A conceptual design and evaluation framework for ADCS for CubeSats

Beatriz Camelo Ribeiro Ferreira Alves
beatrizcameloalves@tecnico.ulisboa.pt

Instituto Superior Técnico, Universidade de Lisboa, Portugal

November 2021

Abstract

During the development of CubeSats, the attitude determination and control system (ADCS) is one of the most expensive subsystems and it is usually based on commercial of the shelf options. A design tool capable of simulating several common ADCS architectures for CubeSats allows designers to test and evaluate different options of ADCS architectures resulting in a more educated selection of components. Furthermore, the tool eliminates the need for commercially protected software specifically created for each individual ADCS architecture option. To this end, a simulation tool based on Matlab/Simulink has been developed. The tool models a realistic orbital space environment, and the CubeSat physical constraints to test and evaluate a wide range of both passive and active ADCS selected and includes a range of computational algorithms for attitude estimation and control. Three different passive attitude stabilization options are currently available: deployable gravity gradient booms and aerodynamic stabilizing panels, permanent magnets and hysteresis rods. Moreover, the user can experiment with both fine and coarse sun sensors, magnetometers, gyros, Earth sensors and star trackers, and actuators such as magnetorquers, reaction and momentum wheels and single gimbal control momentum gyros. Different popular cluster architectures of momentum devices are available to the user. The user can also choose between various conventional static determination, recursive estimation and control algorithms. The tool was used to evaluate the ADCS for the ORCASAT. Alternative ADCS architectures were tested and compared in terms of weight, volume, power, pointing and estimation error. **Keywords:** CubeSat, ADCS, attitude determination, attitude control

1. Introduction

During the design of the attitude determination and control system (ADCS) the ORCASAT team struggled with the validation of the chosen ADCS solution. With the increasing innovation in the ADCS leads to the necessity of a tool capable of realistically simulate the behavior of the satellite in orbit when equipped with different subsystem solutions.

The design of a CubeSat ADCS simulator, able to simulate different realistic ADCS architectures, both passive and active, would enable satellite teams to experiment with different possible solutions before committing to a specific ADCS architecture. More innovative can also be tested with "non-ideal" models without requiring teams big time and monetary investments.

2. Reference Frames

The attitude of a body can be described as the orientation of a reference frame with respect to another [1]. The main reference frames used in this work will be described in this section.

2.1. Earth Centered Inertial Frame

The ECI reference frame is constituted by the set of axes $\{\hat{\mathbf{i}}_1, \hat{\mathbf{i}}_2, \hat{\mathbf{i}}_3\}$, with $\hat{\mathbf{i}}_3$, pointing towards the Earth's geographic North pole, $\hat{\mathbf{i}}_1$, directed towards the vernal equinox. The last axis $\hat{\mathbf{i}}_2$ completes the right-hand triad. The ECI frame's origin coincides with the Earth's center of mass (CM).

2.2. Local-Vertical/Local-Horizon Frame

The Local-Vertical/Local-Horizon (LVLH) frame is a non-inertial reference frame, centered on the satellite's center of mass and defined by the orthogonal triad $\{\hat{\mathbf{o}}_1, \hat{\mathbf{o}}_2, \hat{\mathbf{o}}_3\}$. The axis $\hat{\mathbf{o}}_3$ is aligned with the nadir direction (directed towards the center of mass of the Earth), while the axis $\hat{\mathbf{o}}_2$ is oriented along the orbit normal in the direction opposite to orbital angular momentum vector. $\hat{\mathbf{o}}_1$ completes the right hand triad.

2.3. Body Frame

The body frame is defined by a set of cartesian axes and a origin set on a specified point of the spacecraft. The origin is typically chosen to coin-

cide with the center of mass of the spacecraft and its axis $\{\hat{\mathbf{b}}_\mathbf{1}, \hat{\mathbf{b}}_\mathbf{2}, \hat{\mathbf{b}}_\mathbf{3}\}$ are normally chosen so as to align with the principal axis of inertia of the body [1]. For the ORCASat, the body frame was chosen so that, in nominal mode, it's payload is pointing towards nadir, that is, that for null pointing error the body frame is aligned with the orbit frame.

3. Attitude Parameters

The attitude of a body can be represented by several different parameters. In this section, the main representation methods utilized in this work are presented.

3.1. Direction Cosine Matrix (DCM)

Given two different reference frames $\mathbf{F} = \{\hat{\mathbf{f}}_\mathbf{1}, \hat{\mathbf{f}}_\mathbf{2}, \hat{\mathbf{f}}_\mathbf{3}\}$ and $\mathbf{G} = \{\hat{\mathbf{g}}_\mathbf{1}, \hat{\mathbf{g}}_\mathbf{2}, \hat{\mathbf{g}}_\mathbf{3}\}$, there is always a rotation matrix $\mathbf{A_{GF}}$, such that:

$$\mathbf{k_g} = \mathbf{A_{GF}}\mathbf{k_f} \tag{1}$$

where $\mathbf{k_g}$ and $\mathbf{k_f}$ are representations of the same vector in reference frames $\mathbf{G}$ and $\mathbf{F}$, respectively. The rotation matrix $\mathbf{A_{GF}}$, known as the attitude matrix or DCM, transforms the coordinate reference frame $\mathbf{F}$ to $\mathbf{G}$ and is defined as:

$$\mathbf{A_{GF}} = \begin{bmatrix} \hat{\mathbf{g}}_\mathbf{1}.\hat{\mathbf{f}}_\mathbf{1} & \hat{\mathbf{g}}_\mathbf{1}.\hat{\mathbf{f}}_\mathbf{2} & \hat{\mathbf{g}}_\mathbf{1}.\hat{\mathbf{f}}_\mathbf{3} \\ \hat{\mathbf{g}}_\mathbf{2}.\hat{\mathbf{f}}_\mathbf{1} & \hat{\mathbf{g}}_\mathbf{2}.\hat{\mathbf{f}}_\mathbf{2} & \hat{\mathbf{g}}_\mathbf{2}.\hat{\mathbf{f}}_\mathbf{3} \\ \hat{\mathbf{g}}_\mathbf{3}.\hat{\mathbf{f}}_\mathbf{1} & \hat{\mathbf{g}}_\mathbf{3}.\hat{\mathbf{f}}_\mathbf{2} & \hat{\mathbf{g}}_\mathbf{3}.\hat{\mathbf{f}}_\mathbf{3} \end{bmatrix} \tag{2}$$

3.2. Euler Angles

An angular displacement can always be described as a sequence of rotations. Considered the 3-2-1 sequence, with roll (ϕ) associated with the axis e_1, pitch (θ) with e_2 and yaw (ψ) with e_3:

$$\mathbf{A_{321}} = \begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ -c\phi s\psi + s\phi s\theta c\psi & c\phi c\psi + s\phi s\theta s\psi & s\phi c\theta \\ s\phi s\psi + c\phi s\theta c\psi & -s\phi c\psi + c\phi s\theta s\psi & c\phi c\theta \end{bmatrix} \tag{3}$$

where c and s represent the trigonometric functions cosine and sine, respectivelly.

3.3. Quaternions

Quaternions can be represented as a 4×1 matrix constituted by a vectorial part, $\mathbf{q_{1:3}}$, and a scalar part, q_4 [2].

$$\mathbf{q} = \begin{bmatrix} \mathbf{q_{1:3}} \\ q_4 \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{4}$$

For attitude purposes, the unit quaternion is used:

$$\|\mathbf{q}\| = \mathbf{q}^T\mathbf{q} = \sqrt{q_1^2 + q_2^2 + q_3^2 + q_4^2} = 1 \tag{5}$$

3.3.1 Quaternion Algebra

The complex conjugate of a quaternion is defined as:

$$\mathbf{q}^* = \mathbf{q} = \begin{bmatrix} -\mathbf{q_{1:3}} \\ q_4 \end{bmatrix} \tag{6}$$

And the inverse can be defined as:

$$\mathbf{q^{-1}} = \mathbf{q} = \frac{\mathbf{q}^*}{q} \tag{7}$$

Consider two quaternions $\mathbf{q}$ and $\mathbf{p}$, one way to define the product between the two is:

$$[\mathbf{q}\odot] = \begin{bmatrix} \Xi(\mathbf{q}) & \mathbf{q} \end{bmatrix} = \begin{bmatrix} q_4 I_3 + [\mathbf{q_{1:3}}\times] & \mathbf{q_{1:3}} \\ -\mathbf{q_{1:3}}^T & q_4 \end{bmatrix} \tag{8}$$

where:

$$\Xi(\mathbf{q}) = \begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix} \tag{9}$$

4. Attitude Kinematics

Given two different reference frames $\mathbf{F}$ and $\mathbf{G}$, the angular rate of $\mathbf{G}$ with respect to $\mathbf{F}$ is represented by $\boldsymbol{\omega^{GF}}$, then

$$\dot{\mathbf{A}}_\mathbf{GF} = -\boldsymbol{\omega}_\mathbf{G}^\mathbf{GF}(t) \times \mathbf{A_{GF}} \tag{10}$$

with

$$[\boldsymbol{\omega}_\mathbf{G}^\mathbf{GF}\times] = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \tag{11}$$

is the kinematic differential equation that maps the relative orientation of frame $\mathbf{G}$ with respect to frame $\mathbf{F}$ over time, with $\boldsymbol{\omega}_\mathbf{G}^\mathbf{GF} = [\omega_1, \omega_2, \omega_3]^T$ being the representation of $\boldsymbol{\omega}^{GF}$ in coordinates of the $\mathbf{G}$ frame.

In quaternion representation the equivalent differential equation is given by [1]:

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q}\odot\boldsymbol{\omega} = \frac{1}{2}\Xi(\mathbf{q})\boldsymbol{\omega} = \frac{1}{2}\boldsymbol{\omega}\otimes\mathbf{q} = \frac{1}{2}\Omega(\boldsymbol{\omega})\mathbf{q} \tag{12}$$

where

$$\Omega(\boldsymbol{\omega}) = \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} \tag{13}$$

and $\Xi(\mathbf{q})$ is defined in section 3.3.

5. Attitude Dynamics

Taking the center of mass of the body as the reference point and the ECI frame as the inertial reference frame:

$$\dot{\mathbf{h}}_\mathbf{I}^c = \boldsymbol{\tau}_\mathbf{extI} \tag{14}$$

Under the rigid body assumption,

$$\mathbf{h}_I^c = \mathbf{J}_I^c\boldsymbol{\omega}_\mathbf{I}^\mathbf{BI} \tag{15}$$

where J represents the inertia tensor of the spacecraft.

J is a symmetric matrix, dependent on the mass and shape of the satellite. J is defined as:

$$\mathbf{J_b} = \begin{bmatrix} \int_B (y_p^2 + z_p^2)\, dm & \int_B -(x_p y_p)\, dm & \int_B -(x_p z_p)\, dm \\ \int_B -(x_p y_p)\, dm & \int_B (x_p^2 + z_p^2)\, dm & \int_B -(z_p y_p)\, dm \\ \int_B -(x_p z_p)\, dm & \int_B -(z_p y_p)\, dm & \int_B (y_p^2 + x_p^2)\, dm \end{bmatrix} \quad (16)$$

where P represents a generic body point with infinitesimal mass dm and coordinates $\{x_p, y_p, z_p\}$ in the reference frame. Since the coordinates of P will differ for different frames of reference, J will also differ.

Combining equations (14) and (11), the fundamental equation of angular motion expressed in body frame coordinates becomes

$$\dot{\mathbf{h}} = \boldsymbol{\tau_{ext}} - \boldsymbol{\omega} \times \mathbf{h} \quad (17)$$

where $\boldsymbol{\omega}$ represents the angular velocity of the body frame with respect to the inertial frame represented in the body frame, that is $\boldsymbol{\omega}_B^{BI}$. The subscript B indicating the body reference frame has been omitted for simplicity of notation. Alternatively, from (15)

$$\mathbf{J}\dot{\boldsymbol{\omega}} = \tau_{\mathbf{ext}} - \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} \quad (18)$$

where J is defined in the body frame. Even though J is usually time varying, it's representation in the body frame is constant.

5.1. Attitude Dynamics with Momentum Devices

When momentum exchange devices are present, such as reaction/momentum wheels [3] or single gimbal control moment gyros [4, 5], the total angular momentum of the system "spacecraft+momentum devices" about its center of mass is given by:

$$\mathbf{h} = \mathbf{J}\boldsymbol{\omega} + \mathbf{h}^{med} \quad (19)$$

where $\mathbf{h}^{med}$ represents the angular momentum of momentum exchange device and J represents the moment of inertia of the system "spacecraft+momentum exchange device". Substituting in (18), the equation of angular moment in the body frame, becomes:

$$\mathbf{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times (\mathbf{J}\boldsymbol{\omega} + \mathbf{h^{med}}) - \boldsymbol{\tau_{cp}} - \dot{\mathbf{h}}^{\mathbf{med}} \quad (20)$$

With $\boldsymbol{\tau_{cp}}$ comprising both perturbation and other control torques.

6. Simulation Environment

The Simulink simulation tool is divided in two main areas: an upper area that comprises all the environment, spacecraft, sensor and actuator models, as well as a model of the on board computer and estimation and control algorithms and a lower area devoted to graphical analysis and data output. The upper area is divided into a simulator area, which simulates the interaction of the spacecraft with the space environment and the ADCS area, which simulates the attitude subsystem on board the CubeSat. Each of these areas is divided into blocks which are explained in sections 7 to 13.

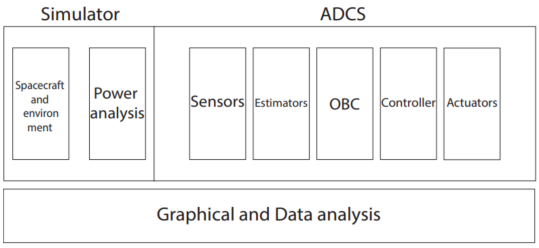

Figure 1: Simulink ADCS Simulator tool

Each section does an overview of functions performed by each block, explaining in depth relevant models. The interaction between each block is shown in figure 2.

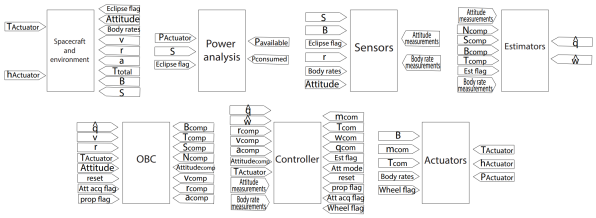

Figure 2: Interaction between blocks

7. Spacecraft Mechanics Simulator Block

This block simulates the space environment experienced in orbit, namely, the geomagnetic field, the gravitational field and sun position. This block also reproduces the translation and rotation motion of the satellite through the orbital and attitude mechanics equations as well as the perturbations. Finally, it contains models of the passive attitude stabilization structures and devices whose models are presented in section 7.1.

7.1. Passive Attitude Stabilization

7.1.1 Gravity Gradient Boom

Gravity gradient booms are small masses attached to a long thin rod, along the axis that is desired to be aligned with the nadir direction.

The small tip masses are modeled as solid spheres. The total moment of inertia in the body frame is given by:

$$\mathbf{J_B} = \mathbf{J_B^{hub}} + \sum_{i=1}^{N} \mathbf{J_B^{tipi}} + \sum_{i=1}^{N} \mathbf{J_B^{rodi}} \quad (21)$$

where $\mathbf{J_B^{hub}}$, $\mathbf{J_B^{tipi}}$ and $\mathbf{J_B^{rodi}}$ are the inertia tensors of the hub, mass tip i and rod i, respectively in the body frame of a spacecraft with n booms.

Both $\mathbf{J_B}$ and the center of mass will change during the deployment process. The gravity gradient boom deployment is modeled as uniform.

7.1.2 Permanent Magnet

Permanent magnets provide a control torque caused by the interaction of a magnetic dipole moment with a magnetic field of the Earth [1]:

$$\mathbf{T} = \mathbf{m} \times \mathbf{B_{geomag}} \tag{22}$$

where $\mathbf{m}$ is the permanent magnet's magnetic moment vector and $\mathbf{B_{geomag}}$ is the magnetic flux density vector.

Permanent magnets are usually accompanied by hysteresis rods and are typically placed orthogonal to them.

7.1.3 Hysteresis Rod

Hysteresis rods provide passive oscillation-damping and de-spin torques. Hysteresis rods are characterized by their hysteresis $B_{rod}(H)$ loop, that is, the component of the geomagnetic field vector, of magnetic field strength H, parallel to the axis of the rod will induce a magnetic flux density (B_{rod}) in the rod that produces a magnetic moment parallel to the axis [6]:

$$\mathbf{m} = \frac{\mathbf{B_r} V_{rod}}{\mu_0} \tag{23}$$

where μ_0 represents the magnetic permeability of vacuum and V_{rod} is the volume of the rod. Similar to the permanent magnet, the magnetic dipole moment will interact with the geomagnetic field and supply a torque given by (22).

Several mathematical models exist to try and replicate with more or less precision the hysteresis loops. The Simulink tool presents the magnetic hysteresis model developed in [7].

7.1.4 Aerodynamic Stabilization Panels

A "shuttlecock" configuration of panels was modeled. Each panel is modeled as a rectangular flat plate. The total contribution of the panels to the total drag force is given by:

$$\mathbf{f_B^d} = \sum_{i=1}^{n} \mathbf{f_B^{di}} = \sum_{i=1}^{n} -\frac{1}{2}\rho v_{rel}^2 C_D A_p \hat{\mathbf{v}}_{\mathbf{relB}} \tag{24}$$

where ρ is the atmospheric density, C_D is a dimensionless drag coefficient, $\mathbf{v_{relB}}$ is the velocity of the spacecraft relative to the air molecules in the atmosphere and A_p is the total projected area of the satellite in direction of motion:

$$A_p = \int_A -H(-\hat{\mathbf{v}}_{\mathbf{rel}} \cdot \hat{\mathbf{n}}^{\mathbf{i}}) \hat{\mathbf{v}}_{\mathbf{rel}} \cdot \hat{\mathbf{n}}^{\mathbf{i}} dA \tag{25}$$

where $H()$ is the Heaviside function and $\hat{\mathbf{n}}^{\mathbf{i}}$ is the unit inward normal of the face i defined as a function of the aperture angle.

The aerodynamic torque $\boldsymbol{\tau}_{\boldsymbol{B}}^{\boldsymbol{d}}$ acting on the spacecraft due to the panels is given by:

$$\boldsymbol{\tau}_{\boldsymbol{B}}^{\boldsymbol{d}} = \sum_{i=1}^{n} \boldsymbol{\tau}_{\boldsymbol{B}}^{\boldsymbol{di}} = \sum_{i=1}^{n} [\mathbf{c_B^{pi}} \times] \mathbf{f_B^{di}} \tag{26}$$

The deployment of aerodynamic panels will impact the tensor of inertia of the spacecraft. The deployment of the panels has been modeled as uniform.

8. Sensors Block

The sensors block models the behavior of typical Cubesat attitude sensors, namely, sun sensors, magnetometers and gyroscopes, which can be found in ORCASat's ADCS and whose models are explained in depth in [8], as well as earth sensors and star trackers.

8.1. Earth Sensor

The Earth sensors detect points in the Earth horizon. The direction of these points in the body frame is given by [9]:

$$\mathbf{y}_e = \begin{bmatrix} \phi_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \phi \\ \theta \end{bmatrix} + \begin{bmatrix} \beta_{xo} \\ \beta_{yo} \end{bmatrix} + \mathbf{w}_e \tag{27}$$

With ϕ_e and θ_e represent the angle between the body roll and pitch axes, respectively, and the horizon direction and ϕ and θ represent the true Euler angles that map the relation between body frame and orbital frame. β_{xo} and β_{yo} are the roll and pitch angular displacements between the Earth's horizon and the orbit frame. $\mathbf{w}_e$ is the white Gaussian noise with null mean and σ_e standard deviation.

Assuming a perfectly spherical Earth with constant, uniform radiation, the relative roll β_{xo} and pitch angle β_{yo} can be determined by [9]:

$$\beta_{xo} = \beta_{yo} = \frac{\pi}{2} - \arcsin \frac{R_E}{R_E + a} \tag{28}$$

where R_E is the equatorial radius of the Earth and a is the orbit altitude.

The measured nadir direction as seen from the body frame is given by:

$$\mathbf{n_B} = \mathbf{A_{BO}} \mathbf{n_O} = \begin{bmatrix} -\sin\theta \\ \sin\theta \sin\phi \\ \cos\theta \cos\phi \end{bmatrix} \tag{29}$$

Note that the knowledge of the yaw angle is not necessary to obtain the nadir direction. Therefore, an estimation of the nadir vector can be obtained from the observed vectors $\mathbf{y_e}$ and the position vector estimated by the OBC (on board computer). Most state of the art nadir sensors are capable of calculating the nadir direction and giving it directly.

The Earth sensor available in the simulation tool was modeled after the CubeSpace's CubeSense Nadir Sensor. This sensor consists of a CMOS based digital camera with a wide field of view.

8.2. Star Tracker

Most state-of-the art star trackers provide the attitude of the satellite directly as a quaternion [1]. The star tracker is modeled as a "black box" and the internal processes of this sensor were not taken into account. This measurement is corrupted by additive Gaussian white noise. Brute force normalization is required to ensure that the output is a unit quaternion.

9. Attitude Estimator Block

This block presents multiple options of both static and recursive algorithms.

9.1. Static Determination

The three options available for the user are the TRIAD algorithm, utilized in the ORCASat [8], the QUEST and the Q-method. Both the Q-method and QUEST are based on Whaba's problem [1].

9.1.1 Whaba's Problem

Wahba's problem is posed as a minimization problem where the cost function is posed as:

$$\mathbf{E} = \frac{1}{2}\sum_{k=1}^{N} a_k \|\mathbf{v_{kB}} - \mathbf{A_{BI}}\mathbf{v_{kI}}\|^2 \tag{30}$$

where $\mathbf{v_{1I}}...\mathbf{v_{nI}}$ is a set of reference unit vectors in the ECI reference frame, and $\mathbf{v_{1B}}...\mathbf{v_{nB}}$ are the corresponding observation unit vectors, measured in the body frame and $\mathbf{A}$ is the rotation matrix from the reference frame to the body frame. $a_1...a_n$ is a set of non-negative weights.

9.1.2 Q-method

The loss function (30) can be expanded as:

$$\begin{aligned}\mathbf{E} = \frac{1}{2}\sum_{k=1}^{N} a_k(\mathbf{v_{kB}} - \mathbf{A_{BI}}\mathbf{v_{kI}})^T(\mathbf{v_{kB}} - \mathbf{A_{BI}}\mathbf{v_{kI}}) = \\ \frac{1}{2}\sum_{k=1}^{N} a_k(\mathbf{v_{kB}}^T\mathbf{v_{kB}} + \mathbf{v_{kI}}^T\mathbf{v_{kI}} - 2\mathbf{v_{kB}}^T\mathbf{A_{BI}}\mathbf{v_{kI}})\end{aligned} \tag{31}$$

Since $\mathbf{v_{kB}}$ and $\mathbf{v_{kI}}$ are unit vectors $\mathbf{v_{kB}}^T\mathbf{v_{kB}} = \mathbf{v_{kI}}^T\mathbf{v_{kI}} = 1$. Substituting in (31):

$$\mathbf{E} = \sum_{k=1}^{N} a_k - g(\mathbf{A}) \tag{32}$$

With $g(\mathbf{A}) = \sum_{k=1}^{N} a_k \mathbf{v_{kB}}^T\mathbf{A_{BI}}\mathbf{v_{kI}}$. Minimizing $\mathbf{J}$ is the same as maximizing $g(\mathbf{A})$.

The attitude matrix $\mathbf{A_{BI}}$ can be expressed in terms of quaternions so the cost function can be re-written in terms of quaternions as [10]:

$$g(\mathbf{q}) = \mathbf{q}^T\mathbf{K}(\mathbf{B'})\mathbf{q} \tag{33}$$

where $\mathbf{K}(\mathbf{B},)$ is the traceless symmetric matrix:

$$\mathbf{K}(\mathbf{B'}) = \begin{bmatrix}\mathbf{B'} + \mathbf{B'^T} - tr(\mathbf{B'})\mathbf{I_3} & -\mathbf{Z} \\ \mathbf{Z}^T & tr(\mathbf{B},)\end{bmatrix} \tag{34}$$

with,

$$\mathbf{B'} = \sum_{i=1}^{N} a_k \mathbf{v_{kB}}\mathbf{v_{kI}}^T \tag{35}$$

and

$$\mathbf{Z} = \begin{bmatrix}B'_{23} - B'_{32} \\ B'_{31} - B'_{13} \\ B'_{12} - B'_{21}\end{bmatrix} \tag{36}$$

To find the value of $\mathbf{q}$ that maximizes $g(\mathbf{q})$, its derivative with respect to $\mathbf{q}$ must be evaluated. Since the four elements of the quaternion are not independent, a new gain function is created with this constraint in mind through the addition of a Lagrange multiplier [10]:

$$g(\mathbf{q})' = \mathbf{q}^T\mathbf{K}(\mathbf{B'})\mathbf{q} - \lambda\mathbf{q}^T\mathbf{q} \tag{37}$$

The differentiation of this gain function yields a stationary point for:

$$\mathbf{q}\mathbf{K}(\mathbf{B'}) = \lambda\mathbf{q} \tag{38}$$

The optimal quaternion is given by to the eigenvector that corresponds to the largest eigenvalue of $\mathbf{K}$.

9.1.3 QUEST

Recalling equation (32) and substituting for a maximum value of $g(\mathbf{A}) = \lambda_{opt}$:

$$\lambda_{opt} = \sum_{k=1}^{N} a_k - \mathbf{E} \tag{39}$$

The optimal eigenvalue of $\mathbf{K}$ the cost function $\mathbf{E}$ should be small, such that [10]:

$$\lambda_{opt} \approx \sum a_k \tag{40}$$

Starting from this assumption (λ_0 given by (40)) and since the QUEST algorithm obtains an estimate value λ_{opt} through a Newton-Raphson iteration process [10]:

$$\lambda_i = \lambda_{i-1} - \frac{f(\lambda_{i-1})}{f\prime(\lambda_{i-1})} \tag{41}$$

where $f(\lambda)$ is the characteristic equation:

$$f(\lambda) = \det\left(\mathbf{K} - \lambda\mathbf{I_{4\times4}}\right) \tag{42}$$

For most cases, only a single iteration is required [1]. By combining (34) and (38) and substituting the

known optimal eigenvalue λ_{opt}, the optimal quaternion is given by:

$$\frac{\mathbf{q_{1:3}}}{q_4} = [(\lambda_{opt} + tr\mathbf{B}')\mathbf{I_3} - \mathbf{S}]^{-1}\mathbf{Z} \tag{43}$$

$$\|\mathbf{q}\| = 1 \tag{44}$$

where (44) states the unit quaternion condition.

The QUEST method encounters a singularity for rotations of 180^o about any of the axis ($q_4 = 0$) [1]. Nevertheless, this singularity can be avoided by applying a method of sequential rotations.

9.2. Extended Kalman Filters

The EKF algorithms used in this thesis are summarized in table 9.2.

Table 1: General EKF algorithm

Step	
1.	Calculate $\mathbf{F}(\hat{\mathbf{x}}^+_{k-1})$ and $+\mathbf{B}\mathbf{u}_{k-1}$
2.	Calculate $\hat{\mathbf{x}}^-_{\mathbf{k}} = \mathbf{F}(\mathbf{x}^+_{\mathbf{k-1}})\mathbf{x}^+_{\mathbf{k-1}} + \mathbf{B}\mathbf{u}_{\mathbf{k-1}}$
3.	Calculate $\mathbf{P}^-_k = \mathbf{F}^+_{k-1}\mathbf{P}^+_{k-1}(\mathbf{F}^+_{k-1})^T + \mathbf{G}\mathbf{Q}\mathbf{G}^T$
4.	Calculate $\mathbf{H}_k$
5.	Calculate $\mathbf{K}_k = \mathbf{P}^-_k\mathbf{H}^T_k(\mathbf{H}_k\mathbf{P}^-_k\mathbf{H}^T_k + \mathbf{R})^{-1}$
6.	Calculate $\Delta\hat{\mathbf{x}} = \mathbf{K}_k[\mathbf{y}_k - \mathbf{H}_k\mathbf{x}^-_k]$
7.	Calculate $\hat{\mathbf{x}}^+_k = \hat{\mathbf{x}}^-_k + \Delta\hat{\mathbf{x}}$
8.	Calculate $\mathbf{P}^+_k = (\mathbf{I}_n - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}^-_k$

For each iteration, steps 1. to 8. are followed. The first 3 steps correspond to the predict phase and the remaining to the update phase. When the quaternion is part of the state vector, $\hat{\mathbf{q}}^+_k$ is divided by its norm in step 7. in order to maintain the unit value:

$$\hat{\mathbf{q}}^+_{kcorrected} = \frac{\hat{\mathbf{q}}^+_k}{\|\hat{\mathbf{q}}^+_k\|} \tag{45}$$

$\mathbf{F}(\hat{\mathbf{x}}^+_{k-1})$ and $+\mathbf{B}\mathbf{u}_{k-1}$ depend on the process and on the state vector $\mathbf{x}$ and $\mathbf{H}_k$ depends on the sensor measurements and the state vector. Its calculations will be particularized for the star tracker additive EKF developed in the following subsection.

9.2.1 Additive Extended Kalman Filter

The additive extended Kalman filter treats the four elements of the quaternion as independent of each other, calculating the quaternion error as [11]:

$$\mathbf{q} = \hat{\mathbf{q}} + \mathbf{\Delta q} \tag{46}$$

where $\hat{\mathbf{q}}$ is the estimated value of $\mathbf{q}$ and $\mathbf{\Delta q}$ is the estimation error. The normalization condition is enforced in the update state using the "brute force" approach expressed in (46).

The AEKF estimates the state vector:

$$\mathbf{x} = \begin{bmatrix}\mathbf{q}\\ \boldsymbol{\omega}\end{bmatrix} \tag{47}$$

where $\mathbf{q}$ stands for the unit quaternion and $\boldsymbol{\omega}$ represents the body rates. The value of $\dot{\mathbf{x}}$ can be calculated by combining the previously stated equations of kinematics and dynamics:

$$\dot{\mathbf{q}} = \frac{1}{2}\Omega(\boldsymbol{\omega})\mathbf{q} \tag{48}$$

$$\mathbf{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times (\mathbf{J}\boldsymbol{\omega} + \mathbf{h}) - \tau - \dot{\mathbf{h}} \tag{49}$$

where $\mathbf{h}$ is the angular momentum exchange device $\mathbf{h^{med}}$, when reaction wheels, momentum wheels or control moment gyros cluster are present. When the satellite is not equipped with moment devices $\mathbf{h} = \mathbf{0}$.

9.2.2 Predict

The predict step follows the steps 1-3 in table 9.2, where the error transition matrix is obtained by substituting (48):

$$\mathbf{F_{k-1}} = \mathbf{I_7} + \begin{bmatrix}\mathbf{F_{11}} & \mathbf{F_{21}}\\ \mathbf{F_{12}} & \mathbf{F_{22}}\end{bmatrix} \tag{50}$$

$$\mathbf{F_{11}} = \frac{\Delta t}{4}\begin{bmatrix} 0 & (\omega_z)^+_{k-1} & -(\omega_y)^+_{k-1} & (\omega_x)^+_{k-1}\\ -(\omega_z)^+_{k-1} & 0 & (\omega_x)^+_{k-1} & (\omega_y)^+_{k-1}\\ (\omega_y)^+_{k-1} & -(\omega_x)^+_{k-1} & 0 & (\omega_z)^+_{k-1}\\ -(\omega_x)^+_{k-1} & -(\omega_y)^+_{k-1} & -(\omega_z)^+_{k-1} & 0\end{bmatrix} \tag{51}$$

$$\mathbf{F_{12}} = \frac{\Delta t}{4}\begin{bmatrix} (q_4)^+_{k-1} & -(q_3)^+_{k-1} & (q_2)^+_{k-1}\\ (q_3)^+_{k-1} & (q_4)^+_{k-1} & -(q_1)^+_{k-1}\\ -(q_2)^+_{k-1} & (q_1)^+_{k-1} & (q_4)^+_{k-1}\\ -(q_1)^+_{k-1} & -(q_2)^+_{k-1} & -(q_3)^+_{k-1}\end{bmatrix} \tag{52}$$

$$\mathbf{F_{21}} = \mathbf{0_{3\times 4}} \tag{53}$$

$$\mathbf{F_{22}} = \Delta t\begin{bmatrix} 0 & \frac{(\omega_z)^+_{k-1}(J_y - J_z) - (h_z)^+_{k-1}}{2J_x} & \frac{(\omega_y)^+_{k-1}(J_y - J_z) - (h_y)^+_{k-1}}{2J_x}\\ \frac{(\omega_z)^+_{k-1}(J_z - J_x) - (h_z)^+_{k-1} - H_0}{2J_y} & 0 & \frac{(\omega_x)^+_{k-1}(J_z - J_x) - (h_x)^+_{k-1}}{2J_y}\\ \frac{(\omega_y)^+_{k-1}(J_x - J_y) - (h_y)^+_{k-1} + H_0}{2J_z} & \frac{(\omega_x)^+_{k-1}(J_x - J_y) - (h_x)^+_{k-1}}{2J_z} & 0\end{bmatrix} \tag{54}$$

where H_0 is the orbit angular momentum.

As previously stated in 9.2:

$$\hat{\mathbf{x}}^-_{\mathbf{k}} = \mathbf{F}(\mathbf{x}^+_{\mathbf{k-1}})\mathbf{x}^+_{\mathbf{k-1}} + \mathbf{B}\mathbf{u}_{\mathbf{k-1}} \tag{55}$$

With

$$\mathbf{B}\mathbf{u}_{k-1} = \begin{bmatrix}\mathbf{0}_{4\times 1}\\ \mathbf{J}^{-1}[\tau_{k-1} - \dot{\mathbf{h}}_{k-1}]\delta t\end{bmatrix} \tag{56}$$

9.2.3 Update

Most commercially available star trackers output directly the attitude of the satellite as a quaternion:

$$\hat{\mathbf{y}}_{\mathbf{k}} = \mathbf{h}(\mathbf{x}_k) = \hat{\mathbf{q}}^-_{\mathbf{k}} \tag{57}$$

The sensitivity matrix in step 4. of table 9.2 is simply given by [9]:

$$\mathbf{H_k} = \begin{bmatrix}\mathbf{I_{4x4}} & \mathbf{0_{4x3}}\end{bmatrix} \tag{58}$$

After this calculation, the algorithm follows steps 5. to 8. of table 9.2, with normalization of $\hat{\mathbf{q}}^+_k$ (equation (46)), for each iteration.

10. On Board Computer Block
The on board computer (OBC) block uses data obtained by the attitude sensors and/or the attitude determination algorithms in conjunction to position measurements obtained by the GPS to compute an estimate of all the kinematic and dynamic variables.

This block contains an environmental model subsystem that, similar to the spacecraft environment block. A propagator subsystem propagates both the position and attitude of the spacecraft along its orbit. This propagator is reset at regular intervals by the GPS measurements in order to reduce the accumulation of estimation errors propagated over time.

11. Power Analysis Block
This block analyses the power consumption and availability of the satellite. The power consumption is obtained by adding the power consumed by all the actuators at any instance in time. The power availability of a spacecraft is a function of the number of solar panels, their peak power for the expected solar power density and the angle between the sun line of sight and the inward normal of the panel.

12. Actuators Block
This block models different attitude actuators available for CubeSats, namely, magnetometers and momentum wheels [8], reaction wheels and control moment gyros (CMG), whose models are going to be further explained in sections 12.1 and 12.2, respectively. Several cluster architectures for each momentum exchange device.

12.1. Reaction Wheel Model
Reaction wheels consist of disks with null nominal angular momentum which are spun up by a motor driver. Configurations of one or two reaction wheels, are commonly used, but full three-axis attitude control requires the use of three or more wheels. The actuator model allows the user to simulate clusters of up to 3 orthogonal reaction wheels.

The angular momentum of a cluster of reaction wheels given in the body frame by:

$$\boldsymbol{h} = \sum_{i=1}^{N} \boldsymbol{h} = \sum_{i=1}^{N} \mathbf{J^{RW}} \Omega^{RW} \hat{\mathbf{g}}_{\mathbf{si}} \tag{59}$$

where $\mathbf{J_i^{RW}}$, Ω^{RW} and $\hat{\mathbf{g}}_{\mathbf{si}}$ represent, respectively, the moment of inertia, the magnitude of the spin rate and a unit vector in the direction of the spin axis of the i^{th} wheel in the body frame.

Applying (14) the torque of a cluster of reaction wheels in the body frame is:

$$\boldsymbol{\tau} = \sum_{i=1}^{N} \boldsymbol{\tau_i} = \mathbf{J^{RW}} \dot{\Omega}^{RW} \hat{\mathbf{g}}_{\mathbf{si}} \tag{60}$$

12.1.1 Motor Model

A basic direct current motor can be modeled as an armature circuit, constituted of a resistance (R_a) and an inductance (L_a) connected in series, and a voltage source (e_b) associated with the back electromotive force (emf) induced in the armature during rotation [12, 13]. Combining Newton's and Kirchoff's laws, the behavior of the motor can be described by:

$$L_a \frac{di_a}{dt} + R_a i_a = e_a - K_b \frac{d\theta}{dt} \tag{61}$$

$$J_m \frac{d^2\theta}{dt^2} + \tau_d = K_i i_a \tag{62}$$

where τ_d is the friction disturbance torque modeled as a combination of viscous and Coulomb friction contributions [1]:

$$\tau_d = B_m \frac{d\theta_{rel}}{dt} + csign(\frac{d\theta_{rel}}{dt}) \tag{63}$$

where B_m is the viscous friction coefficient, c is the Coulomb friction coefficient. And $\frac{d\theta_{rel}}{dt}$ is the relative velocity between the moving parts of the motor and the static ones. Disturbances caused by eccentricity of the wheel have been modeled as gaussian white noise.

The power consumption is calculated as:

$$P = R_a i_a^2 \tag{64}$$

The motor drives of reaction wheels typically accept a torque command [1]. A proportional integral derivative controller was developed for this effect.

12.2. Single Gimbal Control Moment Gyro model
To better understand the behavior of a cluster of single gimbal control moment gyros it is important to introduce two reference frames.

- **CMG reference frame:** given by the triad of unit vectors $\{\hat{\mathbf{g}}_\mathbf{s}, \hat{\mathbf{g}}_\mathbf{t}, \hat{\mathbf{g}}_\mathbf{g}\}$. $\hat{\mathbf{g}}_\mathbf{s}$ is a vector in the direction of spin of the spinning wheel, $\hat{\mathbf{g}}_\mathbf{g}$ has the direction of the gimbal axis and $\hat{\mathbf{g}}_\mathbf{t}$ completes the right hand triad. Out of this three unit vectors, only $\hat{\mathbf{g}}_\mathbf{g}$ is constant in time. Given an initial gimbal angle δ_0 [3]:

$$\hat{\mathbf{g}}_\mathbf{s}(t) = \cos(\delta(t) - \delta_0)\hat{\mathbf{g}}_\mathbf{s}(t_0) + \sin(\delta(t) - \delta_0)\hat{\mathbf{g}}_\mathbf{t}(t_0) \tag{65}$$

$$\hat{\mathbf{g}}_\mathbf{t}(t) = -\sin(\delta(t) - \delta_0)\hat{\mathbf{g}}_\mathbf{s}(t_0) + \cos(\delta(t) - \delta_0)\hat{\mathbf{g}}_\mathbf{t}(t_0) \tag{66}$$

- **Cluster reference frame:** common reference frame for all the CMGs in the cluster. It is constant in time and relates to the individual CMG reference frames through the current gimbal angle and constant shape parameters.

Typical CMG clusters are constituted by equal individual CMG. In the cluster frame:

$$\mathbf{h_C} = \sum_{i=1}^{N} \mathbf{h_{iC}} \tag{67}$$

where $\mathbf{h_{iC}}$ ($i = 1, ..n$ where n is the number of CMGs of the cluster) is given by [4]:

$$\begin{gathered}\mathbf{h_{iC}} = \mathbf{h^f_{iC}} + \mathbf{h^g_{iC}} = h^f_{iC}\hat{\mathbf{g}}_{\mathbf{siC}} + h^g_i\hat{\mathbf{g}}_{\mathbf{giC}} = \\ \mathbf{J^f_{iC}}\boldsymbol{\omega}^{\boldsymbol{f}}_{\boldsymbol{iC}} + \mathbf{J^g_{iC}}\dot{\boldsymbol{\delta}}_{\boldsymbol{i}} = \mathbf{J^f_{iC}}\omega^f_{iC}\hat{\mathbf{g}}_{\mathbf{siC}} + \mathbf{J^g_{iC}}\dot{\delta}_i\hat{\mathbf{g}}_{\mathbf{giC}}\end{gathered} \tag{68}$$

where $\mathbf{J^f_{iC}}$ and $\mathbf{J^g_{iC}}$ represent the inertia of the flywheel (spinning disk) and the gimbal, respectively and $\mathbf{h^f_{iC}}$ and $\mathbf{h^g_{iC}}$ represent the angular momentum of the flywheel and gimbal. $\hat{\mathbf{g}}_{\mathbf{siC}}$ and $\hat{\mathbf{g}}_{\mathbf{giC}}$ are the spin and gimbal axis.

Substituting in (14), the torque acting on the system can be obtained as:

$$\begin{gathered}\frac{\mathrm{d}\mathbf{h_i}}{\mathrm{d}t} = \mathbf{J^f_{iC}}\dot{\omega^f_{iC}}\hat{\mathbf{g}}_{\mathbf{siC}} + \mathbf{J^f_{iC}}\omega^f_i\dot{\hat{\mathbf{g}}}_{\mathbf{siC}} + \mathbf{J^g_{iC}}\ddot{\delta}_i\hat{\mathbf{g}}_{\mathbf{giC}} = \\ \mathbf{J^f_{iC}}\dot{\omega^f_{iC}}\hat{\mathbf{g}}_{\mathbf{siC}} + \dot{\delta}_i\mathbf{J^f_{iC}}\omega^f_{iC}\hat{\mathbf{g}}_{\mathbf{tiC}} + \mathbf{J^g_{iC}}\ddot{\delta}_i\hat{\mathbf{g}}_{\mathbf{giC}}\end{gathered} \tag{69}$$

$$\frac{\mathrm{d}\mathbf{h_{iC}}}{\mathrm{d}t} = \mathbf{J^f_{iC}}\dot{\boldsymbol{\omega}}^{\boldsymbol{f}}_{\boldsymbol{iC}} + \mathbf{J^g_{iC}}\ddot{\boldsymbol{\delta}}_{\boldsymbol{iC}} + \dot{\boldsymbol{\delta}}_{\boldsymbol{iC}} \times (\mathbf{J^f_{iC}}\boldsymbol{\omega}^{\boldsymbol{f}}_{\boldsymbol{iC}}) = \boldsymbol{\tau}^{\boldsymbol{d}}_{\boldsymbol{iC}} \tag{70}$$

where $\mathbf{J^f_{iC}}\dot{\boldsymbol{\omega}}^{\boldsymbol{f}}_{\boldsymbol{iC}}$ is the torque due to the acceleration of the flywheel i, $\mathbf{J^g_{iC}}\ddot{\boldsymbol{\delta}}_{\boldsymbol{iC}}$ is the torque due to the acceleration of the gimbal i, $\dot{\boldsymbol{\delta}}_{\boldsymbol{iC}} \times (\mathbf{J^f_{iC}}\boldsymbol{\omega}^{\boldsymbol{f}}_{\boldsymbol{iC}})$ is the control torque of the CMG i.

Real motors are subject to friction and other disturbances (subsection 12.1.1), the total torque acting on the spacecraft due to one CMG is:

$$\boldsymbol{\tau}_{\boldsymbol{i}} = \boldsymbol{\tau}^{\boldsymbol{control}} + \boldsymbol{\tau}^{\boldsymbol{fa}}_{\boldsymbol{i}} + \boldsymbol{\tau}^{\boldsymbol{ga}}_{\boldsymbol{i}} + \boldsymbol{\tau}^{\boldsymbol{ff}}_{\boldsymbol{i}} + \boldsymbol{\tau}^{\boldsymbol{gf}}_{\boldsymbol{i}} \tag{71}$$

$\boldsymbol{\tau}^{\boldsymbol{ff}}_{\boldsymbol{i}}$ and $\boldsymbol{\tau}^{\boldsymbol{gf}}_{\boldsymbol{i}}$ are the disturbance torques experienced by the flywheel and the gimbal. The total torque due to the cluster of N CMGs is:

$$\boldsymbol{\tau} = \sum_{i=1}^{N} \boldsymbol{\tau}_{\boldsymbol{i}} \tag{72}$$

For a cluster of CMG acting with the same angular momentum, h_{cmg}, (67) is reduced to:

$$\mathbf{h^f_C} = h_{cmg}\mathbf{M} \tag{73}$$

where $\mathbf{M}$ is a $3\times n$ matrix, n is the number of CMGs of the cluster, resulting by summing the directional matrices of each individual CMG axis. That is:

$$\mathbf{M} = \begin{bmatrix}\hat{\mathbf{g}}_{\mathbf{s1}} & \hat{\mathbf{g}}_{\mathbf{s2}} & .... & \hat{\mathbf{g}}_{\mathbf{sn}}\end{bmatrix} \tag{74}$$

Finally, the control torque can also be obtained as:

$$\boldsymbol{\tau}^{\boldsymbol{control}}_{\boldsymbol{C}} = \sum_{i=1}^{N} \mathbf{J_{iC}}(\delta_i)\dot{\delta}_i \tag{75}$$

For a cluster of CMG acting with the same angular momentum, (75) is reduced to:

$$\boldsymbol{\tau}^{\boldsymbol{control}}_{\boldsymbol{C}} = \mathbf{J}^*(\delta)\dot{\boldsymbol{\delta}} \tag{76}$$

where $\mathbf{J}^*$ is the Jacobian Matrix:

$$\mathbf{J}^* = \frac{\partial \mathbf{h^f_C}}{\partial \delta} \tag{77}$$

A big disadvantage of CMGs is that singularities may arise for certain values of $\mathbf{J}^*$. Internal singularities can be partially or totally avoided by employing the correct steering laws.

12.2.1 Steering Law and Motor Control

Table 2 summarizes the steering laws available in the Simulink tool.

Table 2: Steering laws available in the actuator block [1]

Architecture	Moore Penrose pseudo-inverse	Singularity robust inverse	Singularity passing inverse
Single	×		
Twin	×		×
Scissored Pair			
Pyramid	×	×	
Box-90	×	×	

The gimbal motor drives of reaction wheels typically accept a speed command, so a linear quadratic regulator controller (LQR) was designed.

13. Attitude Control

This block contains both an OBC controller and actuator controllers.

The set of available controllers can be divided into detumbling and nominal pointing algorithms. The actuator block is equipped with 4 detumbling algorithms developed in [8], as well as two nominal pointing algorithms: a constant gain linear quadratic regulator developed by [14] and a feedback controller for CMG equipped spacecrafts further detailed in subsection 13.1.

13.1. Feedback Controller for CMG

Recall the dynamics equation (49). A feedback control law was developed in [15] such that:

$$\boldsymbol{\tau}_{\boldsymbol{req}} = -\mathbf{K_p}\mathbf{q^e_{1:3}} - \mathbf{K_d}\boldsymbol{\omega}^{\boldsymbol{e}} \tag{78}$$

with

$$-\boldsymbol{\tau} = \boldsymbol{\omega} \times \mathbf{h} + \dot{\mathbf{h}} \tag{79}$$

Where $\boldsymbol{\tau_{req}}$ stands for the requested torque, $\mathbf{q^e_{1:3}}$ and $\boldsymbol{\omega^e}$ are the quaternion and body-rate vector errors of the spacecraft:

$$\mathbf{q^e} = \mathbf{q_{BI}} \otimes \mathbf{q^{-1}_{desired}} = \begin{bmatrix} \mathbf{q^e_{1:3}} \\ q^e_4 \end{bmatrix} \tag{80}$$

$$\boldsymbol{\omega^e} = \boldsymbol{\omega_B^{BI}} - \boldsymbol{\omega_B^{desired}} \tag{81}$$

Where $\mathbf{q_{BI}}$ and $\boldsymbol{\omega_B^{BI}}$ describe the current attitude and body rates of the spacecraft and $\mathbf{q^{-1}_{desired}}$ and $\boldsymbol{\omega_B^{desired}}$ refer to their desired values.

$\mathbf{K_p}$ and $\mathbf{K_d}$ are the gain matrices of the controller and were determined with a Lyapunov stability analysis.

14. Simulations and Results

The ORCASat is going to be analyzed using the simulator tool. Section 15 summarizes ORCASat's ADCS solution and section 17 compares ORCASat's ADCS solution with possible alternative solutions.

15. ORCASat's ADCS

15.1. ORCASat's ADCS

ORCASat utilizes a COTS solution by Cubespace. This solution comprises a 3-axis Fluxgate Magnetometer, a 180^o FOV CubeSense sun sensor with bore sight axis pointing towards $-\hat{\mathbf{b}}_\mathbf{3}$, aided by 10 photodiods, as well as a 3-Axis MEMS rate sensor. Regarding actuators, ORCASat is equipped with 3-axis magnetic torquers as well as Y-axis CubeWheel Small momentum wheel.

As suggested in [8], the TRIAD algorithm makes use of the sun sensor and magnetometer measurements to initialize a AEKF. Finally, detumbling is achieved with a Fast detumbling B-dot algorithm purposed in [8]. In nominal mode, control is achieved via a LQR purposed in [14].

16. Initial Conditions

The simulation base scenario used for the simulations is summarized in table 3.

Table 3: Base simulation conditions

Parameter	Value
Semimajor axis	$6768.24km$
Eccentricity	0.00005
Inclination	51.64^o
Right ascension of the ascending node	117.76^o
Argument of the Perigee	34.8^o
Initial Mean Anomaly	60^o
Initial quaternion orientation	$[1 \quad 0 \quad 0 \quad 0]^T$
Initial angular rates	$[0.2 \quad 0.2 \quad 0.2]^T$
Mass	$3.6kg$
Inertia Matrix	$\begin{bmatrix} 0.003 & 0 & 0 \\ 0 & 0.007 & 0 \\ 0 & 0 & 0.008 \end{bmatrix} kgm^2$
Parasitic dipole moment	$[0.00707 \quad 0 \quad 0.00707]^T \; Am^2$
Initial epoch	7pm, 1 January 2022

Even though most initial conditions impact the detumbling time and the power consumption during detumble, these do not significantly impact the mission requirements for a realistic range of values. The exception is the initial body rates that seems to have an impact on the estimation error. It was therefore decided that in order to better study the estimation performance of different architectures, it would be necessary to analyze their performance for different initial body rates.

17. Alternative Architectures

In subsection 17.1 alternatives to the sun sensor are tested whereas in subsection 17.2 an alternative to CubeSense's Cubewheel is tested.

17.1. Sensor Alternatives

The mission requirements have been shown to be mostly satisfied in case of sun sensor failure, but also, this sensor does not provide redundancy in case of magnetometer failure. This result lead to the study of two possible alternative architectures: one with only a magnetometer and one where the sun sensor is substituted by the Cubesense nadir sensor (aligned with the $+\hat{\mathbf{b}}_\mathbf{3}$ axis) [16].

Table 4 summarizes the obtained results:

Table 4: Different architectures comparison [16]

Architecture	Mass difference	Mean estimation error	Mean pointing error
Sun sensor + Magnetometers	$0g$	0.54^o	3.8^o
Magnetometers	$-30g$	0.54^o	3.9^o
Earth sensor + Magnetometers	$0g$	0.80^o	4.1^o

For the solution equipped with a nadir sensor the redundancy capabilities of the solution in case of failure of the magnetometer in nominal phase where studied. Even after magnetometer failure in the nominal phase of the mission, when equipped with a Earth sensor instead of a sun sensor, the spacecraft is capable of achieving the mission requirements.

Since the present sun sensor does not provide redundancy in case of magnetometer failure and its performance is not significantly better than the architecture with only magnetometers, this second option would be preferable due to the reduced mass and volume.

However, the nadir sensor could be a preferable option as the redundancy provide by this sensor reduces the risk of mission failure.

17.2. Cubewheel Alternative

In this subsection the Cubewheel momentum wheel is compared with a possible alternative flywheel (RW210 flywheel by Hyperion Technologies) Table 5 compares the two solutions:

Table 5: Different flywheels comparison [17]

Flywheel	Mass	Mean power consumption	Mean estimation error	Mean pointing error
Cubewheel	$60g$	$0.36W$	0.45^o	3.76^o
RW210	$21g$	$0.47W$	0.57^o	5.1^o

The smaller flywheel is enough to mostly maintain pointing and estimation requirements with substantial mass and volume savings, however, it can be seen that the average power consumption is larger in the RW210 flywheel architecture.

18. Conclusions

The present research had two main goals: the development of a conceptual design framework tool capable of simulating several different ADCS architectures and using this tool, analyzing ORCASats current ADCS solution and compare it with possible alternative architectures.

Regarding the first objective, the tool created is capable of modeling most conventional and state-of-the-art CubeSat ADCS architectures. Regarding the second objectives, alternative solutions to the current architecture where studied. It has been shown that the presence of the sun sensor does not significantly impact both pointing and estimation accuracy. Moreover, it has been shown that in spite of not being as accurate, the Cubesense nadir sensor provides redundancy in case of failure of the magnetometer in nominal mission phase, while having the same mass and volume of the Cubesense sun sensor.

Finally, the RW210 flywheel by Hyperion Technologies was tested as an alternative to Cubewheel. It was shown that even though the pointing and estimation accuracy are not strongly impacted by this alteration, the total average power consumption increases.

19. Future Work

The current developed tool is already capable of modeling with detail multiple models, however, the current star tracker model could use some improvement and the estimator and controller block could be equipped with extra alternative algorithms.

Finally, it would be interesting to adapt and complete the current design tool with a hardware-in-the-loop test bed for the ADCS hardware.

References

[1] F. L. Markley and J. L. Crassidis. *Fundamentals of spacecraft attitude determination and control*. Springer, 2014.

[2] B. Wie. *Space vehicle dynamics and control*. Aiaa, 1998.

[3] H. Schaub. Attitude dynamics fundamentals. *Encyclopedia of Aerospace Engineering*, 2010.

[4] V. Nagabhushan. *Development of control moment gyroscopes for attitude control of small satellites*. PhD thesis, University of Florida, 2009.

[5] A. N. Pechev. Feedback-based steering law for control moment gyros. *Journal of Guidance, Control, and Dynamics*, 30(3):848–855, 2007.

[6] D. T. Gerhardt and S. E. Palo. Passive Magnetic Attitude Control for CubeSat Spacecraft. *Small Satellite Conference*, page 10, 2010.

[7] T. Flatley and D. A. Henretty. A magnetic Hysteresis Model. Small Satellite Conference, 2010.

[8] Z. Pavanello. Simulation of a commercial off-the-shelf adcs with design of a pitch sun tracking attitude mode for the orcasat. Master's thesis, Università degli Studi di Padova, Instituto Superior Técnico, 2020.

[9] J. Tuthill. *Naval Postgraduate Thesis Design and Simulation of a Nano-Satellite Attitude Determination System*. PhD thesis, Naval Postgraduate School, 2009.

[10] G. Giorgi. *Attitude Determination*, pages 781–809. Springer International Publishing, "2017".

[11] F. L. Markley. Multiplicative vs. additive filtering for spacecraft attitude determination. *Dynamics and Control of Systems and Structures in Space*, (467-474):48, 2004.

[12] K. Ogata. *Ingenieria de control moderna*. Pearson Educación, 2003.

[13] R. M. A.-M. Hummadi. Simulation of optimal speed control for a dc motor using linear quadratic regulator (lqr). *Journal of Engineering*, 18(3), 2012.

[14] B. Sabino. Attitude determination and control system of the orcasat. Master's thesis, Instituto Superior Técnico, 2020.

[15] H. Okubo and S. Kuwamoto. Agile attitude control and vibration suppression of flexible spacecraft using control moment gyros. *IFAC Proceedings Volumes*, 46(19):417–422, 2013.

[16] CubeSpace. Adcs components. https://www.cubespace.co.za/products/adcs-components/cubesense/. Accessed Oct 2021.

[17] H. Technologies. High precision magnetometer. https://hyperiontechnologies.nl/products/mm200/. Accessed Oct 2021.