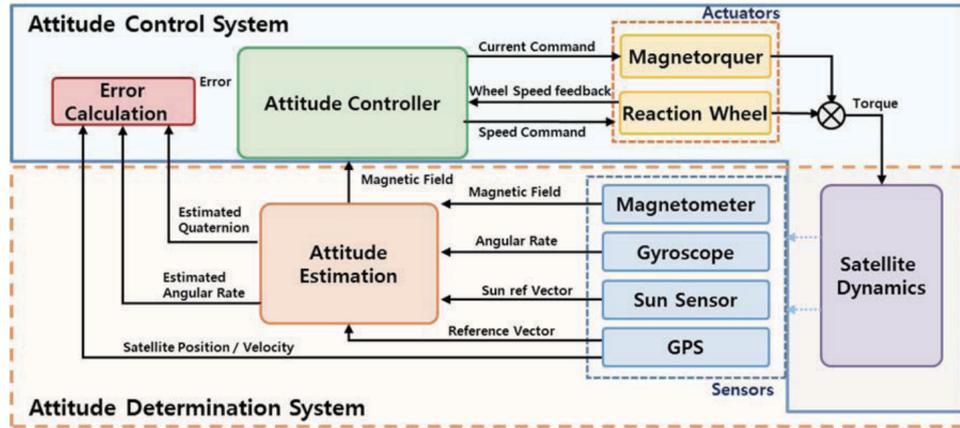


Figure 1.3: ADC functioning loop [10]

As can be seen, the size of the complete subsystem is extremely contained. It should also be noted that the depicted system represents the State of the Art in terms of technologies which are exhaustively reported in [11].

1.1.3 Test bench for ADCS state of the Art

Small satellites have many possible points of failure. CubeSat typically have dozens of microprocessors and related electronic circuitry. These components are used to control power management, solar cell operation, attitude control, telemetry, antennae pointing, propulsion systems, star trackers, sun sensors, earth sensors, cameras, and telescopes. A malfunction in any one of these systems can cause a full or partial failure of the mission. As a result, only one (or in the worst case none) of the mission's objectives will be satisfactorily completed. Considering the fact that most CubeSat do not use radiation hardened electronic components, small satellite failure is not a rare occurrence as reported in figure 1.5. A study observed that between the years 2000 to 2016, 41.3% of all small satellites launched experienced total or partial mission failure. Of these, 6.1% were launch vehicle failures, 11% were partial mission failures, and 24.2% were total mission failures [13].

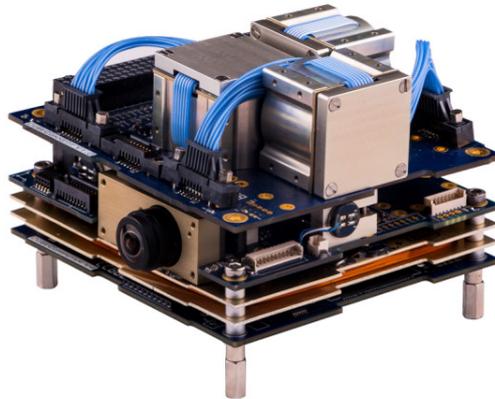


Figure 1.4: Example of full ADCS board for CubeSat applications [12]

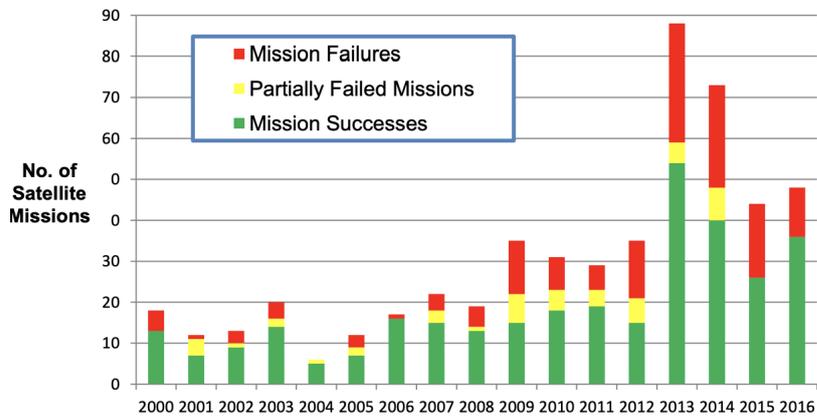


Figure 1.5: Failure rate of CubeSat mission over the year [14]

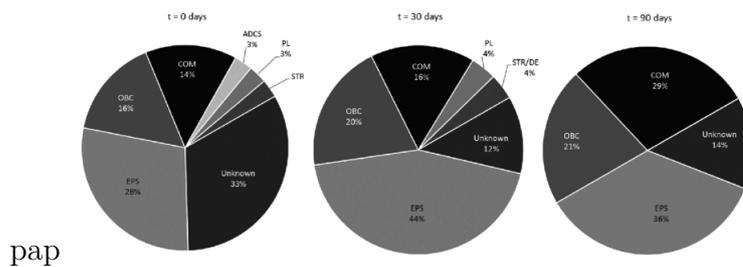


Figure 1.6: Failure rate of CubeSat mission divided in subsystem failure [15]

Hardware and software **Pre-flight ground verification** strives to prove and certify the functionality of the systems, components, and safety features of the

satellite. The satellite must be placed in a test environment that is similar to those found in orbit to reach a reliable test condition. Small companies and academic institutions lack ground testing because they are prevented from access to expensive, small-sat's specialised testing facilities. One of the causes of the high failure rate of nanosatellites is insufficient testing [15]. Over the last two decades, several test criteria have been created to give small-sat's developers acceptable testing standards as in [16].

This thesis focuses on the *improvement of a facility suitable to perform test* on the Attitude Determination and Control subsystem. This is one of the most complicated subsystems within the satellite, as key functions for mission accomplishment depend on it and involve an extremely high level of hardware and software complexity. The high complexity and importance of the subsystem require an adequate testing campaign and consequently adequate testing system.

The focus is on ADCS-related facilities, which may be divided into **simulators** and **testbeds**. The simulators are primarily used to study control laws, whilst the testbeds are used to evaluate integrated hardware devices, called Hardware-In-the-Loop (HIL) testing. Since ADCS may rely on the magnetic field, sunlight, and location of the stars, many components of the on-orbit environment must be recreated. Although facilities for ADCS testing of satellites have been developed in the past, testing of small satellites poses additional challenges due to the *very low residual disturbance torque* values required. To date, there are a few facilities that effectively recreate the space environment compatible with the requirements imposed by small satellites. In fact, the main problem with existing facilities to date is that the residual torque of these is too high for CubeSat's testing. The aim of this thesis is to improve the test facility located at the $\mu 3S$ laboratory in Bologna, which is dedicated to the testing of ADCS system for CubeSats, making it adaptable to ADCS testing for nanosatellites ranging in size from 1 to 3U. Currently, the state of the art sees the use of **Air Bearing** implemented in testing facilities for the reproduction of microgravity conditions, as seen in the facilities [17], [18] and [19]. A different solution is employed in the [20] facility, which sees the use of a robotic arm.

1.2 Objectives

The objectives of the present thesis work are to **update** and **improve** the performances of the Attitude Determination and Control System (ADCS) test facility present at the University of Bologna, specifically in terms of minimization of residual disturbance torques and introduction of a universal mechanical interface for

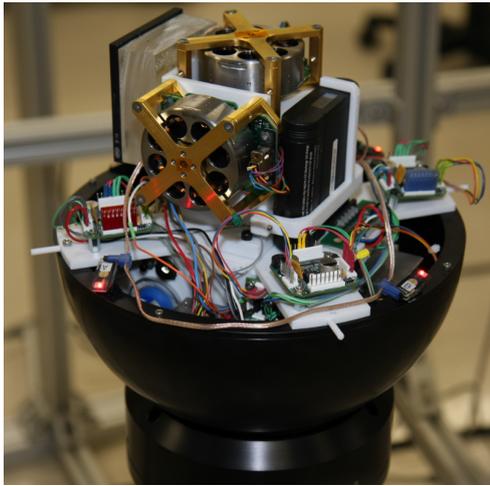


Figure 1.7: CubeTAS facility [21]



Figure 1.8: LAICA facility [22]



Figure 1.9: Facility of Montpellier [23]



Figure 1.10: $\mu 3S$ Facility

hosting of nanosatellites compliant with the CubeSat standards. The first part of the work involves designing and **selecting** new components for the implementation of the updated facility, **verifying**, **integrating**, and **validating** them. The focus

is on minimizing disturbance torque due to gravity and misalignment between the center of rotation and center of mass. To reduce the off-set a balancing procedure is employed, using three sliding masses moved by three dedicated stepper motors. Additionally, an embedded PCB of the electronic components included in the base plate is designed to reduce the presence of moving cables that could affect negatively the disturbance torque.

The second part of the thesis involves **sizing** magnetic actuators, implementing a single-axis control law in the ADCS board and verifying its effectiveness through **HIL** (Hardware In the Loop) tests. The work also includes testing torque generation by air-core magnetic actuator, in order to set back the system from a tilted angle configuration to the equatorial one.

1.3 Outline

Chapter 2 presents the theoretical problem of estimating and determining the attitude of a satellite and a mathematical description of the reference systems used for the facility. **Chapter 3** offers a detailed analysis of the facility's components and their function, with detail on the design of the host CubeSat. **Chapter 4** describes the integration, calibration and verification procedure for the components and the assembled facility. In **Chapter 5**, results are proposed for the balancing procedure using a dummy mass model for the subsystem and the residual torque estimation is carried out. To conclude, in **Chapter 6**, the test results for the magnetic actuation are proposed.

Chapter 2

Preliminaries

2.1 Reference Frames

To study satellite attitude, it is necessary to introduce a series of three-dimensional reference systems that describe the behavior and positioning of the satellite with respect to an inertial system. In this section, the main reference systems required for attitude analysis are analyzed following [24]. The reference systems introduced are:

- Spacecraft Body Frame
- Inertial Reference Frames
- Earth-Centred/Earth-Fixed Frame
- Local-Vertical/Local-Horizontal Frame

2.1.1 Spacecraft Body Frame

A spacecraft body frame is defined by an origin at a specified point in the spacecraft body, usually the CM, and three Cartesian axes. A body frame is used to align the various components during spacecraft assembly. It is quite common to define the body coordinate system operationally as the orientation of some sufficiently rigid navigation base, which is a subsystem of the spacecraft including the most critical attitude sensors and payload instruments. The navigation base often takes the form of a specially constructed optical bench, with its attached sensors and payload components. The purpose of attitude estimation and attitude control is to ascertain and to control the orientation of the navigation base relative to some external reference frame.

2.1.2 Inertial Reference Frames

An inertial reference frame is a frame in which Newton's laws of motion are valid. Any frame moving at constant velocity and without rotation with respect to an inertial frame is also inertial.

The *Earth-Centered Inertial Frame* (\mathcal{F}_{ECI}), represented in figure 2.1, is used to describe the orbital motion of the spacecraft around Earth center.

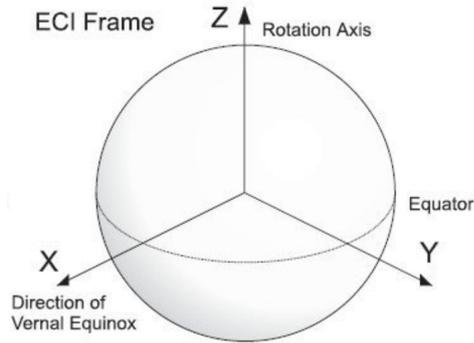


Figure 2.1: ECI Reference Frame

This coordinate frame is also known as Geocentric Inertial Frame (GCI). It is an inertial reference frame. The orbital frame \mathcal{F}_{ECI} is defined as:

- The origin O_{ECI} is the Earth's center;
- The Z_{ECI} axis, whose unit vector is \hat{K} , is aligned with the Earth's North pole, normal to the equatorial plane;
- The X_{ECI} axis, whose unit vector is \hat{I} , lies in the equatorial plane and points toward an inertial reference direction which is the vernal equinox;
- The Y_{ECI} , whose unit vector is \hat{J} , lies in the equatorial plane and it is defined such that it completes the right-handed triad;

2.1.3 Earth-Centered/Earth-Fixed Frame

The *Earth-Centered Earth-Fixed Frame* (\mathcal{F}_{ECEF}), represented in figure 2.2, is similar to the ECI frame with $Z_{ECEF} \equiv Z_{ECI}$ and the origin is again the centre of the Earth $O_{ECEF} \equiv O_{ECI}$. The difference is that the ECEF frame rotates together with the Earth with the angular velocity $\omega_{\oplus} \simeq 7.29218 \cdot 10^{-5} \text{rad/s}$. The orbital frame \mathcal{F}_{ECEF} is defined as:

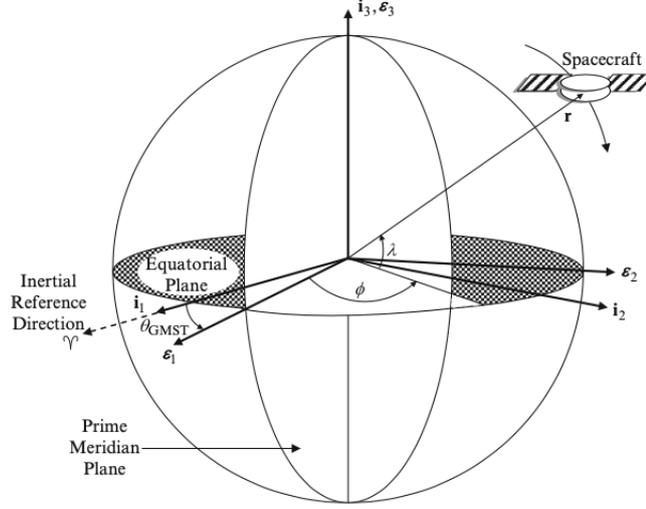


Figure 2.2: ECEF reference frame

- The origin O_{ECEF} is the Earth's center;
- The X_{ECEF} axis lies in the equatorial plane and points toward the Earth's prime meridian which is the Greenwich meridian;
- The Z_{ECEF} axis is normal to the equatorial plane and points toward the Earth's North pole;
- The Y_{ECEF} , axis lies in the equatorial plane and it complete the right-handed system;

The rotation angle is known as the Greenwich Mean Sidereal Time (GMST) angle and is denoted by θ_{GMST} . Determining the GMST angle requires the Julian Date, JD. For a given year Y (between 1901 and 2099), month M, day D, hour h, minute m, and second s, the Julian Date is calculated by equation 2.1.

$$\begin{aligned}
 JD(Y, M, D, h, m, s) = & 1,721,013.5 + 367Y - \text{INT} \left\{ \frac{7}{4} \left[Y + \text{INT} \left(\frac{M+9}{12} \right) \right] \right\} \\
 & + \text{INT} \left(\frac{275M}{9} \right) + D + \frac{60h + m + s/60^*}{1440}
 \end{aligned} \tag{2.1}$$

where INT denotes the integer part and 60^* denotes using 61 s for days with a leap second. We need to compute T_0 , the number of Julian centuries elapsed from the epoch J2000.0 to zero hours of the date in question 2.2.

$$T_0 = \frac{JD(Y, M, D, 0, 0, 0) - 2,451,545}{36,525} \tag{2.2}$$

The GCI coordinate system is fixed relative to the stars, not the Sun, so the GMST angle is the mean sidereal time at zero longitude. A sidereal day is the length of time that passes between successive crossings of a given projected meridian by a given fixed star in the sky. It is approximately 3 min and 56 s shorter than a solar day of 86400 s, which is the length of time that elapses between the Sun reaching its highest point in the sky two consecutive times. Therefore, θ_{GMST} in units of seconds is calculated by equation 2.3.

$$\begin{aligned} \theta_{GMST} = & 24,110.54841 + 8,640,184.812866T_0 + 0.093104T_0^2 \\ & - 6.2 \times 10^{-6}T_0^3 + 1.002737909350795(3600h + 60m + s) \end{aligned} \quad (2.3)$$

This quantity is next reduced to a range from 0 to 86400 s by adding/subtracting multiples of 86400. Then θ_{GMST} in degrees is obtained by dividing by 240, because $1s=1/240^\circ$.

2.1.4 Local-Vertical/Local-Horizontal Frame

The *Local-Vertical/Local-Horizontal Frame (LVLH)*, also known as *Local Orbital Frame* reported in figure 2.3 (\mathcal{F}_O) is used to describe motions with respect to the moving position and direction towards the center of the Earth of an orbiting body. This reference frame is convenient especially for Earth-pointing spacecraft.

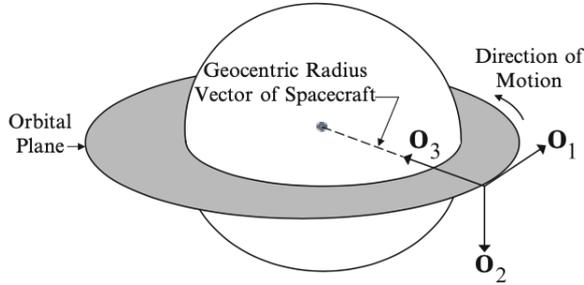


Figure 2.3: Local Vertical-Local Horizontal frame

The orbital frame \mathcal{F}_O is defined as:

- The origin O_o is the center of mass of the spacecraft;
- The z_O axis points along the nadir vector, directly toward the center of the Earth from the spacecraft;
- The y_O axis pointing along the negative orbit normal, in the direction opposite to the spacecraft's orbital angular velocity;
- The x_O axis completes the right-handed triad;

2.2 Attitude Representation

To describe the attitude of the satellite and subsequently describe its evolution over the time, it is necessary to solve the Euler equation of motion. For this purpose, the position of the spacecraft body frame must be known with respect to the local orbital frame. Mathematically, the \mathcal{F}_O system can be described using a coordinate transformation matrix which is defined with 3 parameters.

2.2.1 Euler Axis/Angle Representation

Representation by means of Euler angles are one of the possible solutions to indicate the rotation of one reference system with respect to another, specifically the sequence of rotations to be carried out to make the initial reference system coincide with the final one (fixed frame coincide with the body frame). In the discussion, the verses of the reference system will be referred to as \hat{E}_1 , \hat{E}_2 and \hat{E}_3 , while the verses of the body reference system will be referred to as \hat{e}_1 , \hat{e}_2 and \hat{e}_3 . The original rotation sequence proposed by Euler was the 3 – 1 – 3 one, which is characterized by:

- **1° rotation:** the first rotation is about the third axis of the initial frame, that is \hat{E}_3 , in our case, and takes the first axis \hat{E}_1 to the direction \hat{e}'_1 perpendicular to the plane determined by the unit vectors \hat{E}_3 and \hat{e}_3 ; \hat{E}_2 is rotated onto \hat{e}'_2 ; the rotation angle is called *precession angle* Ψ ;
- **2° rotation:** the second rotation is about the first axis transformed after the first rotation, \hat{e}'_1 , and takes the axis \hat{e}'_3 into the position of \hat{e}_3 ; \hat{e}'_2 is moved onto \hat{e}''_2 ; the rotation angle is called
- **3° rotation:** the last rotation is the one of \hat{e}_3 , and it end up bringing the $\hat{e}''_1 = \hat{e}'_1$, this rotation is named *spin angle* Φ

The listed rotations are represented as in [25] in figure 2.4. The angles Ψ , Θ , Φ , which represent the amplitude of the three successive rotations (around the third, first and third axes respectively) can be used to represent the attitude of the body reference system \mathcal{F}_B : The nutation angle represents the inclination of the third axis of the body (\hat{e}_3) with respect to the local vertical of the fixed reference system; the precession angle represents the angle between the first inertial axis \hat{E}_1 and the line of nodes ξ , i.e. the intersection between the planes perpendicular to \hat{e}_3 and \hat{E}_3 ; the rotation angle is the rotation around the third axis of the body.

It should be mentioned that other sequences, such as Bryant's angle 3-2-1 sequence or Cardan's angle 1-2-3 rotation, can be used, du rotation to bring \mathcal{F}_I with \mathcal{F}_B .

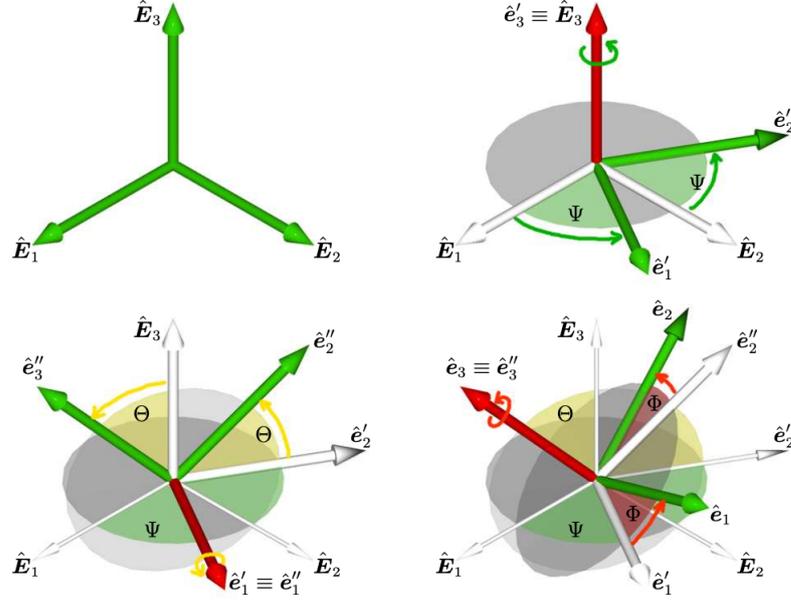


Figure 2.4: Euler Rotation 3-1-3

One problem with 3-dimensional representations is that of singularities. Following the 3-1-3 rotations we consider the case where $\Theta = 0$. This condition is singular because the spin and precession rotation turn out to be around the same axis $\hat{E}_3 = \hat{e}_3$. This means that all the triplets $(\Psi, 0, \Phi)$ for which $\Psi + \Phi$ is constant represent the same change of reference frame.

Each rotation of the 3-1-3 sequence can be translated into an elementary rotation around an axis, which remains fixed during the rotation. The 3 rotation matrices corresponding to the 3 elementary rotations are represented in the equation 2.4, 2.5 and 2.6.

$$\mathbf{R}_3(\Psi) = \begin{bmatrix} \cos(\Psi) & \sin(\Psi) & 0 \\ -\sin(\Psi) & \cos(\Psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

$$\mathbf{R}_1(\Theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\Theta) & \sin(\Theta) \\ 0 & -\sin(\Theta) & \cos(\Theta) \end{bmatrix} \quad (2.5)$$

$$\mathbf{R}_3(\Phi) = \begin{bmatrix} \cos(\Phi) & \sin(\Phi) & 0 \\ -\sin(\Phi) & \cos(\Phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

2.2.2 Quaternion Representation

To solve the singularity problem (necessary to implement numerical implementations of the spacecraft dynamics) introduced with the representation by Euler angles, the representation by Euler angles or quaternions is introduced. Quaternions were first devised by William Rowan Hamilton, a 19th-century Irish mathematician.

Euler's eigenaxis rotation theorem states that it is possible to rotate a fixed frame \mathcal{F}_I onto any arbitrary frame \mathcal{F}_B with a simple rotation around an axis $\hat{\mathbf{a}}$ that is fixed in both frames, called the **Euler's rotation axis** or eigenaxis, the direction cosines of which are the same in the two considered frame. A 4-variable representation of the attitude is proposed as quaternion representation. The major characteristic of this representation is the one to not present singularities. A quaternion $\mathbf{q} \in \mathbb{H}$ is defined as in 2.7:

$$\mathbf{q} = [q_1, q_2, q_3, q_4]^T = \begin{bmatrix} \mathbf{q}_e \\ q_4 \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{e}} \sin \frac{\Phi}{2} \\ \cos \frac{\Phi}{2} \end{bmatrix} \quad (2.7)$$

With the first 3 components of the vector \mathbf{q} we have information about the axes of instantaneous rotation, while with q_4 we have information about the cosine of the angle of rotation around the Euler axis. Note that the relation $q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1$ must be respected. With quaternions representation the Direct Cosine Matrix can be written as in equation 2.8.

$$R(\mathbf{q}) = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \quad (2.8)$$

The coordinate transformation matrix can also be written in a compact form as reported in 2.9.

$$\mathbf{A}(\mathbf{q}) = (q_0^2 - \vec{q} \cdot \vec{q}) \mathbf{1} + 2\vec{q}\vec{q}^T - 2q_0\mathbf{S}(\vec{q}) \quad (2.9)$$

Where $\mathbf{S}(\vec{q})$ represent the cross-product matrix equivalent or skew matrix.

$$\mathbf{S}(\vec{q}) = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix} \quad (2.10)$$

2.3 Attitude Dynamics and Kinematics

In this section of Chapter 2, the dynamics and kinematics relating to satellite attitude are introduced.

2.3.1 Attitude Kinematics

As far as kinematics is concerned, this differs from dynamics in that in dynamics disturbance torques are considered, whereas in kinematics they are not. The kinematics of a rotating body is the representation of the orientation of the body that is rotating. By introducing angular velocity as $\boldsymbol{\omega}_{OB}^B = [\omega_1, \omega_2, \omega_3]^T$, using quaternions as attitude representation, the attitude is given by integrating the following equations 2.11, 2.12 and 2.13.

$$\dot{\mathbf{q}} = \frac{1}{2}\boldsymbol{\Omega}\mathbf{q} = \frac{1}{2} \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} \mathbf{q} \quad (2.11)$$

$$\dot{\mathbf{q}}_e = -\frac{1}{2}(\boldsymbol{\omega} \times \mathbf{q}_e) + \frac{1}{2}q_4\boldsymbol{\omega} \quad (2.12)$$

$$\dot{q}_4 = -\frac{1}{2}\boldsymbol{\omega}^T \mathbf{q}_e \quad (2.13)$$

2.3.2 Attitude Dynamics

The dynamics of a rigid body is now analyzed in detail. The Euler equation, introduced at the beginning of the chapter, is to be determined to describe the dynamic of a rigid body about its center of mass with respect to an inertial frame. To do this, it must be introduced the equation of the angular momentum of a rigid body with respect to an inertial reference frame, in this case the ECI one. Euler's equation is reported in 2.14.

$$\dot{\bar{\mathbf{h}}}^B + \bar{\boldsymbol{\omega}}_{IB}^B \times \bar{\mathbf{h}}^B = \bar{\boldsymbol{\tau}}_{tot}^B \quad (2.14)$$

In 2.14 $\bar{\boldsymbol{\omega}}_{IB}^B$ represent the angular velocity of the SC with respect to the inertial reference frame, in body frame \mathcal{F}_B . $\bar{\mathbf{h}}^B$ is the SC angular momentum vector in \mathcal{F}_B . $\bar{\boldsymbol{\tau}}_{tot}^B$ represent the vector that includes the sum of all the external torques acting on the SC with respect to his CM, expressed in \mathcal{F}_B . Consider the definition of angular momentum $\mathbf{h} = \mathbf{J}\boldsymbol{\omega}$, where \mathbf{J} is the inertia matrix. The assumption of rigid body implies that the time derivative of the inertia matrix is zero, so it does not vary with time. This implies that the time derivative of the angular momentum is expressed as 2.15.

$$\dot{\bar{\mathbf{h}}}^B = \mathbf{J}\dot{\bar{\boldsymbol{\omega}}}_{IB}^B \quad (2.15)$$

Substituting the Euler's equation in 2.15, it is possible to calculate angular accelerations as: $\dot{\bar{\boldsymbol{\omega}}}_{IB}^B = \mathbf{I}^{-1}(\bar{\boldsymbol{\tau}}_{tot}^B - \bar{\boldsymbol{\omega}}_{IB}^B \times \mathbf{I}\bar{\boldsymbol{\omega}}_{IB}^B)$. Integrating the equation in time, angular velocities are obtained.

2.3.3 Disturbance Torques

The space environment is characterized by a series of environmental features that interact with the satellite and generate disturbance torques. It is important to consider all the torques as they perturb the spacecraft's dynamic. Considering an Earth-like orbital environment, non-negligible disturbance pairs result:

- Magnetic Field Torques
- Solar Pressure Torques
- Torques due to Aerodynamic Resistance
- Gravitational Gradient Torques

Magnetic Field Torques

In the context of Earth orbits, the magnetic field generated by the planet must be considered to determine its interaction with the satellite. The earth's magnetic field plays a fundamental role in the satellite's dynamics. The satellite generates a magnetic dipole due primarily to the magnetic control systems and secondarily to any current loops inevitably present in the on-board electronics.

To verify the performance of the determination algorithms, it is necessary to have a model of the magnetic field to simulate the operation of the magnetometer whose measurements are used to determine the attitude. The International Geomagnetic Reference Field (IGRF) model is a geomagnetic field model released by the International Association of Geomagnetism and Aeronomy (IAGA). The earth's magnetic field strength can be calculated as in 2.16, where V denotes the scalar potential, function of the orbital radius r_s , of the co-elevation θ and the longitude λ .

$$\mathbf{B} = -\nabla V \quad (2.16)$$

The absence of magnetic monopoles implies that the divergence equation is zero, so the Laplace's equation is:

$$\nabla^2 V = 0 \quad (2.17)$$

The scalar potential function can be expressed as a series of spherical harmonic expansions as in 2.18.

$$V(r_s, \theta, \lambda) = R_E \sum_{n=1}^k \left(\frac{R_E}{r_s} \right)^{n+1} \sum_{m=0}^n (g_n^m \cos(m\lambda) + h_n^m \sin(m\lambda)) P_n^m(\theta) \quad (2.18)$$

where g_n^m and h_n^m express the coefficients defined by the IGRF (known as Gaussian Coefficients), k is the order of the expansion and $P_n^m(\theta)$ represent the Lagrange

Function with respect to m, n and θ . The IGRF is a set of spherical harmonic coefficients which can be used in mathematical model to describe the large-scale, time-varying portion of Earth's internal magnetic field.

Due to the coupling between the residual dipole of the satellite and the Earth's magnetic field, a disturbing torque is generated. The residual dipole depends on the size of the spacecraft and the arrangement of electronic components, the circuits of which create closed current loops resulting in the generation of a magnetic dipole. When the magnetic dipole is not aligned with the earth's magnetic field, a torque tends to realign the two vectors. The magnetic disturbance torque varies according to the formulation shown in figure 2.19.

$$\bar{\tau} = \bar{m} \times \bar{\mathbf{B}} \quad (2.19)$$

where \bar{m} is the residual dipole of the satellite and $\bar{\mathbf{B}}$ is the Earth's magnetic field, both in body frame.

Solar Pressure Torques

Solar radiation consists of photons, which carry energy according to Planck's law. Interacting with the satellite, this energy is exchanged with the satellite, generating a disturbance pair. In LEO, the effects of SRP are non-negligible. The satellite is modelled as a set of N flat plates of area S_i , with normal \mathbf{n}_B^i in body axes, specular reflection coefficient R_{spec}^i , scattering coefficient R_{diff}^i and absorption coefficient R_{abs}^i . The coefficients must respect the relation $R_{spec}^i + R_{diff}^i + R_{abs}^i = 1$. The unit vector Spacecraft-to-Sun is introduced as $\mathbf{s} = \mathbf{A} \mathbf{e}_{sat\odot}$, where \mathbf{A} is the attitude matrix and $\mathbf{e}_{sat\odot}$ is the spacecraft-to-Sun vector in ECI frame. The SRP force is expressed as in equation 2.20.

$$\mathbf{F}_{SRP} = -P_{\odot} S_i \left[2 \left(\frac{R_{diff}^i}{3} + R_{spec}^i \cos \theta_{SRP}^i \right) \mathbf{n}_B^i + (1 - R_{spec}^i) \mathbf{s} \right] \max(\cos \theta_{SRP}^i, 0) \quad (2.20)$$

where $\cos \theta_{SRP}^i = \mathbf{n}_B^i \cdot \mathbf{s}$ is the angle between the Sun vector and the normal to the i -th plate of the satellite, and P_{\odot} is the Solar radiation Pressure. Once the force due to solar radiation is known, it is possible to calculate the disturbance torque as in 2.21.

$$\tau_{SRP}^i = \sum_{i=1}^N \mathbf{r}^i \times \mathbf{F}_{SRP}^i \quad (2.21)$$

where \mathbf{r}^i is the vector from the spacecraft CM and the center of pressure of the SRP on the i -th plate.

Torques due to Aerodynamic Resistance

In LEO orbits, the presence of the upper atmosphere results in an interaction with the satellite that leads to torque decay over time. It should be noted that atmospheric behavior is strongly influenced by the phase of the solar cycle occurring, which causes a variation in density and thus in the resistance generated. In general, the torque due to aerodynamic drag generates a disturbing torque whenever there is an offset between the satellite's aerodynamic center and the center of mass. Modeling the SC again as a sum of N flat plates of area S_i having normal \mathbf{n}_B^i expressed in body axes, the aerodynamic disturbance pair is determined. This torque depends on the velocity of the SC in the atmosphere. This velocity can be expressed as 2.22:

$$\mathbf{v}_{\text{rel},ECI} = \mathbf{v}_{ECI} + [\boldsymbol{\omega}_{\oplus} \times] \mathbf{r}_{ECI} \quad (2.22)$$

The Earth's angular velocity vector is $\boldsymbol{\omega}_{\oplus I} = \omega_{\oplus} [0 \ 0 \ 1]^T$. Inserting this $\boldsymbol{\omega}_{\oplus I}$ give the relative velocity in the body frame as:

$$\mathbf{v}_{\text{rel}B} = A \begin{bmatrix} \dot{x} + \omega_{\oplus} y \\ \dot{y} - \omega_{\oplus} x \\ \dot{z} \end{bmatrix} \quad (2.23)$$

where A is the attitude matrix. The inclination of the i -th plate to the relative velocity is given by:

$$\cos \theta_{\text{aero}}^i = \frac{\mathbf{n}_B^i \cdot \mathbf{v}_{\text{rel}B}}{\|\mathbf{v}_{\text{rel}}\|} \quad (2.24)$$

so, the aerodynamic force is 2.25.

$$\mathbf{F}_{\text{aero}}^i = -\frac{1}{2} \rho C_D \|\mathbf{v}_{\text{rel}}\| \mathbf{v}_{\text{rel}B} S_i \max(\cos \theta_{\text{aero}}^i, 0) \quad (2.25)$$

where C_D is the drag coefficient and ρ is the atmospheric density. The total aerodynamic torque is:

$$\mathbf{L}_{\text{aero}}^i = \sum_{i=1}^N \mathbf{r}^i \times \mathbf{F}_{\text{aero}}^i \quad (2.26)$$

where \mathbf{r}^i is the vector from the spacecraft center of mass to the center of pressure of the i -th plate.

Gravitational Gradient Torques

The earth's gravitational field is the last environmental characteristic we will consider regarding the generation of disturbances. The gravity-gradient torque is generated whenever an object's center gravity is not aligned with its center of

mass, with respect to the local vertical and the Earth's gravitational force is not constant with distance from the Earth's center. Considering that the spacecraft mass is infinitesimally smaller than the Earth's one, and considering that the distance between SC and planet is higher than the spacecraft dimensions, the gravity gradient torque can be calculated as:

$$\boldsymbol{\tau}_{gg} = 3\omega_o^2 (\hat{\mathbf{e}}_{z,b} \times J\hat{\mathbf{e}}_{z,b}) = \frac{\mu}{R_{gd}^3} (\hat{\mathbf{e}}_{z,b} \times J\hat{\mathbf{e}}_{z,b}) \quad (2.27)$$

where ω_o is the orbital angular velocity of the spacecraft and $\hat{\mathbf{e}}_{z,b}$. In 2.30 μ is the gravitational parameter and R_{gd} is the geocentric distance. The maximum experienced gravitational torque is:

$$\|\boldsymbol{\tau}_{gg}\|_\infty = \frac{3}{2}\omega_o^2 \max(\|J_3 - J_2\|, \|J_3 - J_1\|, \|J_1 - J_2\|) \quad (2.28)$$

where J is the inertia matrix of the SC.

2.3.4 Testbed Dynamics

In the present section of Chapter2 the dynamics of an air-bearing platform with automatic balancing system is analyzed in detail.

The in-plane balancing control problem is equivalent to stabilizing the hanging equilibrium point of a three-dimensional pendulum, where the rotating platform is assimilated to a rigid body with point masses in linear motion. The CR is fixed to a point in the inertial coordinate system \mathcal{F}_i , whose z-axis is considered parallel to the local vertical. The IMU, which in the facility under analysis is located below the base plane, is aligned with the platform reference frame \mathcal{F}_b , centered on the CR, and the equilibrium masses can only move along a set of mutually orthogonal unit axes defined by \mathcal{F}_a , which represents a reference system fixed with respect to the reference frame of the platform. The relative orientation between frames is described by the quaternion \mathbf{q} (or equivalently by a rotation matrix R), and the rotational kinematics of the platform is a function of the absolute angular velocity $\boldsymbol{\omega}$. The gravity vector \mathbf{g} is expressed in the body axis reference frame as in 2.29.

$$\mathbf{g} = g\hat{\mathbf{g}} = gR(\mathbf{q})\hat{\mathbf{e}}_{i,z} \quad (2.29)$$

where g is the magnitude of the gravity vector and $\hat{\mathbf{e}}_{i,z} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$ is the z-axis versor. Considering that the Microsatellites and Space Microsystems Laboratory implements a *spherical air bearing* the offset position between CM and CR is calculated as $\mathbf{r}_{off} = \mathbf{r}_{CM} - \mathbf{r}_{CR}$. In the formulation \mathbf{r}_{CM} represent the position vector of the Center of Mass, meanwhile \mathbf{r}_{CR} is the position vector of the Center

of Rotation. Note that \mathbf{r}_{CR} is inertially fixed, so it is placed as the origin of the reference frame, implying that the formulation introduced previously for \mathbf{r}_{off} becomes $\mathbf{r}_{off} = \mathbf{r}_{CM}$. Due to the offset present between the center of rotation and the center of mass, a gravitational torque is generated which has the expression 2.30, where m_{tot} indicates the 'satellite' total mass.

$$\boldsymbol{\tau}_{CM} = m_{tot}\mathbf{g} \times \mathbf{r}_{CM} \quad (2.30)$$

To compensate for the gravitational torque, and thus minimize it, a mass $m_{b,tot}$ is introduced at a distance \mathbf{r}_b . This compensating mass will in turn generate a torque:

$$\boldsymbol{\tau}_u = m_{b,tot}\mathbf{g} \times \mathbf{r}_b \quad (2.31)$$

where the value of r_b must be such that we obtain $\tau_b = \tau_{CM}$. It is now necessary to write the Euler's equation of dynamics considering the effect of τ_{CM} and τ_u . Considering punctiform balancing masses the total angular momentum is:

$$\mathbf{h} = J\boldsymbol{\omega} + \sum_{i=1}^3 \mathbf{r}_{b,i} \times m_{b,i}\dot{\mathbf{r}}_{b,i} \quad (2.32)$$

where J is:

$$J = J_S + \sum_{i=1}^3 (-m_{b,i} [\mathbf{r}_{b,i} \times] [\mathbf{r}_{b,i} \times]) \quad (2.33)$$

In 2.33 J_S is the inertia without the balance masses, while J is the total inertia matrix. In equation 2.32 the further right term can be neglected if the masses are sufficiently small .

The rotational dynamics of the simulator about CR, subject to gravity torque can be expressed as in 2.34.

$$J\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times J\boldsymbol{\omega} + m_{tot} \mathbf{g} \mathbf{r}_{CM} \times R^T \hat{\mathbf{e}}_3 + \boldsymbol{\tau}_u \quad (2.34)$$

where the only method to control the rotational dynamic is to vary the position of the balancing masses. The control mass displacement components \mathbf{r}_b necessary to produce the correct torque can be calculated in accordance with the selected control law. Further information about the dynamic of the testbed are analyzed in [26].

Chapter 3

Facility Description

This section will introduce the facility present in Bologna laboratory by describing the included components, placing special attention and focus on the functionality of each of them.

3.1 Test bench description

3.1.1 Microsatellites and Space Microsystems Laboratory

Given the importance of the ADCS subsystem and its complexity, it is essential to carry out a thorough testing campaign before launching the satellite. To do this, an appropriate test facility is required. Dynamic simulation facilities have been built since the dawn of space missions, by emulating the space environment and microgravity. It is precisely this last feature, the reduced gravity compared to that on earth, that deserves special attention.

The larger the satellite, the lower the impact of the facility's disturbance torque on it. With the advent of nanosatellites, it has become increasingly important to ensure a disturbance torque compatible with the characteristics of the satellite under test. This implies that a characterising element of the facility is the disturbance torque it generates. This torque is a consequence of the different methodologies used to recreate microgravity.

The methods for recreating microgravity are many: they range from **cable suspension**, to the use of **robotic arms**, or **air bearings**. In MSM Laboratory it is implemented an air bearing type facility. The method of operation is extremely simple: a pressurized air flow is used to generate a film between two corresponding surfaces. This film will act as a lubricant between the surfaces, substantially reducing friction. The types of surfaces used can be *two plates*, or *two spherical*

surfaces. Spherical air bearings are the most widely used in testing to date, as they allow frictionless rotational dynamics to be generated on one or more axes, depending on the bearing configuration. Most common air-bearing setups can be categorized into three main configurations, namely **table-top**, **dumbbell** and **umbrella**. The different types of air bearing set-up mentioned above are shown in figure 3.1.

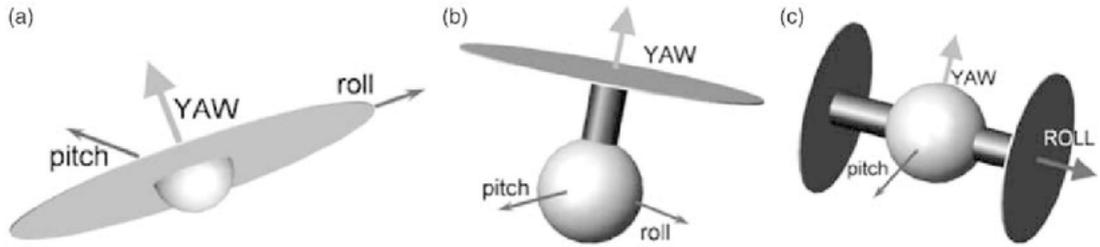


Figure 3.1: Air bearing spherical setups: table-top (a) , umbrella (b), and dumbbell (c) [27]

In the case object of study, the table-top configuration is used, as it is the lightest and least expensive, at compromise of a limited tilt angles.

On-orbit environment aspect	Usage	Recreation method
<i>Sunlight</i>	Sun sensors	From correctly sized lamp to precisely calibrated light sources and specific optical systems
<i>Stars</i>	Star tracker	Not recreated due to the complexity and high cost of hardware and software required
<i>Magnetic Field</i>	- Magnetometer - Magnetic Torquer	Helmholtz cage
<i>Microgravity</i>	-	Compressor

Table 3.1: On-orbit environment aspects that needs to be simulated

The Bologna’s facility consists of two main groups of components: components that reproduce the spatial environment, and testing components that enable the subsystem under study to be analyzed. In the first category of components are allocated:

- Helmholtz cage
- Sun simulator
- Air Bearing

In the second category of components are placed:

- Air Bearing interface plate
- Stepper Motors
- Mechanical interface with the subsystem

3.1.2 Space environment reproduction components

Helmholtz cage

The working principle of the Helmholtz cage is that a circular coil of radius R and N windings, with a current i running through it, at a distance x from the center, generates a magnetic field B according to the formulation:

$$B = \frac{\mu_0 i N R^2}{2(x^2 + R^2)^{\frac{3}{2}}} \quad (3.1)$$

This magnetic field must perform two basic functions: the first is to counteract the earth's magnetic field present in the laboratory where the test is taking place, and the second is to simulate the magnetic condition that occurs in the orbital environment. The Helmholtz cage is composed of the following hardware components:

- 3 orthogonal pairs of coils with 1300 mm diameter, which correspond to the x, y and z axes of the reference system, respectively.
- 1 fluxgate magnetometer
- Programmable power supply
- Arduino Uno for closed-loop control

By means of the components listed above, it is possible to generate an arbitrary magnetic field in the range of $\pm 10 Gauss$ ($1 Gauss = 10^{-4} Tesla$). The cage makes also possible to track a desired magnetic field with $0.5 mG$ accuracy, moreover a 1% homogeneity of the field in a $404 mm^3$ spherical volume is achievable. In figure 3.2 the magnetic field that is generated between two coils of the HC is pictured. As it is seen the maximum magnetic field is located nearby the coils and it decreases its intensity with the cube of the distance. In the center of the HC it is installed a pedestal, which has the function to keep the base platform of the air bearing at the center of the cage, and so at the center of the generated magnetic field.

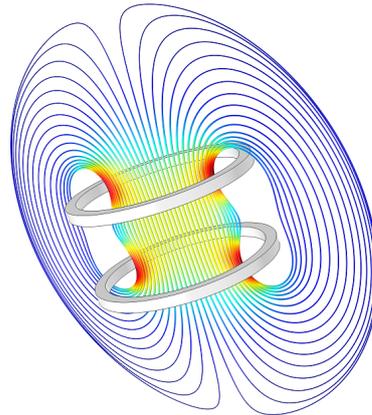


Figure 3.2: Magnetic field generated from two co-axial coils

Sun simulator

The function of the sun simulator is to replicate, even if only partially, the solar radiation characterising the satellite’s orbit. This radiation makes it possible to assess two fundamental aspects for the ADC subsystem. The first relates to the satellite’s attitude determination: for small satellites such as CubeSats, attitude determination systems that exploit solar radiation are often employed. Such systems can be active or passive, depending on mission requirements and available budget, it is determined which sensor is preferable to use. Low-priced but also low-precision sensors include photodiodes, which consist of sun radiation sensitive cells. By implementing a adequate number of sensors it is possible to coarsely determine the satellite’s attitude with respect to the sun: by means of the detection of current generation or not, from at least three of them. If a finer determination is required, active sun sensors are used. The second purpose of the sun simulator is to allow the simulation of disturbance torque due to solar radiation. This torque must be counterbalanced by the components that make up the attitude control part of the subsystem. The simulator consisting of a **COST LED Studio light**, with a 300W phosphor-coated LED as light source. The collimation lens is a 400mm diameter Fresnel. The purpose of the simulator is to collimate the light of the lamp to replicate sunlight. Through a focus test, the distance between the Fresnel lens and the light source has been tuned to maximise beam collimation. A Sun simulator is often categorised using the following three factors:

- spectral matching
- homogeneity of the space
- steadiness over time

Other factors need to be considered when testing a Sun sensor, such as the collimation of the light beam over a broad area, which needs to be kept within 0.53° , or the apparent angular diameter of the Sun at 1 AU, and the power flux level (about 1367 W/m^2 at 1 AU) at the nominal target distance, or in our case, the distance between the illuminated target and the LED source, which is about 0.75 m. The solar radiation cannot be replicated in its entirety due to the power and size limitations of the simulation system's focusing lens.

A LED source was chosen for our simulator, taking advantage of its inherently flicker-free emission, high efficiency, and good match in the visible part of the solar radiation spectrum. The main disadvantage is the near absence of emission in the IR and UV bands, so that spectral correspondence with sunlight outside the visible band is lost. This, however, is not considered a limiting factor in our application, as most of the existing nanosatellite solar sensors are built on CMOS, CCD, PSD, whose response is maximum in the visible band and falls rapidly in the IR and UV wavelengths. On the other hand, coarse solar sensors based on photocells would suffer more from the lack of simulation of the IR and UV bands. Note that the response of these sensors is also altered by the earth's albedo, which is not modelled in the facility anyway. By using an LED source, the aim is not to match the overall solar irradiance, as a large part of this is in the IR and UV bands. Rather, the LED power was chosen to match the extra-atmospheric solar illuminance which amounts to approximately 130,000 lux at 1 A.U [28]. As shown in the figure 3.3,

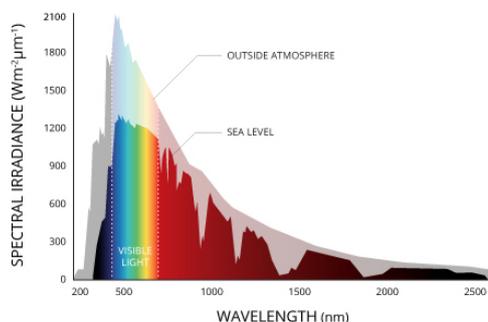


Figure 3.3: Solar Radiation



Figure 3.4: Sun simulator

the visible portion is only a fraction of the full spectrum of solar radiation. As for figure 3.4, we observe in detail the sun simulator implemented into the facility.

Air Bearing

As introduced above, an Air Bearing system was implemented to simulate the microgravity condition characterising orbital environment. This system, reported

in figure 3.5, consists of two parts: the base which is fixed at the pedestal, and the hemispherical one which is movable. Between two a thin layer of air is generated to minimize friction. Ideally, the disturbance torques affecting the platform should



Figure 3.5: Air Bearing system implemented in $\mu 3S$ facility consisted of an articulated stand (Ferronato® ULTAS-1 from Serviciencia, Toledo, Spain), a spherical air bearing (150mm of diameter) by Physical Instrumentns and tabletop-style platform

be kept in the same order of magnitude as those in orbit: in case of LEO orbit at 700 km altitude, with a 1U satellite, these orbital disturbance torques are in the order of $10^{-6}Nm$. The disturbance torques affecting an air-bearing platform can have different causes. In [29] the torques affecting an air-bearing platform are subdivided into: platform-derived torques, bearing-derived torques, ambient-caused torques and, finally, torques from the test system.

The torques arising from the platform include gravity torque and anisoelasticity, a torque caused by the deformation of the platform at different tilt angles. In the case under study, the torque due to gravity is the main torque acting on the platform, while anisoelasticity is not an issue as it is applied at small facility sizes.

Torques coming from the bearing include the aerodynamic torque of the turbine, i.e. the torque caused by the non-precisely symmetrical airflow. This is mitigated by using high-end COTS components with extremely low friction. Torques from the laboratory environment include aerodynamic torque, residual dipole torque and vibrations. Aerodynamic torque is neglected due to reduced angular velocities and facility's small size, residual dipole torque is minimized by using a-magnetic

materials for the platform, and vibration are minimized through robust mounting. Torques due to test environments, such as mass displacement in the bearing and loose couplings, are important in large test platforms, but can be neglected here, while other torques can be minimized by carefully designing the testing platform. Gravitational torque is present whenever there is misalignment between the CM and CR of the platform. The imbalance vector is determined by the separation between CM and CR. Theoretically, the gravitational torque can be completely adjusted if the CM-CR distance is set to zero. The accuracy of the Inertial Measuring Unit (IMU) positioned on the satellite platform, however, limits the accuracy to which this distance may be determined in practice. It is possible to create manual balancing methods, but these types of procedures takes time and does not ensure performance at an a-priori level. **Automatic balancing** techniques must be utilised to overcome these restrictions. To perform an automatic balancing technique, the tested platform's centre of mass and inertia parameters must be identified. Their identification is a problem in identifying dynamic parameters. The air bearing platform at the MSM includes an ABS that uses three stepper motors that drive three balancing masses, allowing the CM to be moved within a region that includes the CR. Due to the extremely high position of the CM caused by the fairly high weight of the components on the base plate, for the present case of study it was necessary to place additional masses in the hemispherical part of the bearing to low the CM position. Through this action, the weight of the satellite, and therefore its inertia, were certainly increased, but the correct positioning of the CM under the CR as a starting condition was guarantee. The total mass added in the bearing is $1.811[kg]$. To minimise the oscillations of these masses within the bearing, it was necessary to implement a fastening system consisting of a screw and an upper plate attached to the masses and the hemisphere.

The minimization of the distance between CM and CR is achieved by means of a balancing procedure carried out in two distinct phases. The first is that of coarse balancing: in this procedure, the host CubeSat is positioned on the base platform already placed in a microgravity condition by means of the mechanical connection to the mobile bearing hemisphere, and a manual balancing is made, searching for a point that sees the system globally balanced even if in a rough way. It is necessary to proceed with then second part of the balancing procedure: the automatic balancing, which is achieved by dynamically moving the 3 installed masses, commanded by a closed loop.

3.1.3 Testing Components

In this section of the thesis, the elements that constitute the interface with the testing facility and the actual tested system will be introduced in a complete and

detailed way.

Air Bearing interface plate

The base plate interface between the hemispherical part of the bearing and the ADCS subsystem is now discussed in detail. The ultimate purpose of this base plate is to mechanically interface with the base platform, to which it is screwed by means of through-bolts that are fixed in specific holes in the bearing. This plate also serves as the allocation of all the components necessary for automatic balancing and the ADC Subsystem.

The base plate is made of **7075 Aluminium** with a size of 268x268x6 mm. The longitudinal and transverse dimensions of the component were determined by the footprint of the stepper motors used, while the thickness was determined by the production requirements imposed by the workshop. The plate was then specifically drilled where it is to be fixed with the electronic elements, according to the CAD model of the structure. In addition, a special allocation for the motor along the z-axis was determined. In the central part, four slots are provided for attaching the host CubeSat to the top of the platform. The platform including the implemented electronics and the two mounted motors, as reported in figure 3.6, has a total weight of 1.961[kg], constituting 31% of the mass of the system to be tested. This fairly high weight leads to an increase in the overall weight of the facility, which in turn entails to extremely high inertia of the structure and thus substantial disturbance torques caused by the facility. Given the fundamental importance of minimizing the mass of the system to be tested, solutions have been developed to reduce it. The first one is to reduce the thickness of the base plate, opting for a solution that still guarantees adequate structural stability and integrity. A further solution for weight reduction could be to reduce the longitudinal and transverse dimensions of the plate: to do this, however, it is necessary to change the motors used for automatic balancing, preferring a type with a shorter stroke, and consequently smaller dimensions.

Stepper Motors

There are three stepper motors included in the design of the structure, one for each platform body axis (x,y,z), and their purpose is to move the balancing masses attached to them to automatically balance the platform once in microgravity conditions.

Stepper motors are DC motors that move in discrete steps. They have multiple coils that are organized in groups called *phases*. By energizing each phase in sequence, the motor will rotate, one step at a time.

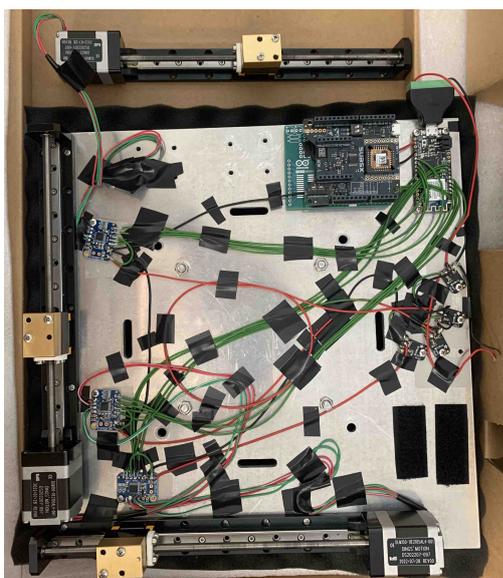


Figure 3.6: Interface plate with electrical components and stepper motors on x,y axes mounted

With a computer controlled stepping you can achieve very precise positioning and/or speed control. For this reason, stepper motors are the motor of choice for many precision motion control applications. A stepper motor may have any number of coils, but these are connected in groups called "phases". All the coils in a phase are energized together. Stepper motors are characterised by an anisotropic magnetic circuit. Considering a single winding, the magnetic geometry varies with position, the rotor moves to the minimum position for reluctance, the rotor moves to the maximum position for $L(\theta_m)$, where θ_m is the angle shown in 3.7. To calculate the delivered torque, the starting point is the electromagnetic induction equation 3.2.

$$v - \frac{d\varphi_c}{dt} = v - \frac{d(Li)}{dt} = Ri \quad (3.2)$$

Calculating the derivative, since inductance is a function of angular position θ , it is obtained:

$$v = L \frac{di}{dt} + i \frac{dL}{d\theta_m} \frac{d\theta_m}{dt} + Ri = L \frac{di}{dt} + i \frac{dL}{d\theta_m} \omega_m + Ri \quad (3.3)$$

The winding equation 3.4 is like that of a DC machine and includes the same terms.

$$v = L \frac{di}{dt} + i \frac{dL}{d\vartheta} \omega_m + Ri \quad (3.4)$$

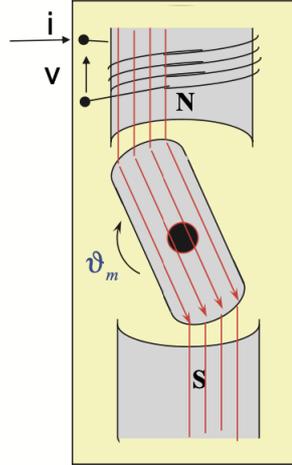


Figure 3.7: Single winding functioning for a variable reluctance stepper motor

The energy balance for the phase is carried out:

$$vi = Li \frac{di}{dt} + i^2 \frac{dL}{d\theta_m} \omega_m + Ri^2 \quad (3.5)$$

There is the input power and Ri^2 is the power dissipated by the Joule effect. Since L depends on position, the expression of magnetic energy must first be developed to interpret the other terms correctly:

$$\frac{d}{dt} \left(\frac{1}{2} Li^2 \right) = \frac{1}{2} \frac{dL}{d\theta_m} \omega_m i^2 + \frac{1}{2} L \left(2i \frac{di}{dt} \right) \quad (3.6)$$

To determine the torque, the term 3.7 has to be calculated.

$$Li \frac{di}{dt} = \frac{d}{dt} \left(\frac{1}{2} Li^2 \right) - \frac{1}{2} \frac{dL}{d\theta_m} \omega_m i^2 \quad (3.7)$$

The balance of powers is rewritten as in 3.8.

$$vi = \frac{d}{dt} \left(\frac{1}{2} Li^2 \right) + \frac{1}{2} i^2 \frac{dL}{d\theta_m} \omega_m + Ri^2 \quad (3.8)$$

It can therefore be deduced that the mechanical power is that indicated in 3.9.

$$P_m = \frac{1}{2} \frac{dL}{d\theta_m} \omega_m i^2 \quad (3.9)$$

So the pair has a value of:

$$C_m = \frac{P_m}{\omega_m} = \frac{1}{2} \frac{dL}{d\theta_m} i^2 \quad (3.10)$$

Bipolar drivers use H-bridge circuitry to reverse the current flow through the phases. By energizing the phases with alternating polarity, all the coils can be put to work one after the other, turning the motor. A two-phase bipolar motor has 2 groups of coils, as shown in figure 3.8, and therefore it will have 4 wires - 2 for each phase.

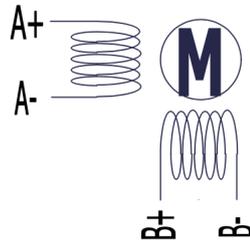


Figure 3.8: Bipolar functioning scheme for an External Linear stepper motor

Driving a stepper motor is a bit more complicated than driving a regular brushed DC motor. Stepper motors require a **stepper controller** to energize the phases in a timely sequence to make the motor turn. Driving a bipolar motor requires 2 full H-bridges so it can reverse the current to the phases. About the motor drivers chosen for the realisation of the two H-bridges, it was decided to use the Adafruit DRV8833 DC/Stepper Motor Driver Breakout Board, which will be introduced in detail in the following section on component integration and testing.

For this application, it was decided to use DINGS brand External Linear bi-polar stepper motors reported in figure 3.9, with the characteristics shown in table 3.2.



Figure 3.9: DINGS' Stepper motor implemented in the facility

Regarding the third motor driver, a bracket had to be implemented in order to fix it in a vertical position on the platform. This support will then be considered in the total mass calculation for the stepper motors. Each **motor** has a total weight of 322[gr] relative to the motor alone, to which the weight of the **moved mass**, equal to 45[gr], must be added. For the motor on the z-axis, a weight of 129[gr] has to be added for the support and the fixing screws.

<i>Voltage[V]</i>	4.55
<i>Current[A]</i>	0.5
<i>Resistance[Ω]</i>	9.1
<i>Inductance[mH]</i>	6
<i>Motor Length[mm]</i>	33.35
<i>Lead Wire No.</i>	4
<i>Screw Dia.[mm]</i>	4.77
<i>Lead[mm]</i>	0.3175
<i>Trave Per Step @ 1.8° [mm]</i>	0.0016

Table 3.2: Size 28mm DLM characteristics

Mechanical interface with the ADCS

An analysis of the solutions implemented to realise the mechanical interface with the ADCS subsystem and the base plate is made. The starting point was the necessity to design a system that was as *universal* as possible. In fact, the purpose of the facility is to test the ADC subsystem of any nanosatellite of size from 1U to 3U . For this reason, two fixing options were considered:

- First option involves the use of an interference locking system that interact directly on the rectangular case containing the subsystem, without any modification to the host system. This configuration is shown in figure 3.10.

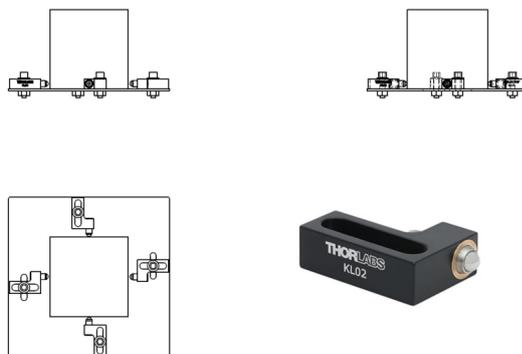


Figure 3.10: Blocking with interference solution

- Second option involves fixing the host CubeSat by means of L-shaped blocking system, which therefore requires special pods on which to be locked, and

consequently requires the design of an external case equipped with pods as it is shown in figure 3.11.

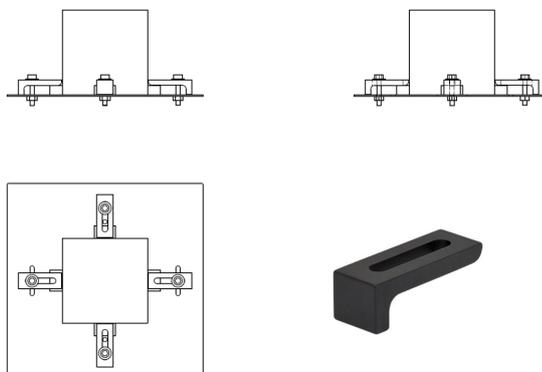


Figure 3.11: Blocking with L-shaped solution

Although the direct-interference solution on the case is the one that allows the most flexibility for the external users of the facility (who have no constraints other than submitting a case of the specified dimensions), L-shaped pods blocking system was implemented. This choice was guided by the greater ease with which the parts could be manufactured, including by means of 3D printing, thus making it possible to lighten the component, as well as reducing the total cost of the system itself. This solution, however, requires the facility user to adapt an external case specifically designed to fit with the locking solution. The pods mounting system, consisting of a through-screw and bolt and the pods itself, allows for variable positioning of the CubeSat on the base plate. This will be of fundamental importance during the rough, hand-made balancing procedure, which will be introduced in detail in the following chapters of this discussion.

There are four pods implemented in the facility, one on each side of the CubeSat. The weight of the single pod is $15[gr]$, including screw and bolt. This results in a total weight of the fastening system of $30[gr]$.

Hosted CubeSat

For now, the system under test will be treated as a black box: it will be studied in detail in the following section. The development of the subsystem's facility and interface is in fact unrelated to what is inside the case containing the subsystem. The requirements imposed are those of the maximum size of the external case, which must not exceed 105×105 mm, while keeping the weight as low as possible. In addition, locking pods must be attached to the outer case. The final weight of

the hosted CubeSat is **467 grams**, which constitutes 7% of the overall weight of the mobile testing facility component.

Mass Distribution

The total mass distribution of the 'satellite', i.e. the full system just illustrated (air bearing, base plate and host CubeSat), is shown below 3.12.

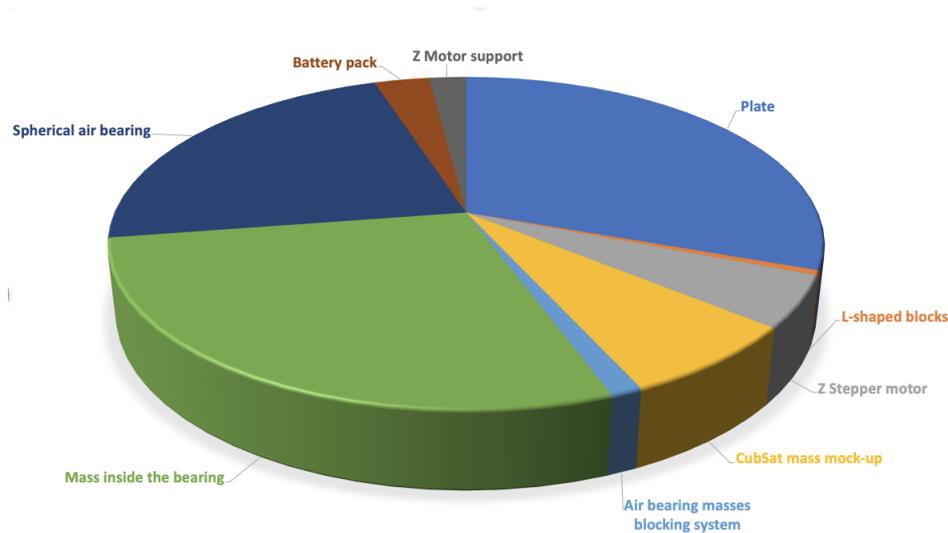


Figure 3.12: Mass distribution on the satellite

As can be seen from the graph, the most influential components are the air bearing with the additional weights, the base plate with all the electronic components and the two stepper motors responsible for movement in the x and y axis implemented. The masses, in kilograms, of the components just described are shown in the table 3.3.

3.2 Design of the interface system

3.2.1 CubeSat prototype

In this section, it is studied in detail the realisation of the host CubeSat, i.e. the structure containing the functioning subsystem of ADCS that was the focus of the tests. This component has its own section, since up to now it has been referred to as a component to be mounted on the facility, but it is necessary to focus on the complete development of the subsystem to make it testable. Initially, a mass

<i>Plate</i>	1.961
<i>L-shaped blocks</i>	30
<i>Z Stepper motor</i>	322
<i>CubeSat mass mock-up</i>	467
<i>Air bearing masses blocking system</i>	79
<i>Mass inside the bearing</i>	1.811
<i>Spherical air bearing</i>	1.444
<i>Battery pack</i>	191
<i>Z Motor support</i>	129
<i>Final satellite weight</i>	6.434

Table 3.3: Distribution of masses of the satellite

mock-up of the CubeSat had to be made to be able to calculate the residual torque of the facility and to verify the correct functioning of the *satellite* components. This dummy model was implemented by including the **ADCS board**, **functional board**, and **structure**.

ADCS Board

It was decided to test an already complete ADCS board in the ownership of the CubeSat student team at the Politecnico di Torino. Since a duplicate of the board itself was available, it was implemented directly in the mock-up. The subsystem board involves the implementation of:

- Drivers for the powering of magnetic actuators
- Connector
- Microprocessor
- Inertial platform

This platform is therefore the final one, which will be used during the final test phase for the verification of magnetic actuation, once the control law software has been loaded.

Functional Board

The ADCS system tested needs to be independent in terms of power supply and data transmission to the ground. This need arises from the limitation imposed by the type of facility, which allows only the mechanical interface with the base

platform. Like in nominal orbital operating condition, therefore, the system must be autonomous. A *functional board* is therefore developed, having the indispensable functions of **powering the ADCS board** and **data communication** via Wi-Fi. The choice of perform communication via Wi-Fi means that there is no need for USB component implemented on the board and thus meets the key requirements considered for the CubeSat design, i.e minimal weight and maximum accessibility. Figure 3.13 shows the high-level functions from which the functional analysis and design of the board started. The physical components required to accomplish the

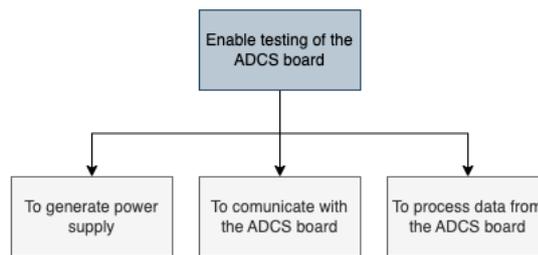


Figure 3.13: High level functions of the functional board

above-mentioned functions were analysed and chosen. It was decided to adopt the elements shown in the table 3.4, where the function performed can be found for each of them. Once the type of component to be used were determined, commercial components has been selected from the options available on the market. The **block diagram** of the board was then produced, as depicted in the figure 3.14. The buses used for IMU data communication with the functional board microprocessor are: *CAN Bus, I2C, SPI, RS232*. Starting from the block diagram, an external party was commissioned to produce the schematic and print the board.

Mechanical interface Structure

From the mechanical interface conditions imposed by the facility, the structure of the two subsystem boards was designed developing a Solidworks 3D model. An initial solution was to create the interface in one piece, so that it that could contain the functional board and the ADCS board, while maintaining dimensions consistent with the CubeSat 1U standard. The first design iteration is shown in figure 3.16. The case has a total volume of 500 cm^3 . To minimize the weight of the structure, a hollow case with 2 mm thick walls was realised with an in-house 3D printer. To have a case that could interface with both initially assumed mounting solutions, it was decided to implement external removable pods. In this way, the fixing system can be easily changed from the configuration with L-shaped pods to that with interference on the structure itself. To maintain a low weight, it was decided to

Component	Function
<i>Batteries 7.4V</i>	power generation
<i>PI0-W processor</i>	data processing/OBC functions: handles telemetry data relating to the status of components on the base board, first processing data from the ADC board
<i>Hardline connector</i>	For uploading software to the microprocessor
<i>Analogue to digital converter</i>	Conversion of analogue battery data into digital data to be sent to the microprocessor
<i>Connector 104</i>	To power the ADCS board with 5V and 3.3V, as well as data communication with the microprocessor
<i>3.3V regulator circuit</i>	power supply required for ADCS components requiring such voltage
<i>5V Regulator circuit</i>	power supply required for ADCS components requiring such voltage
<i>SD memory 2Gb</i>	For redundancy and memory expansion integrated in the microprocessor

Table 3.4: Function board components and their functions

print the proposed model in PLA, a material that provides sufficient structural strength with low density, resulting in the minimum structural weight.

Once the case was printed, it has been assembled with the two boards of the subsystem to obtain a mass mock-up of the host CubeSat. Having a mass mock-up allows to perform the tests relating to the calculation of the inertial characterisation, computation of the residual torque of the facility, minimization of the residual torque and thus the characterization of the test facility to be carried out during the software development period for the subsystem boards.

The first step in the assembly procedure is to secure the functional board to the structural base. For this purpose, spacers of sized 1.7 mm are implemented to accommodate the battery pack situated below the board itself. The subsystem board is then fastened above the functional one, using 1.7 mm spacers.

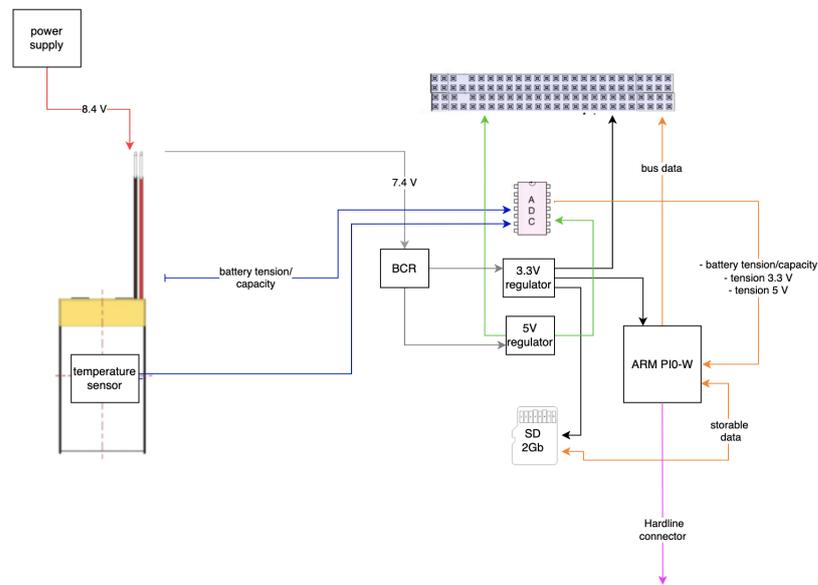


Figure 3.14: Functional Board Block Diagram

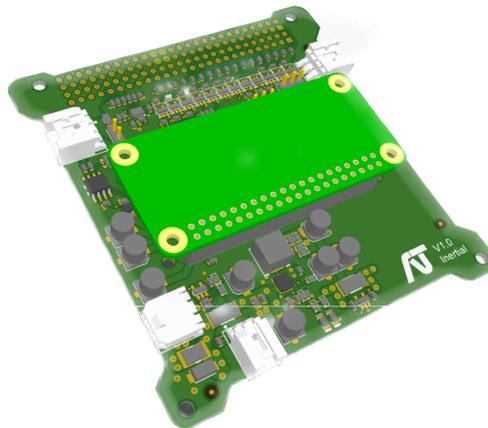


Figure 3.15: Functional Board

Finally, a reversible connection is made for the pods, which are secured at the structure by means of M2 screws fixed with threaded drill holes in the case. This first configuration results in an overall weight of **340 g**, including the components of the two board and the actuator.

From the realisation of this first structural solution, problems arose relating to:

- Fixing of the components: it was very difficult to screw the various components

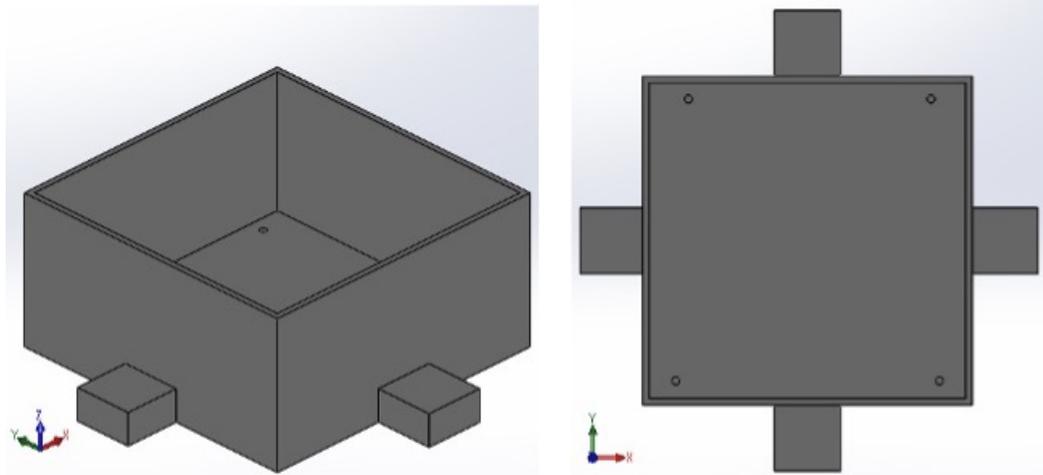


Figure 3.16: Initial structural solution for the host CubeSat

together due to the small size of the structure and the one-piece design

- The CM of the system is rather high due to the large spacers used for both functional and system board fixing
- The dimensional tolerances of the base plate were not considered

A second design iteration was therefore carried out, in which the modularity of the testbed components was implemented to have easier access for battery charging and component assembly. A solution was also devised to reduce the number of spacers integrated, with a consequent reduction the overall structure weight. At the same time, the center of gravity of the CubeSat was lowered to have greater stability during the balancing procedures.

These results were obtained by constructing a case made of two parts: an external one with pods (removable if required) and an internal one. The external part of the structure allows the actual mechanical interface with the base plate of the Bologna facility, while the internal structural part contains the subsystem components. The results of the second design iteration are reported in figure 3.17, 3.18, 3.19 and 3.20. It can be seen how accessibility is increased by the fact that the outer structural part is only screwed to the inner part. This solution make it easy to perform operations on the system board such as battery charging or assembly operations. Furthermore, by directly attaching the functional board to the PLA risers pods, the overall weight of the structure can be reduced as the number of spacers implemented is lower.

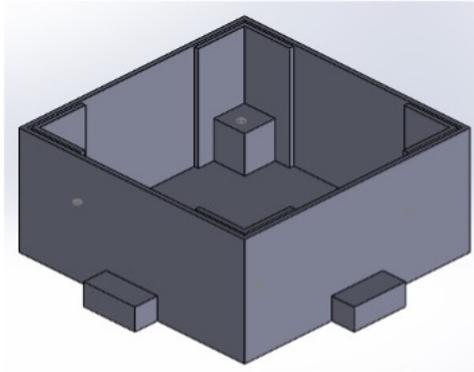


Figure 3.17: Isometric view

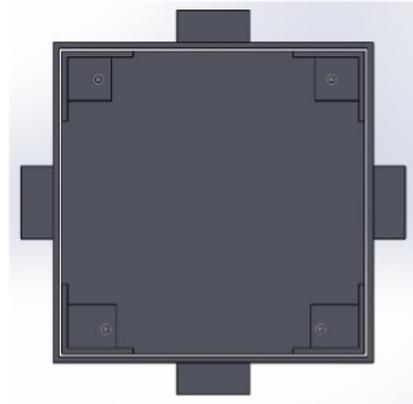


Figure 3.18: Top view

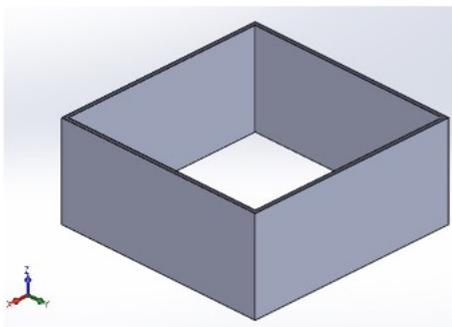


Figure 3.19: External case

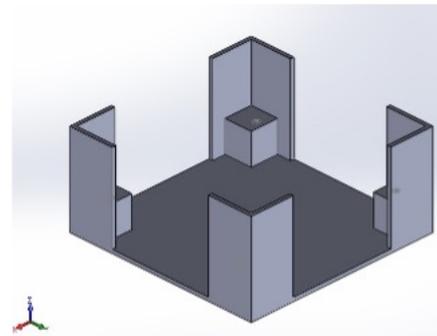


Figure 3.20: Internal structure

Together with the facility developers in Bologna, it was decided to realise an interface with slightly larger dimensions than the CubeSat standard ones, to have an easier housing for the boards.

For the latest structural design, two mass budgets were carried out to identify areas of possible improvement in terms of mass reduction. In table 3.5, are listed the components present within the mass mock-up, with their weights. The total weight of the CubeSat is **467 gr**. This weight is mainly attributable to the battery pack, which accounts for 31% of the total, and the structure, which accounts for 29% of the total weight. In order to decrease the overall mass of the CubeSat, it is therefore possible to act on two fronts: a lightening of the structure, a change in the power supply strategy. Relatively to the structural weight reduction, the component that makes up the largest percentage of it is the internal base on which the two boards set.

Component name	Total mass [gr]
<i>Structure</i>	130,3
<i>Base board</i>	70
<i>Spacer 17 mm</i>	10,4
<i>Spacer 6 mm</i>	3,6
<i>Screw M2</i>	1,2
<i>E-STAR ADCS board</i>	76
<i>battery</i>	140
<i>M2 bolts</i>	5,5
<i>MT</i>	30
Total weight	467

Table 3.5: Distribution of masses between the components of the mass mock-up

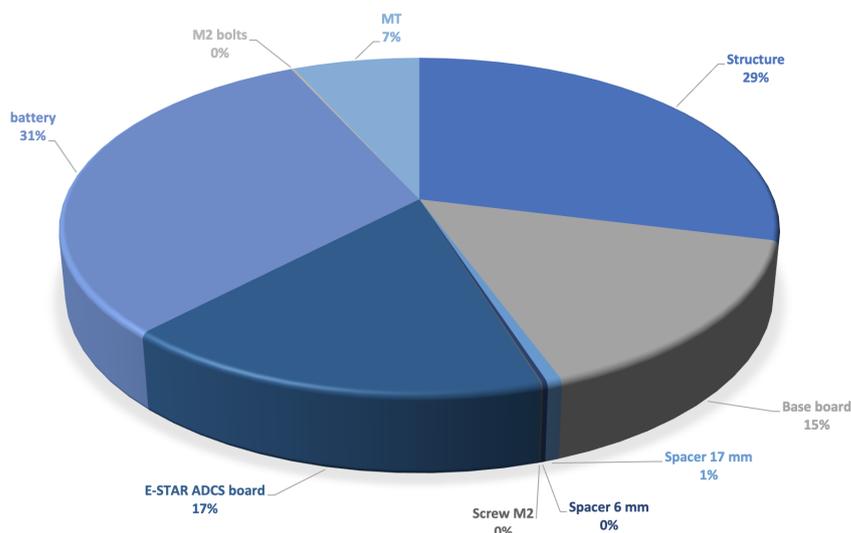


Figure 3.21: Mass distribution for the mass mock-up

To reduce its mass, a design with smaller base pods can be considered, keeping them compatible with the allocation of the battery pack, in conjunction with a lower side wall height, keeping it under the 5cm limit. With regard to the thickness of the walls, those could be reduced to 1.5mm, considering that the priority remains to have structural integrity. As far as the variation of the power supply system is concerned, one could opt for the use of thin battery packs, to be able to allocate them between the two boards and at the same time reduce the weight of the battery and the structure (which would need smaller base elevations). This solution has

the only constraint of providing a reduced power supply autonomy to the system. The final configuration of the host CubeSat in figure 3.22, and of the satellite in its final configuration in figure 3.23 are shown below.



Figure 3.22: Final mass mock-up

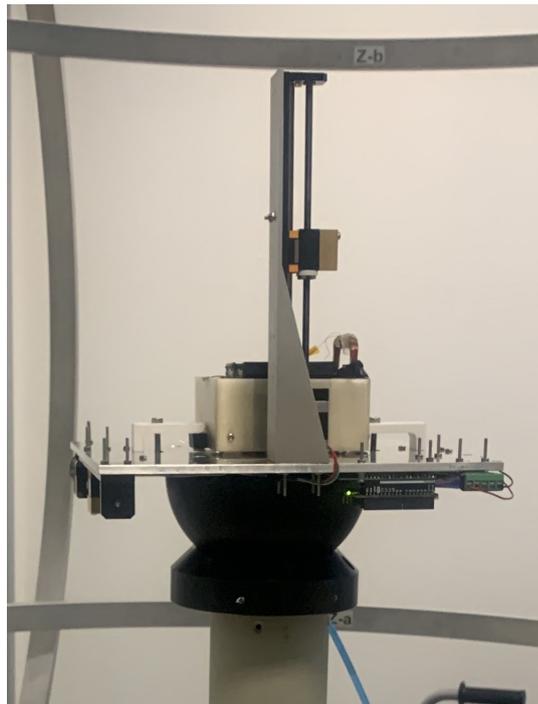


Figure 3.23: Satellite components of the facility

Chapter 4

Integration, calibration, and verification

This section of the thesis proposes a detailed analysis of the **calibration, integration** and **verification** process followed to get the testing system together, preparing it to perform the balancing procedures described in chapter 5. An initial verification of the purchased commercial components is then carried out. Once the appropriate corrections arising from the verification tests have been made, the mechanical and electronic integration of the components takes place. Following this procedure, the fully integrated system is verified and validated.

The process just described can be introduced into the V model of system engineering, which is represented in figure 4.1. There are two phases of system engineering:

- **Design phase:** top-down, its purpose is to start from the stakeholders' needs, translate them into requirements and arrive at a final system that meets the customer's needs. The process is reported as top-down because the requirements, initially functional requirements at high-level, are defined with increasing detail as they are allocated from the general function to the system, continuing to the subsystem and finally to the components;
- **Integration and Verification phase:** bottom-up, its purpose is to verify and integrate the components defined in the design phase. The process is defined as bottom-up because it envisages that verification of compliance with the design requirements has to be conducted starting from the components requirements of the subsystems, proceeding with the integration of the verified components into subsystems and then integrating the subsystem into system. The verification of compliance with the requirements is an iterative procedure;

It is in the second part of the V model that this chapter is introduced.

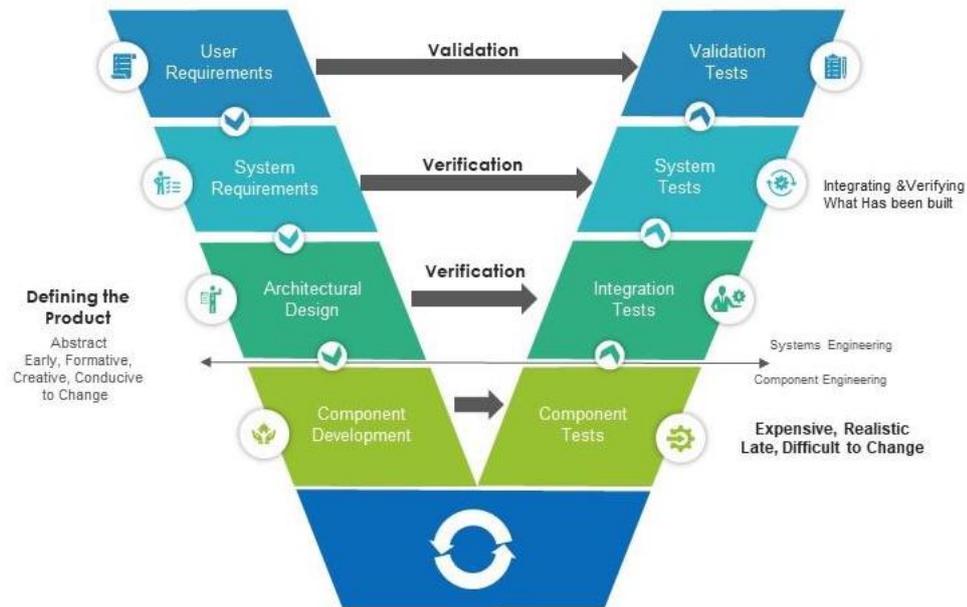


Figure 4.1: System Engineering V-model [30]

4.1 Verification

The verification procedure is used to ensure that the component operates in accordance with the declared operating specifications. As a result of the design phase, the components necessary to fulfil the functional requirements of the testing platform were selected.

4.1.1 Stepper Motors and Motor Drivers

As introduced in the 3.1.3 section, the type of motors used are bipolar stepper motors, with technical characteristics shown in table 3.2. A *motor drive* must be used to guarantee the control of the stepper motor operation.

'Stepper motors' or 'steppers' motors are synchronous electric motors whose feed rate is a single step corresponding to a specific angle of rotation, so they can be rotated at a precise angle at pleasure. A stepper is capable of precisely maintaining rotation speed and position without the need for feedback transducers such as tach generators or encoders. Compared to other types of electric motors, the stepper motor does not change the speed of rotation depending on the torque to which

the rotating shaft is subjected, it keeps it constant and if the torque increases, the motor stops. To be driven, stepper motors require a sequence of pulses generated by special electronic control circuits. The stepper motor is also referred to as an electromechanical transducer in that it converts electrical command impulses into elementary step (steps) of a fixed angular amplitude depending on the type of motor. To operate, this type of electric motor must be connected to a specific power supply circuit and an electronic control system capable of governing the angle and speed of rotation. As with all electric motors, the stepper motor also consists of a fixed part called the stator and a moving part called the **rotor** to which a rotating shaft is connected. The stator contains copper windings, consisting of enamelled (insulated) copper coils, while the rotor consists of magnetic material. The stator windings are connected in such a way as to create a two-phase system. Bipolar stepper motors have the current in the stator windings flowing in both directions.

The wires constituting the two poles are distinguished by different colours and the stepper motor control must be carried out by a **bipolar driver**. The excitation sequence will be realised by a *microcontroller*. The bipolar driver can be compared in a nutshell to 8 switches: i.e. 4 switches for each stator coil as pictured in figure 4.2.

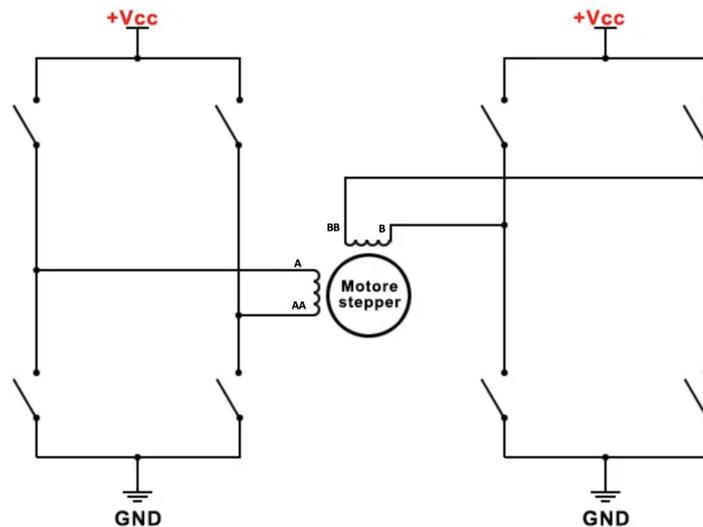


Figure 4.2: H-bridge microcontroller scheme

Electronically, the bipolar driver consists of a double H-bridge electronic circuit which allows the direction of the current in the stator coils to be reversed. The switches in the previous circuit are realised with mosfet or bjt transients. By polarising pairs of transients for each H-bridge, it will be possible to supply the

coils with different polarities. Using the two digital outputs of a microcontroller, by sending HIGH and LOW values (5V and 0V voltages respectively) to pins IN1 and IN2 it is possible to control the polarisation of the transistors and the consequent direction of the current flowing in the motor's stator coil. When there is a HIGH value at the IN1 point and a LOW value at the IN2 point, transistors **Tr4** and **Tr2** will be in conduction, causing the motor to rotate in one direction, as seen in figure 4.3 image D. When at IN1 the signal is LOW and on IN2 the signal is HIGH the transistors that will go into conduction will be **Tr1** and **Tr3**, causing rotation in the opposite direction to the previous step, as shown in figure 4.3 C [31].

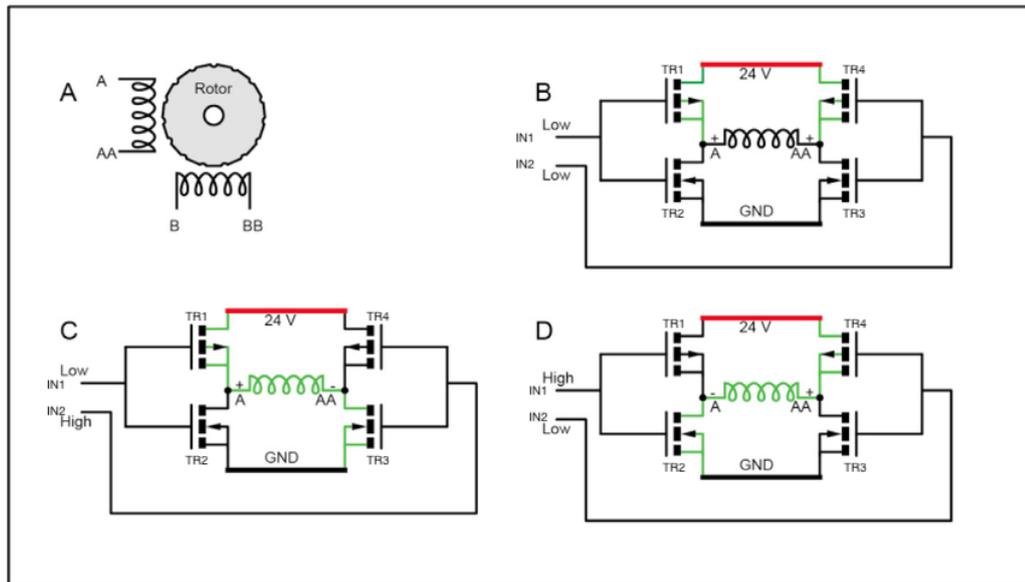


Figure 4.3: H-bridge functioning sequence

The circuit connection mode prevents the transistor pair from being powered simultaneously, thus preventing short circuits. The motor driver initially selected for this application is the **Adafruit 2448**.

Verification board implementation

Once the working principle of stepper motors and motor drivers has been introduced, it is time to proceed to *stepper motor verification*. To this end, a power supply circuit must be created, and a control algorithm implemented to verify and control the stepper motor. An Arduino-Uno board was used, connected to the driver, which was initially fixed on a breadboard and then connected in turn to the stepper motor. It is essential to include a voltage regulator in the power supply circuit. It takes a voltage of up to 21V as input and outputs 5V voltage with a maximum current of 1.5 [A]. The voltage regulator selected for the present application is the

Adafruit MPM3610 Breakout, which runs at 2-MHz fixed-frequency in PWM mode. The final configuration is shown in figure 4.4.

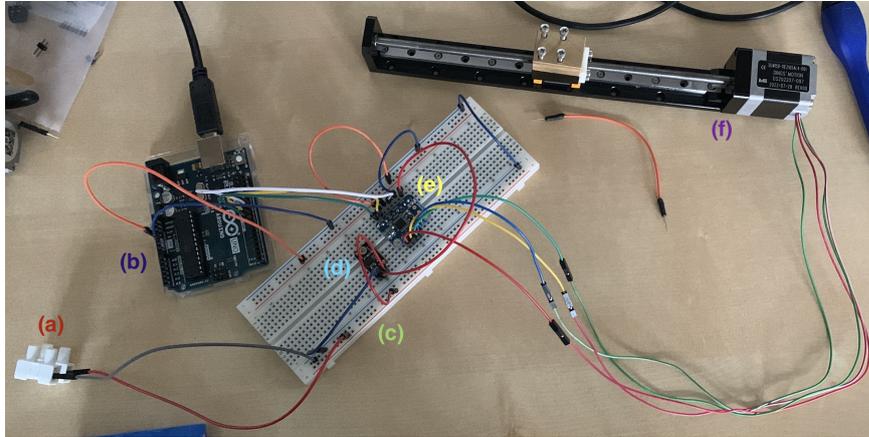


Figure 4.4: Arduino Uno set up for stepper motors verification

In the figure 4.4, it is indicated:

- (a): connection for power supply, via 7.4V battery pack;
- (b): Arduino Uno board, on which the verification software is loaded;
- (c): Breadboard, used to have a reversible electrical connection;
- (d): Voltage regulator to adjust the supply voltage;
- (e): Motor Driver, for adjusting the voltage inputs to the stepper motor;
- (f): Stepper motor under verification;

Verification procedure

Once the circuit was implemented, a dedicated software was written to verify firstly the correct functionality of the stepper motor, and secondly its operation characteristics at variable speeds. The purpose of this second operation is to find the most appropriate speed range of the motor, that will be subsequently implemented in the balancing code.

Verification Code

In accordance with the Arduino guidelines for testing stepper motors, a script for testing stepper motors was implemented by means of **AccelStepper.h** library, imposing the initial values for maximum speed with *Stepper.setMaxSpeed*, maximum

acceleration with *Stepper.setAcceleration*, imposed speed with *Stepper.setSpeed* and final position in terms of number of steps with *Stepper.moveTo*.

The first results of the functional test reported no rotation of the stepper motor, with intense vibrations. It was therefore necessary to proceed with an analysis of the stepper motor datasheet [32], focusing on the sequence of HIGH and LOW inputs required for rotation in clockwise sense, reported in 4.5.

EXCITING SEQUENCE				
CW	→			
Red A+	+		-	
Red/wht A-	-		+	
Green B+		+		-
Gm/wht B-		-		+
CCW	←			

Figure 4.5: Voltage sequence for DINGS' stepper motors

In accordance with the given pulse sequence, the set up in the library *AccelStepper.h* was varied. Downstream the proposed corrections, the motor rotates but still shows excessive vibrations. Stepper motor vibration can be decreased in a variety of strategies. The methods can generally be divided into **mechanical** and **electrical** dumpings, with the electrical category including the example of current ripple reduction. A starting point to solve the motor vibrations problem was going through the most popular electrical noise and vibration reduction techniques.

- **Resolving vibrations independently:** by addressing each vibration type separately. In fact certain drivers can deliver low vibration performance at all speeds. Individual phase current detectors make it possible for the driver to control current exceedingly well and guarantee a highly uniform waveform;
- **Microstepping driver:** The best recommendation for the reduction of vibration in stepper motors is microstepping. Here, the motor's current is divided between its phases using drivers. As a result, vibration, torque ripple, and audible noise are significantly diminished. This enables the driver to accurately manage the amount of torque and the angular position;

For the present application, it was decided to apply the second methodology, carrying out microstepping implemented via the fast and slow decay functions of the motor driver. However, the Adafruit 2448 Motor Driver does not provide for this functionality, so a different driver had to be selected. After careful analysis of

different drivers datasheets, it was decided to use the **Adafruit DRV8833 Motor Driver**. The excitation sequences for the two motor drivers in 4.6 is compared.

Input				Output		
IN1	IN2	PWM	STBY	OUT1	OUT2	Mode
H	H	H/L	H	L	L	Short brake
L	H	H	H	L	H	CCW
		L	H	L	L	Short brake
H	L	H	H	H	L	CW
		L	H	L	L	Short brake
L	L	H	H	OFF (High impedance)		Stop
H/L	H/L	H/L	L	OFF (High impedance)		Standby

xIN1	xIN2	xOUT1	xOUT2	FUNCTION
0	0	Z	Z	Coast/fast decay
0	1	L	H	Reverse
1	0	H	L	Forward
1	1	L	L	Brake/slow decay

Figure 4.6: On the right is the voltage sequence proposed by the MD 2448, on the left that of the MD 8833

As can be seen in figure, via MD 8833 it is possible to vary the type of decay, which can be either *fast* or *slow*. This possibility is not provided by MD2448. Stepper motors rotate by 200 steps per revolution, or 1.8° , on average. When it is needed to make small motions, this could be an issue. With micro-stepping, users can have smaller movements and more than 200 steps per rotation. The stepper motor driver output appears as a square signal and results in jerky movements when controlling stepper motors with complete steps. The output signal more closely resembles a sine wave and the stepper motor runs more smoothly as the micro-stepping size increases. However, there is a drawback: the *torque decreases significantly* as the micro-stepping value rises, and if the value is too high, it is possible that the motor won't have enough torque to turn at all. Typically, values of 1/4, 1/8, or even 1/16 can still create smooth movements, producing still enough torque, in figure 4.7 it is cited [33].

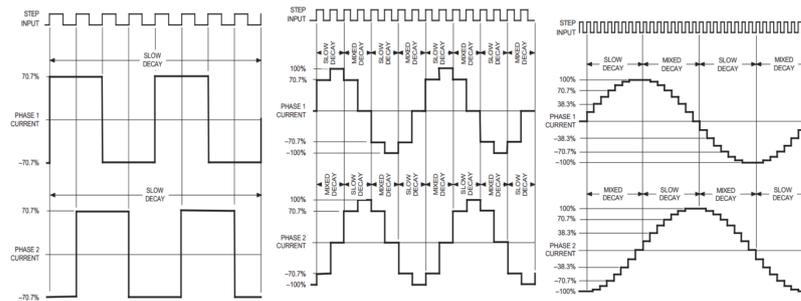


Figure 4.7: Current waves while increasing micro-stepping

In figure 4.7 is represented how the waves of current change while using micro-stepping: it can be seen curves are moving towards increasingly smooth, with sinusoidal trend.

In MD DRV8833 the inputs can be utilised to control the motor speed using PWM as well. The inductive nature of the motor necessitates that the drive current continues to flow while controlling a winding with PWM even if the drive current is interrupted. The term associated with this type of current is *recirculation current*. The H-bridge can function in either a fast decay or a slow decay mode to manage this recirculation current. In slow decay mode, the motor winding is shorted; in fast decay mode, the H-bridge is disabled and recirculation current passes via the body diodes. When employing slow decay, one xIN pin is held high while the PWM signal is applied to one while the other is held low. In figure 4.8 are reported the **PWM Control of Motor Speed**.

xIN1	xIN2	FUNCTION
PWM	0	Forward PWM, fast decay
1	PWM	Forward PWM, slow decay
0	PWM	Reverse PWM, fast decay
PWM	1	Reverse PWM, slow decay

Figure 4.8: PWM Control of Motor Speed

Figure 4.9 shows the current paths in different drive and decay modes [34].

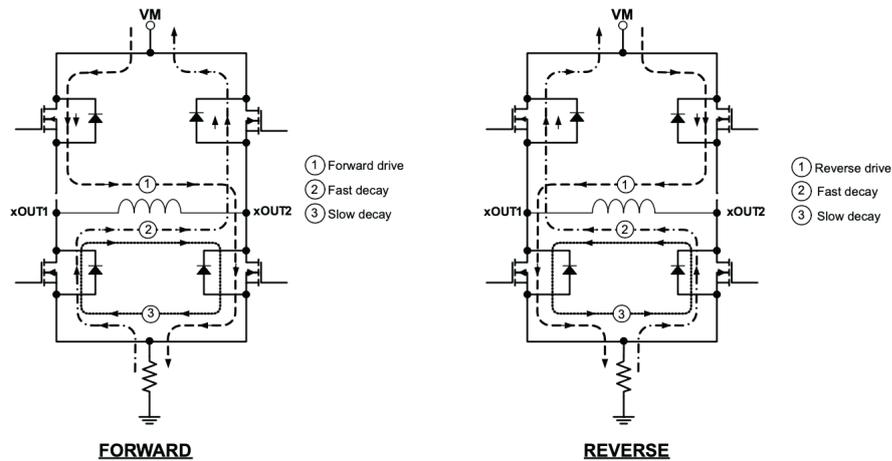


Figure 4.9: Current paths in different drive and decay modes

Once the motor driver has been replaced, a simplified Arduino script was first implemented to test the motor rotation only. Following the positive result of the test, operation with the original **AccelStep.h** library was verified, and it was seen that the motor functioned correctly and vibrations were significantly reduced: the implemented solution was suitable for achieving the desired result. Once the operation of the motors had been verified and optimised, the stepper's characteristic speed range was investigated.

An **empirical** study of the ranges of *acceleration* and *speed* that allow for the greatest possible reduction in vibration, while avoiding stall, was conducted. The results are shown in the diagram in figure 4.10.

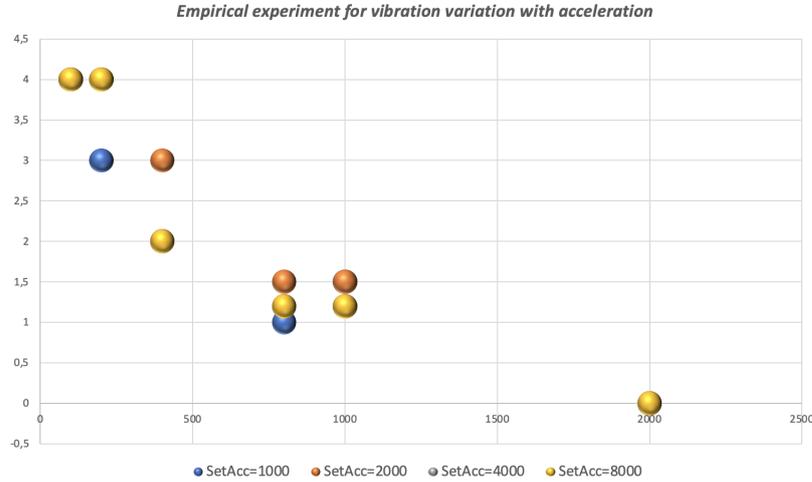


Figure 4.10: Curves representing the change in vibration intensity as the imposed speed and acceleration vary

The vibration level was assessed qualitatively by assigning a value from **0**, corresponding to a stall condition (the shaft no longer rotates), up to **4**, corresponding to an extremely intense vibration condition and difficulties in rotation. The test was carried out for four different acceleration values: i.e. 1000, 2000, 4000, 8000 [reps/s²]. Variable velocities were considered for each tested acceleration, in particular velocities of: 100, 200, 400, 800, 1000 and finally 2000 [reps/s]. The analysis shows that for each acceleration tested, the stall occurs at a velocity of 2000 [reps/s]. It can also be seen that vibrations decrease with the increase in speed, more markedly for higher speeds. It was performed the same type of test as above, but with increasing in the accelerations level up to 40000 [rep/s²]. A condition of decreasing vibrations was found up to a value of 20000 [rep/s²], and then stabilised. A velocity of 1290 [reps/s] results to be the maximum speed without stalling.

The purpose of the empirical evaluation is to identify the most appropriate range, in terms of smoothness of rotation and minimization of vibrations, that can be included within the Arduino script to be loaded into the Feather M0 (base plate microprocessor) to realise the updated platform balance. The speed selected for this purpose is 1000 [rep/s], while the acceleration selected is 20000 [rep/s²] as it was verified to be an acceleration compatible with the implemented operations. Even though it is recognized that the empirical vibration index was assigned in a purely subjective and qualitative manner, it was, nonetheless, the only tool available for

the above purpose.

4.1.2 CubeSat boards verification

Regarding the functional verification of the boards constituting the system under test, a greater level of detail is required on the development methodology implemented and their functionalities. These topics will be dealt with in detail in the chapter 6 of this paper, in which a general context will be given regarding the magnetic implementation and the functioning of the ADCS and basic functional board, and the verification tests adopted will be described in detail.

4.2 Integration

After having verified the functionality of the single components, the implementation of the testing platform is carried out. This process can be divided into **electronic integration** and **mechanical integration**.

Electronic integration

Initially, the electronic components to be integrated underneath the base-plate were integrated. The electronics to be included are reported in figure 4.11.

In figure are indicated with:

- (a): Feather M0
- (b): Arduino Protoshield Rev3
- (c): Voltage regulator 1
- (d): Voltage regulator 2
- (e): Voltage regulator 3
- (f): Voltage regulator 4
- (g): Motor Driver 1
- (h): Motor Driver 2
- (i): Motor Driver 3
- (l): Stepper motor 1
- (m): Stepper motor 2

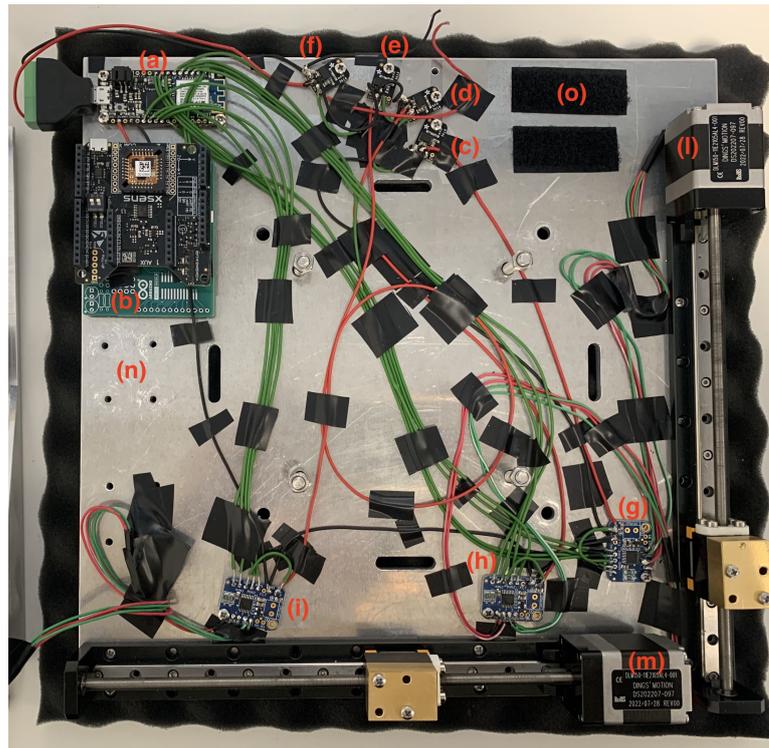


Figure 4.11: Electronic scheme

- (n): Stepper motor 3
- (o): Battery accommodation

The sequence of connections followed to make the power supply circuit for all components is outlined below.

Electrical connections

1. Connect power (+) to all V_{in} of the four Voltage Regulator
2. Connect power (-) to all GND of the electronic components listed above
3. Connect Voltage regulator 4 exit output 5V to Feather M0 USB input
4. Connect Feather M0 to Arduino Protoshield:
 - (a) Connect Feather M0 3V to Arduino Protoshield 3V
 - (b) Connect Feather M0 GND to Arduino Protoshield GND
 - (c) Connect Feather M0 SLC to Arduino Protoshield SLC

(d) Connect Feather M0 *SDA* to Arduino Protoshield *SDA*

5. Connect Stepper Motor 1:

- (a) Connect voltage regulator 1 *5V* to the stepper driver 1 *VM*
- (b) Stepper Motor 1 *A+* (Red wire) to Motor Driver 1 *AOUT1*
- (c) Stepper Motor 1 *A-* (Red and white wire) to Motor Driver 1 *AOUT2*
- (d) Stepper Motor 1 *B+* (Green wire) to Motor Driver 1 *BOUT1*
- (e) Stepper Motor 1 *B-* (Green and white wire) to Motor Driver 1 *BOUT2*
- (f) Motor Driver 1 *AIN1* to Feather M0 Pin #10
- (g) Motor Driver 1 *AIN2* to Feather M0 Pin #11
- (h) Motor Driver 1 *BIN2* to Feather M0 Pin #12
- (i) Motor Driver 1 *BIN1* to Feather M0 Pin #13
- (j) Motor Driver 1 *SLP* to Motor Driver 1 *VM*

6. Connect Stepper Motor 2:

- (a) Connect voltage regulator 2 *5V* to the stepper driver 2 *VM*
- (b) Stepper Motor 2 *A+* (Red wire) to Motor Driver 2 *AOUT1*
- (c) Stepper Motor 2 *A-* (Red and white wire) to Motor Driver 2 *AOUT2*
- (d) Stepper Motor 2 *B+* (Green wire) to Motor Driver 2 *BOUT1*
- (e) Stepper Motor 2 *B-* (Green and white wire) to Motor Driver 2 *BOUT2*
- (f) Motor Driver 2 *AIN1* to Feather M0 Pin #9
- (g) Motor Driver 2 *AIN2* to Feather M0 Pin #6
- (h) Motor Driver 2 *BIN2* to Feather M0 Pin #5
- (i) Motor Driver 2 *BIN1* to Feather M0 Pin #A5
- (j) Motor Driver 2 *SLP* to Motor Driver 2 *VM*

7. Connect Stepper Motor 3:

- (a) Connect voltage regulator 3 *5V* to the stepper driver 3 *VM*
- (b) Stepper Motor 3 *A+* (Red wire) to Motor Driver 3 *AOUT1*
- (c) Stepper Motor 3 *A-* (Red and white wire) to Motor Driver 3 *AOUT2*
- (d) Stepper Motor 3 *B+* (Green wire) to Motor Driver 3 *BOUT1*
- (e) Stepper Motor 3 *B-* (Green and white wire) to Motor Driver 3 *BOUT2*
- (f) Motor Driver 3 *AIN1* to Feather M0 Pin #A1

- (g) Motor Driver 3 *A1N2* to Feather M0 Pin #A2
 - (h) Motor Driver 3 *B1N2* to Feather M0 Pin #A3
 - (i) Motor Driver 3 *B1N1* to Feather M0 Pin #A4
 - (j) Motor Driver 3 *SLP* to Motor Driver 3 *VM*
8. Connect the Xsens to the Feather M0 via the Arduino Protoshield:
- (a) Connect Feather M0 *3V* to Arduino Protoshield *3V*
 - (b) Connect Feather M0 *GND* to Arduino Protoshield *GND*
 - (c) Connect Feather M0 *SLC* to Arduino Protoshield *SLC*
 - (d) Connect Feather M0 *SDA* to Arduino Protoshield *SDA*
 - (e) Connect the Xsense to the Arduino Protoshield pins

Once all soldering has been done as listed, the battery pack is connected, and the components powering is checked.

Feather M0 Verification

To verify the correct functionality of the Feather M0 board, it is first necessary to check whether it allows the movement of the 3 stepper motors and powers the X-Sens. First a simplified software, compared to the final balancing software, is loaded on the board in order to check the correct functioning. The loaded software is the one implementing the Accellsteps.h library, with set values obtained from the range optimisation analysis. A short test campaign was carried out to verify the correct functioning of the motors and connections. Downstream from the tests, some connections were re-soldered as they did not make contact.

The second function of the Feather M0, that of powering the X-sens, was verified by connecting it to the pinout of the Arduino Protoshield board. The green power LED is switched on, so the power supply is successful.

The third and final functionality to be checked for the Feather M0 board is to realise communication to and from the Workstation. It is necessary that data from the X-sens are supplied to the whitening software to be entered, and that mass movement commands can be received to dynamically control the stepper motors. To realise, and verify, such a connection, it is necessary to connect the **Neufbox** modem Ethernet cable to the WorkStation PC. Once connected, it is necessary to launch the desktop software for balancing. Once the user interface is open, the IP generated by the modem is entered. After that the correct IP has been set, press the 'Connect' button: one is now remotely connected to Feather M0. Successful connection is indicated by the interface window updating with data from the IMU.

X-sens Verification

Once the full condition of the Feather M0 has been assessed, the X-sens is verified. The IMU used is an Xsens MTi-3, which gives magnetic field readings, orientation data, and angular velocity data. The specifications of the selected component are reported in table 4.1.

Sensors Performance	
Sensors Performance	0.5 deg RMS
Yaw/Heading	2 deg RMS
Gyroscope	
Standard full range	2000 deg/s
In-run bias stability	6 deg/h
Bandwidth (-3dB)	230 Hz
Noise Density	$0.003 \text{ } ^\circ/\text{s}/\sqrt{\text{Hz}}$
Accelerometer	
Standard full range	16 g
In-run bias stability	$40 \mu\text{g}$
Bandwidth (-3dB)	230 Hz
Noise Density	$70 \mu\text{g}/\sqrt{\text{Hz}}$
Magnetometer	
Standard full range	8 G
Total RMS noise	0.5 mG
Non-linearity	0.2%
Resolution	0.25 mG

Table 4.1: X-sens specifics

In order to verify the correct functionality of the IMU, it is necessary to integrate the board on the base platform. Once the connection to the graphical desktop interface has been established, via feather M0, it is possible to observe the updating of the data coming from the IMU, and to observe its functioning.

Mechanical integration

The importance of checking the electrical components before mechanically integrating the satellite was a must, as the electronic components are no longer accessible once the mechanical connections are made. Mechanical integration is then carried out according to the platform's *CAD Solidworks model*. The sequence used for integration is the following:

- Mount the stepper motor 3 (n) on the top of the platform using bolts;

- Mount the round plate on the air bearing convex hemisphere using bolts;
- Link the base-plate with the round plate using bolts;
- Mount the host CubeSat on top of the platform using L-shaped pods and bolts;
- Secure the battery in the appropriate location;

The images of the Solidwork CAD model used as reference for the integration according to the sequence above are shown in 4.12.

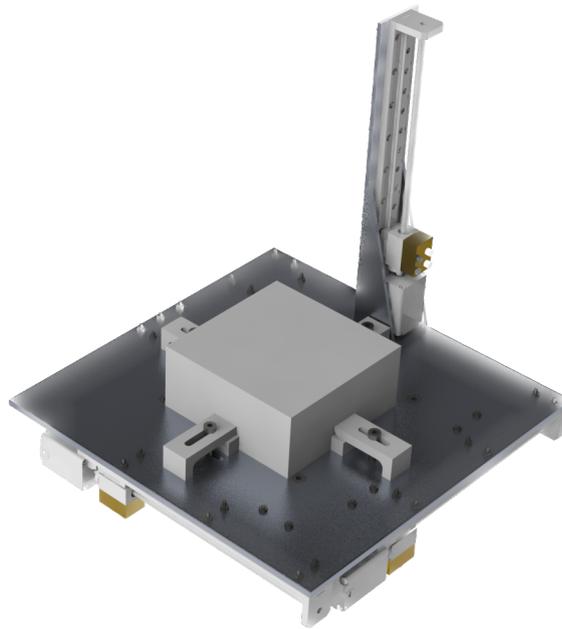


Figure 4.12: Facility CAD Solidworks global model

Below are the main views of the system to better understand what the structure is made up of. As can be seen in the figure 4.14, the location of the battery is fixed in such a way that it does not interfere with the base plate, so it will protrude from the base plate. A possible solution is to fix the battery at the top of the base plate.

During mechanical implementation, it is essential to make extremely firm connections. Care must be taken to have connections that cannot vibrate. In fact, even minimal vibration would cause a disturbance that would increase the residual torque of the facility.

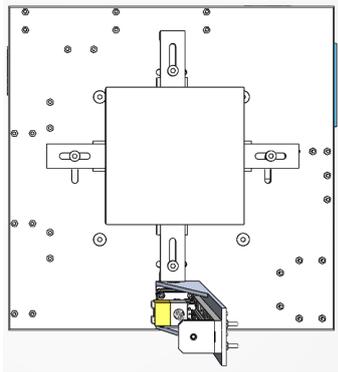


Figure 4.13: Upper view

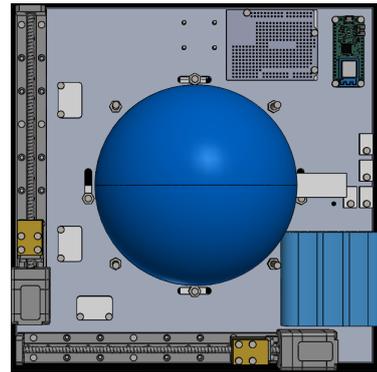


Figure 4.14: Bottom view

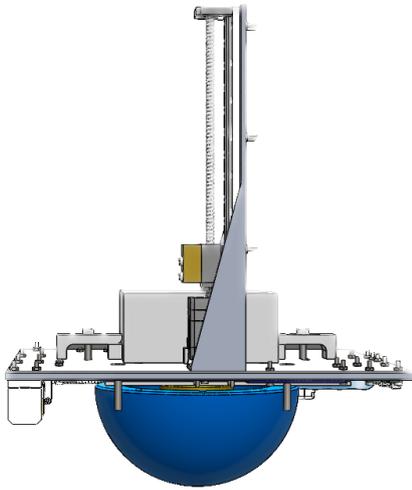


Figure 4.15: Lateral view

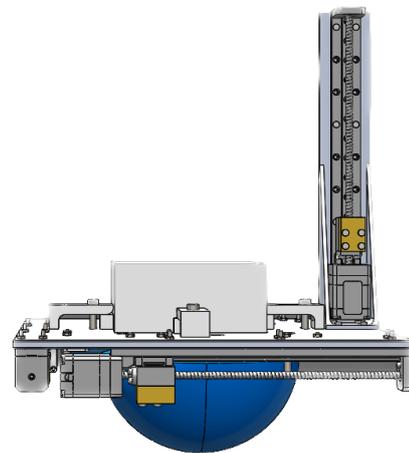


Figure 4.16: Frontal view

4.3 Validation

During validation, the user verifies that a process's outputs adhere to a predetermined standard. Although it could also be necessary to comply with internal standards, this requirement is normally enforced by an outside agency. In this application case, it is necessary to comply with internal operating requirements.

To determine the internal requirements, the starting point is the functional analysis relative to the testing platform. A high-level functional tree is shown in figure 4.17, in which are depicted the **main functions** the system must accomplish.

Regarding the first high-level function, it is broken down into its sub-functions.

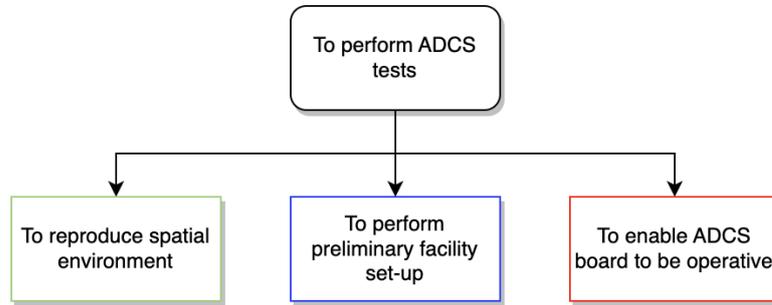


Figure 4.17: Main functional tree

This breakdown is shown in figure 4.18.

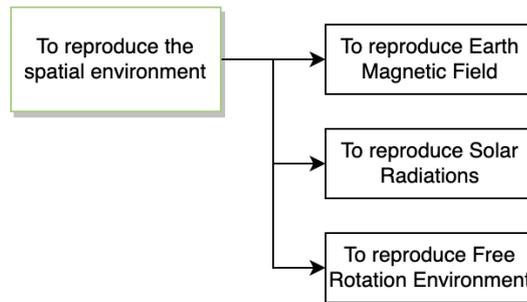


Figure 4.18: Space environment ramification

These sub-functions can be allocated to the functioning of the systems introduced in the chapter 3.1, in which it was seen which elements of the orbital environment are reproduced in the facility and the challenges these involve. As for the second function, relating to the set-up of the facility in order to be able to carry out the tests, this requires a more detailed analysis. As in the case of the first high-level function, this too is divided into sub-functions shown in figure 4.19. The requirements reported in table 4.2 were defined starting from the sub-functions defined. In order to determine the compliance with the internally defined requirements reported in table 4.2, the platform must be tested by means of balancing procedure which theory is reported in 5. More specifically, the results are examined in detail in 5.3.

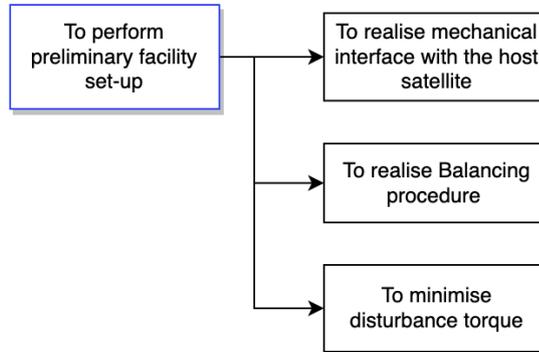


Figure 4.19: Set up ramification

Description	
Mechanical Interface	The interface shall be designed to accommodate a 1U CubeSat weighing 350 [g] according to CubeSat standard specifications
Balancing accuracy	The platform should be designed to realise balancing functions with a residual torque of 10^{-5} [Nm].

Table 4.2: Technical requirements specification related with the facility set-up

Chapter 5

Balancing Procedure

In this chapter, the satellite balancing process is analysed in detail. In particular, the mathematical basis of this procedure and the iteration followed will be discussed. Satellite balancing consists of four main stages:

- **Rough balancing:** in which the host CubeSat is secured on the platform, placed in a low-friction condition by means of the compressor, so that an initial balancing condition is achieved;
- **In-Plane fine balancing:** in this second balancing phase, the motors moving the masses are actuated using a PID control algorithm;
- **Inertia estimation:** through which the inertial characterisation of the platform is obtained. This estimation is used to calculate the offset on z axis between CM and CR;
- **Determination of disturbance torque:** through which the torque characterising the facility is determined;

The process of disturbance torque minimization is analysed in further detail, and finally a solution is proposed for decreasing torque variability due to moving cables.

5.1 Manual Balancing

Once the structure has been integrated, as seen in the chapter 4, an initial balancing called *coarse* is carried out. This definition is due to the fact that this balancing is carried out manually and in a completely qualitative manner. In fact, the procedure involves placing the integrated system on a functioning air bearing: free rotation of the system is therefore guaranteed, with minimum friction. The platform will tend to be unbalanced, and this is evident in a freely rotating condition. The

purpose of the rough balancing is to place the host CubeSat above the platform in such a position that initial balance can be achieved. The platform will not be in a horizontal position, but what is desired is that the facility assume a stable pendulum behaviour: this occurs when the CM is below the CR. A further method of achieving the above objective is to vary the vertical spacing between the upper platform and the spherical bearing, using the 4 mechanical connection screws.

The manual balancing process is therefore complete when the CM is within the reachable workspace of the ABS. The type of balancing that is carried out is to decrease the off-set between the components of \hat{z} and \hat{g} only in the direction perpendicular to \hat{g} . Physically speaking, the torque that the balancing masses are capable of producing is restricted to directions perpendicular to the gravitational field.

5.2 Automatic balancing

The automatic balancing procedure is introduced and analysed in detail in this section. The followed strategy consists of two steps:

- **1° step:** In-plane balancing realised via feedback loop, in which the off-set of two of the 3 off-set vector components is realised;
- **2° step:** vertical off-set estimation and compensation, after inertia estimation;

The procedure can be divided into 4 sub-steps, 3 of which require an iterative process.

- *In-plane balancing*
- *Inertia estimation*
- *Residual torque estimation*
- *Vertical offset balancing*

In the present discussion, the inertia tensor is considered constant during the balancing procedure, although this is not actually the case, because of the sliding masses movement, the residual torque estimation still has a satisfactory level of safety. Inertia estimation is the only non-iterative action of the process. This estimation is carried out using information obtained by subjecting the platform to free oscillation. In contrast to the inertial estimation, it results that: in-plane balancing, residual torque estimation and vertical offset balancing are highly iterative actions. In particular, in-plane balancing is performed by employing a PID (Proportional Integrative Derivative) control algorithm in order to move the

masses generating the balancing torques in the x-y plane of the platform. The residual torque is estimated downstream of the in-plane balancing. This information together with that relating to the type of pendulum (normal or inverse) is used in order to determine, and consequently compensate for, the vertical off-set between CM and CR. The iterative procedure is interrupted when variations in the estimated residual torque are no longer appreciable.

In-plane Balancing

The purpose of balancing in the plane is to align the vector \hat{e}_z with the acceleration of gravity vector \hat{g} . In order to realise the automatic balancing, a closed-loop control law of the PID type is used. As stated in [35], and reported here for completeness of the present discussion, the first step is to express a framework to be able to describe the dynamics of the system, which is assumed to be a rigid body with moving point masses. The center of the inertial coordinate reference system \mathcal{F}_i is located in CR. The inertial reference system is shown in figure 5.1.

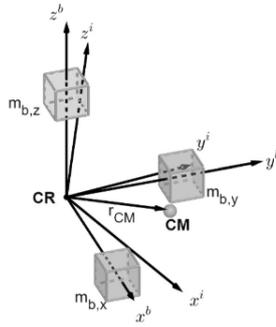


Figure 5.1: Inertial and Body frame representation

The axes of the IMU must correspond with those of the platform reference system \mathcal{F}_b also centred in the CR. Since the IMU is fixed to the bottom of the base plate, it is necessary to invert the axis \vec{e}_z : this correction is implemented in terms of control law by overwriting the IMU reference system so that the direction of the \mathbf{z} vector agrees with the direction of the local vertical. The mutual orientation of the two frames is represented by quaternions. The matrix R_b^i representing the attitude of \mathcal{F}_b with respect to the inertial frame \mathcal{F}_i is given in 5.1.

$$R_b^i = \begin{bmatrix} 1 - 2(q_2^2 + q_3^2) & 2(q_1q_2 - q_4q_3) & 2(q_1q_3 + q_4q_2) \\ 2(q_1q_2 + q_4q_3) & 1 - 2(q_1^2 + q_3^2) & 2(q_2q_3 - q_4q_1) \\ 2(q_1q_3 - q_4q_2) & 2(q_2q_3 + q_4q_1) & 1 - 2(q_1^2 + q_2^2) \end{bmatrix} \quad (5.1)$$

One can therefore express the gravity vector in body axes as $\mathbf{g}_b = R_b^i \mathbf{g}_i$. Defining the distance between the CM and the CR, in the body reference system, with

\mathbf{r}_{CM} as in figure 5.1, it is possible to calculate the residual torque due to the non-coincidence of the two centers as:

$$\boldsymbol{\tau}_{CM} = m_{tot} \mathbf{g}_b \times \mathbf{r}_{CM} \quad (5.2)$$

where m_{tot} denotes the total mass of the free to rotate system, in section 3 the elements included have been specified in detail. The torque $\boldsymbol{\tau}_{CM}$ must be balanced by the three shifting masses, which generate a pair $\boldsymbol{\tau}_b$, with expression is reported in 5.3. In order to achieve an in-plane balance, the condition that must be reached is $\boldsymbol{\tau}_{CM} = \boldsymbol{\tau}_b$.

$$\boldsymbol{\tau}_b = m_{tot,b} \mathbf{g}_b \times \mathbf{r}_b \quad (5.3)$$

As stated in [36], the effect of the three sliding masses, which fulfil the condition $m_{b,1} = m_{b,2} = m_{b,3} = m_b$, is the same as that obtained by using a single mass moving in three dimensions by a vector equal to the sum of the displacement vectors corresponding to the divided masses. The rotational dynamics of the simulator, with respect to the center of rotation, is expressed by the Euler equation of dynamics suitably adapted as in 5.4.

$$J\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times J\boldsymbol{\omega} + m_{tot} \mathbf{g} \mathbf{r}_{CM} \times R^T \mathbf{e}_3 + \boldsymbol{\tau}_u \quad (5.4)$$

Where J denotes the inertia matrix of the simulator. It consists of the inertia of the platform in the reference system \mathcal{F}_b , without considering the balancing masses, added to the inertia due to the balancing masses. The first contribution is denoted by J_s , while the second contribution is expressed by summing the cross-product matrix function of the position vector of the balance mass multiplied by the i th balancing mass. The final formulation is given in 5.5.

$$J = J_s + \sum_{i=1}^3 (-m_{b,i} [\mathbf{r}_{b,i} \times] [\mathbf{r}_{b,i} \times]) \quad (5.5)$$

Through the assumption of parallelism between body axes and principal axes of inertia, the inertia matrix is **diagonal**, and can be expressed as $J = \text{diag}(J_x, J_y, J_z)$. The effect of masses on angular momentum is assumed to be negligible as the displacement velocity imparted by stepper motors is very small. The equation of dynamics can therefore be written as in 5.6:

$$\begin{aligned} J_x \dot{\omega}_x^b + (J_z - J_y) \omega_y^b \omega_z^b &= m_{tot} (r_{CR,z} g_z^b - r_{CR,y} g_z^b) + \tau_{u,x} \\ J_y \dot{\omega}_y^b + (J_x - J_z) \omega_x^b \omega_z^b &= m_{tot} (r_{CR,x} g_z^b - r_{CR,z} g_x^b) + \tau_{u,y} \\ J_z \dot{\omega}_z^b + (J_y - J_x) \omega_x^b \omega_y^b &= m_{tot} (r_{CR,y} g_x^b - r_{CR,x} g_y^b) + \tau_{u,z} \end{aligned} \quad (5.6)$$

In the final balancing configuration, only the first two equations will be non-zero, as the term $\tau_{u,z}$ is cancelled. It is now introduced the Euler angles for the

representation of the platform attitude as $[\phi, \theta, \psi]^T$ where θ is the pitch angle, ψ is the yaw angle and ϕ is the roll angle. By assuming small angles and velocities, it is possible to linearize the equations of non-zero dynamics. In this case, the control is solved with two controllers PID_x and PID_y . These controllers are intended to stabilise the second-order equations given in 5.7.

$$\begin{aligned} J_x \ddot{\phi} &= m_{\text{tot}} \left(r_{\text{CM},z} g_y^b - r_{\text{CM},y} g_z^b \right) + \tau_{b,x} \\ J_y \ddot{\theta} &= m_{\text{tot}} \left(r_{\text{CM},x} g_z^b - r_{\text{CM},z} g_x^b \right) + \tau_{b,y} \end{aligned} \quad (5.7)$$

where the vectors $\tau_{b,x}$ and $\tau_{b,y}$ are the proportional derivative feedback of the roll and pitch angles, to which is added a term proportional to the component in the plane of angular velocity.

$$\begin{aligned} \tau_{b,x} &= -k_p \phi - k_i \int_0^t \phi dt + k_\omega \omega_x \\ \tau_{b,y} &= -k_p \theta - k_i \int_0^t \theta dt - k_\omega \omega_y \end{aligned} \quad (5.8)$$

Making the assumption of small angles, the error vector given in the output can be expressed as:

$$\mathbf{y}_e = \left[\begin{array}{cc} \frac{g_y^b}{g^b} & \frac{g_x^b}{g^b} \end{array} \right]^T \quad (5.9)$$

The vector \vec{y}_e is used as feedback in the loop. In equilibrium conditions we have $g_x^b = 0$ and $g_y^b = 0$, thus the terms $J_x \ddot{\phi} = J_y \ddot{\theta} = 0$. This implies that any in-plane unbalance, given by $(r_{\text{CM},x}, r_{\text{CM},y})$, results in a disturbance torque. This latter has to be compensated by the **integral action** of the controller. The position to be assumed by the masses is calculated from the desired equilibrium condition, such as:

$$\begin{aligned} r_x &= \frac{\tau_{b,y}}{m_{\text{tot}} g_z} \\ r_y &= \frac{\tau_{b,x}}{m_{\text{tot}} g_z} \end{aligned} \quad (5.10)$$

Inertial Estimation

The approach used to determine the inertia matrix and the offset between CM and CR is **hybrid** to those proposed so far in the literature. Typically is used a one-step approach, or a dual-step approach in which the inertia is assumed as known and then the vertical offset is estimated. In our case initially a balancing is performed in the x-y plane. By this first procedure a partial knowledge of the unbalancing vector is used to get a more accurate estimation of the components of the offset vector between CM and CR and of the inertia tensor. In the case under study, there are no actuation forces acting apart from the moving masses, so the

only acting torque is the gravitational disturbance torque. The approach used is therefore to *sample free oscillations*. The identification problem can be reported in a least-squares system described by:

$$H\mathbf{x} = \mathbf{b}(\boldsymbol{\tau}_{ext}) \quad (5.11)$$

where H is the observation matrix, $\boldsymbol{\tau}_{ext}$ is the external torque and $\boldsymbol{\xi}$, reported in 5.12, is the dynamic parameter that has to be determined.

$$\mathbf{x} = \left[\mathbf{j}^T \quad \mathbf{r}_{CM}^T \right]^T = \left[J_{xx}, J_{yy}, J_{zz}, J_{xy}, J_{xz}, J_{yz}, r_{CM,x}, r_{CM,y}, r_{CM,z} \right]^T \quad (5.12)$$

Without implementation forces, $\boldsymbol{\tau}_{ext}$ is null. If it is considered the equation updated to our operational case, i.e. $H\mathbf{x} = 0$, it is possible to estimate \mathbf{x} only up to an unknown scaling factor. In the application case under study, the scalar factor problem is not present since a partial knowledge of the unbalance vector is known from the x-y balancing, downstream of which we have $r_{CM,x}, r_{CM,y}$ null, ideally. Therefore, by applying a known displacement to the two balancing masses x and y, we determine the two vectors $r_{CM,x}, r_{CM,y}$. A matrix formulation of the dynamic system is thus derived as:

$$[\boldsymbol{\Omega}(\dot{\boldsymbol{\omega}}) + [\tilde{\omega}]\boldsymbol{\Omega}(\boldsymbol{\omega}) \mid m_{tot}[\tilde{g}]] \begin{bmatrix} \mathbf{j} \\ \mathbf{r}_{CM} \end{bmatrix} = \mathbf{0} \quad (5.13)$$

Where $\boldsymbol{\Omega}$ means a matrix that rearranges the elements of velocity and angular acceleration. Rewriting the determination problem as:

$$B\mathbf{x} = \mathbf{c} \quad (5.14)$$

and imposing the $r_{CM,x}, r_{CM,y}$ constraint:

$$B = \begin{bmatrix} 0_{1 \times 6} & 1 & 0 & 0 \\ 0_{1 \times 6} & 0 & 1 & 0 \end{bmatrix}, \mathbf{c} = \begin{bmatrix} r_{CM,x} \\ r_{CM,y} \end{bmatrix} \quad (5.15)$$

This method is strongly subject to the noise to which angular velocities are subject. Therefore, it is preferable to implement a different formulation that includes the 5.14. This solution is robust with respect to noise due to angular velocity estimates as it is not intended to be used.

$$\left[\boldsymbol{\Omega}(\boldsymbol{\omega}^b) + \int [\tilde{\omega}]\boldsymbol{\Omega}(\boldsymbol{\omega}^b) \mid m_{tot} \int [\tilde{g}] \right] \begin{bmatrix} \mathbf{j} \\ \mathbf{r}_{CM} \end{bmatrix} = 0 \quad (5.16)$$

Residual Torque Estimation and Vertical Offset Balancing

Once the in-plane balancing procedure has been completed, the residual torque due to the residual distance between CM and CR is estimated and then compensated by moving the balancing mass in the z-axis. In order to consider the contribution due to the unbalance vector, it is necessary to assume:

- Small angular speed;
- Air-bearing friction and aerodynamic torque are negligible;
- Magnetic Interaction Torque is compensated for through the use of the Helmholtz Cage;

Under these assumptions, the predominant torque is the one due to the unbalance vector, and consequently, its estimation provides an estimate of the effectiveness of the balancing procedure. In order to determine the disturbance torque acting on the facility, it is necessary to solve the equation of dynamics 5.17, considering the change in angular momentum. In 5.17, it is denoted by τ_d the disturbance pair under study.

$$\tau_d = J\dot{\omega} + \omega \times J\omega \quad (5.17)$$

The information on the change in angular velocity cannot be directly derived from the IMU data. It is necessary to apply the **Savitzky-Golay** filter to differentiate angular velocities. To derive the necessary data for torque estimation, information is gathered about the platform once it has been placed in a free-swinging condition. Free oscillations can be generated manually in two ways:

- Manually tilting the platform and afterwards realise it;
- Once already tilted, returning manually the platform to a horizontal position and realising it;

It is necessary to introduce two methods for the initialization of free oscillations because, depending on the reciprocal position of CM relative to CR, the system can behave in two ways: rigid *stable* or *unstable* pendulum.

- **Stable pendulum behaviour:** the system has this behaviour when the CM is below the CR. As can be seen in 5.2 [37], the torque that tends to be generated for the decomposition of the weight force is a balancing one, in the sense that it tends to return the system to its original condition.

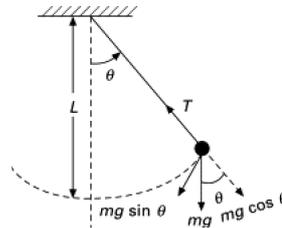


Figure 5.2: Stable Pendulum condition

In this case, the oscillation is generated by manually tilting the platform, and using its stable behaviour to initialise the oscillatory behavior;

- **Inverted pendulum behaviour:** This behaviour occurs when the CM is above the CR, in which case, the gravitational force tends to destabilise the system, as seen in 5.3. In reverse pendulum conditions, the platform once placed in a tilted condition will tend to be totally unbalanced to one side, without returning to its initial operating condition. This means that an oscillatory motion will not be induced. In this case, the oscillation will be shorter and will be triggered by returning the platform to the horizontal position and leaving the system free.

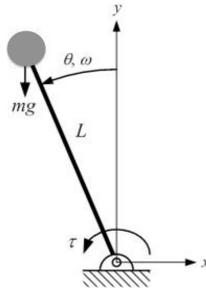


Figure 5.3: Inverted pendulum behaviour

Depending on the type of pendulum representing the system, the vertical balance mass is moved: towards a position further away from the base plate of the platform if the pendulum is normal, in the opposite direction if the pendulum is inverted. An iterative procedure results in an off-set of the z-mass comparable to the residual imbalance in the x-y plane. At this point, the behaviour of the platform is no longer well defined, i.e. the type of pendulum is no longer easily identifiable. At this point, one proceeds to determine the residual torque and evaluate, in successive iterations, a possible offset on z to decrease torque. Once this decrease is no longer appreciable, the balancing process is considered to be complete.

5.3 Balancing Results

An analysis of the data obtained through the above balancing procedures is proposed in this section.

Inertia Estimation

In order to carry out a first iteration of the balancing process, it is necessary to have an initial estimate of inertia. To do this, the CAD model of the platform previously introduced in the integration chapter was used. The estimated inertia

matrix is as follows:

$$J_{CR} = \begin{bmatrix} 0.0333 & 0.0061 & 0.0007 \\ 0.0061 & 0.0458 & -0.0009 \\ 0.0007 & -0.0009 & 0.0384 \end{bmatrix} \text{ kg} \cdot \text{m}^2 \quad (5.18)$$

According to the IMU characteristics in table 4.1, errors corrupt angular rates, attitude, and gravity data before they are given to the control and estimation algorithms. According to the restrictions experimentally discovered for the stepper motors, the actuators' maximum rate and acceleration of the balancing mass displacement are limited. The setting for mass displacement resolution is corresponding to a motor step. In addition to gravity, air friction is considered as a disturbance torque and is supposed to be proportional to the square of the platform angular rate. By applying the inertia calculation procedure described in 5.2, the inertia of the platform was estimated. This procedure was repeated several times in order to have a reliable estimation. The final inertial matrix is:

$$J_{est} = \begin{bmatrix} 0.025 & 0 & 0 \\ 0 & 0.02 & 0 \\ 0 & 0 & 0.017 \end{bmatrix} \text{ kg} \cdot \text{m}^2 \quad (5.19)$$

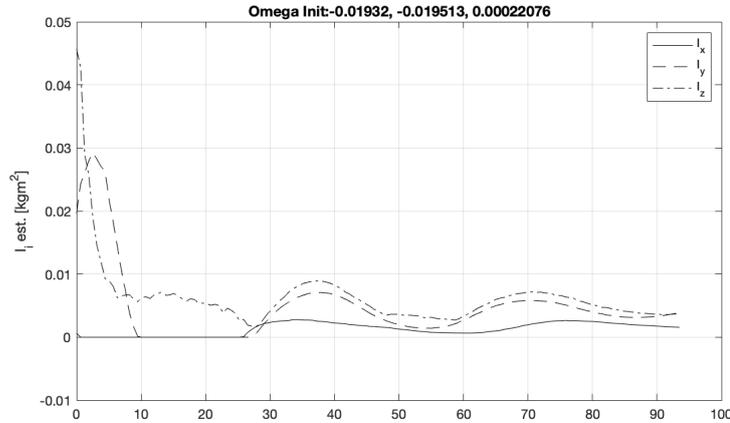


Figure 5.4: Representation of the gravitational vector variation

This result was obtained by imposing a compressor pressure of $80[psi]$, a z mass off-set of $77.5624[mm] = 77.5624 \div 0.0016 = 48476[steps]$ from the base of the stepper motor and an unbalance vector of $[-120, -120, 0]$ steps to induce the free oscillation. In order to assess the quality of the imposed oscillation, the variation of the vector \vec{g} in space is reported as in 5.5: 'good oscillation' is defined as that

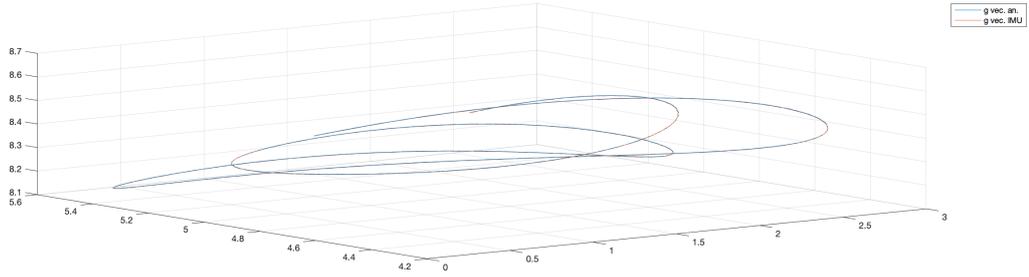


Figure 5.5: Representation of the gravitational vector variation

which allows the displacement of \vec{g} in the widest range of directions. As can be seen in figure 5.5, the oscillation used for inertia estimation is optimal. The proposed inertia estimate is realised for an off-set between CM and CR shown in figure 5.6.

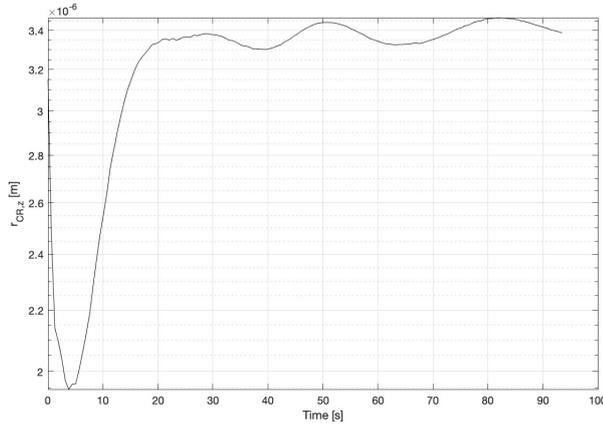


Figure 5.6: $R_{CR,z}$ offset estimation

With regard to the angular velocity filtering settings, it is set the order of the interpolating polynomial equal to **3**, and the number of samples used for interpolation, indicated by frame length, equal to **121**. Angular, raw and filtered velocities are shown in figure 5.7. As a check on the correctness of the filter data, it is good to note the correspondence of the filtered data with the raw data: the filter must reduce the fluctuations as much as possible while respecting the overall trend of the raw data.

Taking samples, one should make a trade-off between two competing demands when gathering data for problem equations: on the one hand, having a sufficient number of data allows a very good observability of the phenomenon; on the other hand,

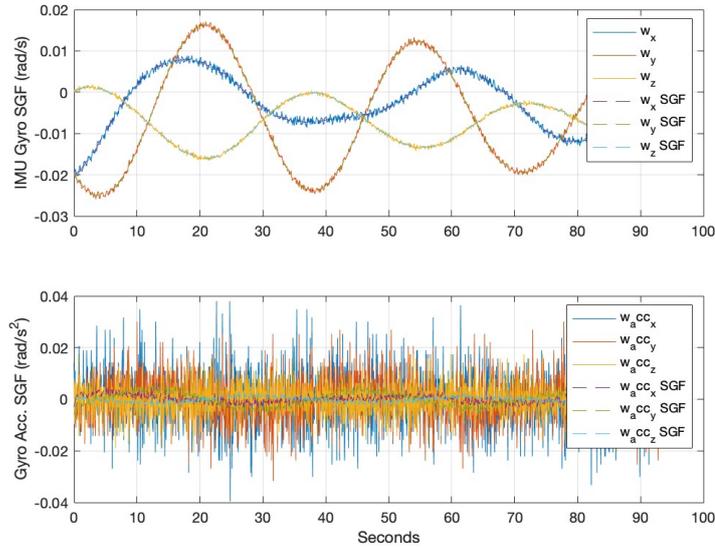


Figure 5.7: Angular velocities and acceleration proceeds from IMU and accelerometers

however, the estimation may be compromised if the period of sampling collection is too long due to the growing influence of the un-modeled torques.

Balancing procedure

The results obtained for the procedure of automatic balancing performed by using only the shifting masses are shown below.

Since the commanded position for the balancing moving masses is characterised by random noise due to the gyroscope data, a solution is proposed to minimise vibrations by tuning the gains used to achieve balancing in the x-y plane. First, it is necessary to introduce the action of each individual gain on the behaviour of the platform.

- **Derivative gain:** This gain has the effect of causing the angular velocities ω_x and ω_y to tend to zero;
- **Proportional gain:** is the proportional gain gives an action that is proportional to the error in trim. When the error becomes small, the filter action is correspondingly small and may not be sufficient to change the pitch, so integrative gain comes into action;

- **Integrative gain:** which sums the error to give the action. That is: if the inclination at instant $t=0$ is 30° , the action given by the integrative gain is equal to a fictitious value 3. If at instant $t=2$ the inclination is still 30° , the proportional gain continues to give the same correction, while the integrative gain now gives a correction value of 6. The correction action is therefore the greater, the more time passes if the position error remains. The integrative gain compensates for the error as it compensates for the action if the system remains in the same position as time passes;

The gains allowing the least possible oscillation of the masses' position have been found experimentally after several iterations of the balancing process. The results are shown in the table 5.1. The displacement commanded to the masses during

K_d	K_p	K_i
0.007	0.0008	0.00007

Table 5.1: Gain adopted for x-y balancing

the balancing procedure is shown in figure 5.8.

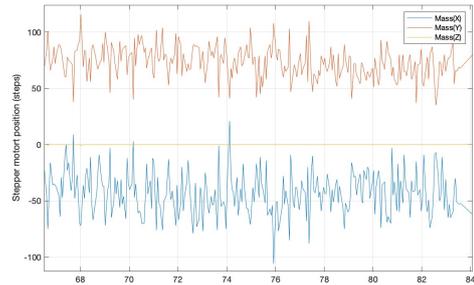
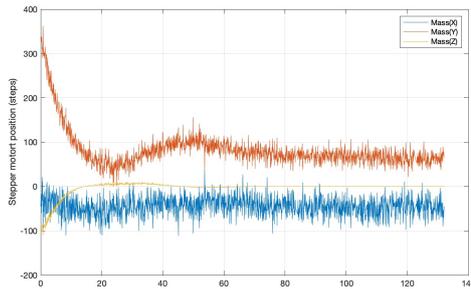


Figure 5.8: Shifting masses positions **Figure 5.9:** Oscillations of x,y masses

The choice of stepper motors with a step size of $0.0016[mm]$ allows for higher balancing accuracy than the one that could be achieved with the previous platform, which implemented motors with step sizes of $0.002[mm]$. As can be seen in figure 5.9, the shifting masses have a position variation of ± 50 steps.

This oscillation is a source of disturbance during the balancing procedure as it causes significant vibrations. It is therefore conceivable to further reduce this type of oscillation by means of a noise-reducing filter for angular velocity data. A first filter is already applied to the angular velocities during the post processing elaborations, so it is possible to compare non-filtered and filtered velocities as

shown in figure 5.10. More precisely 5.10 shows the raw angular velocities, while figure 5.11 shows the same velocities to which the Savitzky-Golay filter is applied.

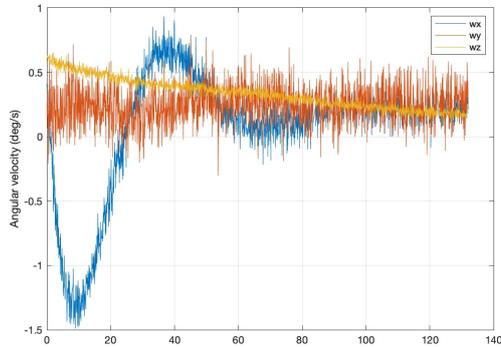


Figure 5.10: Non filtered

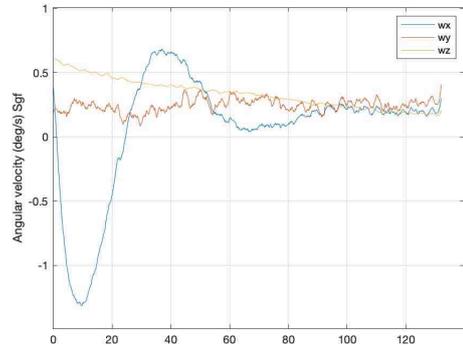


Figure 5.11: Filtered

It is clear that the filtering has already considerably reduced the noise due to IMU measurements, however, considerable fluctuations can still be present, especially in relation to the measurements of ω_x and ω_y . The balancing procedure in the x-y plane is considered complete when the platform has the gravity vector aligned with the \mathbf{z} vector: this is verified experimentally by observing the vector \vec{g}_z whose components must be as close to 1 as possible, while the components \vec{g}_x and \vec{g}_y must be brought as close to zero as possible. The variation of the components of the gravity vector as time varies are shown in figure 5.12.

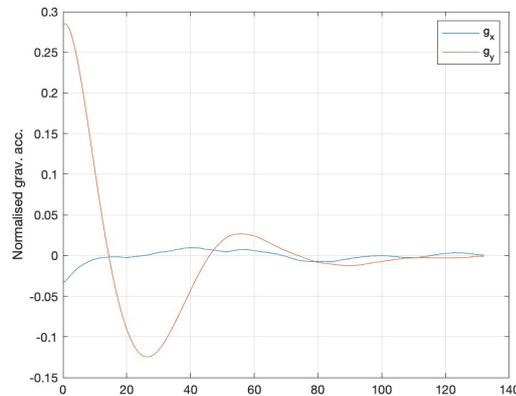


Figure 5.12: Gravitational vector reduction during time

Once the balancing is complete, the following step is to calculate the torque characterising the facility due to the residual distance between CM and CR. Figure 5.13 shows the results from the samplings obtained using the second oscillation methodology, as the estimate is shown after various iterative procedures of balancing and

mass correction on z and the behaviour of the pendulum is not clearly identifiable.

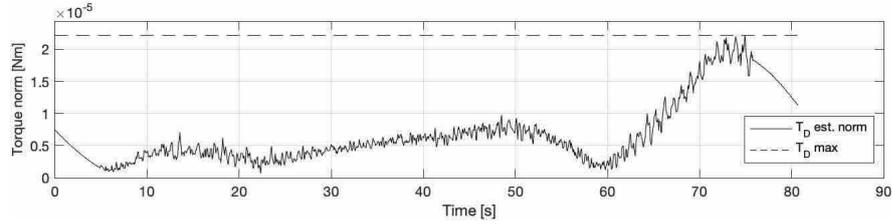


Figure 5.13: Torque estimation results

In figure 5.13 it can be seen that the maximum residual torque is $2.5 \cdot 10^{-5} [Nm]$, while the RMS is equal to $8.5 \cdot 10^{-6}$. These values are in line with those obtained from other facilities similar to the one under study as reported and exhaustively analyzed in [26]. Angular velocity data from the IMU and angular accelerations from the gyroscope are given in 5.14.

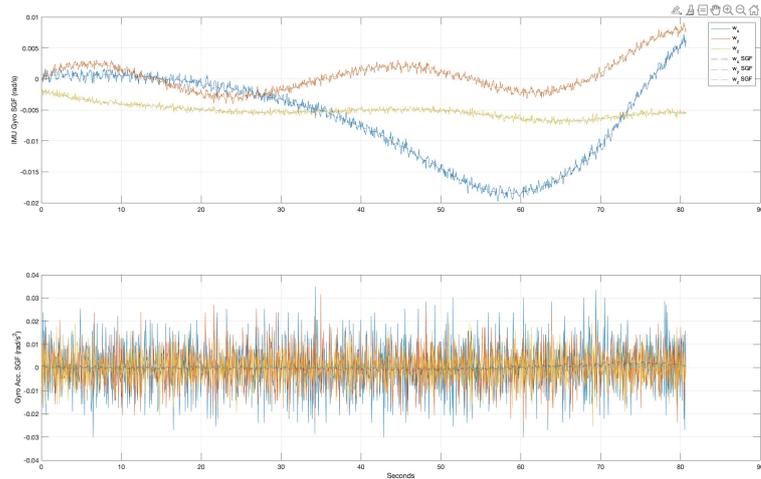


Figure 5.14: Torque estimation filtered data

In the angular velocity graph, data filtered by setting integration order equal to 3 and number of samples equal to 51 are shown as a dashed curve. As can be seen, these sample values allow for optimal data filtering as the trend of angular velocities is respected, without following the perturbations. With regard to angular accelerations obtained by means of gyroscopes, it must be remembered that these are affected by a drift that increases with the passage of time. It will therefore be necessary to perform a reset, via the user interface, after a certain amount of time.

5.4 Electrical Board PCB

As introduced above, the need that has been followed as a guideline for the realization of the platform is to implement solutions to achieve a residual torque as low as possible, ideally of the same order of magnitude as the orbital disturbances, i.e. $10^{-6}Nm$. From the experimental data, illustrated in 5.3, it can be seen that further torque minimization strategies are necessary in order to meet the required standard. In this section, a possible strategy to further minimise disturbance torques due to the facility is analyzed.

When residual torques are of the order of 10^{-5} , even the slightest change in mass distribution on the platform causes a change in torque. The presence of free moving cables, such as those used during the integration of electronic components on the base platform, is a source of mass distribution variability whenever cables are accidentally moved during platform relocation or during the balancing procedure. To reduce the occurrence of moving components, an **embedded board** was designed. Starting from the same electronic components implemented in the base plate of the platform, two different types of boards were designed:

- **Fully Embedded Board:** in this first design, an integral components embedding was opted for;
- **Pin Board:** in this second, simplified design, it was decided to reproduce the pinout of the electronic components on the board and only realise the connections between the various pins in an embedded form;

Fully Embedded Board

In first instance, the design of a fully embedded board was realised using **Eagle** software. In order to design this, the starting point was the schematics of the commercial components implemented in the facility. Once the individual elements of each commercial component were imported into the same schematic, the connections indicated by the integration procedure were implemented. Figure 5.15 shows the portion of the schematic relating to the motor driver with its associated voltage regulator, while figure 5.16 shows the section of the schematic relating to the Feather M0 and the Arduino Protoshield board connections.

Once the schematic was realised, the PCB (Printed Circuit Board) was produced. The software automatically reports the components introduced with their respective air-routes, which represent the connections designated via the schematic. The next step is to realise the connection paths. In the realisation of the board, the following design choices were made:

- The **Ground** is realised via **polygon**;

Balancing Procedure

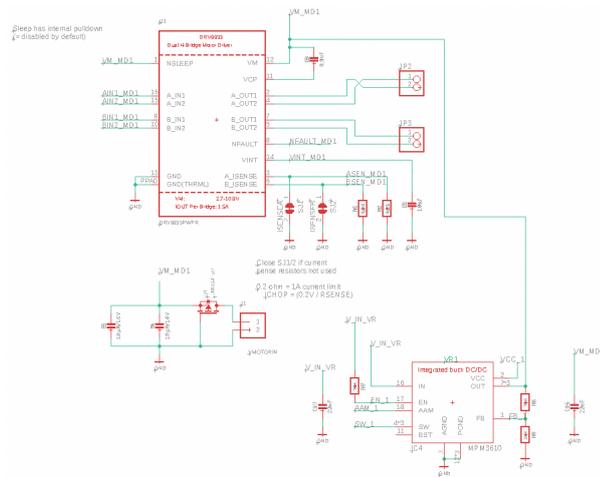


Figure 5.15: Schematic of Motor Driver and Voltage Regulator for the fully integrated board

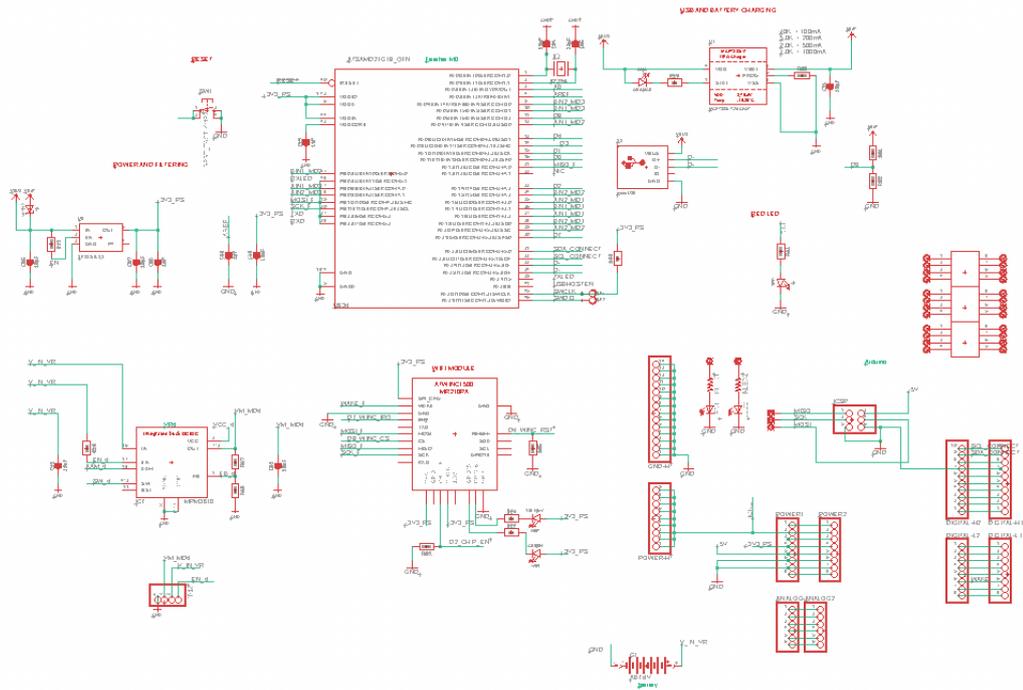


Figure 5.16: Schematic of battery pack, Feather M0 and Arduino Protoshield for the fully integrated board

- The size of the **routes** for supplying the motor drivers is $0.5[mm]$, in order

to guarantee the correct high voltage supply, without causing damage to the circuit;

- The dimension of the **Vias** is at least of 0.6mm in order to facilitate a possible choice of construction of the board in-house;

The PCB of the final boards is shown in figures 5.17.

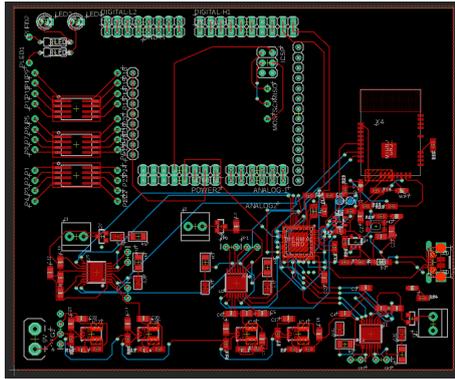


Figure 5.17: Fully integrated PCB

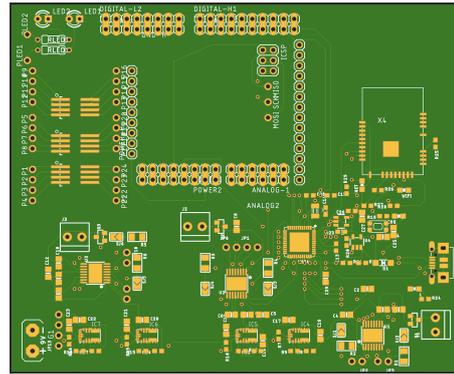


Figure 5.18: Printed board

Building a fully integrated board has a number of advantages and disadvantages. Among the **advantages** are the possibility of minimizing weight, volume and manual integration. The main **disadvantages** related to the in-house realisation of the component are: the sale of individual parts of commercial boards is not guaranteed, and the complexity of the project is evident. In addition, budget constraints and working time imposed for the project must be considered. In order to realise the board in the shortest time and with the lowest costs, considering that the commercial elements are already available for this platform, it was decided to realise a second design of the board, which would only include the integration of the pins.

Pin Board

As far as the pin board is concerned, this only provides for embedded connections external to the respective boards of the commercial components. For this reason, appropriate pin holes are allocated for motor drivers, voltage regulators, etc., from which the embedded connection tracks will start. The same steps as previously introduced are followed. Initially, the schematic was made from those of commercial components: in this case, it was not necessary to completely import the component, but only the break-out pins.

The final schematic is extremely simplified compared to the previous case, as can be seen in figure 5.19 and 5.20.

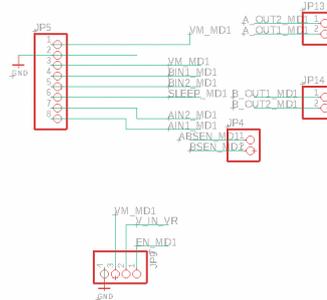


Figure 5.19: Schematic of Motor Driver and Voltage Regulator for the pin board

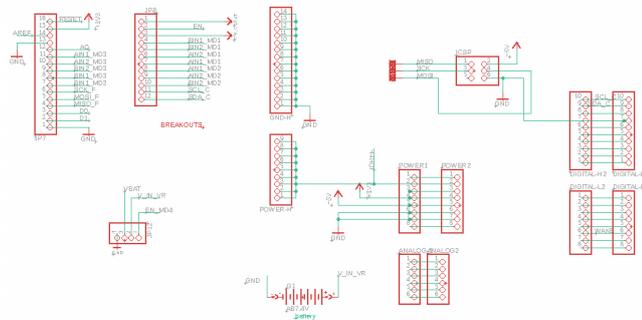


Figure 5.20: Schematic of battery pack, Feather M0 and Arduino Protoshield for the pin board

Once the schematic has been completed, the following step is to realize the board PCB. In addition to the considerations introduced for the fully integrated board, it must be borne in mind that the final plugged in components comply with the dimensions given in the respective datasheet, so the overall dimensions of each must be carefully assessed and modelled so as not to have overlapping or interference. The design of the final board is shown in figure 5.21. As can be seen in 5.21, the complexity of the board is considerably reduced compared to that in 5.17. This solution makes it possible to realise the board within the university laboratory in a very short time, while maintaining almost zero costs due to in-house production without the need to buy additional components.

By means of the present solution, an initial torque reduction solution has been designated.

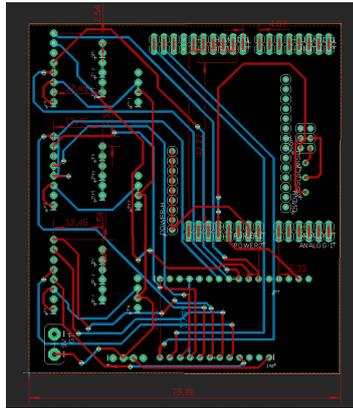


Figure 5.21: Functional Board with pin integration PCB

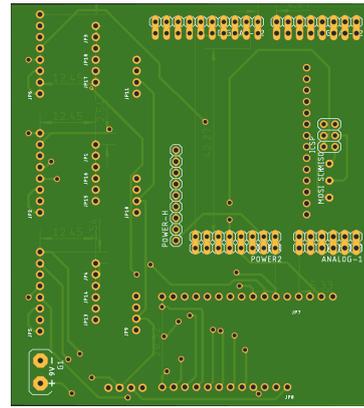


Figure 5.22: Functional Board with pin integration printed board

However, its effectiveness must be verified through realisation and implementation in the facility. These activities are left as future development to be applied for the next design iteration of the platform.

Chapter 6

Final tests

In this chapter, the procedure followed for the sizing of the magnetic actuator implemented in the host CubeSat under test is reported. The implemented verification procedure is then examined in detail.

6.1 Magnetic Torquers

As introduced in 1, the ADCS subsystem has a **determination** functionality and a **control** functionality. This section will focus on the latter. With regard to the implementation methodology chosen for the present application, the *magnetic* one was chosen.

6.1.1 Introduction to Magnetic Actuation

The typology of actuation that is integrated and tested in the ADCS subsystem is the **magnetic one**. The type of magnetic actuators used are magnetic torquers, which are key components of attitude control systems.

In order to control the attitude, actuators must generate a torque. The magnetic torquers consists of windings made of conducting material: when the current flows through the winding, it generates a magnetic dipole in the normal direction to the plane of the winding itself. The magnetic dipole vector generated is:

$$\vec{m} = N \cdot A \cdot I \cdot \vec{n} \quad (6.1)$$

Where \vec{m} denotes the dipole moment vector in [$A \cdot mm^2$], N is the number of turns of the actuator, I is the current flowing through the solenoid in [*Ampere*], A is the area of the torquer in [mm^2] and \vec{n} is the normal vector to the plane containing the MT. The dipole torque tends to couple with the external magnetic field, bringing

the system back into alignment with the magnetic field lines themselves. The actuation torque is expressed as:

$$\vec{T} = \vec{m} \times \vec{B} \quad (6.2)$$

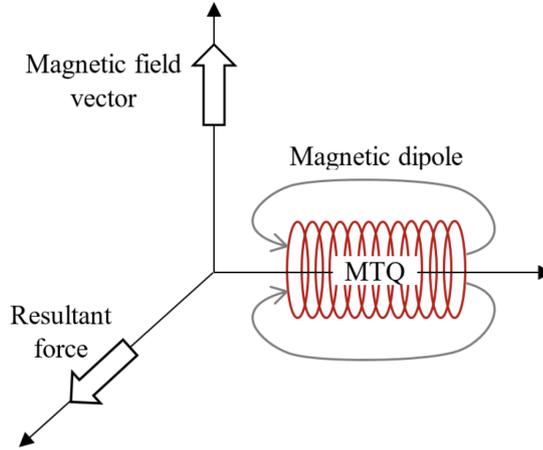


Figure 6.1: MT principle of operation [38]

In equation 6.2 we denote by τ the resulting torque vector and by \vec{B} the external magnetic field vector. In terms of actuator's space application, \vec{B} denotes the earth's magnetic field. As can be seen in figure 6.1, the direction of the resultant force is perpendicular to the plane containing the actuator and the magnetic field vector. Consequently, three different actuators (one for each axis) must be used if 3-axis control is required. This type of control is not always necessary as a single-axis control (as the one tested in this discussion) may be sufficient for the desired control applications. Further investigations will be proposed in section 6.2. Developing the vector product, it can be seen that the control torque is greatest when the magnetic dipole and the magnetic field are perpendicular, while the torque cancels out when the two vectors become parallel. The **intensity** of the control torque is also determined by the intensity of the dipole and the magnetic field. In an orbital operating condition, the magnetic field will be the earth's magnetic field, which has a decreasing intensity as one moves away from the surface, while in a laboratory testing context, the magnetic field is that generated by the Helmholtz Cage. The vector \vec{B} will be set by the user and modulated/scaled in intensity and direction according to the needs of the performed test. This magnetic field variability has a dual purpose during the testing process of the ADCS system:

- To allow **compensation** for the local terrestrial magnetic field present in the laboratory;

- To allow **modulation** of the control torque without over-sizing the actuator;

As introduced earlier, ideally the residual torque of the testing facility should be of the order of magnitude of the orbital disturbance torque (order 10^{-6}). However, the order of the residual torque is 10^{-5} . The control torque must therefore be at least greater than the disturbance torque induced by the facility: more precisely, it must be equal to the disturbance torque plus the control/implementation torque required for the satellite attitude control. There are two ways to accomplish the desired objective:

- **size the actuator** according to the total torque required;
- reproduce a **magnetic field greater** than the orbital one;

In order to be able to comply with the limitations in terms of weight, volume and consumption required in orbit, and to be able to test its effectiveness, the solution involving modulation of the magnetic field is preferred.

The advantage of magnetic actuation lies in the ease of implementation and high reliability of the actuators, combined with low mass and power consumption. The main disadvantages are a functionality that is strongly correlated to the magnetic condition outside the satellite, as well as the low efficiency of the actuator. This type of actuator has applicability both in the detumbling phase of a space mission and during operational phases, in order to allow desaturation of reaction wheels without affecting the satellite's attitude. There are three types of magnetic actuators:

- **Air Core Magnetic Torquer:** this type of actuator consists of a coil, with a certain number of turns N , of ferromagnetic material (usually copper). This actuator is intended to have the largest possible area, and is called an *air core* precisely because there is nothing inside the winding. Generally, this type of actuator is installed on the walls of the Cube Sat, so its size is limited at the top to be 10×10 cm;
- **Embedded Magnetic Torquer:** The PCB-type solution implements the air coil actuator concept in an embedded manner within the satellite's enclosure and cover panels. The copper cable is an embedded track within the panel itself. The shape is rectangular with spirals narrowing towards the centre;
- **Torquerod Magnetic Torquer:** this third type of torquer is based on the same principle as air coils, but with increased torque produced. This is achieved by placing a conductive material inside the winding, which amplifies the effectiveness of the system;

The type of actuator used in this thesis project is the *air core type*. The sizing methodology is hereafter explained.



Figure 6.2: Air Core MT [39]

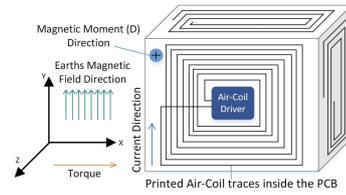


Figure 6.3: PCB MT [40]



Figure 6.4: Rods MT [39]

6.1.2 Sizing

With regard to the sizing of a magnetic actuator, particularly of the air core type, several design aspects must be taken into account. The most stringent ones are:

- Mass;
- Magnetic dipole generated;
- Occupied Volume;
- Power consumption;

Sizing preliminaries

As introduced earlier, the control torque, generated by the interaction of the control magnetic dipole and magnetic field, tends to bring the control torque vector to coincide with the direction of the magnetic field. It is therefore useful to maximize the magnetic dipole produced for each winding: in this way, maximization of torque is achieved against minimization of mass. Looking at the equation 6.1, an ideal design aims to maximize the number of winding, the supply current I , or the actuator area A . As already seen, the area A is limited by the size of the CubeSat, so N and I must be taken as variables. Considering a **square-shaped** Air core and

assuming it to be an ideal electrical conductor with a constant wire cross-section and uniform mass distribution, the resistance induced by the winding wire is equal to:

$$R = \sigma \frac{L}{S} = \sigma \frac{4aN}{S} \Rightarrow N = \frac{RS}{4a\sigma} \quad (6.3)$$

Where L is the length of the wire and S is the cross-sectional area, calculated as $S = \pi \cdot r^2$ [mm^2]. In equation 6.3 it is also denote by a the size of the actuator, while σ denotes the electrical resistivity of the material of which the actuator is made. The following formulation can be used to calculate the mass of the actuator:

$$M = \rho V = \rho LS = 4a\rho NS \Rightarrow N = \frac{M}{4a\rho S} \quad (6.4)$$

where V is the volume of the air core actuator and ρ is the density of the conductive material. Taking the last of the sizing parameters into consideration, it has to be considered that the power consumed can be derived from the resistance and the current supplied as $P = RI^2$. Considering the set of equations introduced above, an alternative formulation for calculating the magnetic dipole generated by the actuator can be derived as:

$$m = A\sqrt{N^2 I^2} = a^2 \sqrt{\frac{RS}{4a\sigma} \frac{M}{4a\rho S} I^2} = \frac{a}{4} \sqrt{\frac{PM}{\sigma\rho}} \quad (6.5)$$

This formulation is useful for determining the drivers which effect the dipole variation. It varies linearly with a , so maximum use must be made of the dimension provided by the standards. It should also be noted that the energy consumed can be reduced by an increase in mass of the actuator: if one increases, the other decreases. The ideal condition is that for which there is minimum energy consumption and minimum mass, it follows that since the two are inversely proportional, a trade-off will have to be made.

Sizing Results

For the present application, it was necessary to start from a pre-existing active magnetic actuator. However, the characteristics of this actuator are not known in advance. Therefore, following the formulation introduced above, we started from the physical data of the actuator, proceeding backwards until the number of turns and current value were derived, and consequently the dipole produced. Information on the given actuator are listed in the table 6.1. where σ denotes the electrical density of the material, while ρ denotes the volumetric density. Using a multimeter, the resistance of the component was determined. Once known, considering Ohm's law for which $I = V/R$, the supply current $I = 0.1587$ [A] was calculated. It was then necessary to determine the number of turns of the actuator.

Power supply	
Voltage	5 [V]
Geometrical	
M	30 [gr]
R	31.5 [Ohm]
a	74 [mm]
A	5476 [mm ²]
B	$5 \cdot 10^{-4}$ [T]
S	0.0314 [mm ²]
Material	
Material	Copper
σ	$1.7 \cdot 10^{-5}$ [Ohm · mm]
ρ	0.0089 [gr/mm ³]

Table 6.1: Air core magnetic torquer known characteristics

To do this, it was implemented the formulation 6.4, of which every value except N was known. The further step was to proceed with the derivation of the inverse formulation as:

$$N = \frac{M}{4a\rho S} = 360 \text{ turns} \quad (6.6)$$

It can be done a cross-check for the correctness of the calculation just performed using the formulation derived from 6.3. Once the supply current and the number of turns are known, it is possible to calculate the **magnetic dipole** according to the formulation given in 6.5. Once the magnetic dipole is computed as just described, the implementation torque τ can be calculated. If the vector product is developed, it can be seen that the intensity of the magnetic actuation torque varies with $\sin(\theta)$, where θ denotes the angle between the magnetic field vector \vec{B} and the normal to the actuator \vec{n} . With regard to the torque, its maximum value can be calculated: it is considered the torque developed when there is orthogonality between the two vectors. Although an optimal case of study is being used to estimate the couple, considering a real application with 3 actuators, as the angle θ decreases for one actuator, it increases for the other two: when the first actuator has zero angle with the magnetic field, the next one will be perpendicular to it. Thus, the calculated torque is always guaranteed. The results obtained for the actuator introduced above are shown in table 6.2.

As can be seen, the torque obtained with a magnetic field of 5 Gauss (1 order of magnitude larger than the one in orbit) is greater than the facility's residual torque (obtained in 5.3) by 1 order of magnitude.

Sizing Results	
\vec{B}	5[Gauss]
m	0.3137[Am ²]
τ	$1.5683 \cdot 10^{-4}$ [Nm]

Table 6.2: Sizing Results for Air Core Magnetic actuator

Considering the increase in the magnetic field up to a maximum of 10 [Gauss] as a further margin to guarantee the adequate torque for attitude variation manoeuvre, the effectiveness of the actuator is theoretically guaranteed. At the same time, the generated magnetic dipole is kept low, in accordance with the required in orbit application levels.

Dimensioning of in orbit disturbances

With regard to torquer sizing results, it is now verified whether the torque provided agrees with what is required for the contrast of orbital disturbances. To do this, a preliminary estimate of the disturbances acting on a CubeSat 1U located at an **altitude of 700 km** is made. In order to calculate the pairs characterising the ambient of a low earth orbit, the formulae introduced in section 2.3.3 were used.

Dimensions and weights given by the CubeSat standard were considered as spacecraft characteristics. It is assumed to have a circular orbit and an **inclination of 71°**. As far as attitude control is concerned, a pure magnetic actuation is considered, carried out by means of MT: no reaction wheel desaturation torques are therefore being included. For torquer sizing, **1.5 times the external disturbance** torque must be provided: this margin is intended to ensure effective detumbling and a good safety margin, without over-dimensioning the actuator. The result for the disturbance torques are shown in table 6.3.

External Torques	
T_g	$2.1798 \cdot 10^{-11}$ [Nm]
T_{sp}	$2.0.361 \cdot 10^{-9}$ [Nm]
T_m	$4.4496 \cdot 10^{-7}$ [Nm]
T_a	$5.5586 \cdot 10^{-8}$ [Nm]
T_{tot}	$4.4843 \cdot 10^{-7}$ [Nm]

Table 6.3: Disturbances estimation for 71°, 700 Km 1U CubeSat application

In table 6.3 T_g denotes the torque due to the gravity gradient, T_{sp} is the torque due to solar radiation, T_m is the torque due to the Earth's magnetic field in orbit, T_a is the aerodynamic torque and finally T_{ext} is the total disturbance torque.

Using $1.5 \times T_{ext} = 6.7264 \cdot 10^{-7}$ as the dimensioning torque, the total dipole required for MT was obtained. The dipole required for each torquer is $0.0151[Am^2]$. This value takes into account the disturbance pairs only, however, an appropriate analysis of the effectiveness of the implementation during the detumbling phase will be necessary.

Final Observations

As can be seen from the values obtained above, the actuator used for the tests generates a magnetic dipole of $0.3137[Am^2]$, as opposed to the required $0.1[Am^2]$ (taking into account the increase due to the detumbling phase). However, for an application related to **testing** the effectiveness of control algorithms, it can be useful to have an oversized actuator. This is because the inertia of the platform is very high, and consequently the disturbance torque due to the facility that has to be counteracted. Once the algorithms have been verified, MTs of dimensions in agreement with the ones emerged from the sizing procedure will be implemented.

6.2 Magnetic Actuation Tests

This section provides a detailed presentation of the ADCS subsystem and all the preliminary set-up activities that preceded the magnetic test activities carried out at the microsatellite laboratory in Bologna. Once the set-up is complete, a detailed analysis of the results obtained during the test phase is carried out.

6.2.1 Preliminary Set-up

In the phase preceding the magnetic tests, the ADCS board is analysed in detail: the components and their function are introduced. Subsequently, the control laws under test are developed. Two different types of control are proposed in this thesis: **PD controller** for a basic control to assess the effectiveness of the selected actuator and **Y-dot controller** to assess the single axis energy reduction achievable with the MT. It is then analysed how the basic functional board and the upper subsystem board interface each other in terms of data packet exchange and power supply. Finally, a **MATLAB SIMULINK** model, used for an initial simulation of control operation, is reported and analyzed.

ADCS board description

The ADCS board, which was previously introduced in chapter 3.2.1, is now analyzed in detail. The components listed in 3.2.1 can be seen in figure 6.5.

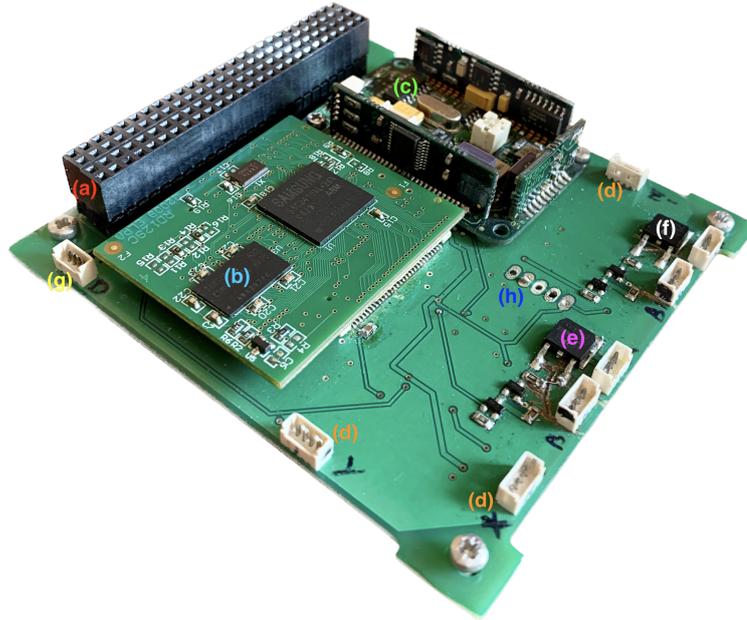


Figure 6.5: ADCS board components

In figure 6.5 the ADCS components are denoted as:

- (a): *connector 104*, the purpose of which is to establish the connections to the basic functional board in order to power the system board components;
- (b): *ELPA RD129 microprocessor*. It is an embedded CPU board with an ARM9 microprocessor. The circuit board features a watchdog timer, and requires a 3.3V power supply. The microcontroller supports the required communication interfaces (SPI, CAN bus). Also 4 timer PWM are available. The main characteristics of the microprocessor are reported in table 6.4;
- (c): *inertial platform*. It implements an X-sens type **MTi** of the Xsens Technologies. The purpose of the inertial platform is to determine the magnetic field in which the board is located via magnetometer, while the angular position conditions are determined via accelerometers. The needed IMU specifics are reported in 6.5;
- (d): *PWM outputs* to which the magnetic actuators are attached. The air core MT implemented in this application is connected to **timer 1**, indicated

Component	Characteristics
<i>CPU</i>	32 bit of class ARM9 at 240MHz
<i>Memory</i>	32MB of SDRAM at 120MHz, 32bit interface
<i>Flashboard memory</i>	64 MB on-board
<i>Serial Interfaces</i>	3 one of which compatible with IrDA
<i>USB interfaces</i>	2 (1 of which can be turned into a USB device)
<i>Synchronous serial interface</i>	for codec audio stereo at 16 bits
<i>Video output</i>	for LCD STN o TFT
<i>ADC convertor</i>	at 10 bits
<i>Interfaces</i>	IICbus, SPI
<i>PWM timer</i>	4 internal timers
<i>Whatchdog timer</i>	included in the board
<i>Numbers of pin</i>	90 I/O
<i>Power consumption</i>	0.5W typical, 1W max; single power supply required at 3.3Vdc 5%.
<i>Dimensions</i>	45×40×8 mm
<i>Connections</i>	2 x 60-pin smd connectors 0.8 mm pitch
<i>Temperature range</i>	from -25C to +85C

Table 6.4: Microprocessor ELPA RD129 technical characteristics

by **z**. The timer is regulated according to the datasheet by the RD129 microprocessor. The PWM function is to control the power supply of magnetic actuators so that they can generate a magnetic dipole that varies according to the need for actuation, controlled by the microprocessor;

- **(e),(f)**: antenna opening circuits, not used for this board;
- **(g)**: j-tag, STDOUT of the processor, for displaying the outputs of the PIO-W. The 3 active outputs are Tx (write), Rx (read) and GND;
- **(h)**: connection for IMU serial;

	rate of turn	acceleration	magnetic field	temperature
Unit	$[deg/s]$	$[m/s^2]$	$[mGauss]$	$[^{\circ}C]$
Dimensions	3 axes	3 axes	3 axes	–
Full Scale $[units]$	± 300	± 50	± 750	$-55.. + 125$
Linearity $[\%of FS]$	0.1	0.2	0.2	< 1
Bias stability $[units1\theta]^{11}$	1	0.02	0.1	0.5^{12}
Scale stability $[factor]$	$[\%1\theta]^{11}$	–	0.03	0.5
Noise density $[units/\sqrt{Hz}]$	0.0513	0.002	$0.5(1\theta)^{14}$	–
Alignment error $[deg]$	0.1	0.1	0.1	–
Bandwidth $[Hz]$	40	30	10	–
A/D resolution $[bits]$	16	16	16	12

Table 6.5: MTi specifics

PD controller

Once the subsystem board has been analysed, the single-axis control laws to be tested by magnetic actuation are implemented. The first proposed control law is the **Proportional Derivative (PD)**. A PD-type controller has an operating scheme as shown in figure 6.6.

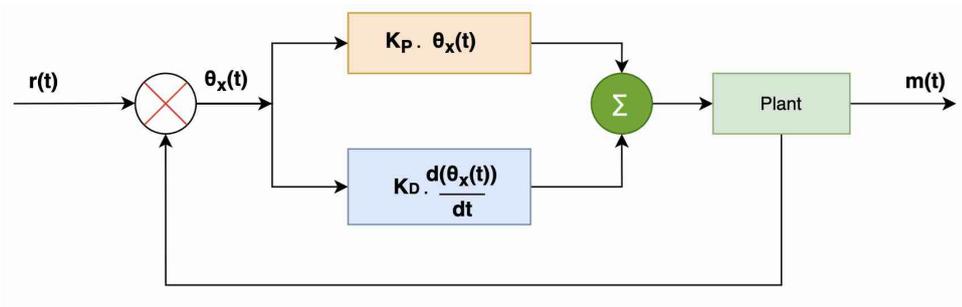


Figure 6.6: Control System with PD Controller

The proportional derivative controller is a form of controller in a control loop whose output varies proportionally to the error signal as well as with its derivative.

$$m_z(t) = K_d \omega_x(t) - K_p \theta_x(t) \quad (6.7)$$

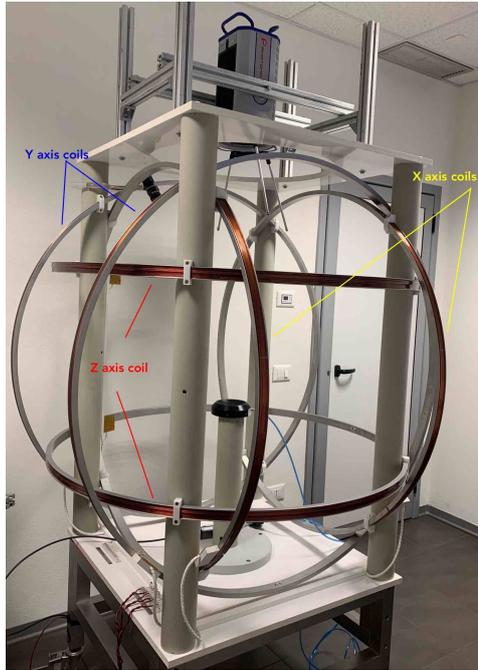


Figure 6.7: HC reference frame

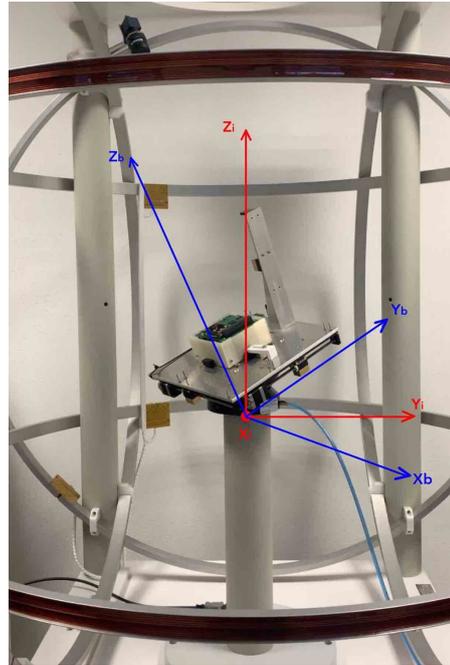


Figure 6.8: Body and Inertial RF

In 6.7 K_p denotes the **proportional gain** of the control law and K_d the **derivative gain**. A **Proportional Controller** is a type of controller in which the output varies *in proportion* with the *input*. While a **Derivative Controller** provides an action where the output is *proportional* to the rate with which the *error signal is changing over time*. In the control equation ω_x denotes the angular velocity respect to the x-body frame axis, while $\theta_x = \arcsin(g_y/|\mathbf{g}|)$ is the angle between y axis of the body frame and y axis of the inertial frame. In θ_x formulation g_y is the y body component of the gravitational acceleration \mathbf{g} , and $|\mathbf{g}|$ is the norm of \mathbf{g} . The control law takes the angle formed between the local vertical and the vertical of the inertial system, corresponding to that of the Helmholtz cage, and via the control loop calculates the magnetic dipole to be generated in order to bring the magnetic field component z and the torquer normal vector aligned. In this way, actuating torque becomes zero, and the system is maintained in the desired position.

The desired magnetic control action in this application case is one for which the platform starts in a tilt configuration around the x-axis of the platform, and by means of magnetic actuation the system is required to be brought back to horizontal equilibrium conditions. To do this, a magnetic field of type B_z is generated by HC at the local vertical of the horizontal plane. The actuator must therefore be positioned aligned with the x-y plane of the platform.

Y-Dot controller

The second control law implemented in the ADCS subsystem board is the Y-dot. This type of control originates as a single-axis version of the more complex Bdot law. Both controllers are designed for platform detumbling, i.e. the reduction of the satellite's kinetic energy in all directions (via Bdot) or on a single axis (via Y-dot). Control is achieved by activating the magnetic torquer air-core aligned with the pitch axis of the platform. Specifically, a magnetic dipole proportional to the time derivative of the pitch component of the magnetic field in the body reference system is controlled. The control magnetic dipole is calculated as in 6.8.

$$\mathbf{m} = -k_y \dot{b}_2 \hat{\mathbf{e}}_2 \quad (6.8)$$

where $\hat{\mathbf{e}}_2$ is the unit vector aligned to the pitch angle, k_y is the control gain and \dot{b}_2 is the derivative of the magnetic field in y direction. A more comprehensive discussion can be found in [41]. The control law implemented for testing is that given in equation 6.9.

$$\mathbf{m}_z = -K_d \cdot \frac{dB_z}{dt} \quad (6.9)$$

The y-dot control law was implemented as a solution to the detumbling problem by implementing a single actuator. This control law was simulated using the MATLAB Simulink model developed for the tested system, however, it was not tested experimentally.

6.2.2 ADCS software control software

The ADCS board implements the two single-axis control laws under test introduced above. Regardless of the control law in action, the **ADCS software algorithm** is structured as reported below:

- After the board boots up, the program is launched: variables are initialized, the GPIO pins are set, and the UART ports are opened and configured for communication;
- After initialization, the ADCS system begins its operation;

- In an infinite while loop, the software reads measurements from the IMU, which are 16-bit integers representing the sensor's logical state (**raw**);
- The calibration formula in matrix form is applied to obtain the real measurement vectors: $\mathbf{measure} = \mathbf{K} \cdot (\mathbf{raw} - \mathbf{bias})$, where **raw** is the vector of the raw sensors state (16-bit unsigned integers), while the **bias** and the **K** gain matrix are calibration parameters given by the producer;
- The derivative of the magnetic field B and the angle between the body y-axis and the inertial y-axis (of the Helmholtz cage) are subsequently calculated;
- The ADCS software then applies the selected control law;
- A duty cycle is calculated by dividing the dipole moment commanded from the control law by the maximum possible moment (from the torquer specifications), then the signal is modulated using the built in PWM timers and sent to the torquer;
- A packet containing IMU data, control commands (dipole moment and resulting PWM duty cycle) and a progressive counter are sent to the base board: total execution time is calculated from the counter as $time = dt \cdot counter$, with dt fixed at 0.5[s];
- At the end of the loop, the program is halted for a period of time equal to 0.5[s] minus the code execution time.

Communication strategy

Data from the subsystem board must be supplied to the basic functional board and then to the end user via a wi-fi communication.

For the communication with the OBC from the ADCS board, at the end of each functioning cycle, a data packet of **39 bytes**, in total is sent on the UART serial pin, with the following structure:

- **5 bytes** of header consisting of the 5 characters 'S' 'T' 'A' 'R' 'T';
- **30 bytes** of data, of which:
 - **18 bytes** of raw data from the sensors of the IMU, i.e. 9 *16-bit unsigned integers* representing the raw IMU data;
 - **4 bytes** making up a *32-bit floating point number*, representing the duty cycle written to the virtual file of the RD129's PWM timer: a number between 1 and 999 is written, representing a duty cycle ranging from

0% to 100%. The frequency of the PWM is set to 100kHz but it can be changed afterwards;

- **4 bytes** which make up a *32-bit unsigned integer*, which represents a progressive counter of the number of cycles completed by the ADCS: to obtain the time we multiply this number by the time-step, which is set a priori to 0.5[s], but eventually modifiable.

- **4 bytes** of closer consisting of the 5 characters 'S' 'T' 'O' 'P';

Simulink model

In order to be able to carry out an initial verification of the control laws, after a physical integration and test campaign, a virtual model of the ADCS system under test is developed. The model was developed in *Matlab Simulink*.

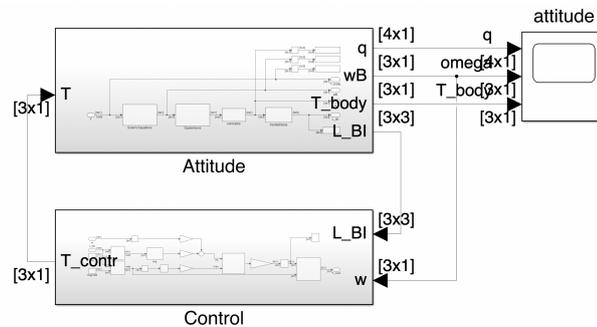


Figure 6.9: Global ADCS Simulink Model

In figure 6.9, the global model of the system is shown. It consists of two main parts: the one representing the satellite attitude, and the one implementing the control laws. The first block is shown in figure 6.10, and the second in 6.11.

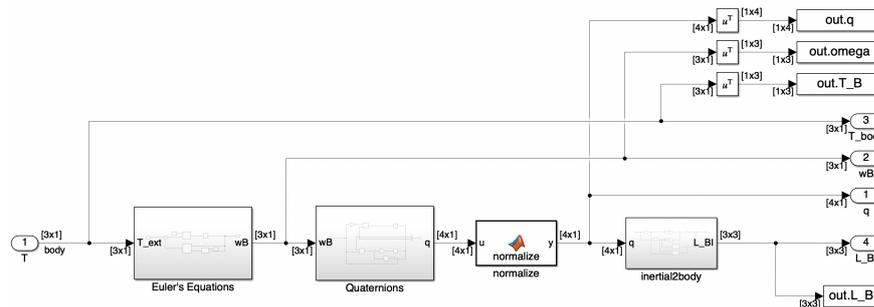


Figure 6.10: Attitude model

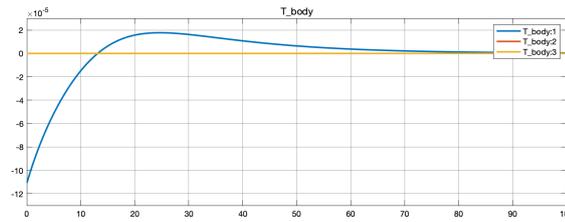


Figure 6.13: MT air-core magnetic torque actuation variation

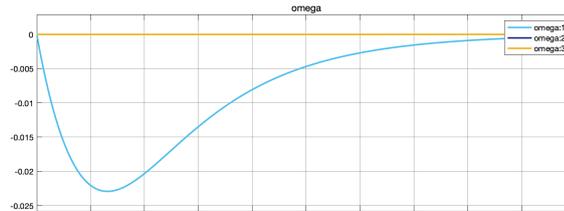


Figure 6.14: Angular velocity variation during magnetic actuation

6.3 Verification Program

This section of the chapter introduces the verification tests performed on the boards constituting the complex of functional and subsystem boards introduced in the previous chapters.

6.3.1 Functional verification of ADCS board, base functional board and connected boards

The verification tests performed on the boards constituting the host CubeSat are introduced in detail. The two boards were initially tested separately, and then the communication and correct power supply from the functional board to the upper system board was verified.

Starting with functional verification tests of the basic functional board: in particular, we analysed the voltage level of the *UART Tx* and *Rx* pins of the Raspberry microprocessor. The verification test of the ADCS subsystem board proceeded, starting with the analysis of the *RD129 microprocessor*: power-up, loading of the kernel, booting of the operating system and loading of the command line were all objectives of the test. Finally, a test was carried out to verify the successful communication between the two boards: communication between the two microprocessors was verified through the initial use of a dev board and then through direct connection of the two boards.

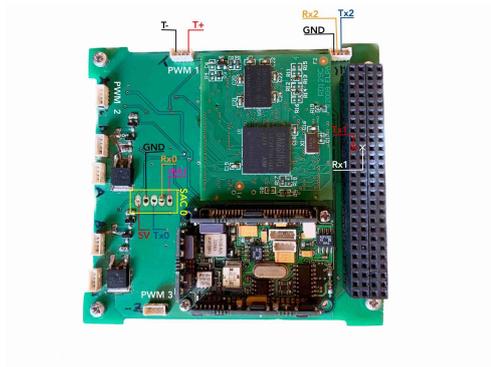


Figure 6.15: ADCS pinout

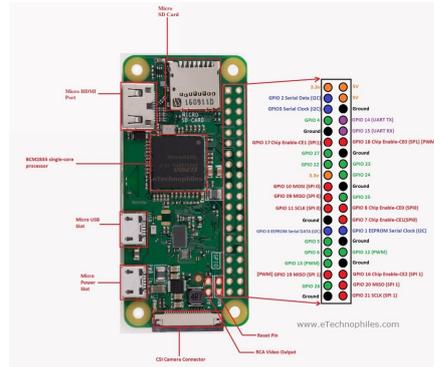


Figure 6.16: Raspberry pinout [42]

Raspberry PI0-W power supply and communication test

The objective of the Raspberry power supply and communication test is to verify that once the processor is integrated on the basic functional board, it is powered correctly. The voltage level of the UART Tx and Rx pins is also checked, in order to compare the result obtained with the Tx and Rx values of the ADCS board.

Test Set-Up In figure 6.17, one can see the set-up realised for the power supply

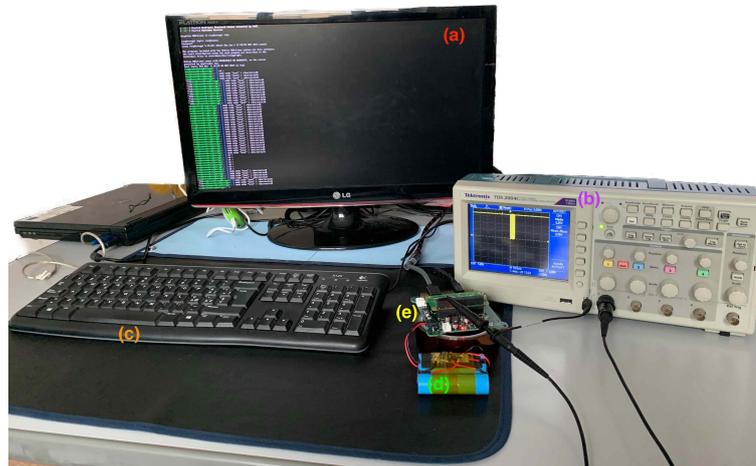


Figure 6.17: Verification test set up

and communication test on the *Raspberry PI0-W* microprocessor. In order to

carry out the test, the necessary components are listed below:

- **(a)**: graphic interface screen to view the output of the microprocessor, connected to the microprocessor via HDMI;
- **(b)**: Oscilloscope, connected between the Tx and Rx pins, makes it possible to determine the voltage present between the two, and during communication it makes it possible to identify the data packet sent, represented in the figure by the yellow bar shown on the screen;
- **(c)**: keyboard for input, connected to the microprocessor via USB output;
- **(d)**: Lithium-ion battery pack for powering the board and consequently the microprocessor;
- **(e)**: basic functional board, subject of the test, on which the microprocessor is plugged in;

Test execution consists of an initial phase to assess the correct power supply of the board and microprocessor. Attaching the batteries, by means of the appropriate connector, verifies that the board and the microprocessor are correctly functioning by switching on their respective LEDs which indicate that the power supply was correctly carried out. A multimeter is then used to assess that the microprocessor's supply current is properly regulated by means of the integrated 3.3V regulator. The second step is to connect the write (Tx) and read (Rx) pins of the PI0-W, following the schematic. Between the two cables connecting Tx and Rx, the oscilloscope is attached in order to read the voltage value present during the writing/reading of a command entered via the keyboard. By connecting the microprocessor to the video output, the entire initialisation phase of the processor and the input data during the test is displayed on the screen. Following the initial loading, a 'test' string can be sent for reading and writing on the same pin. The successful writing and reading of the string can be verified using the oscilloscope: a data packet (yellow band) appears on the screen for each 'test' string sent to the microprocessor.

Test results are **positive**, in fact, not only was the correct voltage was verified on the UART write and read pins which was found to be HIGH level at 3.3V, but it was also possible to assess the correct operation of the write and read functions as indicated above.

RD129 microprocessor initialization test

The objective of RD129 microprocessor initialization test is to test the functionality of the microprocessor implemented on the subsystem board. The functions being verified are:

- correct power supply by the basic function board;
- processor initialization;
- Kernel loading;
- booting of the operating system;
- processing commands from terminal, via Linux command line;

Test Set-Up In figure 6.18, one can see the set-up realised for the functional

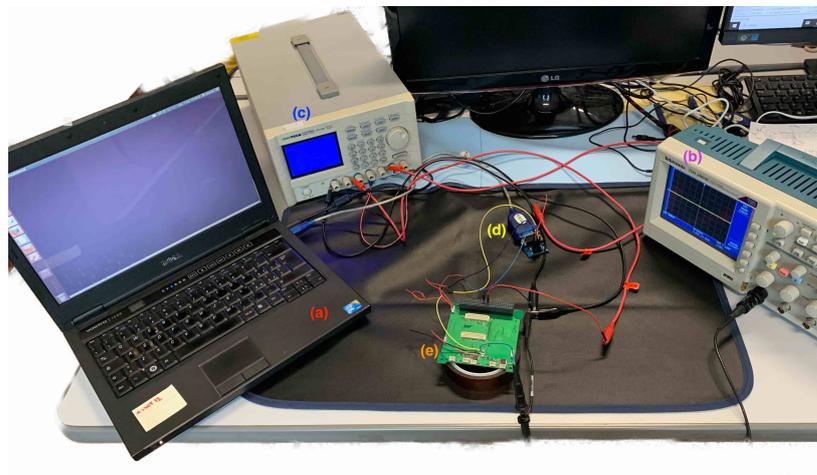


Figure 6.18: Set up for RD129 verification test

verification test of the **RD129** microprocessor. In order to carry out the test, the necessary components are listed below:

- **(a)**: laptop with cross compiler for RD129 microprocessor;
- **(b)**: oscilloscope;
- **(c)**: bench power supply, to generate the two power channels required for the ADCS board, i.e. a 3.3V and a 5V channel;
- **(d)**: RS232/TTL 3-5.5V serial adapter to modulate the output from the processor from 3.3V to 5V in order to display the microprocessor output on the screen;
- **(e)**: ADCS board, with Tx, Rx and GND output connected to the serial adapter, powered at 5V via the power supply channel;

Test execution consists of supplying the board with 5V, and the level adapter with 3.3V. The **j-tag STDOUT** cables for the processor display are then connected to the **RS232/TTL** adapter pins in order to see the processor data on the screen. The correct voltage level of Tx and Rx on the j-tag is verified using an oscilloscope. Once this is done, the microprocessor is plugged in. Once the correct power supply has been evaluated by oscilloscope, it is verified that the microprocessor is sending the initialisation packets to the Tx output. This verification can be seen on the screen by the succession of yellow bands which constitute the automatic post plug-in initialisation packets of the processor. The result of loading the kernel and booting the operating system can be seen on the screen. At this point, the loading of commands via the **Linux** command line can be verified. Due to a cold solder related to the microprocessor, the test was repeated by connecting the microprocessor to a *dev board*.

Test results show that the correct passage of packets out of the microprocessor is verified. Once the system is connected to the dev board, the verification steps are shown on the screen: successful opening of the channels, loading of the kernel, booting of the operating system, and correct execution of commands via the **Linux** terminal.

Communication test

The objective of the communication test between the two boards is to verify the correct communication between the subsystem board and the basic functional board.

Test Set-Up This last functional test was carried out in two set-up configurations: the first using the dev board and the second using the ADCS board directly. The *first set-up* is shown in figure 6.19, where the components used are indicated by:

- **(a)**: laptop device;
- **(b)**: oscilloscope;
- **(c)**: dev board, which allows I/O operations through serial terminal on the connected laptop device;
- **(d)**: battery pack for powering the base board and thus the RD129 microprocessor via connection cables;
- **(e)**: basic functional board;
- **(f)**: RD129 microprocessor;

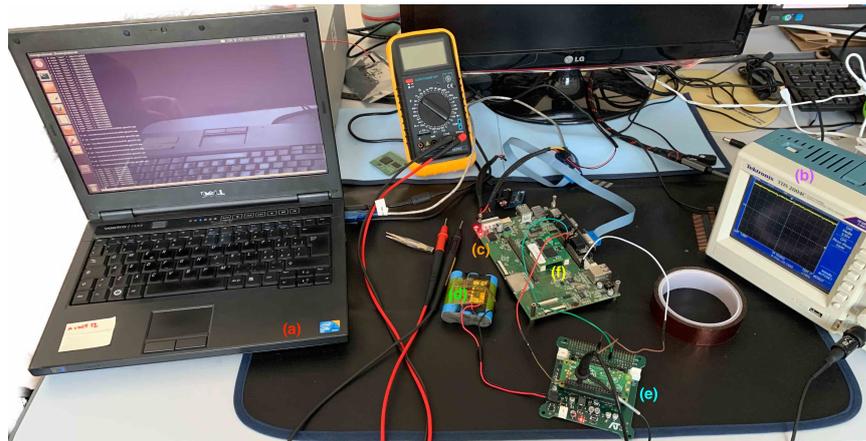


Figure 6.19: First verification set-up for communication test

The *second set-up* is shown in figure 6.20, where the components used are indicated by:

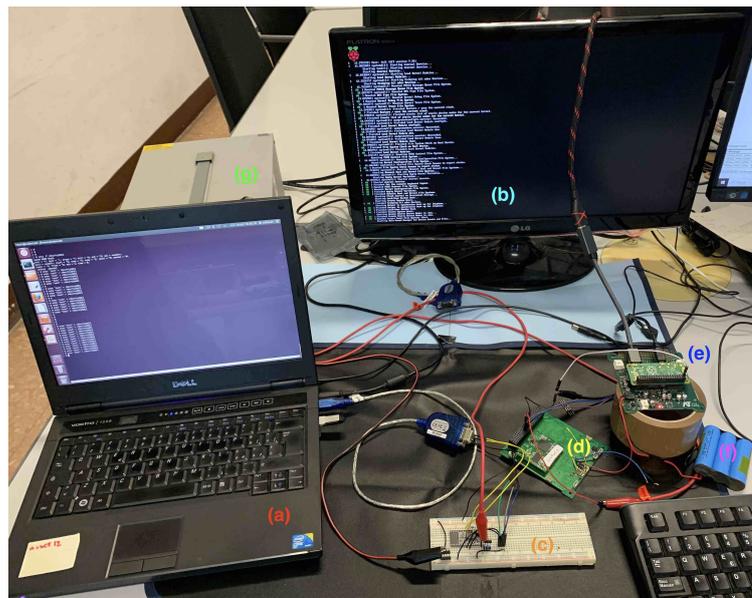


Figure 6.20: Second verification set-up for communication test

- (a): laptop device;
- (b): graphic interface screen to view the output of the microprocessor, connected to the microprocessor via HDMI;
- (c): bread board with RS232 to TTL Serial Adapter;

- **(d)**: ADCS board with RD129 microprocessor integrated;
- **(e)**: basic functional board;
- **(f)**: battery pack for powering the base board and thus the RD129 microprocessor via connection cables;
- **(g)**: Power supply;

In this second configuration, the microprocessor is directly connected to the ADCS board, so the level adapter is introduced again in order to see the output on the screen. At this stage, a remote WiFi connection on is established with the Raspberry, allowing wireless communication with the system through the 'ssh' command and exchange of files with the 'scp' command.

Test execution to provide the read and write functions between the two boards, the Tx of the base board is connected to the Rx of the functional board, and the Rx of the base board is connected to the Tx of the functional base board. The video output of the dev board is connected to the laptop, and the functional base board is powered by connecting batteries. The dev board and the microprocessor are powered by connecting the GND and 5V pins from the base board to the dev board. Communication is verified when data packets, the passage of which is verified by oscilloscope, pass between the two boards in both read and write. The execution of the test with the second set-up configuration is the same, the only difference being that the same connections are made on the dev board's UART channels.

Test results communication is verified from raspberry to the microprocessor on the dev board: the 'TEST' package arrives from the RD129 microprocessor to the Raspberry. With regard to the tests carried out on the ADCS board, the test could not be completed due to a short circuit of the RD129 microprocessor. The functional diagram was therefore rearranged so that the magnetic tests could still be carried out.

6.3.2 Updated boards configuration for final magnetic tests

After the **Communication test** it was necessary to implement a configuration of the two boards such that a functioning system could be realised using only the processor of the basic functional board, bypassing the microprocessor of the subsystem board. Firstly, through careful analysis of the PI0-W's pinout, shown in figure 6.16, it was found that the output of the GPIO 18 pin, which has a PWM function, is available. This pin produces a PWM driver signal between 0V and 3.3V, which must be sent to the ADCS board in order to control the magnetic actuators with variable voltage, and therefore torque, according to the one calculated by the

control law. However, it must be taken into account that the desired voltage for proper actuation must be variable between $-5V$ and $5V$: the necessary circuit for this conversion, and for power supply to the MT, is on the subsystem board.



Figure 6.21: New power circuit



Figure 6.22: ADCS's IMU connections

The connection between the Raspberry GPIO 18 pin to the power circuit was realised through a cable connected to the PWM pin on one end and soldered directly to the proper trace on the subsystem board on the other.

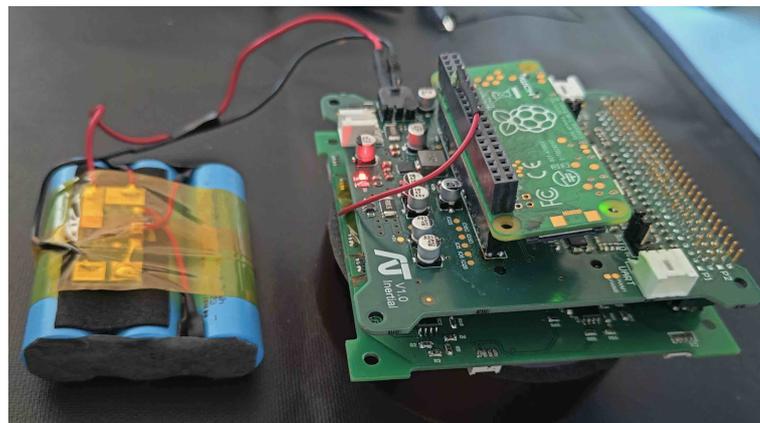


Figure 6.23: Card system updated by applying corrections to bypass RD129

Having solved the problem of powering the actuators, the problem of IMU data transmission to the Raspberry remains. First of all, as the UART signal transmitted to the Raspberry through the 104 connector must be adapted from a signal between $-6V$ and $6V$ to 0 and $3.3V$. The necessary adapter circuit is present on the subsystem board, but its output signal is between 0 and $5V$, as the $3.3V$ regulator chip is inside the RD129. To bypass the microprocessor, the supplied voltage had to be changed soldering a cable carrying the correct voltage to the adapter circuit, and

then soldering two cables connecting the output Tx and Rx traces directly to the corresponding 104 connector pins. This way the IMU data signals travel on the serial channels that were previously assigned to the communication packets between the two boards, allowing the Raspberry to read the measurements. The end board package system is illustrated in figure 6.23. In parallel to these changes, the ADCS code was converted in Python, adapted to the new architecture, and loaded on the Raspberry. Once the above changes have been made on the two boards, verification of the operation of the modified board package is restarted. Tests performed on the new platform are: serial data reception from IMU, PWM signal test, ADCS code functional test on Raspberry. The **serial data reception from IMU** test objective is to verify the transmission of IMU data packet to the Rx UART pin of the PI0-W microprocessor.

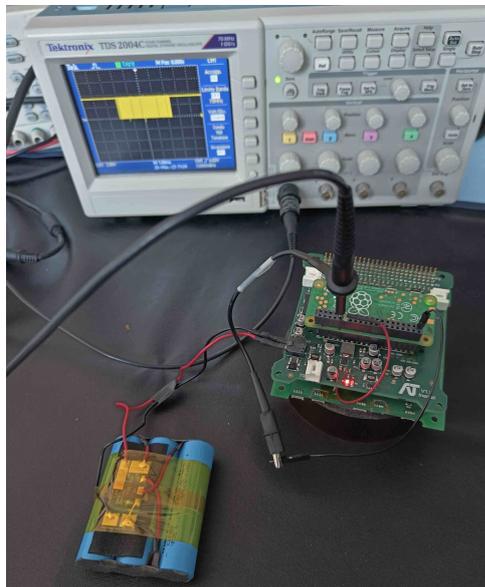


Figure 6.24: Serial data reception from IMU test set up

The set up is reported in figure 6.24, and the execution provides a simple powering of the functional board so that the IMU and the microprocessor are working. As it can be seen in 6.24 data packets are sent from IMU to Raspberry which reads them correctly. As for the **PWM Test**, the objective is to test the output voltage from the PWM GPIO 18 pin of the microprocessor, evaluating its time-varying voltage level. The test set-up is shown in figure 6.25. During the execution of the test, the oscilloscope shows the PWM driver output of the Raspberry, while the multimeter reads the state of the modulated differential voltage entering the magnetic torquer. As can be seen, the modularity of the voltage is verified. The last functional verification test related to the board system is the **ADCS code**

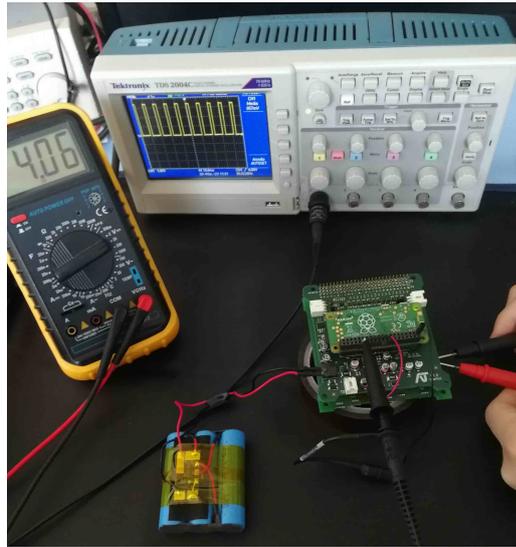


Figure 6.25: PWM signal output

functional test. The aim of this test is to verify that the *Python* code loaded on PI0-W sends the relevant data packets from the IMU to the remote terminal which is the user interface, while the sensor data is read and the actuation command, in form of the PWM signal, is sent. The set-up is shown in figure 6.26.



Figure 6.26: Set up for Python code function test

During execution of the test, the relevant measurement and control data is correctly printed out, verifying correct functioning of the program, the remote communication and the reading of the IMU sensor. On the oscilloscope the command signal to the torquer is seen at the output, verifying the correct output of the actuation

signal. The correct functioning of the ADCS software is thus completed. Once the verification campaign is completed, the boards are integrated into the mechanical interface structure with the test facility of Bologna. The order of integration is outlined below:

- Connect the magnetic torquer to the base of the structural platform, taking care to limit cable mobility;
- Mount the basic functional board with the appropriate spacers
- Ensure the battery pack above the base board, taking care to insulate it appropriately so as not to create shorts;
- Plug in the ADCS board to the base board;
- Secure the external case with the previously mounted pods to the internal structure

6.3.3 Magnetic Test

Once the system constituting the host CubeSat has been implemented, a **Test Readiness Review (TRR)** is carried out following the approach provided by *ESA* [43], including an initial development of the **Assembly, Integration and Test (AIT)** plan. With regard to assembly, it must be remembered that the CubeSat system is transported from Turin to Bologna after integration, so it is necessary to carry out a functional test of the system to verify that no damage has occurred during transport. A post-transport functional test of the platform under test is therefore planned. Once the CubeSat has been verified, integration on the balancing platform is performed. The following step is to perform the balancing procedure described in detail in chapter 5. The last test of the AIV plan carried out is the magnetic actuation one, during which a tuning of the control law gains is carried out and a verification of the achievement of the pointing objective is evaluate. It is important to keep in mind that the purpose of the magnetic control is to bring the platform to have the body axis z vector aligned with the z vector of the inertial reference frame.

Post Transport Functional test

Test Objective of the *Post Transportation Functional test* is to verify that the functionality of the host CubeSat is not deteriorated downstream of the system transport from Turin to Bologna. The test consists of 3 different steps, each of which is necessary to verify a different aspect of the functionality of the system.

- **Step 1:** powering up the functional board and checking the correct supply of power with a multimeter, check of working conditions of the Raspberry Pi;
- **Step 2:** remote connection to the Raspberry Pi, and check of execution of commands on the terminal;
- **Step 3:** functional test, launching appropriate Python scripts;

Test Set Up is shown in figure 6.27. On the right of the picture is the remote

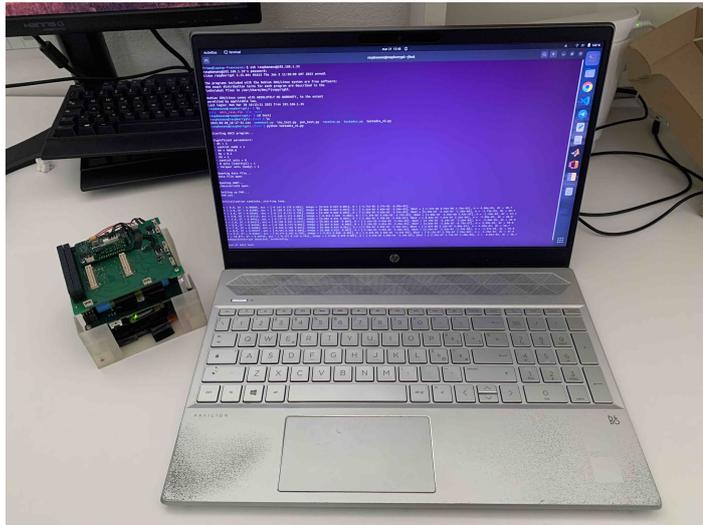


Figure 6.27: Post transfer Functional test set up

user interface *terminal*, while on the left is the *fully assembled* and battery-powered CubeSat system. In order to carry out the above tests, both devices must be connected to the same network.

Test Execution provides that once the basic functional board is powered, the raspberry and the system board, on which the IMU is implemented, are powered. Step 1 is executed connecting the batteries to the functional board. Using a multimeter, supply of 5V and 3.3V is checked on the relevant pins using a multimeter. Then the system is prepared for setup of remote connection with a USB keyboard and a display connected through HDMI cable. This way, verification of correct boot and loading of the operating system is also performed, allowing I/O and execution of commands from the terminal. Step 2 involves connecting the Raspberry Pi to the local WiFi network, and finding it's local ip address running the 'ifconfig' command. Once done, a remote connection can be established with a PC to the Raspberry Pi terminal using the 'ssh' command. Basic commands are executed to

navigate directories and listing files, verifying correct operation of the system. Step 3 involves running similar test scripts as described in 6.3.2, in particular a IMU test script to print on screen the sensor data, a PWM test script which allows the measurement of a correctly modulated differential voltage on the MT connector with a multimeter, and the final test of the complete ADCS code after verification of sensing and actuating components.

Test Results of the post-transport verification of the fully implemented system are positive on all three implemented steps. So, the boards are correctly powered, the supply and communication lines are working, the remote connection is established and therefore the system can be controlled and code executed, and the PI0-W correctly reads data from the IMU and outputs a command signal to the actuators through PWM modulation. Therefore, the complete system is in working order.

Balancing procedure results

Once the functional verification tests on the host CubeSat were completed, it was implemented on the test platform. The first step performed was the manual balancing on the platform. It was immediately apparent that the system was unstable despite the fact that the position of the mass on the z's was set close to the base of the platform, thus favouring the simple pendulum behaviour of the overall system. It can be observed that the mass mock-up used in the 5, is indeed faithful in terms of the total mass of the system (the mock-up weight is **467 g** and the final system weight **469 g**), but it is not representative of the mass distribution. In order to be able to allocate the Raspberry, connected by means of a (bulky) connector to the base board, it was necessary to rethink the allocation of the battery, which was positioned between the functional board and the system board, imposing a greater distance between the two boards than initially planned. Given the considerable weight of the battery, the upward shift of the centre of mass is such that once the CubeSat is implemented on the platform, the system is always unstable. In order to solve this problem, it is necessary to lower the CM by allocating the battery as low as possible. The final configuration of the CubeSat is therefore rethought, switching from the 6.28 to the 6.29 configuration. To make space for the battery at the bottom, the PI0-W is moved and fixed to the outer edge in the middle of the boards, and the necessary pins are connected via jumper cables to the two boards. Once this change had been made, the lowering of CM was sufficient to be able to proceed with the inertia estimation and then the estimation of the residual torque of the platform. This torque is desired to be as low as possible since it is the couple that has to be overcome during the next phase of magnetic actuation.

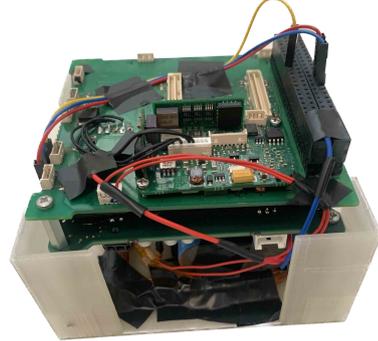
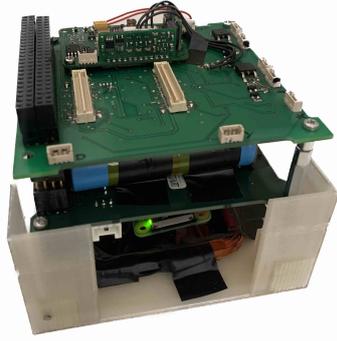


Figure 6.28: High CM configuration **Figure 6.29:** Low CM configuration

The updated inertia matrix obtained by replicating the procedure given in 5 is:

$$J_{est} = \begin{bmatrix} 0.01 & 0 & 0 \\ 0 & 0.009 & 0 \\ 0 & 0 & 0.005 \end{bmatrix} \text{ kg} \cdot \text{m}^2 \quad (6.10)$$

Once the inertia was estimated, the iterative procedure of estimating the vertical offset and correcting it was implemented, until a condition of indeterminate platform behaviour was reached. At that point, the residual torque was estimated, with results shown in figure 6.30. The **maximum disturbance torque** of the facility is estimated to be 5.8×10^{-5} , with a **root mean square** of 4.26×10^{-5} .

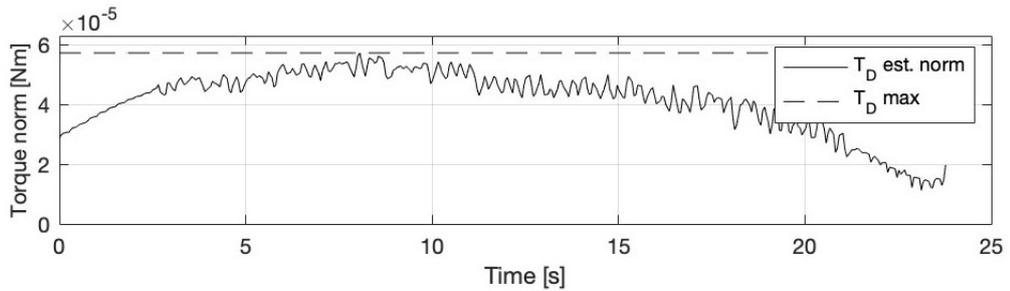


Figure 6.30: Residual Torque estimation

Once the torque characterising the facility and the inertia of the system had been determined, the Simulink model was updated, through which the effectiveness of the magnetic actuation could be verified. By means of matlab simulator, it is possible to determine the initial values of the gains implemented in the PD control law. The initial settled gains are $K_P = 1$ and $K_D = 15$.

Magnetic Control test

Test Objective of magnetic control tests is to verify magnetic actuation. Once the Helmholtz cage is turned on, the desired magnetic field is reproduced along one or more axes, and the PD control law is tested through the **ADCS** program. The aim is to achieve realignment of the test platform to the local vertical by means of magnetic actuation, through rotation about a single axis (in this case the body x-axis).

Test Set Up is shown in the figure 6.31.

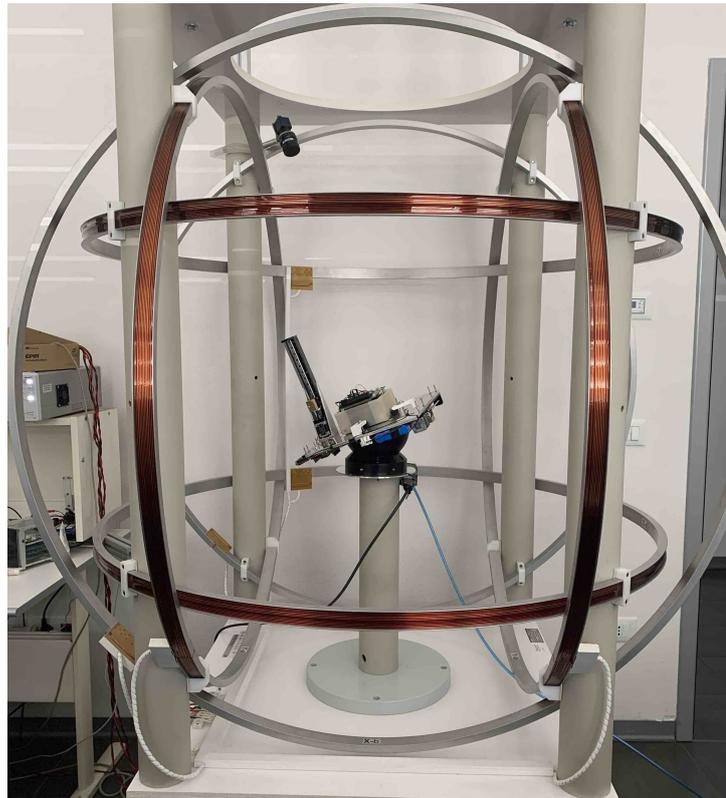


Figure 6.31: Magnetic Control Test set up

The configuration provides for a single-axis control, hence only one torquer is available to generate the control torque. The actuator is arranged so that its magnetic dipole is aligned with the local vertical at the moment when the platform is balanced in the equatorial plane.

Test Execution The procedure involves imposing a magnetic field through a Helmholtz cage. The imposed magnetic field is only in the Y direction of the

inertial reference system (i.e., that of the cage itself). The type of control imposed is indeed a PD type, with proportional action to the control variable and a derivative term, which provides for maximum torque control under equilibrium conditions, that is when the platform is aligned with the local vertical and hence the magnetic dipole is perpendicular to the external magnetic field. After launching the program containing the control law, the magnetic torquer is powered with a variable duty cycle based on the angular position between the body y-axis and the inertial y-axis calculated using data collected from the IMU, which is fed into the control loop. The inertial and body reference frames are aligned, and a 45-degree manual rotation of the platform around the x-axis is applied and released. The system should realign itself to the local vertical within approximately 60 seconds (as simulated) using the control torque exerted by the magnetic torquer.

Test Results indicate that magnetic actuation occurs; however, due to the possible reasons reported below, the resulting effect of actuation is not as desired.

- The ideal execution of the test assumes that there are no rotations out of the x-z plane. This assumption does not correspond to the application reality as inevitably disturbances acting on the facility will cause the platform to move out of the x-z plane;
- The bar used to fixed the masses inside the bearing was of unknown material. The possible non a-magnetic material of the bar may have given a magnetic torque (this was confirmed later) which reduced the effect of the torque generated by the actuator;
- The estimate of residual torque is inaccurate;
- The torquer generates a lower dipole than that estimated during the preliminary sizing procedure;

The reasons are not mutually exclusive. However, the proper functionality of the torquer was further verified by manually applying the maximum positive and negative tension to the actuator via command line. MT functioning in both direction is verified as its magnetic field is picked up by the on-board IMU and movement of the platform can be observed. In order to demonstrate the effectiveness of the magnetic control law and the actual operation of the actuator, two variations are implemented: further residual torque minimization operations are performed, and secondly, the test objective is modified to minimize the influence of off-axis actuation oscillations.

Modified Magnetic Control test

Test Objective of the magnetic control test is to realign the system with the direction of the magnetic field by rotating the system around a single axis, namely around the inertial z-axis (in this case coinciding with the body z-axis).

Test Set Up is shown in figure 6.32.

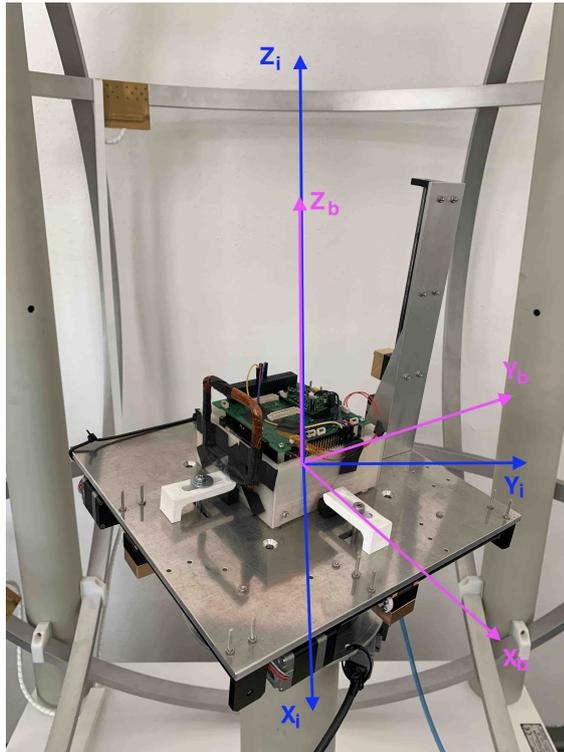


Figure 6.32: Modified Magnetic Control test set up

It can be noted that it was necessary to reposition the magnetic torquer so that its axis was aligned with the body y-axis. Furthermore, in order to reduce oscillations outside the plane as much as possible, the balancing mass in the z-direction was lowered, making the system a strongly stable pendulum. This solution allows for obtaining a system with extremely high realignment torque to the local vertical, but this does not affect the actuation as a pure rotation around the z-axis is desired.

Test Execution It is planned that the Helmholtz cage is activated with a magnetic field value in the positive x-direction equal to 5 Gauss. The platform is placed so that the axis of the actuator forms an angle of 45 degrees with the direction of the

magnetic field. The control law remains mathematically the same as the previous test allowing rotational control of the platform around the z-axis until the axis of the torquer is aligned with the inertial y-direction. The control law was changed as follows:

$$m_y(t) = K_d * \omega_z(t) - K_p * \theta_z(t) \quad (6.11)$$

where $\theta_z = \arcsin(B_y/|\mathbf{B}|)$.

Test Results The results are consistent with what was expected. However, a tuning operation of the gains was necessary in order to have a contrast of the platform's inertia, which tends to make the platform rotate beyond the equilibrium point. After the tuning, the gain values of $K_P = 5$ and $K_D = 500$ were set. Despite the reduction in overshoot, it is still present. A finer tuning operation should be planned for future applications, however, the purpose of the present test has been achieved: the control law needs to be refined, but its effectiveness has been verified. The results obtained from the post-processing of the IMU data are reported below.

In figure 6.33 the duty cycle commanded by the control law to the actuator is

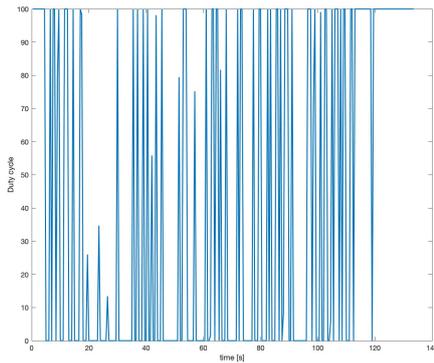


Figure 6.33: MT duty cycle

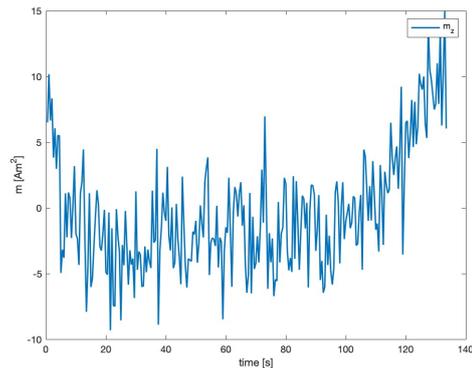


Figure 6.34: MT Magnetic dipole

shown. The duty cycle is defined in this case as $\mathbf{dc} = m/m_{max}$: output differential voltage, as modulated by the circuit on the subsystem board, ranges from -5V, corresponding to a dc of 0%, and +5V, corresponding to a dc of 100%. Therefore, at the beginning of the test, dc should be 100, while it will tend to return to 0 when realigned with the inertial Y direction. Based on \mathbf{dc} , there is a variable magnetic dipole, which is shown in figure 6.34. It is noted that the dipole has a positive sign at the beginning of the test, in order to accelerate the platform, and becomes negative when the system has to be decelerated. With regard to the figure 6.35, the three components of the gravity vector can be observed. It can be seen that as

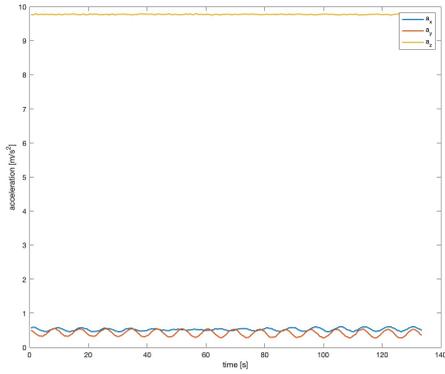


Figure 6.35: g components

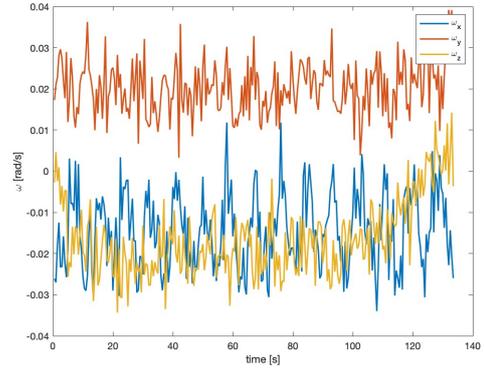


Figure 6.36: Angular velocities

the component g_z is equal to $9.81[m/s^2]$, i.e. the platform is aligned to the local vertical. The components g_x and g_y should ideally be zero, however due to the pendulum's imperfect stability condition in the x-y plane, out-of-plane oscillations occur which are shown in the figure. The same phenomenon is found in figure 6.36, in which the angular velocities on the 3 axes are shown. It can be seen that ideally there should only be one rotation around the z-axis, so only ω_z should be non-zero. However, due to the out-of-plane oscillations, it can be seen that the angular velocities ω_x and ω_y also have a non-zero value.

Chapter 7

Conclusions

The work presented in this thesis aims to introduce and investigate the problem related to the attitude control of nanosatellites, specifically CubeSats. Due to their increasing use, the optimization of the attitude control subsystem is of fundamental importance to improve the performance and overall success of CubeSat missions. In order to have a high-performing ADCS subsystem, it is necessary to verify its pre-launch characteristics. To do this, the system must be subjected to an adequate verification campaign, which is strongly correlated with the availability of facilities that allow the reproduction of the environmental conditions and disturbances that characterize the typical orbits in which nanosatellites are applied.

In the first part of the thesis, the subsystem under analysis was introduced, presenting its complexity and state of the art. The problem of attitude control was subsequently introduced, starting from the fundamental reference systems and continuing to evaluate the disturbance couples characterizing the space environment for a low Earth orbit. In order to carry out a verification campaign on the ADCS subsystem, it is necessary to reproduce the space operational environment, and to this end, the elements implemented in the test facility for nanosatellite subsystems at the Microsatellite Laboratory of the University of Bologna were introduced. The first objective of the work is to adapt the already implemented facility to improve its performance and enable its housing on the ADCS subsystem platform compliant with CubeSat standards for systems with dimensions between 1U and 3U. To do this, a mechanical interface composed of 4 L-shaped pods was integrated into the facility, which allows hosting a system with a maximum size of 110x115 cm, thus expanding the applicability of the facility to any external entity that ensures that its test system has adequate external dimensions. Furthermore, in order to improve the facility's performance in terms of reducing the residual gravitational disturbance couple generated by the vertical offset between CM and CR, stepper motors with a greater displacement definition were implemented. This design variation allowed

for more precise balancing in the plane and out of the plane than achieved with the previously used stepper motors, thus reducing the residual couple to values of the order of 10^{-5} , as required for facilities of this type. To further increase the balancing precision, a PCB board was designed that integrated the electronic components present on the platform, thus ensuring the reduction of moving components that could cause variations in the mass distribution of the platform during the balancing procedure and thus a reduction in performance. Once the performance of the new configuration of the test facility was verified, the second main objective of this study was pursued: to test a functioning ADCS subsystem integrated into the test facility. To accomplish this, a basic functional board was designed to carry out the functions of OBC, ComSys, and EPS, and subsequently sent into production. After evaluating the characteristics of an air core magnetic actuator already available in-house, it was found to provide a maximum dipole of 0.3 Am^2 which, coupled with a magnetic field generated by the Helmholtz cage of 5 Gauss, allows for a control torque of approximately $1.5 \times 10^{-4} \text{ Nm}^2$, deemed sufficient to operate the system and counteract the disturbance torque of the facility. Once the torquer sizing was completed, the package of functional and system boards was tested through bench verifications, which aimed to verify the microprocessors' operation on both boards and the communication of data from the system board's IMU to the basic functional board. Despite technical issues due to the system board's microprocessor failure, a new system configuration was proposed, deciding to directly control the system board with the microprocessor of the basic functional board. To verify the feasibility of the proposed solution, the bench verification procedure was repeated, resulting in a proper functioning of the board package.

Finally, the complete ADCS system, implemented within the CubeSat standard mechanical interface, was fixed onto the test facility to verify the implemented single-axis magnetic control laws. Starting from the verification of the PD law, a setup was created to verify the control effectiveness around the body's x-axis, using a single actuator positioned with the axis aligned to the z-body. The verification test involves tilting the system in the x-z plane by 45 degrees and, through a PD closed loop, controlling the appropriate angular moment to realign the inertial vertical to the local one. Due to the single actuator implementation on the platform, only one axis can be controlled, causing a loss of control for any oscillation outside the x-z body plane. The platform realization involves inevitable oscillations outside the x-z plane, making the magnetic control ineffective. Therefore, the magnetic test configuration was changed: moving the magnetic actuator so that its axis is aligned with the y-body direction, the system's orientation is controlled by inducing a rotation around the inertial z-axis coinciding with the z-body axis. To cope with the oscillations outside the plane, the facility's characteristics are exploited: by bringing the z-balancing mass to its lower limit, the platform becomes a simple

pendulum, which tends to return to the equilibrium condition in which the system is aligned with the local vertical. In this way, once balanced in the plane, the system is stable, and the magnetic actuation is more effective. It is verified that aligning the platform at a 45-degree angle with respect to the magnetic field, imposed in the inertial x-direction, with the MT axis aligned with the y-body axis, thus 45 degrees relative to the inertial axis, results in an effective orientation of the platform to coincide with the inertial and body y-axes.

Despite the fact that the implemented control law is extremely simple and a more refined tuning job is certainly necessary, this thesis has effectively laid the foundation for the verification of the ADCS subsystem. Specifically, an improved facility has been constructed with superior performance compared to previous configurations, which can efficiently accommodate external systems within the dimensional limits imposed. A functioning ADCS subsystem has been created, fully remote-controlled and independent of external sources other than the mechanical interface that fixes it to the facility. Finally, it has been demonstrated that the implementation of both systems has been successfully executed.

Therefore, this thesis provides a solid foundation for expanding the testing and verification capabilities for the attitude control subsystem of nanosatellite systems. This not only allows for experimental verification of the system's effectiveness and control laws, but also enables the testing of various off-nominal configurations, thereby reducing system failures. A future development of great interest involves the implementation of actuators on all three axes of the facility, thereby providing complete control and enabling a multitude of tests to verify control laws that are much more complex than the ones proposed thus far.

Bibliography

- [1] NASA CubeSat Launch Initiative et al. «Cubesat 101: basic concepts and processes for first-time CubeSat developers». In: *no. October* (2017), p. 96 (cit. on p. 1).
- [2] Turtogtokh Tumenjargal and Mengu Cho. «Standardized, flexible interface design for a CubeSat bus system». PhD thesis. 2019 (cit. on p. 1).
- [3] Adriano Camps. «Nanosatellites and applications to commercial and scientific missions». In: *Satell. Mission. Technol. Geosci* (2020), pp. 145–169 (cit. on p. 2).
- [4] Martin Sweeting. «Modern small satellites-changing the economics of space». In: *Proceedings of the IEEE* 106.3 (2018), pp. 343–361 (cit. on p. 2).
- [5] James Richard Wertz, David F Everett, and Jeffery John Puschell. *Space mission engineering: the new SMAD. Number v. 28 in Space technology library*. 2011 (cit. on p. 3).
- [6] Chantal Cappelletti, Simone Battistini, and Benjamin Malphrus. *Cubesat handbook: From mission design to operations*. Academic Press, 2020 (cit. on p. 3).
- [7] Armen Toorian, Ken Diaz, and Simon Lee. «The cubesat approach to space access». In: *2008 IEEE aerospace conference*. IEEE. 2008, pp. 1–14 (cit. on p. 3).
- [8] Ana Carolina Cabral Pimentel de Melo. «A benchmark for assessing nanosatellite on-board computer power-efficiency and performance». In: (2019) (cit. on p. 3).
- [9] Alexander Kramer, Philip Bangert, and Klaus Schilling. «UWE-4: First Electric Propulsion on a 1U CubeSat—In-Orbit Experiments and Characterization». In: *Aerospace* 7.7 (2020), p. 98 (cit. on p. 3).
- [10] Yeona Yoo, Seungkeun Kim, Jinyoung Suk, and Jongrae Kim. «Attitude control system design & verification for CNUSAIL-1 with solar/drag sail». In: *International Journal of Aeronautical and Space Sciences* 17.4 (2016), pp. 579–592 (cit. on p. 4).

- [11] Xiwang Xia, Guowen Sun, Keke Zhang, Shufan Wu, Tian Wang, Lei Xia, and Shanwu Liu. «Nanosats/cubesats adcs survey». In: *2017 29th Chinese Control And Decision Conference (CCDC)*. IEEE. 2017, pp. 5151–5158 (cit. on p. 4).
- [12] *Satsearch*. URL: <https://satsearch.co/products/cubespace-cubeadcs-3-axis> (visited on 10/21/2022) (cit. on p. 5).
- [13] Steven P Neeck, Thomas J Magner, and Granville E Paules. «NASA’s small satellite missions for Earth observation». In: *Acta Astronautica* 56.1-2 (2005), pp. 187–192 (cit. on p. 4).
- [14] Stephen A Jacklin. *Small-satellite mission failure rates*. Tech. rep. 2019 (cit. on p. 5).
- [15] Martin Langer and Jasper Bouwmeester. «Reliability of cubesats-statistical data, developers’ beliefs and the way forward». In: (2016) (cit. on pp. 5, 6).
- [16] *ECSS-E-ST-10-03C Rev.1 – Testing*. URL: <https://ecss.nl/standard/ecss-e-st-10-03c-rev-1-testing-31-may-2022/> (cit. on p. 6).
- [17] P Wang, J Yee, and F Hadaegh. «Experimental study of synchronized rotation of multiple autonomous spacecraft with rule-based controls». In: *Guidance, Navigation, and Control Conference and Exhibit*. 2001, p. 4150 (cit. on p. 6).
- [18] Nemanja Jovanovic, Joshua M Pearce, and Jaan Praks. «Design and Testing of a Low-Cost, Open Source, 3-D Printed Air-Bearing-Based Attitude Simulator for CubeSat Satellites». In: *Journal of Small Satellites* 8.2 (2019), pp. 859–880 (cit. on p. 6).
- [19] C Papakonstantinou, G Moraitis, V Lappas, and V Kostopoulos. *Design of a Low-Cost Air Bearing Testbed for Nano CMG Maneuvers. Aerospace 2022*, 9, 95. 2022 (cit. on p. 6).
- [20] Irina Gavrilovich, Sébastien Krut, Marc Gouttefarde, François Pierrot, and Laurent Dusseau. «Robotic test bench for CubeSat ground testing: Concept and satellite dynamic parameter identification». In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2015, pp. 5447–5453 (cit. on p. 6).
- [21] Simone Chesi, Octavio Perez, and Marcello Romano. «A dynamic, hardware-in-the-loop, three-axis simulator of spacecraft attitude maneuvering with nanosatellite dimensions». In: *Journal of Small Satellites* 4.1 (2015), pp. 315–328 (cit. on p. 7).
- [22] Rodrigo Cardoso da Silva, Igor Seiiti Kinoshita Ishioka, Chantal Cappelletti, Simone Battistini, and Renato Alves Borges. «Helmholtz cage design and validation for nanosatellites HWIL testing». In: *IEEE Transactions on Aerospace and Electronic Systems* 55.6 (2019), pp. 3050–3061 (cit. on p. 7).

- [23] Irina Gavrilovich. «Development of a robotic system for CubeSat Attitude Determination and Control System ground tests». PhD thesis. Université Montpellier, 2016 (cit. on p. 7).
- [24] F Landis Markley and John L Crassidis. «Fundamentals of spacecraft attitude determination and control». In: (2014) (cit. on p. 9).
- [25] Giulio Avanzini. «Spacecraft Attitude Dynamics and Control». In: *Politecnico di Torino* 9 (2008), p. 13 (cit. on p. 13).
- [26] Anton Bahu. «Development of a ground testing facility and attitude control for magnetically actuated nanosatellites». In: (2021) (cit. on pp. 21, 75).
- [27] Irina Gavrilovich, Sébastien Krut, Marc Gouttefarde, François Pierrot, and Laurent Dusseau. «Test bench for nanosatellite attitude determination and control system ground tests». In: *4S: Small Satellites Systems and Services Symposium*. 2014 (cit. on p. 23).
- [28] *Solar Radiation and Photosynthetically Active Radiation*. URL: <https://www.fondriest.com/environmental-measurements/parameters/weather/photosynthetically-active-radiation/> (cit. on p. 26).
- [29] LUCA MARIANI. «Design and development of a small satellites three axis attitude simulation platform». In: (2019) (cit. on p. 27).
- [30] *V shape project model of system engineering*. URL: <https://www.slideteam.net/v-shape-project-model-of-system-engineering.html> (cit. on p. 45).
- [31] *Bipolar Stepper Motor with H bridge functioning scheme*. URL: https://www.researchgate.net/figure/A-Bipolar-Stepper-Motor-B-H-Bridge-stop-position-C-H-Bridge-energised-A-D_fig5_366326832 (cit. on p. 47).
- [32] Ltd. DINGS' ELECTRICAL MECHANICAL Co. «ENGINEERING AND MANUFACTURING FOR MOTION CONTROL PRODUCTS». In: (2020) (cit. on p. 49).
- [33] *Stepper motor driver – explanation*. URL: <https://3dcncafrica.co.za/stepper-motor-driver-explanation/> (cit. on p. 50).
- [34] *DRV8833 Dual H-Bridge Motor Driver*. URL: <https://www.ti.com/lit/ds/symlink/drv8833.pdf> (cit. on p. 51).
- [35] Anton Bahu and Dario Modenini. «Automatic mass balancing system for a dynamic CubeSat attitude simulator: development and experimental validation». In: *CEAS Space Journal* 12 (2020), pp. 597–611 (cit. on p. 64).

- [36] Simone Chesi, Qi Gong, Veronica Pellegrini, Roberto Cristi, and Marcello Romano. «Automatic mass balancing of a spacecraft three-axis simulator: Analysis and experimentation». In: *Journal of Guidance, Control, and Dynamics* 37.1 (2014), pp. 197–206 (cit. on p. 65).
- [37] Sundaramoorthy Rajasekaran. «Structural dynamics of earthquake engineering: theory and application using MATHEMATICA and MATLAB». In: (2009) (cit. on p. 68).
- [38] Aekjira Kuyyakanont, Suwat Kuntanapreeda, and Nisai H Fuengwarodsakul. «On verifying magnetic dipole moment of a magnetic torquer by experiments». In: *IOP Conference Series: Materials Science and Engineering*. Vol. 297. 1. IOP Publishing. 2018, p. 012011 (cit. on p. 82).
- [39] *iMTQ Magnetorquer Board*. URL: <https://www.isispace.nl/product/isis-magnetorquer-board/> (cit. on p. 84).
- [40] Shoaib Ahmed Khan, Anwar Ali, Yang Shiyong, Jijun Tong, and Josep M Guerrero. «Optimized design of embedded air coil for small satellites with various dimensions». In: *Journal of Aerospace Information Systems* 18.5 (2021), pp. 269–279 (cit. on p. 84).
- [41] Alessandro Zavoli, Fabrizio Giuliotti, Giulio Avanzini, and Guido De Matteis. «Spacecraft dynamics under the action of Y-dot magnetic control law». In: *Acta Astronautica* 122 (2016), pp. 146–158 (cit. on p. 93).
- [42] *Raspberry PI Zero pinout, features and Specifications*. URL: <https://www.electrorules.com/raspberry-pi-zero-pinout-features-and-specifications/> (visited on 01/12/2022) (cit. on p. 98).
- [43] *ECSS-E-ST-20-01C-DIR1 : Public review from 27 February - 26 April 2019*. URL: https://ecss.nl/item/?glossary_id=4633 (cit. on p. 107).