UNIVERSITAT POLITÈCNICA
DE CATALUNYA

MASTER THESIS

Development of Attitude Determination Control System for nanosatellites

Emilio Liaño de la Fuente

DIRECTED BY
ADRIANO JOSE CAMPS CARMONA

TUTORED BY
JORDI GUTIÉRREZ CABELLO

Universitat Politècnica de Catalunya
Master Aerospace Science & Technology
July 2022

Development of Attitude Determination Control System for nanosatellites

BY
Emilio Liaño de la Fuente

DIPLOMA THESIS FOR DEGREE
Master in Aerospace Science and Technology

AT
Universitat Politècnica de Catalunya

DIRECTED BY:
Adriano Jose Camps Carmona
Signal Theory and Communications Department

TUTORED BY:
Jordi Gutiérrez Cabello
Physics department

Acknowledgment

To my Cat, my family and my friends. They have always been there to support me through this journey.

Thanks to all the UPC members involved in the development of this thesis. The NanoSat Lab, specially Adriano as the head of the lab and the ^{3}Cat-8 project. Also Jordi, the professor supervising it and the person who help me find this field.

To my colleagues in the AOCS team, without them I wouldn't have been able to do as much work as fast as we have done. Oscar and Pol, I am sure you will achieve great things. It was a pleasure to share both, tasks and beers.

Finally, to all the people who has ever worked in the space field. You inloopd and motivated me to make this change in my career, and I have found in the space industry my real passion. Whether a famous astronaut, or the anonymous assembly technician, your achievements have inloopd me, and will hopefully inloop many people in the future.

Abstract

The Attitude Determination and Control System, or ADCS, is an essential part of nearly every satellite mission. In a CubeSat, there are additional constrains to the system. This thesis presents a design of the system for the first stage of a mission in the NanoSat Lab, the ^{3}Cat-8.

As the determination system was previously developed on the NanoSat Lab, there is a special focus on the control algorithm and the sizing of the actuators. For the control algorithm, it is proposed a classic control based on quaternions. The actuators selected are 3 magnetorquers, and a single reaction wheel. The final solution is tested an presented in the expected orbit conditions of the mission.

Table of Contents

List of Figures

Introduction

CubeSats have become very popular for various applications. They are specially abundant in innovation due to their lower cost, as compared to other satellites, shorter developing time and reduced launch cost [2]. According to the Nanosats Database, up to the first of June of 2022, there have been 1862 CubeSats launched [3]. This have resulted in an increase of the commercial activity around CubeSats and small launches [4, 5].

In this environment, the motivation for missions trying new CubeSat technologies is clear. On one side, innovation and education missions, lead by universities and other research centers are interested to test the newest technologies possible. On the other, companies in the sector want a qualification of their new modules in space environment, also, obtaining flight heritage for their modules in the process.

For these missions, the limitation of space and the characteristics of the payloads usually require systems different from the traditionally used in satellites. In the case of control, this means that a full scale attitude control system is not possible to implement [6]. This forces the development of specific solutions for different missions.

The control systems have two main components, the actuators, and the control algorithm. For CubeSats, the actuators that have been used more extensively are, either magnetorquers or reaction wheels [7]. The control algorithm used is the PD or PID, but investigation with modifications of these controllers have been already done, like the application of fuzzy controllers [8].

The attitude control is dependant on the information it receives, the attitude determination. In this case, the determination algorithm used is the Optimal Request developed for the ^{3}Cat-4 mission in the NanoSat Lab [9]. The sensors are a set of magnetometers in the 3 axes, a 3-axes gyroscope, and Sun sensors in the 6 faces of the CubeSat.

Chapter 1

Definition of the mission

The development of the ADCS considered in this work, is within the frame of the mission ^{3}Cat-8, developed by the NanoSat Lab in the UPC. The requirements of the mission set the foundations for the ADCS design. This mission is a technical and scientific demonstration. As such, the main goal is to successfully use and study technologies and techniques for the first time.

1.1 General objectives

The mission has four main goals:

1. The deployment of two PocketQubes.
2. Orbit rising maneuver with a new thruster for CubeSats.
3. The study of electromagnetic events in the atmosphere.
4. Deployment of a triangular antenna behind the satellite.

The deployment of the PocketQubes will be done using a spring in the rear of the satellite. Once the satellite is pointing to nadir, the cables restraining the spring will be melt, and the spring will push the PocketQubes out of the satellite at an approximated speed of 2 m/s relative to the satellite.

Once the PocketQubes have been released, the ion thruster, on the tip of the spring, will stay exposed for the rest of the mission. This allows for its use raising the orbit. Not only will be the first flight of this thruster, but also a successful raising will extend the life span of the mission. This is very important due to the high drag expected from the deployed antenna.

The study of the electromagnetic fields will be done using a hyperspectral camera, and the deployed antenna. The main events that are meant to be studied are sprites, the large-scale electric discharges that occur high above thunderstorm clouds, and the aurora borealis in the polar regions.

The study of these events using cameras has been already undertaken by other research groups (e.g.[10]). The main role of the camera is to detect these events happening in front of the satellite, so they can be measured with the antenna once the location is behind the satellite.

For these measurements the deployment of the antenna has to be successful. After the orbit raising, the thruster will be turned off to deploy the antenna. The antenna will be nested in a triangular net of roughly one meter of side. Considering that the frontal face of the satellite is 10 x 20 mm, the area of the antenna will significantly increase the drag of the satellite. This resulting in two main consequences, a reduced life span of the mission, and a high momentum produced by the atmospheric drag.

The orbit of the satellite is defined by the constrains stemming from these objectives. Crossing the poles requires a polar orbit. Using the body frame axes defined by the 6U CubeSat Design Specification Revision 1.0 [1], shown in Figure 1.1. The faces of the satellite in the Z axis are used for the payloads, so the solar panels could only be allocated in the X or Y axis faces. As the faces in the Y axes are wider, those were selected to have the solar panels. To ensure the panels will be facing the Sun, a Sun-synchronous orbit with the Y axis perpendicular to the sun vector is needed.

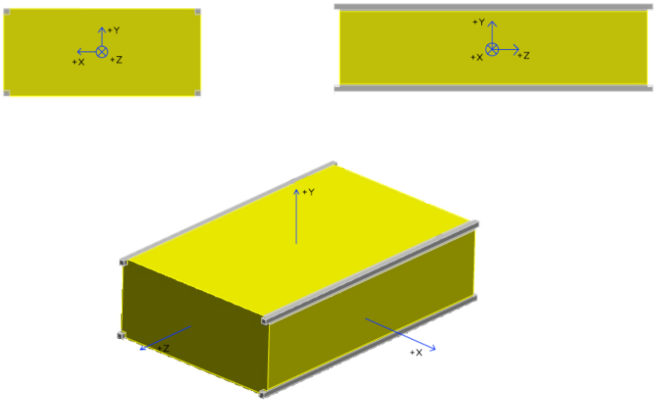


Figure 1.1: 6U CubeSat Design Specification Drawing [1]

1.2 ADCS objectives

The objective of the ADCS subsystem is to provide the mission with controllers that allow the successful accomplishment of the mission goals. In the case of the ^{3}Cat-8 mission, the most important objective is to maintain nadir pointing during the development of the mission objectives. Which means the vertical axis of the satellite points to the Earth orthogonally.

The nadir pointing in the X axes is required by the camera in the perpendicular axes, that has to point to the horizon during the mission. Also, the antenna communicating with the ground station is in X axis to allow communications while doing the nadir pointing. Despite this mode being required during all the mission the controllers will have to change, as the conditions of the system will change from the original stowed CubeSat, to the configuration with the thruster activated or the antenna deployed.

Additionally to the nadir pointing, other control modes could be required in different events:

1. The use of a reaction wheel, requires a de-saturation mode.
2. In the event of low battery levels, the satellite will start Sun pointing.
3. Finally, in the case that the thruster is not aligned with the Z axis, a mode to set the thrust vector and the speed vector in the same direction will be designed.

The de-saturation mode has two aims: to prevent reaching too high or too low angular rates, and also to ensure an efficient consumption of power. The reaction wheel was added to control the rotation of the Z axes. When the magnetic field is parallel to one of the axes of the magnetorquers, the rotation in that axes is impossible to achieve, and therefore one Degree of Freedom (DoF) is lost. Due to the polar orbit of the mission, when crossing the equator the magnetic field will be nearly parallel to the Z axis. The reaction wheel will not be necessary out of this zone, and so the de-saturation mode will be enabled whenever the reaction wheel is not in use.

In the case of a perfect polar Sun-synchronous orbit, nadir pointing mode will be equivalent to Sun pointing. Expecting the Orbit to not be ideal, the Sun pointing mode will trade off a perfect nadir pointing to achieve perpendicularity between the sunlight and the panels. This will only be needed to recharge batteries in case the power consumption is too high.

The Thrust Vector Pointing, TVP, is a mode aiming to maximize the orbit raising when the thruster is working. In the case of a thrust vector in the same direction of the velocity, this mode will act the same as the nadir pointing. Being the thruster off-center, this generates a momentum on the satellite. To minimize the torque generated, the thruster could be rotated towards the center of mass. In this case, this mode will rotate the satellite, so the thrust vector can be made parallel again with the velocity vector.

Chapter 2

Simulation

The physical testing of a satellite is difficult and expensive, and this is also the case for CubeSats [6]. Then, a simulation is needed as an approximation before the real implementation of the control system, thus allowing for a cheap, easy to replicate, and iterative method to calculate the variables of the actuators and the controller. The accuracy of this simulation is critical for the results to be usable later on physical test benches.

The simulation is composed of several elements:

1. The environment configuration, which includes the configuration of the orbit environment and the properties of the satellite itself.
2. The attitude determination, for the sensor readings, and the determination algorithm.
3. The control system, that calculates the action required by the control algorithm and the result of the actuators.
4. A disturbances model, including all the environmental disturbances that affect the satellite every iteration.
5. A dynamic solver, that calculates the new position and angular rate of the satellite based on all the information previously calculated.

2.1 Environment configuration

The configuration parameters are created before the simulation, stored in files ’*.mat’, and then loaded for the simulation. This was made to separate the process of calculating these parameters from the simulation itself, thus reducing the simulation time.

The configuration of the environment is divided in two main parts: The orbit propagation and the satellite configuration.

2.1.1 Orbit propagation

In the orbit propagation it is included the calculation of all the parameters of the orbit, and the environment in that orbit. First, the orbit is defined using a TLE (Two Lines Element), a date for the simulation, the time between points, and the number of revolutions.

Using this data, the difference of time between the simulation and the epoch of the TLE is calculated. Then, the propagation of the orbit in the ECI (Earth Centered Inertia) frame is calculated using the SGP4 model [11]. The ECI frame and Orbit frame are the coordinates systems used in the simulation, but the ECI coordinates are also transformed to ECEF (Earth Centered Fixed Earth) and LLH (Longitude Latitude Height) frames for later calculations.

With the propagated orbit, the environment in each of the points of the orbit is calculated. The atmospheric density is determined using the NRLMSISE-00 model [12]. The magnetic field is calculated using the IGRF-13 model [13]. The solar irradiance is obtained considering the air mass equal to 0, EAM0. Finally, the Sun and Moon position, and the eclipse information of the orbit, are calculated using the Julian Date and the position of the orbit.

2.1.2 Satellite configuration

The satellite has three possible configurations. Initially stowed, representing the satellite in the beginning of the mission, before anything has been deployed. Stowed without PocketQubes, after the satellite has ejected the two masses out. Last, the Antenna deployed, as the final stage of the mission.

The dimensions for all the configurations are those of a 6U CubeSat, $20 \times 10 \times 30$ cm in the axes X, Y and Z. The mass of the satellite structure and permanent components is estimated to be 6 kg, as the mechanical design, and then it is not completely finished is not possible to use more accurate data. The mass of the PocketQubes is of 0.25 kg each, which is 0.5 kg in total. Finally, the mass of the antenna and its support structure is estimated at 2 kg.

For the initial stowed configuration, it is considered that the mass is of 9 kg, and that the center of masses is in the geometric center of the satellite. Then the inertia tensor and the areas of the satellite are calculated, assuming all parts as uniform solid bodies. This will be the configuration used for the simulations of detumbling and initial nadir pointing.

In the deployed configuration, the PocketQubes have already been ejected out of the satellite, so the total is mass is down to 8 kg. Also the antenna and its structure of three cylinders are out of the satellite, and are separately accounted for the calculation of the center of mass the body of the satellite and those. The changes in the allocation of masses also affect the calculation of the inertia tensor.

The configuration stowed without PocketQubes, was considered out of the scope of this report. Being a secondary stage of the initially stowed configuration, it can be developed later.

2.2 Attitude determination

The real attitude of the satellite is initially set as a parameter, and later calculated by the dynamic solver. Nevertheless, the control system can not access directly to these real variables, they can only be read through the sensors. After the sensors reading of these variables, a determination algorithm reduces the errors caused by noise and other disturbances.

2.2.1 Sensors readings

The sensors are observing the system from the body frame. So, the first step is to convert the variables in ECI and orbit frames to the body frame. From the propagation of the orbit, the magnetic, Sun, and Moon vectors are calculated in body frame. Also, the quaternion and the angular rate of the satellite are calculated in this frame.

The reading of the sensors is simulated using these vectors and adding the bias, an additive white Gaussian noise, and the thermal drift. All these options can be disabled in the simulation. Eliminating random noise is used to get deterministic simulations, but is later added to test the robustness of the control algorithm.

A set of unitary vectors is required: for the magnetometer, the magnetic vector; for the gyroscope, the angular rate of the satellite; and, for the Sun sensors, the Sun vector and whether if there is eclipse or not is used. These last sensors are used to calculated the current attitude of the satellite.

2.2.2 Optimal request

Optimal request is a recursive determination algorithm for least-square estimation of the quaternion of a rigid body using the vector measurements [14]. The algorithm was developed in previous works in the NanoSat Lab, and is implemented as it is described in the MSc Thesis by Ángel Navarro [15].

This kind of observer uses the values in the previous iterations to calculate the new quaternion of this step. Some changes have been added to the code to reduce the execution time.

2.3 Control system

The control system uses the information of the determination system to calculate the appropriate response, and execute it through the actuators.

2.3.1 Control algorithm

The control selects the mode based on the inputs. Both, a manual selection of the mode, and automatic change between modes are possible. The selected mode then calculates the ideal torque required from the actuators.

The algorithm control will be discussed in depth in chapter 3.

2.3.2 Actuators

Being the actuators the magnetometers, the ideal torque is translated into the magnetic dipole that need to be generated, using the magnetic field read by the magnetometer. Then, this magnetic dipole generated by the magnetorquers interacts with the magnetic field of the Earth, to get the real torque applied to the satellite.

In the case of the reaction wheel, the required torque is translated into angular acceleration using the inertia tensor of the wheel. Due to the higher current consumption, the use of the reaction wheel is reduced as much as possible.

The equations of the actuators, and how they are being characterized for the mission, will be discussed in chapter 4.

2.4 Disturbances

For the simulation, there has been four disturbances considered affecting the attitude:

1. The disturbance of the gravity gradient.
2. The torque generated by the aerodynamic forces.
3. The solar radiation pressure.
4. The torque generated by the residual magnetic torque.

The torque generated by the gravity gradient is calculated as it is shown in (2.1), where μ is the Earth gravitational parameter, $\vec{r}$ is the radius vector from the center of the Earth to the satellite in body frame, and $\vec{I}$ is the inertia tensor of the satellite.

$$\vec{\tau}_{\text{GG}} = \frac{3\mu}{\|r\|^3}(\vec{r} \times \vec{I}\vec{r}) \tag{2.1}$$

The aerodynamic disturbance is calculated as the drag force applied in each face, considering the point of application is the center of the face. In the case of the antenna being deployed, it is needed first to calculate the shadow of the satellite in the antenna, and then the drag torque that is applied to the antenna. This is added to the whole aerodynamic disturbance.

The equation for the torque produced by the solar radiation pressure is given in (2.2), where $\vec{R}$ is the distance between the point of application in the face, the center, and $\vec{F}$ is the force applied in that face.

$$\vec{\tau}_{\text{SRP}} = \sum \vec{R} \times \vec{F} \tag{2.2}$$

The force applied in a face is calculated as described in the equation (2.3), where $\vec{P}$ is the mean momentum flux of radiation, $\vec{S}$ is the surface of a face, and α and γ are variables that represent the absorbed and reflected radiation.

$$\vec{F} = -\vec{P} \times \vec{S}(\alpha + \gamma) \tag{2.3}$$

The residual magnetic disturbance is modeled as a random distribution between 0 and the maximum spacecraft residual magnetic moment. This produces a non deterministic simulation, so this disturbance is fixed when trying to achieve deterministic results in the simulation.

2.5 Dynamic solver

The dynamic solver operates the dynamic Euler and quaternion-based kinematic equations through the Dorman-Prince method algorithm [16], a method member of the Runge–Kutta family of ODE (Ordinary Differential Equation) solvers. This is a method of order n to solve ordinary differential equations, and in this case it is used of order 5.

Chapter 3

Control algorithms

The control algorithm converts the desired output of the system, in its own dimensions, to the electrical signal that the actuators need to achieve it, normally in voltage units. In the case of a satellite, the desired output is either, an angular rate during detumbling, or an angle while pointing.

Control algorithms can be divided in several categories. Two distinctions will be named, classical controllers and modern ones, or, robust and optimal controllers. In the industry, normally, the classical controllers are favoured, being cheaper to produce, and also it is more common to use robust controllers.

The classical controllers are algorithms based on two assumptions, linearity and a single input single output system. The most used example of a classical controller is the PID. This controller is relatively easy to set up, which makes it cheaper, and is reliable.

Modern controllers deal with non-linear systems, or multiple inputs and outputs, or both. Fuzzy logic controllers, in order to solve non-linearity, set different rules of action for different inputs, in a way that transition between these rules is not abrupt, but gradually [17]. This controller has several advantages against a classical PID, but is much more difficult to set up, as it is needs for a different PID for each rule.

Robust controllers are characterized by a wide range of action. Optimal controllers trade off the range for a better performance in a specific scenario. Classical controllers lose robustness for non-linear systems. Despite that, the PID controller is considered a very robust controller. Fuzzy logic controllers solve the problems of robustness of the classic PID, while also allowing for improvements in the optimization.

Aside from the control algorithm, the controller has to also manage the different modes. The control modes have different setpoints and different control laws. The detumbling mode controls the angular rate, while the pointing modes control the angle itself. Managing the current control mode is automatic, but can be override manually also.

The automatic change between Detumbling and any pointing mode is set by the angular rate of the satellite. A hysteresis loop is defined where, entering a pointing mode means the angular rate of the satellite is 20% under the limit set by the requirements. Going back to the detumbling mode is only done when the angular rate is 20% above said limit.

The changes between different pointing modes are decided based on the state of the mission. The standard mode is the nadir pointing. If the battery is low the Sun pointing mode is initiated. Finally, if the thruster is on, the TVP mode is activated. Note that the Sun pointing has priority over the others. The only change among these modes is the angle for the setpoint.

3.1 Detumbling

The detumbling is the process of reducing the angular rate of the satellite. After the satellite has been set on its orbit, the rotation of the satellite is often too fast to control its attitude, so the first step is to damp all angular velocities of the spacecraft.

In the case of the ^{3}Cat-8 mission, the requirements for the detumbling set the maximum angular rate the satellite could had after launch as 30º/s, and the limit to pass in order to consider the detumbling done, of 2º/s. The only constrain is the time in which this is accomplished, as the mission will start always after these operations have been finished.

The control algorithm used was a proportional controller, a reduced version of the PID controller only using the P. This type of controller multiplies the error by a number, the proportional constant, and that is the input of the actuators. In this case the precision is not the main aim, but the speed. The setpoint is an angular velocity, $\vec{\omega}$, of [0,0,0]º/s, so the angular velocity of the satellite respect to the orbit is considered the error.

In order to test the detumbling control, the simulations were made with a sampling time of 0.1s, and a duration of 5 orbits around the Earth. For the orbit selected, around 500 km of height, the duration is approximately 8 hours.

First, it is analyzed the relation between the value of the proportional controller, and the duration of the detumbling. For a value of the proportional constant of 10^{-5}, results in Figure 3.1, the detumbling is not finished in the period of time assigned the simulation.

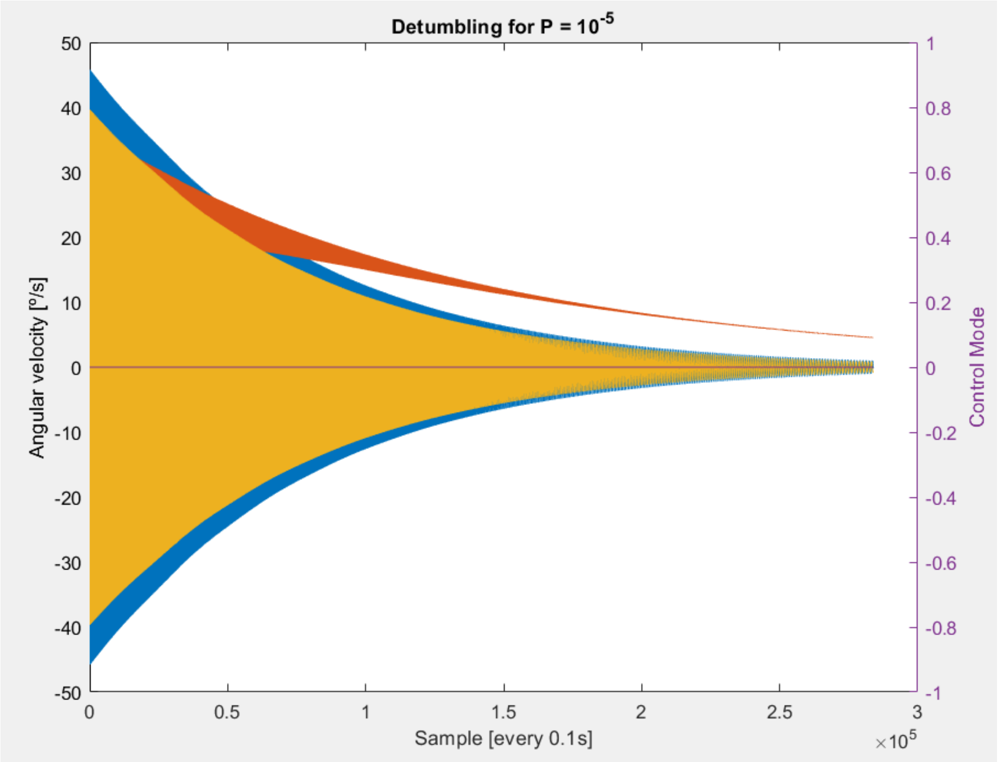


Figure 3.1: Detumbling for a P = 10^{-5} where each color represents a dimension of the $\vec{\omega}$.

For values of the proportional at 10^{-3} or higher, the duration of the detumbling reaches its minimum, around 2.9 hours. The result of this simulation is shown in Figure 3.2. Here, the purple line marks when the mode has changed from detumbling, 0, to nadir pointing, 1.

The reduction of time stops, cause the magnetorquers are saturating during all the detumbling for 10^{-3}, so higher numbers get the same result. The use of the magnetorquers is shown in Figure 3.3. Here it can be seen that during the duration of the detumbling the magnetic dipole generated by the magnetorquers is saturating at 0.8 and -0.8 Am^2, the limits set by the simulation.

In order to further reduce the time of detumbling, the saturation of the magnetorquers has to be increased. The value of the saturation will be finally defined with

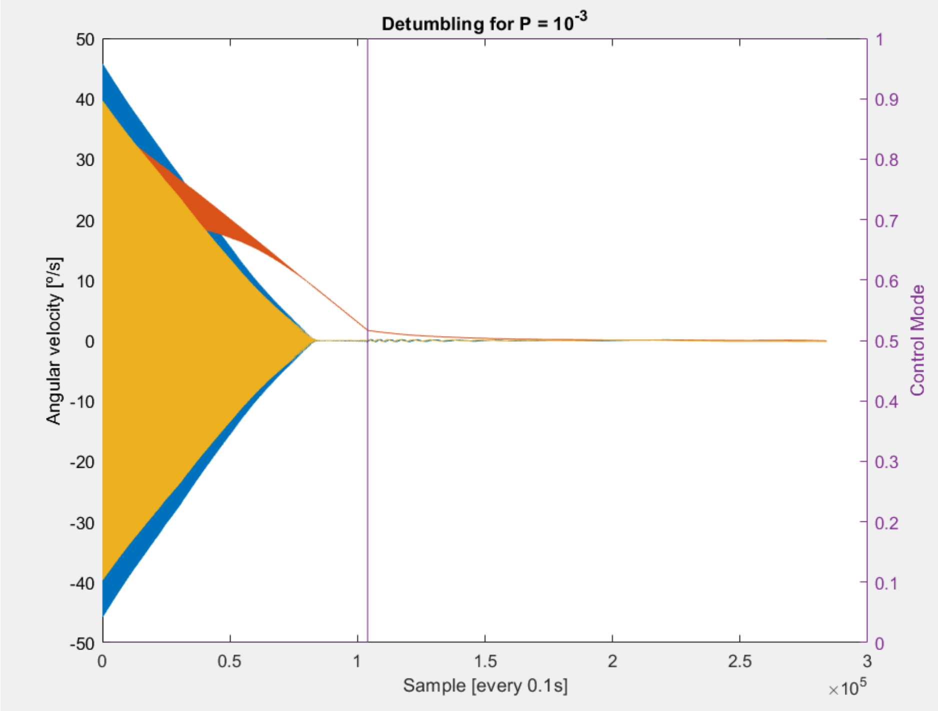


Figure 3.2: Detumbling for a P = 10^{-3}.

the dimension of the magnetorquers in the next chapter, but the range of acceptable values can be extracted from the simulations.

Being the limit of the detumbling duration 3 hours, a saturation of 0.8 Am^2 is the lowest limit acceptable for the magnetorquers to operate. The final value used for the mission should be higher, in case there was any problem. Increasing this value is not free though, as it implies an increase in volume, mass, and power consumption for the actuator.

To study the effect of the saturation value in the detumbling duration, the same method is applied as for the proportional constant, a range of values in the selected interest zone are tried. In this case saturation from 0.8 to 1.2, in steps of 0.1 Am^2.

The results range from 2.5 to 1.9 hours. The more the saturation is increased, the less the time is reduced. To have some margin, the saturation selected as ideal would be 1 Am^2, which is associated with a detumbling duration of 2.3 hours.

This saturation is twice as high the value of the ISIS magnetorquers module, a typical module used for 1U CubeSats, but it could be downsized if required by the mechanical and power specifications.

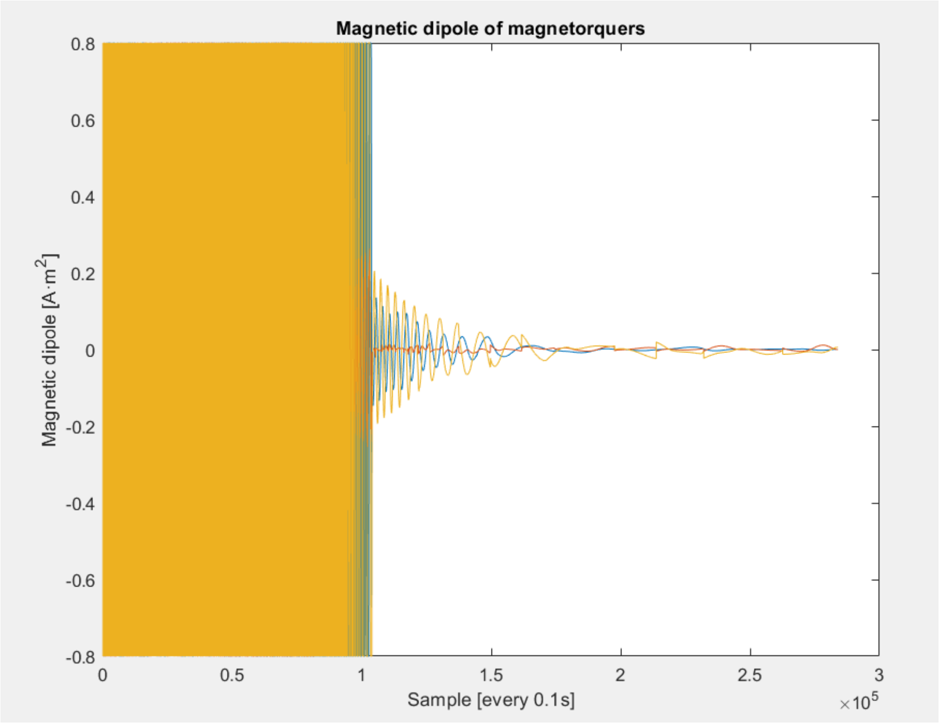


Figure 3.3: Magnetic dipole generated by the magnetorquers.

This value of saturation will be kept for the simulations of the pointing control. It will be studied then if the values of saturation are enough for this control mode. For a pointing control, accuracy is a priority, so saturation should not be reached at any point. In this sense, the pointing could be more restrictive than the detumbling for the saturation.

3.2 Pointing control

Once the satellite rotation imparted on the satellite during the orbital insertion process has been reduced to an acceptable rate, the attitude of the satellite has to be controlled for the duration of the mission. The satellite will be pointing to different directions, depending on the state of the mission. Those required attitudes will be defined as different set-points for the control.

In the case of the ^{3}Cat-8 mission, pointing control is always required to be accurate along the three axes. The asymmetries of the three axes made it necessary to have the three degrees of freedom. The pointing of the camera in the direction of the velocity, the Z axes, requires precision on the angles of the X and Y axes.

For the pointing of the side with solar panels, and the X-band antenna for ground communication, it is required the angle in Z.

To represent the rotation required for a certain attitude, there are three options. The first one is employing Euler angles. This is the rotation expressed as the angle rotated in each of the three axes. The rotation matrix, expresses the rotation in the euclidean plane as a transformation matrix. Finally, unit quaternions are an extension of the complex numbers using 4 elements, that can be used to express a rotation or an attitude.

Quaternions will be used to define the rotations of the control algorithm. Among the advantages of this system as compared to the Euler angles, they are not vulnerable to gimbal lock, which results on the loss of 1 degree of freedom, and also the discontinuity in the Euler angle from 180° to -180° is not present in the quaternions. The main advantage compared to rotation matrices, is that it uses a more compact format, saving computation time, and making it easier to work with [18].

3.2.1 Algorithm Implementation

The controller algorithm chosen for the pointing is a PID. The equation of the PID is shown in (3.1), where, $\ddot{q}$ is the angular acceleration desired in quaternion form, q_{err} is the error in the quaternion calculated as the difference between the set-point and the current attitude quaternion, $\dot{q}$ is the derivative of the quaternion, q_{int} is the integral of the quaternion error, and the K_j represents the three gains of the PID controller.

$$\ddot{q} = q_{\mathsf{err}}K_P + \dot{q}K_{\mathsf{D}} + q_{\mathsf{int}}K_{\mathsf{I}} \tag{3.1}$$

The derivative of the quaternion is obtained in equation (3.2). Here, the angular rate is expressed in quaternion form, and multiplied by the current attitude quaternion, applying the multiplication rules of quaternions [19].

$$\begin{aligned} \dot{q} &= \frac{1}{2}\omega q \\ \omega &= \begin{bmatrix} 0 \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \end{aligned} \tag{3.2}$$

In order to simplify this operation, and save computation time, it is used the term Ω as a 4×4 matrix that can be multiplied by the quaternion 4×1, as it is shown in equation (3.4).

$$\dot{q} = \frac{1}{2}\Omega q \tag{3.3}$$

$$\Omega = \begin{pmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & -\omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{pmatrix} \tag{3.4}$$

The integral of the error is calculated as a discrete integral of order zero, where the integrated value in the new step is the value of the previous step, plus the error in this instant multiplied by the sample time. The equation is in (3.5).

$$q_{int_k} = q_{int_{k-1}} + q_k \, \Delta T \tag{3.5}$$

The second derivative of a quaternion is expressed in equation (3.6). Operating using (3.2), angular acceleration can be obtained as a function of the quaternion and its derivative. It is important to note that, as the quaternions used to express rotations are unitary, the inverse of the quaternion is the conjugate.

$$\begin{aligned} \ddot{q} &= \frac{1}{2}(\alpha q + \omega \dot{q}) \\ \alpha &= 2(\ddot{q}q^* - (\dot{q}q^*)^2) \\ \alpha &= \begin{bmatrix} 0 \\ \alpha_x \\ \alpha_y \\ \alpha_z \end{bmatrix} \end{aligned} \tag{3.6}$$

With the angular acceleration in Euler form, the desired torque can be calculated using the Inertia matrix of the satellite, as in equation (3.7). Separating the calculation of $\ddot{q}$, using the control constants, and the calculation of the final torque using the inertia tensor, isolates the control constants from any changes in this value. So the nadir pointing constants are valid for any inertia tensor, unless the actuators saturate.

$$\tau = I \, \alpha \tag{3.7}$$

The desired actuators output is calculated, for the magnetorquers as the required magnetic dipole, and for the reaction wheel as the required variation of angular rate. For the magnetic dipole, the needed magnetic dipole is calculated using the magnetic field read by the magnetorquers. If these actuators saturate, then the reaction wheel is used to achieve the desired torque.

Magnetorquers interacts with the real magnetic field, which might be different to the one read by the sensors. However, as the micro-controller can only access the magnetic readings through the sensors, the desired toque and the one provided by the magnetorquers in the reality may differ.

3.2.2 Controller tuning

The constants K_P, K_D, and K_I, are obtained using a combination of an heuristic approach, and a Monte Carlo simulation. An analytic solution for a classical controller only considers one input and one output. In this case there are 3 inputs, considering only the magnetorquers, and 4 outputs as each dimension of the quaternion. These 4 outputs are dependant of the 3 actuators, so a change in one output would affect the other 3 outputs.

The tools of classic control theory, like the root locus or Bode diagram, for each of the outputs would disregard the effect of the controller for one output in the others. This process might return an accurate PID controller, considering it as the only active controller. The interaction with the other 3 controllers would change the system this controller was designed for. Despite that, the theoretic concepts behind these tools are used in the heuristic approach.

All of these constants are defined as diagonal matrix. So, when multiplying by the quaternions, each of its dimensions has only one associated constant. The constant matrix is defined in (3.8). The added effect of these three constants define the required action of the actuators as described in (3.1).

$$K = \begin{pmatrix} K_1 & 0 & 0 & 0 \\ 0 & K_2 & 0 & 0 \\ 0 & 0 & K_3 & 0 \\ 0 & 0 & 0 & K_4 \end{pmatrix} \tag{3.8}$$

The proportional constant, K_P, adds a quantity proportional to the error. Increasing the value of K_P increases the accuracy and speed of the controller. This results also in a higher cost of energy. Too high values can lead to the actuators saturation. Also, high K_P values might result on unstable systems.

To tune this value, the most important factor is the error. The value should be raised as much as possible to minimize the error. Once the error obtained by the output is small enough, the other variables have to be tuned to obtain other goals, like a small overshoot, higher stability, or reduced oscillations.

The value of the K_P moves the poles in the root locus. This means that the stability of the system is determined by this value. In systems with high number of poles and zeros, the root locus tend to be curved, with only a part of the curve in the stable region of the graph. So, if rising the value of K_P in one range of values is making the system more unstable, is not a valid assumption that the system will always behave like this.

The derivative constant, K_D, adds a quantity proportional to the derivative of the attitude quaternion. This defines any derivative different from 0 error. So the derivative is trying to slow down the output. Raising the derivative would reduce the overshoot and oscillations of the system. But it also makes the system more susceptible to disturbances.

Ultimately, a higher value of the derivative raises the gain of the controller in the high frequencies, looking at the Bode diagram. This might give good results in the lower frequencies, but also amplifies the vibrations in high frequencies, which are also very difficult to model and capture in a simulation. Another effect of the derivative is that it moves the curves of the root locus, so the value of K_D can be used to make a system stable for a desired K_P.

Once the system is stable and the errors are acceptable, the value of K_D can be slightly increased or decreased to reduce the overshoot or oscillations. It should be noted that the trade-off of increasing this value, increasing the disturbances, might be represented poorly on the simulations. This means the trade-off are not fully observable, and the value should be tuned with this in mind.

The integral constant, K_I, adds a quantity proportional to the integral of the error. The effect of this constant is slower than the other two, but eventually reduces the error to 0. Increasing this value will make the convergence process faster. A value too high may increase oscillations and overreaction to disturbances. This can pair with the increased disturbances of the K_D to destabilize the system.

The integral part also may suffer from non linear issues. The saturation or windup is the most important one. If the system is not able to reduce the error by any physical limitations, the integral will keep accumulating the error. Once the system can act again, or a disturbance increases the error, the system is going to be prone to overact or saturate the actuators if the integral value is too high. This can be solved capping the values of the integral, or setting it back to 0 when a strong change happens.

Tuning this value should be done last. As the system of a satellite is very fast and friction is not high enough to damp oscillations, the integral part is a value that should be kept as low as possible. Considering that the controller already has an acceptable error by the effect of the proportional, The integral should be high enough to close the remaining error, but not to disturb the controller before reaching this point.

The integral also has an effect on the root locus, moving its curves, similar to the way the derivative does. So stability must be kept once the integral part is added. In the Bode diagram the effect is seen as a lag controller, increasing the gain of the low frequencies. So this integral part has the most effect when the system is already slowed down. When the system becomes too fast, the integral part is reset.

The heuristic approach starts with all the variables at the same value, ones, and the integral part multiplied by 0. Then a mapping of the magnitude of the K_P and K_D is made. The slower outputs are selected, as they are closer to being stable. With this system the proportional constant can be increased to match with the error at the expense of stability. The derivative is then adjusted to make the system more stable again. Finally, the integral part is added at a very low value, so the controller is barely affected by it, and adjusted to match the convergence speed desired.

In general, for a system as fast as the one of a CubeSat, the values of K_P and K_D are going to be low. So the initial guess of orders should be between 1e-8 and 1e-5 for K_P, and between 1e-5 and 1e-1 for K_D. The value of K_I should be orders of magnitude smaller.

This is an iterative process, as the changes in one axes affect the other 3. Fully adjusting K_P and K_D of one axes at first might not be possible, for the effect of the other 3 axes destabilizing the system. Also, the axes that have been already adjusted might need be readjusted once the other axes are already adjusted.

Finally, once the system is stable and with acceptable errors, the final tuning of all the parameters is done through Monte Carlo simulation. This is a simulation that, generates a cloud of random points around the point of interest, the current controller adjusted gains.

The new random values are calculated as the value calculated before, times a random value from a normal distribution centered in 1 and with a designated σ calculated through the Matlab function 'normrnd', (3.9). As the σ defines the variance, the method can be done with a higher variance and more simulations for cases far from the optimal solution, or a lower variance and less simulations if the control answer is close to the desired.

$$K_{\text{new}} = K_{\text{old}} \cdot N(1, \sigma) \tag{3.9}$$

From the cloud of points, the controller selected is the one with the smaller absolute error integrated. The integration of the error is calculated independently for each of the dimensions of the quaternion. For a new controller to be selected it is required that the error is lower in all 4 dimensions.

This process can be iterated. Using the solution given by the first Monte Carlo, the simulation is repeated around this new point. The variance can be reduced for each iteration, as the solution is further refined.

3.2.3 Case studies

The pointing control has been tested for two cases. The first one uses the setpoint for the nadir pointing. The second one is a strong change in the attitude. The aim is to check the system either for keeping the current attitude, or the answer to changes in the setpoint.

In all the pointing simulations the same orbit and length are defined as in the detumbling simulation. The change in the initial parameters is the angular rate, set to 0, thus simulating if the satellite is able to perform the pointing control once the detumbling is finished. The setpoint and the control constants are defined differently for both cases.

For the first case, the setpoint is set to the quaternion [1, 0, 0, 0]. The initial position is assigned the same value. This quaternions is equivalent to the Euler angles [0, 0, 0], which expressed as a rotation of the body frame respect the orbit frame, it is the nadir pointing.

Without performing the Monte Carlo analysis, the constants selected for this case are those shown in equation (3.10). To simplify, only the elements of the diagonal shown in equation (3.8) are presented here, as the other values are always 0.

$$\begin{aligned} K_P = [9 \cdot 10^{-6}, 10^{-6}, 1.15 \cdot 10^{-5}, 3 \cdot 10^{-6}] \\ K_I = [10^{-7}, 10^{-11}, 3.9 \cdot 10^{-9}, 10^{-11}] \\ K_D = [10^{-3}, 4 \cdot 10^{-4}, 1.7 \cdot 10^{-3}, 10^{-4}] \end{aligned} \tag{3.10}$$

The attitude quaternion is shown in Figure 3.4. The oscillations shown by one of the axes around the value 0, are a result of the control being inactive in the first instant.

With no error the between the starting point and the setpoint, the error was exactly 0, and there was no need of any action. The disturbances create the oscillation that is later mitigated by the control.

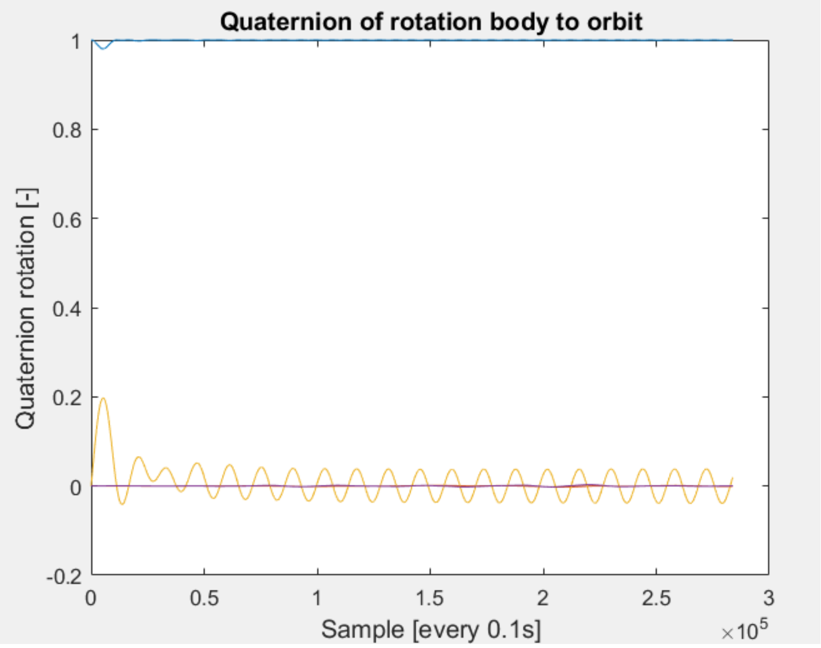


Figure 3.4: Four dimensions (first blue, second yellow, third red, fourth purple) of the quaternion rotation from body to orbit frame.

The equivalent of this rotation in terms of the Euler angles is shown in Figure 3.5. The oscillations are within a (5,-5) range. As all the angles are required to achieve 0º, their value is the error itself. The accepted error is of ±10º. The oscillations have an approximated period of 15 minutes. Being under the error margins and not very fast, these oscillations are acceptable as an initial result.

Magnetorquers' action is shown in Figure 3.6. As no large changes are needed in the direction, the magnetorquers are very far from saturation, 1 Am^2. In terms of saturation, detumbling is the most restrictive of the two modes analysed.

Given that these results are within the requirements for the constants describe in (3.10), the Monte Carlo analysis can be done around this point. After 500 simulations using a variance of σ=0.1, the new values found are given in equation (3.11).

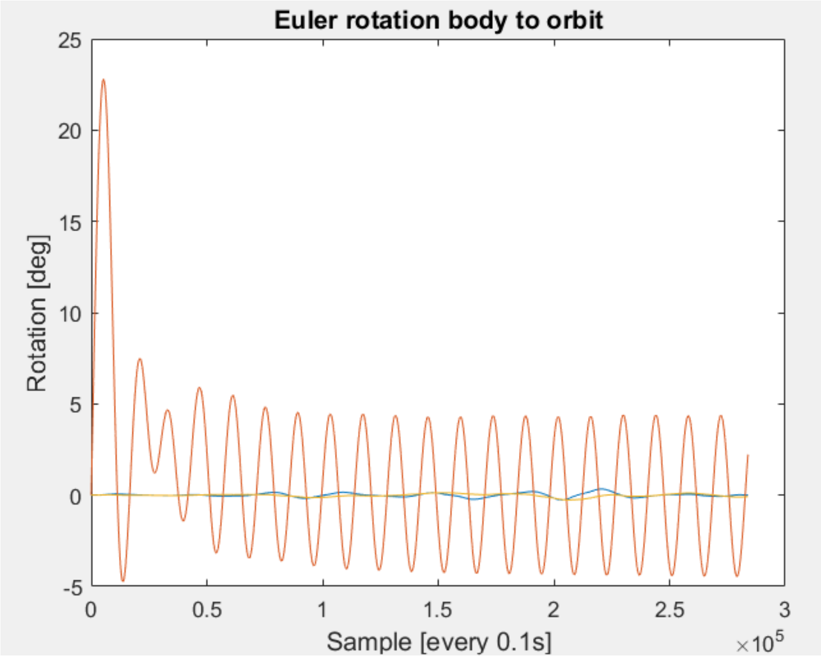


Figure 3.5: Euler angles of the rotation from body to orbit frame.

$$
\begin{aligned}
K_P &= [9.22705831465, 1.05235853264, 9.81912592020, 3.55992167898] \times 10^{-6} \\
K_I &= [7.85094796943, 0.00083647633, 0.43736745953, 0.00103251740] \times 10^{-8} \\
K_D &= [9.57108695577, 4.00463843550, 18.2432414047, 0.91319772372] \times 10^{-4}
\end{aligned}
\tag{3.11}
$$

The answer of the system after applying the new constants is shown in Figure 3.7. As the error was already low, the differences from Figure 3.4 are difficult to spot. Looking at the Euler angles in Figure 3.8, the oscillations have been reduced from ± 5º to ± 4.5º degrees.

As the results were already within the mission requirement, The refined version of the Monte Carlo can be taken without further iteration. The PID found will be the used to keep the nadir pointing once it has been reached.

In the second case, the setpoint is set to the quaternion [0, 1, 0, 0]. The initial position [1, 0, 0, 0]. In Euler rotation going from an angle of [0, 0, 0] to an angle of [0, 180, 180] degrees. This means a full rotation in two of the axes.

Applying the constants obtained earlier, the system is unstable. This means the control of the satellite is a non-linear system. A different set of constants has to be

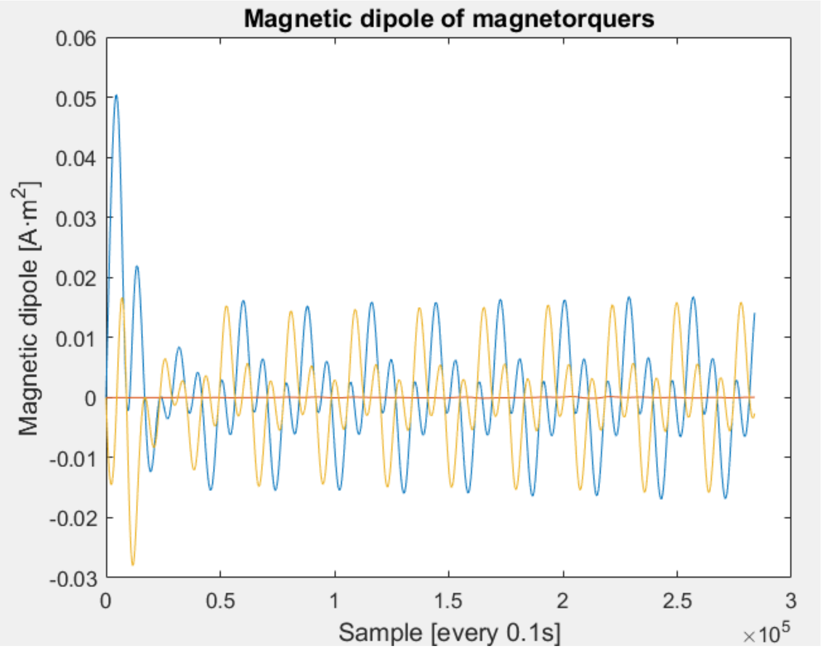


Figure 3.6: Magnetic dipole generated by the magnetorquers (NP).

defined for the extreme turns case. The PID constants achieved through the heuristic method are in equation (3.12).

$$\begin{aligned} K_P &= [9\times10^{-6}, 10^{-6}, 1.15\times10^{-5}, 3\times10^{-6}] \\ K_I &= [10^{-7}, 10^{-11}, 3.9\times10^{-9}, 10^{-11}] \\ K_D &= [10^{-3}, 4\times10^{-4}, 1.7\times10^{-3}, 10^{-4}] \end{aligned} \tag{3.12}$$

The attitude quaternion is shown in Figure 3.9. Here it can be seen that the first and the second dimensions exchange the place of 1 and 0. The curve of the first dimension, blue, has overshoot and an error that is not completely eliminated by the integrator. Although the second dimension sticks to the value of 1, the dimensions 3 and 4 oscillate.

The error shows more clearly the response in the 4 dimensions, from a control theory perspective. Figures 3.10 and 3.11 show respectively the error, and the absolute error. It can be seen that the integral is reducing the error in the first dimension, and the second dimension has a near 0 error. The last two dimensions experience oscillations over time.

Three loops of 500 simulations were used to try further refined the PID, using Monte Carlo for the respective values of σ, 0.1, 0.01 and 0.001. No better PID was found

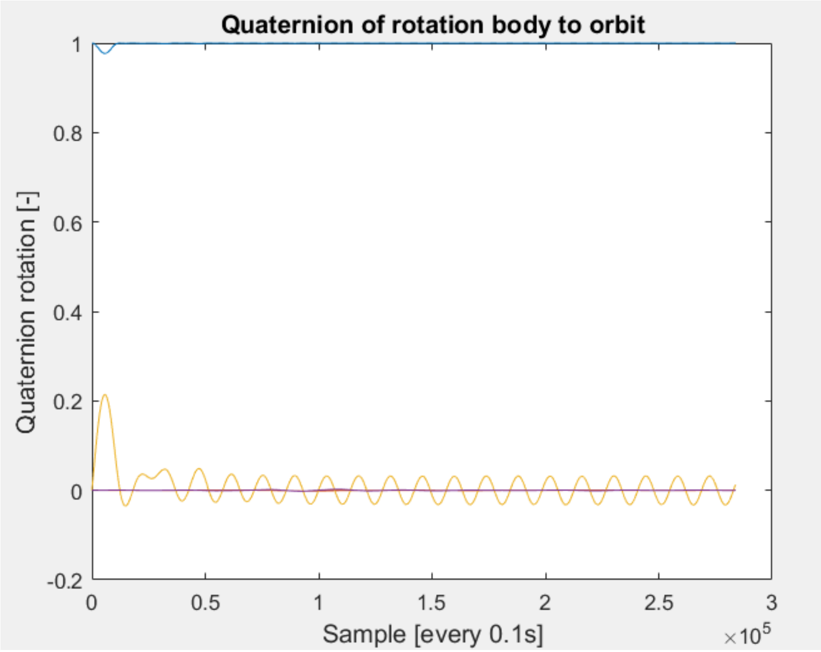


Figure 3.7: Four dimensions of the quaternion rotation after the Monte Carlo analysis.

through this analysis. This might be due to a high proximity between the point found through the heuristic method, and the local minimum of absolute error of the four dimensions.

It must be noted that there might be other minimums with lower errors. However, further optimization of this error is not needed, as it is already within the requirements. As such, the PID in equation (3.12) will be used for the changes in attitude.

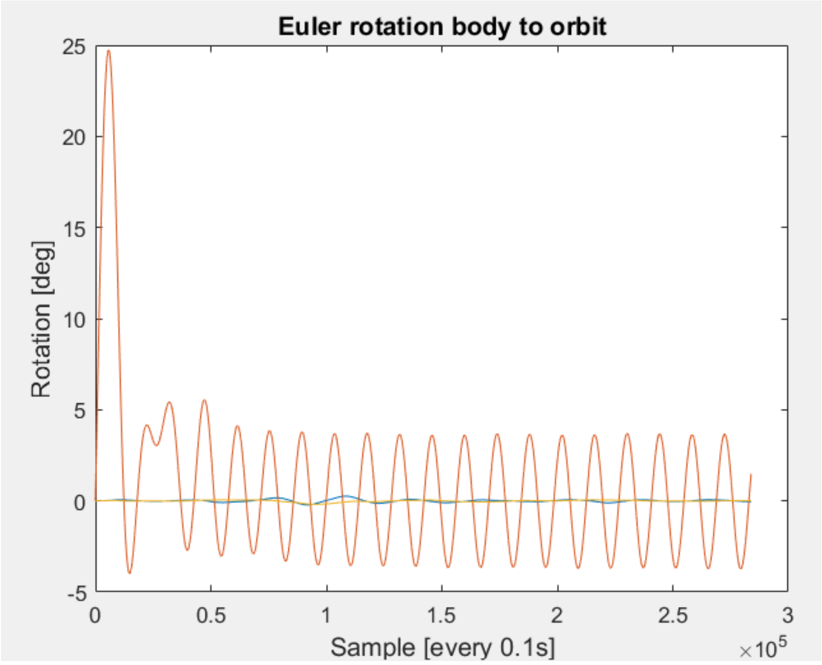


Figure 3.8: The Euler angles of rotation after the Monte Carlo analysis.

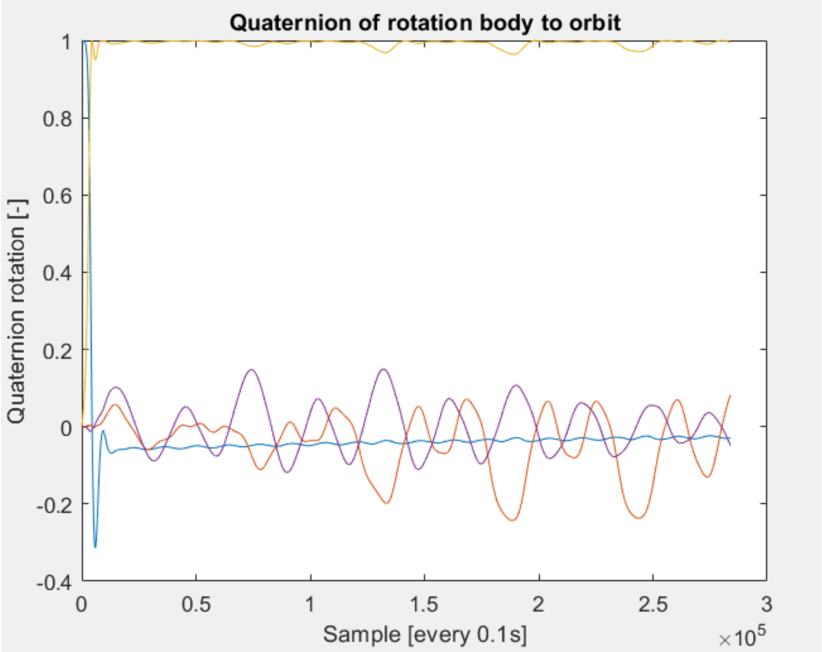


Figure 3.9: Four dimensions of the quaternion rotation from body to orbit frame.

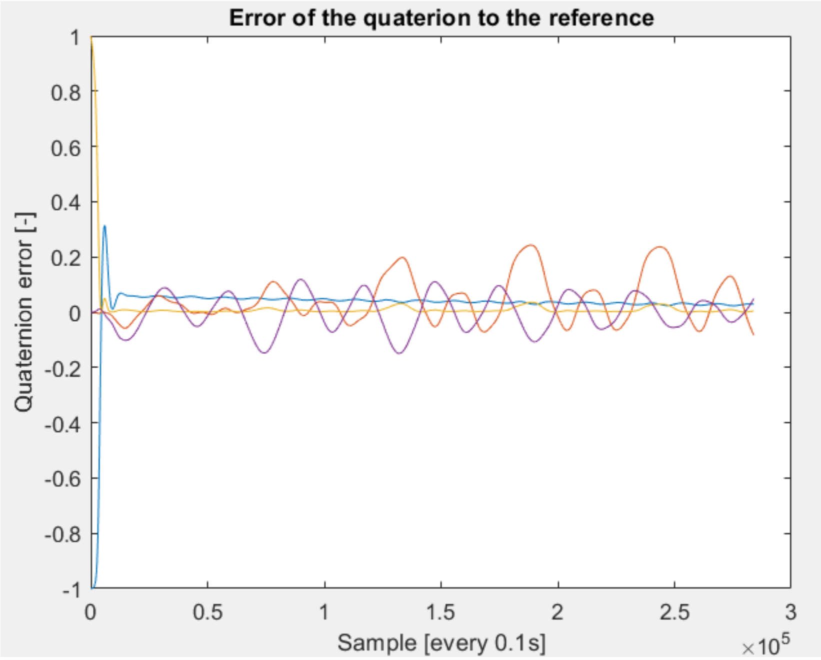


Figure 3.10: Error of the quaternion rotation.

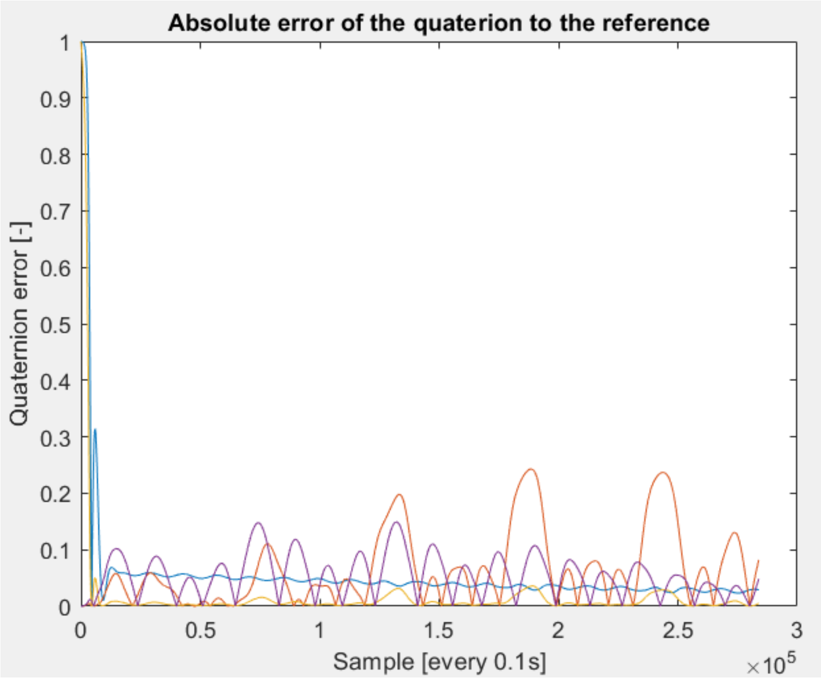


Figure 3.11: Absolute error of the quaternion rotation.

Chapter 4

Actuators

The actuators are the connection between the control algorithm, and the system that is desired to be controlled. In the case of a satellite, as the system is practically in the vacuum, the type of actuators that can be used are very limited. The most common options are magnetorquers, that interact with the magnetic field of the Earth, and reaction wheels that generate torque on their own.

Sizing these actuators implies a trade off between the characteristics requested to accomplish the ADCS requirements and the size or power they consume of the satellite. Based on the results of the detumbling control, and seeing that the nadir pointing is not saturating the magnetorquers, the dimensions of the magnetorquers can be calculated. The dimensions of the reaction wheel are already chosen, as it is already in the lab.

4.1 Magnetorquers

These actuators are coils made of a conductor material. When current circulates through the conductor, a magnetic field is generated in the direction of the coil. This dipole interacts with the magnetic field of the Earth generating a torque, as it is shown in equation (4.1), where d is the dipole generated, and B is the magnetic field of the Earth.

$$\tau = \vec{d} \times \vec{B} \tag{4.1}$$

There are many variables that define a magnetorquer, but the more important ones are the geometric dimensions of the design, as space is very limited in CubeSats, and the saturation, as it marks the limits of the actuation of the magnetorquer.

From the detumbling simulations, it was extracted that the ideal saturation of the magnetorquers is 1 Am^2. How this measurement relates to the dimensions of the magnetorquer is illustrated in equation (4.2), where, Sat is the value of the saturation previously defined, A is the total area of the magnetorquer defined as the number of loops, n, and the surface of each loop, S, and finally, I_{max} is the maximum current that the magnetorquer accepts.

$$\begin{aligned} Sat &= A \cdot I_{\text{max}} \\ A &= n \cdot S \end{aligned} \tag{4.2}$$

As the surface of each loop is defined by the spaces left for the magnetorquers in the satellite, the variables to calculate are the current and the number of loops. The current has been limited at 1 A to avoid high temperatures and power consumption. This left the number of loops to be calculated.

In the X and Y axes, the magnetorquers will be located on the sides of the PocketQubes deployer. The spaces where magnetorquers could fit are of 65 × 75 mm, minus 4 right angle triangles of 20 × 20 mm, for the anchorage to the deployer. This leaves a final area of 65 × 75 - 2 × 20^2 = 4075mm^2. This can be nearly multiplied by two, as there is an additional space, but the second one is a little bit smaller.

The surface for the magnetorquer in the Z axis is bigger, as it is located on the rear part of the deployer. The initial surface is of nearly a unit, at 90 × 100 mm, and in this case the 2 upper right triangles are of 40 × 40 mm, while the lower ones are of 25 × 25 mm. Which results on total surface of 90 × 100 - (40^2 + 25^2) = 6775 mm^2. All the dimensions are shown in Figure 4.1.

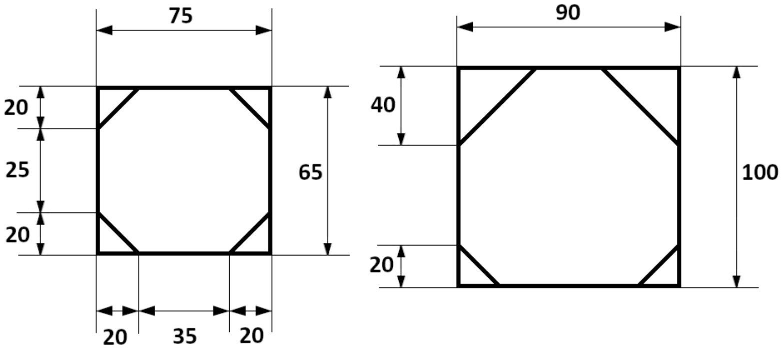


Figure 4.1: Dimensions in mm of the magnetorquers (not to scale).

For the desired saturation, the number of loops respectively are 246 and 148. Once the number of loops is set, the length of the wire is determined. The wire will be of copper, as it is the standard for magnetorquers. To calculate the final dimensions of the magnetorquer and the weight, the section of the wire must be determined.

The sections of copper wire is selected following the AWG, American Wire Gauge, standard. Within this catalogue, the sections studied are 0.321 mm, 0.286 mm, 0.255 mm and 0.227 mm. With the sections, the height and width of the magnetorquer can be calculated. For the magnetorquers in X and Y, the height is set to 5 mm to fit only the coil. For the magnetorquer in Z maximum width is set on 8 mm.

In the case of the X and Y axes, the width is calculated using equation (4.3), where n_{max} is the maximum number of loops that fit for the maximum height, d is the section of the cable, and t is the number of additional turns over the same axes.

$$\begin{aligned} n_{\text{max}} &= h_{\text{max}}/d \\ t &= n/n_{\text{max}} \\ w &= t \cdot d \end{aligned} \tag{4.3}$$

Once the maximum number of loops for that height is computed, the following loops are set on top of the first ones doing again the same turn. Doing this, by the geometry of the cylindrical wires, the next row of loops is placed tangent to two wires. In this case to simplify calculations it is considered that the wires are only tangent to one, so the width is exactly the number of turns times the section of the cable.

For the sections mentioned before and a range of saturation from 0.5 to 2 Am^2, the results of the width calculated are shown in Figure 4.2. The blue line represents the relationship between the width and the saturation of the magnetorquer. The red line is the limit for which the resistance of the wire would not allow the current of 1 A for 15 V.

The relationship of width and saturation is not a continuous function, as the width can only be a multiple of the wire diameter. Also, the width decrements with the lower values of section as expected. Taking the lower section that allows a saturation of 1 Am^2, 0.286 mm, the width is of 4.3 mm. Being lower than 5 mm, a limit for the physical design, this width can be accepted.

With the number of loops, the weight of the magnetorquer can also be obtained. This is calculated as the density of copper times the volume of wire, which is the length of the wire times the section. The results of the weight analysis are in Figure 4.3.

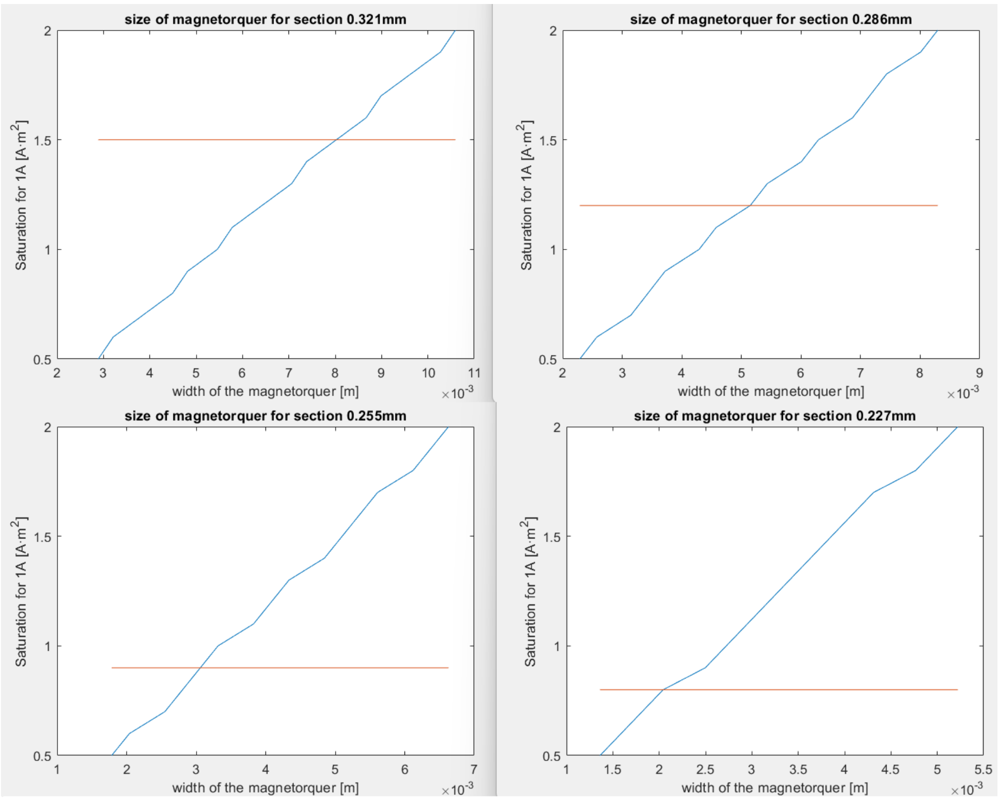


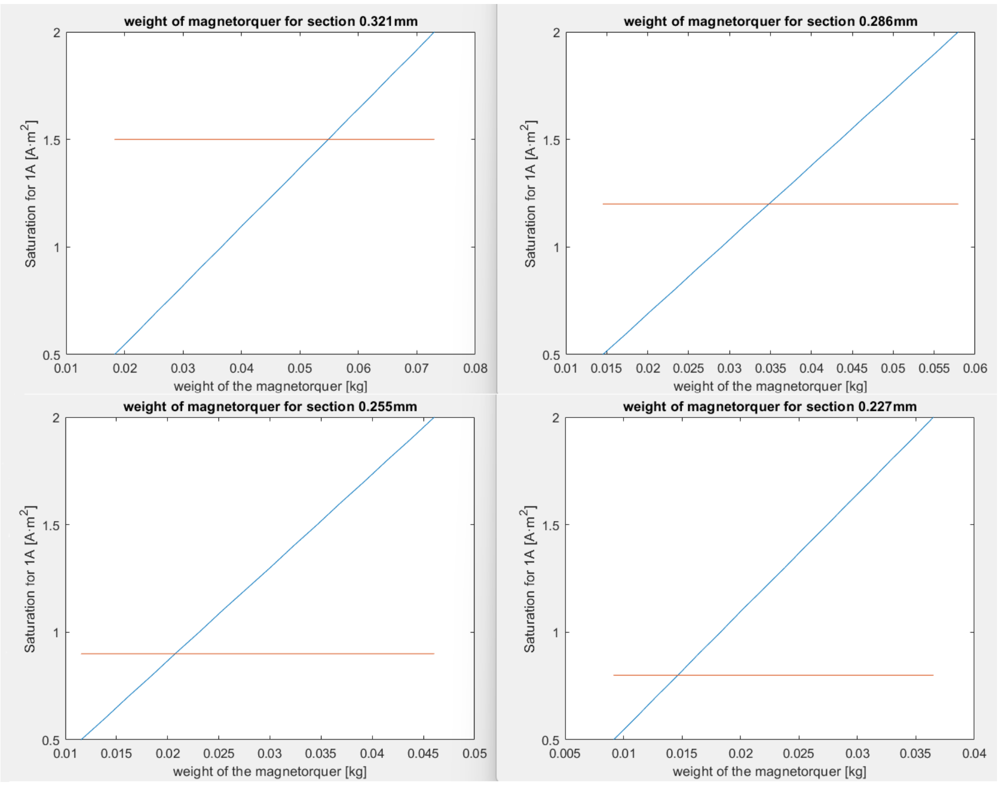
Figure 4.2: Width for the different sections of wire (X/Y).

The weight of the magnetorquer also decreases for lower sections, so the same criteria of choosing the lower section that allows for the desired saturation is applied here. Taking again the wire section of 0.286 mm for 1 Am^2, the weight is approximately 30 g per magnetorquer.

In the case of the Z axes, as the width is the starting value, it is calculated in reverse order. First the number of wire rows that fit in that width is calculated, and then, the height is the factor to be minimized. The analysis of the height is shown in Figure 4.4.

To reduce again the volume and weight of the magnetorquer, the wire of lower section that allows for the desired saturation at 15 V is the one selected. In this case this is the section of 0.227 mm. Due to the higher area of this magnetorquer, the section of the wire can be thinner. The height resultant of using this section is 1.1 mm.

The results of the weight of the magnetorquers are shown in Figure 4.5. For the wire of 0.277 mm of diameter, and 1 Am^2, the mass of the coil is around 11.5 g. The final weight of the three coils of the magnetorquers would be a total of 71.5 g.

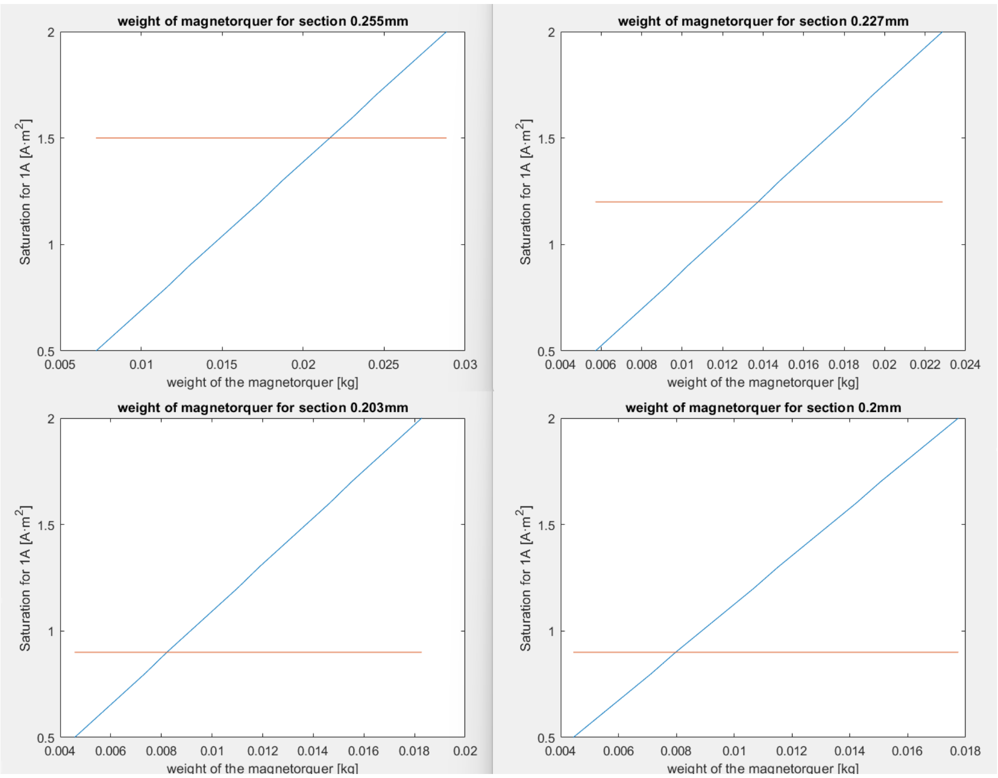

weight of magnetorquer for section 0.321mm

weight of magnetorquer for section 0.286mm

weight of magnetorquer for section 0.255mm

weight of magnetorquer for section 0.227mm

Figure 4.3: Weight for the different sections of wire (X/Y).


weight of magnetorquer for section 0.255mm

weight of magnetorquer for section 0.227mm

weight of magnetorquer for section 0.203mm

weight of magnetorquer for section 0.2mm

Figure 4.5: Weight for the different sections of wire (Z).

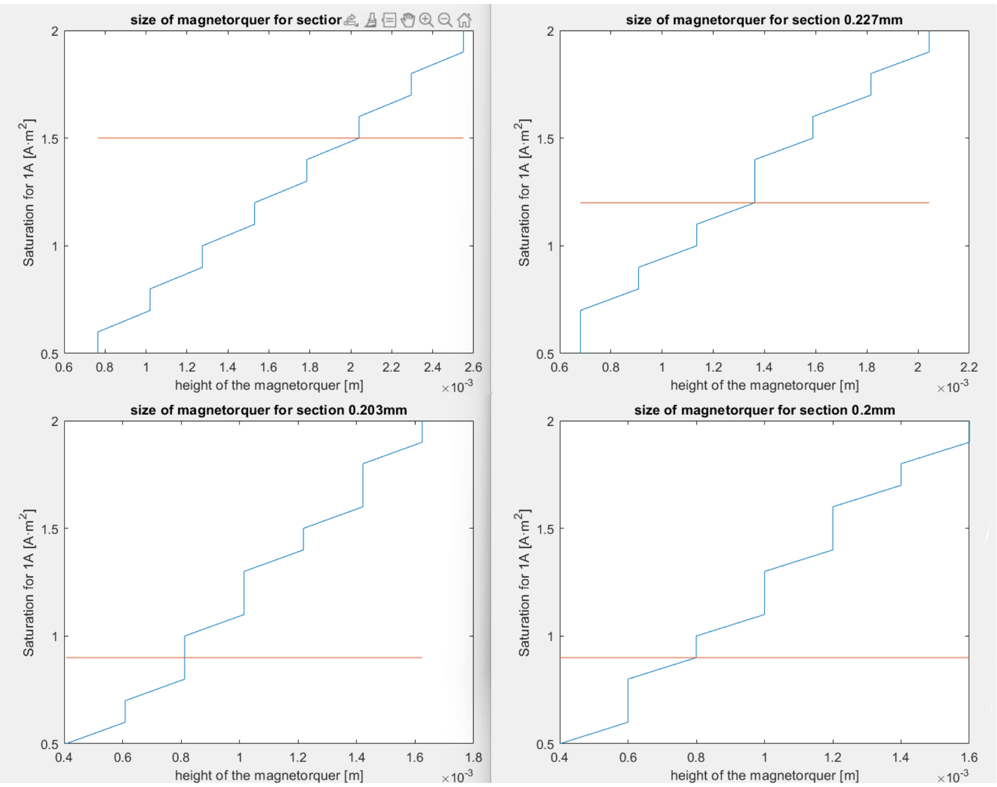


Figure 4.4: Height for the different sections of wire (Z).

Considering the mass of the PCBs needed for each magnetorquer, the physical support, and the power stage, the mass of the whole magnetorquers system will not be higher than 100 g. This is around 1% of the mass of the CubeSat.

A power stage was a necessary addition to the system. The saturation of 1 Am^2 require larger diameters of wire to still deliver 1 A of current. These larger diameters, not only were heavier, but also made up magnetorquers too large for the specific dimensions given. Also a power stage of 30V was studied to reduce the weight and size of the magnetorquers, but the power consumed was also higher, when the requirements were already met with 15 V.

4.2 Reaction wheel

The reaction wheel is a motor with a disc in the shaft. When the engine accelerates it builds up momentum in the disc following equation (4.4), where $\vec{\tau}$ is the momentum generated, I is the inertia tensor of the disc, and $\vec{\alpha}$ is the angular acceleration of the engine.

$$\tau = I\,\alpha \tag{4.4}$$

The reaction wheel is able to generate momentum without interacting with external sources, and the transmission of this moment to the system is as easy as a physical attachment, making it a very good candidate for a satellite actuator. High energy consumption, and the large volume and mass needed to produce enough momentum are the downsides for a CubeSat.

The decision to include the reaction wheel in this mission, despite the cost in volume and mass, was due to the possible loss of 1 degree of freedom when the magnetic field of the Earth aligns with a magnetorquer. When this happens, the rotation on the axes of that magnetorquer cannot be controlled.

Being on a polar orbit, when the satellite passes through the equator, the velocity axis, Z, and the magnetic vector of the Earth are close to aligned. To avoid the loss of control in this axis, as it is needed to keep the power generation, the reaction wheel was added there.

As the reaction wheel was already property of the lab, the dimensions are already set. For the physical dimensions, the reaction wheel fitted inside the deployer, where the spring is located when it is compressed. The mass is around 50 g.

For the magnitudes that affect the control system, the inertia moment is 1.27 $\times$ 10^{-6} kg m^{2}. The maximum angular rate is 8000 rpm, and the nominal one is 2000 rpm. Finally, the maximum momentum storage is 1.7 $\times$ 10^{-3} Nms.

Chapter 5

Conclusions

In this first iteration of the ^{3}Cat-8 ADCS system, it has been demonstrated that, with the current theoretical parameters of the mission, the objectives can be achieved. These objectives were the performance of the two first maneuvers of the mission, detumbling and nadir pointing.

Also, the foundations for the future development of the mission has been set up, the other pointing modes such as Sun pointing and TVP.

In terms of the environment simulation, the previous script has been adapted into a modular design to enhance the usability in future works. It has been generalized so it can be adapted to every future mission in the UPC NanoSat Lab. These changes have ended up in an optimization on the computational time needed, reducing to approximately a third the duration of each simulation.

For the controller algorithms, all the control modes required by the mission, can be done within the current algorithm structure. The set-point in the pointing control can be set to any rotation quaternion needed, allowing for any control mode desired.

The actuators are able to perform the maneuvers required within the missions restrains. In this thesis, the first sizing of the magnetorquers has been calculated to fit within the space constrains, considering the weight and size of the actuators. There is space to improve the capabilities of the magnetorquers if needed, but this comes with an increase in the power consumption. The reaction wheel has a secondary role and can not be modified.

The following extra observations about the system can be done:

- The classic control used is always going to have limitations, as the system is non-linear with multiple inputs and outputs. This may require different control constants for different set-points.

- Due to the control algorithm design, the inertia tensor has been separated from the control constants. So, a change in this value will not affect the constants calculated previously.

All the code developed for this project can be found in the UPC NanoSat Lab repository, for future usage.

5.1 Future works

In order to meet all the requirements of the mission, there are still tasks that should be performed. It is also possible to further optimize the simulation and control algorithms.

The tasks needed to complete the ^{3}Cat-8 mission after this work are:

- Implementation of the Sun pointing and TVP modes.
- Definition of the control constants for all the modes, both in the stowed configuration, and with the antenna deployed.
- A second iteration of the magnetorquers sizing, considering the loss in area for each row inwards.
- A study of the power consumption of the ADCS in all the stages of the mission.

Regarding the optimization, the different possible routes to follow are:

- Optimize the determination algorithm, in terms of computational cost.
- Refine the current algorithm or the constants already calculated for it.
- Transform the classic control algorithm into a modern one. Either, by using fuzzy logic to avoid the non-linearity problem, or an adaptive LQR control to solve both problems.

Bibliography

[1] R. Nugent, “6u cubesat design specification revision 1.0.” Document of public domain, California Polytechnic State University, San Luis Obispo, CA. 2018.

[2] K. Gaber, M. B. El_Mashade, and G. A. A. Aziz, “Hardware-in-the-loop real-time validation of micro-satellite attitude control,” *Computers & Electrical Engineering*, vol. 85, p. 106679, 2020.

[3] “Nanosats database.” `https://www.nanosats.eu/`. Last accessed July 2022.

[4] E. Kulu, “In-space economy in 2021—statistical overview and classification of commercial entities,” in *72nd International Astronautical Congress (IAC 2021), Dubai, United Arab Emirates*, pp. 25–29, 2021.

[5] E. Kulu, “Small launchers-2021 industry survey and market analysis,” 2021.

[6] J. Li, M. Post, T. Wright, and R. Lee, “Design of attitude control systems for cubesat-class nanosatellite,” *Journal of Control Science and Engineering*, vol. 2013, 2013.

[7] H. S. Ousaloo, “Globally asymptotic three-axis attitude control for a two-wheeled small satellite,” *Acta Astronautica*, vol. 157, pp. 17–28, 2019.

[8] A. Bello, K. Olfe, J. Rodríguez, J. Ezquerro, and V. Lapuerta, “Experimental verification and comparison of fuzzy and pid controllers for attitude control of nanosatellites,” *Advances in Space Research*, 2022.

[9] C. D. García, “Study and development of attitude determination and control simulation software and control algorithms for ^{3}cat-4 mission,” Master’s thesis, Universitat Politècnica de Catalunya, Barcelona, 2019.

[10] T. Farges, E. Blanc, and J.-L. Pinçon, “Monitoring lightning from space with taranis,” in *American Geophysical Union*, vol. 2010, pp. pp–1542, 2022.

[11] D. Vallado and P. Crawford, “Sgp4 orbit determination,” in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, p. 6770, 2008.

[12] J. Picone, A. Hedin, D. P. Drob, and A. Aikin, “Nrlmsise-00 empirical model of the atmosphere: Statistical comparisons and scientific issues,” *Journal of Geophysical Research: Space Physics*, vol. 107, no. A12, pp. SIA–15, 2002.

[13] I. Wardinski, D. Saturnino, H. Amit, A. Chambodut, B. Langlais, M. Mandea, and E. Thebault, “Geomagnetic core field models and secular variation forecasts for the 13th international geomagnetic reference field (igrf-13),” *Earth, Planets and Space*, vol. 72, no. 1, pp. 1–22, 2020.

[14] D. Choukroun, I. Y. Bar-Itzhack, and Y. Oshman, “Optimal-request algorithm for attitude determination,” *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 3, pp. 418–425, 2004.

[15] A. Navarro Trastoy, “Attitude determination using quaternion-based algorithms for ^{3}cat-4,” Master’s thesis, Universitat Politècnica de Catalunya, Barcelona, 2019.

[16] C. Engstler and C. Lubich, “Mur8: a multirate extension of the eighth-order dormand-prince method,” *Applied numerical mathematics*, vol. 25, no. 2-3, pp. 185–192, 1997.

[17] J. Xu and L. Xie, *Control and estimation of piecewise affine systems*. Sawston, UK: Elsevier, 2014. pp 169-192.

[18] G. Ledo López, “Development and implementation of a mass balancing system for cubesat attitude hardware-in-the-loop simulations,” Master’s thesis, Luleå University of Technology, Lulea, 2019.

[19] Y.-B. Jia, “Quaternions,” *Com S*, vol. 477, p. 577, 2019.