

GRANASAT-I: KINEMATIC AND DYNAMIC MODEL FOR ATTITUDE DETERMINATION AND CONTROL SYSTEM

Luis Marchante Arjona



UNIVERSIDAD DE GRANADA

Trabajo Fin de Grado en Física

Tutor: Andrés María Roldán Aranda

Departamento de Electrónica y Tecnología de Computadores
Universidad de Granada

Abstract

This Bachelor Thesis introduces a Theoretical Description and Simulink implementation of different ADCS (Attitude Determination and Control System), a fundamental feature of spacecrafts. As it makes possible to control the orientation and stabilize the satellite within its orbit. Which is as complex as fascinating. Mathematical Formalism and Physical Model are presented in order to fully understand this phenomenon and build it to make it work in Simulink, creating a simulation of a Digital Dynamical System that would be present at the on board computer of the satellite GranaSAT-I. In this context, the Aerospace GranaSAT Group at UGR is building the GranaSAT-I.

Este Trabajo de Final de Grado presenta una descripción teórica y su implementación Simulink de diferentes ADCS (sistema de control y determinación de la orientación), una característica fundamental de las naves espaciales. Ya que hace posible controlar la orientación y estabilizar el satélite dentro de su órbita. Algo que resulta tan complejo como fascinante. El formalismo matemático y el modelo físico se presentan para comprender completamente este fenómeno y construir un modelo para hacerlo funcionar en Simulink, creando una simulación de un sistema digital dinámico que estará presente en el ordenador a bordo del satélite GranaSAT-I. En este contexto, el Grupo Aeroespacial GranaSAT de la UGR está construyendo el GranaSAT-I.

Agradecimientos

Este trabajo, como tal, supone cerrar un capítulo de mi formación académica, pero también personal. Física es una carrera que potencia aspectos tales como el espíritu crítico, el razonamiento, creatividad, abstracción... además de fraternidad o trabajar en grupo, pero va más allá de todo eso, en esencia, hace que saque lo mejor y lo peor de uno, haciéndole a uno más humano.

Respecto a este proyecto, quiero reconocer la atenta supervisión y consejos del Profesor Andrés Roldán que ha sabido con mucho esfuerzo, crear la semilla de lo que un día llegará a materializarse como un satélite.

Es absolutamente necesario recordar el apoyo de mis padres, incondicional, sincero y afable. Este trabajo está completamente dedicado a mi Madre y a mi Padre. Con pundonor.

Construir un satélite no es trivial, y en particular un ADCS. Es necesario tener ambición y visión, pero sobretodo paciencia y atención en los detalles. Todo surge como una idea, que se puede explicar con ecuaciones físicas y modelos que describen la realidad hasta cierto punto, pero precisamente, poder modificar una pequeña parte de esta realidad para que haga exactamente lo que uno quiera, es complejo. Es decir, plasmar esas ideas en algo tan real que se puede construir y trascender los límites clásicos de la existencia humana, es algo sencillamente fascinante...

Per Aspera, Ad Astra.

Contents

1	Introduction	6
1.1	Attitude control for GranaSAT-I	6
1.2	Importance of ADCS	6
1.3	Structure of this work	7
2	Mathematical Formalism	8
2.1	Reference Frames	8
2.2	Vector Transformations	9
2.3	Euler angles and Quaternions	10
2.4	Lyapunov stability	10
2.4.1	Positive definite function	11
2.4.2	Stability of autonomous systems	11
2.4.3	Uniform global stability	12
2.4.4	Energy of the satellite	12
2.4.5	Lyapunov Function	13
3	Satellite Model	14
3.1	Orbital Mechanics	14
3.2	Dynamics	16
3.3	Kinematics	18
3.4	Environmental and Actuators Torques	19
3.4.1	Environmental Torques	19
3.4.2	Actuators Torques	22
4	Simulation Results	24
4.1	Angular Velocity Feedback Controller	25
4.2	Attitude Feedback Controller	27
4.3	B dot Controller	29
5	Conclusions	31
5.1	Energy	32
5.2	Magnetic Moment	33
5.3	Stability	34
6	Future Work and Recommendations	36

NOTATION

v	3x1 vector
v^T	vector transpose
v_b	vector expressed in b (body) frame
R_a^b	3x3 rotation matrix from frame a to b
R^T	matrix transpose
q_a^b	4x1 quaternion from frame a to b
ω_{ib}^b	angular velocity from body to inertia expressed in body frame
ω_{ob}^b	angular velocity from body to orbit expressed in body frame
ω_{io}^o	angular velocity from orbit to inertia expressed in orbit frame
ω_{io}^b	angular velocity from orbit to inertia expressed in body frame
I	3x3 inertia tensor
I_{3x3}	3x3 identity matrix
$S(v)$	3x3 skew-symmetric form of a vector

1 Introduction

1.1 Attitude control for GranaSAT-I

The attitude of a spacecraft is its orientation in space. [1]

The attitude of the satellite tends to vary under the action of couples, which may be external, due to radiation pressure or atmospheric drag on solar panels, or internal, due to mechanical motion of the instrument motors. A stabilization system is thus required to maintain the satellite in the right position relative to the local orbital frame.

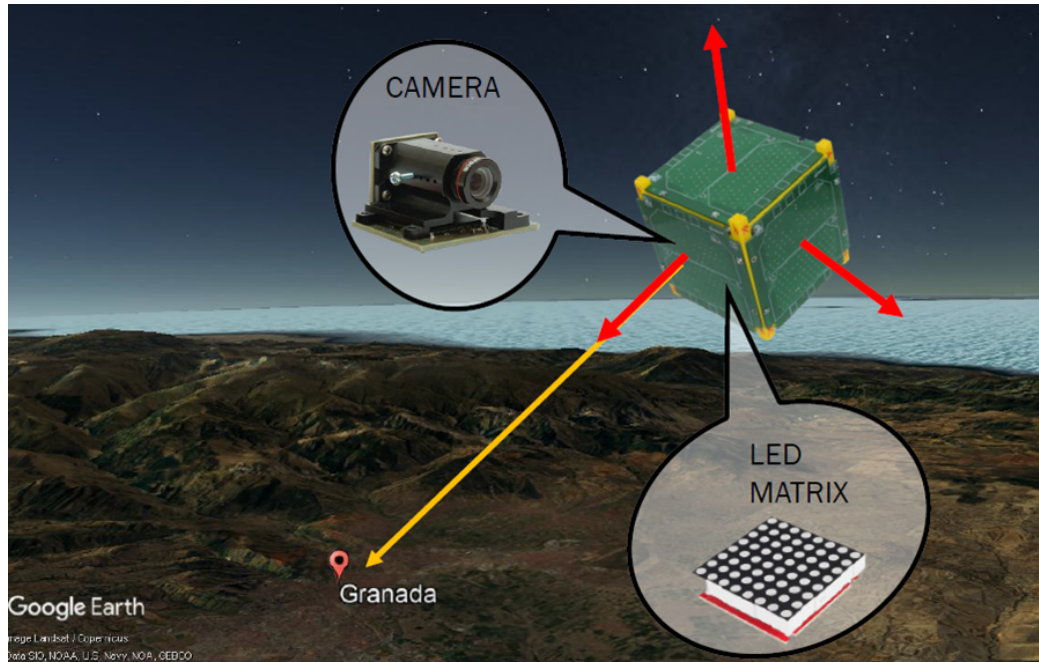


Figure 1: GranaSAT-I pointing ground station. Potential implementation of a LED matrix and a camera in the cubesat.

1.2 Importance of ADCS

The attitude determination and control system (ADCS) plays an indispensable part in satellite on-orbit operation which could greatly affect the satellite's performance.

When the satellite is dropped by its launcher, e.g. ISS, it certainly will not be stable but it will twist upon itself. The first mission of this system is thus to stabilize the satellite.

All in all, ADCS allows us performing precise measurements or observing towards Earth's surface in orbit, which requires the satellite to be stable, as well as for receiving or transmitting telemetry data or collecting sun's energy.

1.3 Structure of this work

For a proper understanding of the movement and orientation of the spacecraft, with **GranaSAT-I characteristics** (4), is natural to define, firstly, a **Reference Frame** in which to study **Kinematic** (equations that express how position changes for a given velocity) and **Dynamic** (equations describing how velocity changes for a given force) properties of the satellite.

Throughout this work there will be defined five different reference frames, being of special importance Orbital frame and **Body Frame, B** (2.1), and from where most of equations are defined.

In the Satellite Model section, using **Euler's Equations** for rigid body dynamics in **B** for computing Attitude Dynamics results to be useful for small maneuvers, but has singularities, therefore **Quaternions are introduced**. (2.3). Then, environmental torques are introduced, important to notice that aerodynamic pressure/drag and solar radiation pressure are not modeled in Simulink.

In Mathematical Model section, environmental and actuators torques are modeled, thus obtaining the equations necessary for **Attitude Determination and Stabilization** through Lyapunov's Stability (2.4).

Finally, using **Simulink** (6) environment to perform various simulations in order to test three different **Attitude Control Algorithms** (4.1), yields consistent results that lead to **Conclusions** (5) about energy, applied torque and stabilization. Additionally, **Recommendations** (6) for future work are presented.

Appendix (6) is *not necessary* to read in order to understand this work.

2 Mathematical Formalism

2.1 Reference Frames

This section introduces different reference frames for representing satellite position and attitude. Essentially, we have:

Inertial frame (ECI): The Earth-centered inertial frame is an inertial frame for terrestrial navigation, which means that it is a non-accelerated reference frame in which Newton's laws of motion apply. The origin of the frame is located at the center of the Earth. Here sub-index i is used.

Orbit frame (O): The orbit frame has its origin in the satellite's center of mass. The x_o -axis points in the normal direction of the orbital plane, while the z_o -axis points to the Earth center and the y_o -axis completes the right-hand orthogonal system. It is a non-inertial reference frame.

Body frame (B): The body-fixed reference frame is a moving coordinate frame which is fixed on the vessel. The axes are locked in the satellite, x_b -axis is forward, z_b -axis is downwards and the y_b -axis completes the right-hand orthogonal system. The origin is situated at the center of mass. This coordinate system rotates with the satellite, x_b , y_b and z_b coincide with principal axes of the moment of inertia tensor.

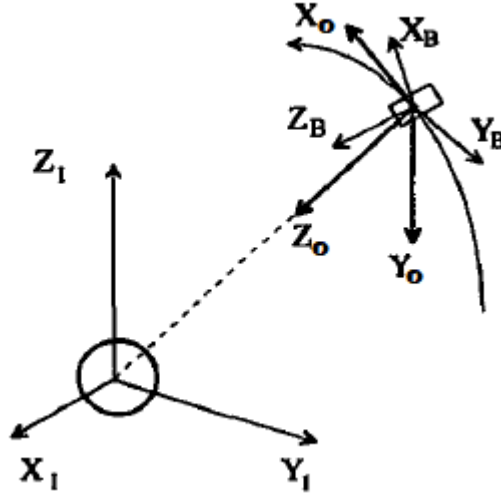


Figure 2: Inertial frame (I), Orbit frame (O) and Body frame (B) representations.

2.2 Vector Transformations

This section contains the main principles when transforming between different reference frames, and introduces unit quaternions and the inertia matrix.

Rotation Matrix

In terms of generic angles between each axes from one reference frame to the other, θ , the rotation matrix may be written:

$$R_a^{a'} = \begin{pmatrix} \cos(\theta_{x'x}) & \cos(\theta_{x'y}) & \cos(\theta_{x'z}) \\ \cos(\theta_{y'x}) & \cos(\theta_{y'y}) & \cos(\theta_{y'z}) \\ \cos(\theta_{z'x}) & \cos(\theta_{z'y}) & \cos(\theta_{z'z}) \end{pmatrix} \quad (1)$$

Cross product operator

The vector cross product is defined by [22] :

$$\lambda \times a = S(\lambda)a \quad (2)$$

where S is defined as:

$$S(\lambda) = -S(\lambda)^T = \begin{bmatrix} 0 & -\lambda_3 & \lambda_2 \\ \lambda_3 & 0 & -\lambda_1 \\ -\lambda_2 & \lambda_1 & 0 \end{bmatrix} ; \lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} \quad (3)$$

The super-index T means the transpose of the matrix. The angle-axis parameterization of the rotation matrix, $R_{\lambda,\theta}$, corresponding to a rotation θ about the λ -axis:

$$R_{\lambda,\theta} = I_{3 \times 3} + \sin(\theta)S(\lambda) + (1 - \cos(\theta))S^2(\lambda) \quad (4)$$

is an useful parameterization.

Rotation matrix differential equation

The time derivative of the rotation matrix between a frame a and a frame b is:

$$\dot{R}_b^a = R_b^a S(\omega_{ba}^a) \quad (5)$$

ω_{ba}^a is the angular velocity of a with respect to b expressed in a .

2.3 Euler angles and Quaternions

Attitude (Euler angles)

The attitude of a satellite can be represented by the roll, pitch and yaw angles [4].

$$\Theta = \begin{pmatrix} \varphi \\ \theta \\ \psi \end{pmatrix} \quad (6)$$

where roll φ is the rotation about the x -axis, along the velocity vector, completing a right-handed orthonormal base there is pitch θ about the y -axis and yaw ψ about the z -axis.

Quaternions

Quaternions, represented by q , which is a complex number with one real part η and three imaginary parts given by the ε vector, where θ is a rotation about the unit vector λ [10]:

$$\eta = \cos\left(\frac{\theta}{2}\right), \quad \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{pmatrix} = \lambda \sin\left(\frac{\theta}{2}\right), \quad q = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix} = \begin{pmatrix} \eta \\ \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{pmatrix} \quad (7)$$

unit quaternion satisfies $q^T q = 1$, which also means that: $\eta^2 + \varepsilon_1^2 + \varepsilon_2^2 + \varepsilon_3^2 = 1$.

Conversions Between Unit Quaternions and Euler Angles

Rotations in three dimensions can be represented using both Euler angles and unit quaternions. An unit quaternion can be described as:

$$\begin{aligned} q_1 &= \cos\left(\frac{\alpha}{2}\right) & q_3 &= \sin\left(\frac{\alpha}{2}\right)\cos(\beta_y) \\ q_2 &= \sin\left(\frac{\alpha}{2}\right)\cos(\beta_x) & q_4 &= \sin\left(\frac{\alpha}{2}\right)\cos(\beta_z) \end{aligned} \quad (8)$$

where α is a simple rotation angle and β_k are the direction of cosines locating the axis of rotation.

2.4 Lyapunov stability

In order to introduce stability as a mathematical concept in order to formulate a control system, the most natural approach would be using Lyapunov stability, which concerns the stability of a system's equilibrium points [1, 22]. An autonomous system is defined:

$$\dot{x} = f(x) \quad (9)$$

A non autonomous system, where $f(\cdot, x)$ is continuous and $f(t, \cdot)$ is locally Lipschitz uniformly in t :

$$\dot{x} = f(t, x) \quad (10)$$

2.4.1 Positive definite function

A function $V(x)$ is positive definite if $V(0) = 0$ and $V(x) > 0$ for $x \neq 0$, and it is positive semidefinite if $V(x) \geq 0$ for $x \neq 0$. A function is negative definite and negative semidefinite if $-V(x)$ is positive definite or positive semidefinite, respectively. A function $V(t, x)$ is positive semidefinite if $V(t, x) \geq 0$. It is positive definite if $V(t, x) \geq W_1(x)$ for some positive definite function $W_1(x)$ and it is radially unbounded if $W_1(x)$ is so, and decrescent if $V(t, x) \leq W_2(x)$.

2.4.2 Stability of autonomous systems

Definition 1

The equilibrium point $x = 0$ of (9) is stable in $D \subset \mathbb{R}^n$ (the domain containing x), if the continuously differentiable function $V : D \rightarrow \mathbb{R}$

$$V(0) = 0 \text{ and } V(x) > 0 \text{ in } D - \{0\} \quad (11)$$

$$\dot{V}(x) \leq 0 \text{ in } D - \{0\} \quad (12)$$

And asymptotically stable if:

$$\dot{V}(x) < 0 \text{ in } D - \{0\} \quad (13)$$

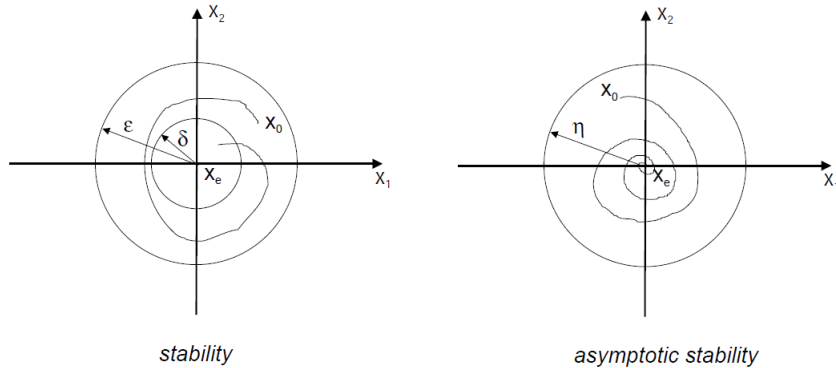


Figure 3: Illustration on both concepts of stability and asymptotic stability.

2.4.3 Uniform global stability

Definition 2

A continuous function $\alpha : [0, a) \rightarrow [0, \infty)$ is said to belong to class κ if it is strictly increasing and $\alpha(0) = 0$. It is said to belong to class κ_∞ if $a = \infty$ and $\alpha(r) \rightarrow \infty$ as $r \rightarrow \infty$.

Definition 3

The origin of a non autonomous system as (10) is said to be uniformly globally stable (UGS) if there exists $\alpha \in \kappa_\infty$ such that, for each $(t_o, x_o) \in \mathbb{R} \times \mathbb{R}^n$ each solution $x(\cdot, t_o, x_o)$ satisfies the following condition:

$$|x(t, t_o, x_o)| \leq \alpha(|x_o|) \quad \forall t \geq t_o \quad (14)$$

2.4.4 Energy of the satellite

The total energy of the satellite is divided into kinetic and potential energy. The kinetic energy is principally a result of the rotation in the inertial and orbit frame. These expressions are obtained in [6].

Kinetic Energy

The expression for the kinetic energy is revolved in body frame with respect to the orbit frame, and assuming a near circular orbit,

$$E_{kin} = \frac{1}{2}(\omega_{ob}^b)^T I \omega_{ob}^b \quad (15)$$

Potential Energy

The potential energy due to the gravity gradient E_{gg} and the potential energy due to revolution of the satellite about Earth, E_{gyro} ,

$$E_{gg} = \frac{3}{2}\omega_o^2((c_3^b)^T I c_3^b - I_z) \quad (16)$$

$$E_{gyro} = \frac{1}{2}\omega_o^2(I_x - (c_1^b)^T I c_1^b) \quad (17)$$

The total energy can be expressed as,

$$E_{tot} = \frac{1}{2}(\omega_{ob}^b)^T I \omega_{ob}^b + \frac{3}{2}\omega_o^2((I_x - I_z)c_{13}^2 + (I_y - I_z)c_{23}^2 + \frac{1}{2}\omega_o^2((I_x - I_y)c_{21}^2 + (I_x - I_z)c_3^2) \quad (18)$$

2.4.5 Lyapunov Function

For studying Lyapunov stability, explained above in this section, we need to define a function known as Lyapunov function. But before, the Lyapunov candidate, a potential Lyapunov function, that has to fulfill some special requirements is presented in the next proposition,

Proposition 1

Lyapunov *candidate*,

$$V(x) = E_{tot} \quad (19)$$

where E_{tot} is defined in (18) satisfies,

$$V(0) = 0 \quad (20)$$

$$V(x) > 0 \quad \forall x \neq 0 \quad (21)$$

Proof

From (18) is clear,

$$x = [\omega_{ob}^b, c_{21}, c_{31}, c_{13}, c_{23}]^T \quad (22)$$

so if $x = 0$ then $V(0) = 0$. To ensure that $V(x)$ is positive, the requirement is then,

$$I_x > I_y > I_z \quad (23)$$

so the energy function is positive definite. This is a **very important result for** model implementation in **Simulink**, otherwise the system might be uncontrollable.

3 Satellite Model

3.1 Orbital Mechanics

Defining the Frame of Reference

Consider a satellite in periodic motion around the Earth. Let us define the frame $(O; x, y, z)$. The origin O is the center of the Earth, which is taken to be a sphere Σ .

The axis Oz is the axis joining the poles, oriented from the south to the north. The plane xOy is the equatorial plane of the Earth, denoted E , which cuts the terrestrial sphere at the equator.

The axis Ox is chosen arbitrarily to point towards a distant star.

The axis Oy is deduced from the other two axes in such a way as to obtain a right-handed orthonormal frame.

The frame associated with this coordinate system is considered to be Galilean and will be denoted by R .

The motion of the satellite is Keplerian, i.e., it occurs on a Keplerian orbit.

In R , the trajectory is a conic section, in this case an ellipse, with one focus at the center of attraction O , and lying in a plane P , the orbital plane.

In this Galilean frame, the orbital plane P is fixed.

Let OZ denote the straight line perpendicular to P at O . The intersection of the planes P and E is a straight line through O , called the line of nodes.

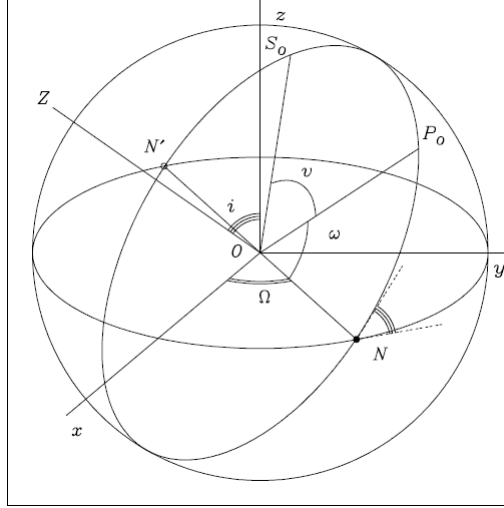


Figure 4: Ground track and orbital elements. The point S_o is the projection of the point S (satellite) and the point P_o is the projection of the perigee P onto the ground track. The point N is the projection of the ascending node and N' is the projection of the descending node. The equatorial plane of the Earth is (xOy, N, N') , normal to Oz . Orbital plane: (P_o, S_o, N, N') , normal to OZ . Three of the orbital elements are the Euler angles: the longitude of the ascending node (Ω), the inclination (i), and the argument of the perigee (ω). The fourth, here the true anomaly (v), specifies the point on the ellipse. The two other parameters (a and e), semi-major axis and eccentricity, serve to define the shape of the ellipse and are not shown here.

Specifying a Point on an Orbit

In order to specify a point in Keplerian motion in space, the first step is to identify the orbit, and then the point on the orbit. We thus define successively:

- (a) the location of the orbital plane in this frame,
- (b) the position of the elliptical orbit in this plane,
- (c) the characteristics of the ellipse,
- (d) the position of the moving point (i.e., the satellite) on the orbit.

We shall find that six parameters are necessary and sufficient to determine the position of the satellite in R .

Keplerian Elements

The parameters discussed above define the orbit and the position of the satellite on the orbit. These parameters constitute the six orbital elements, also known as the Keplerian elements. They are generally organized in the following order:

$$a, e, i, \Omega, \omega, M$$

The parameter a has dimensions of length, whilst the five others (e and the four angles) are dimensionless.

We should ask why there are 6 parameters. Here are two equivalent reasons:

- Three points define the position of a solid in space. Once the point O is fixed, 6 parameters (2 times 3 position coordinates) define the two other points.
- The position of a point (3 position coordinates) and its velocity (3 velocity components) at a given time can provide the initial conditions required to integrate the equations of motion, thereby defining the position of a point on its trajectory.

Since its axes are specified (OP orients the ellipse), the ellipse is characterized by two parameters:

- Length of the semi-major axis and eccentricity, respectively: a and e

Using standard astronomical notation, the three Euler angles are :

- Ω , called here the right ascension of the ascending node or the longitude of the ascending node: $\Omega = (Ox, ON)$.
- i , called here the inclination. This is the dihedral angle $i = (E, P)$ between the equatorial and orbital planes, $i = (Oz, OZ)$.
- ω , called here the argument of the perigee: $\omega = (ON, OP_o)$.

3.2 Dynamics

A satellite can be regarded as an ideal rigid body. The dynamic model of the satellite is derived using a Newton-Euler formulation, where the angular momentum change is related to applied torque.

Euler's second law states that the rate of change of angular momentum L about a point that is fixed in an inertial reference frame (often the mass center of the

body), is equal to the sum of the external moments of force (torques) acting on that body about that point:

$$\frac{d\vec{L}}{dt} = \vec{\tau} \quad (24)$$

Angular momentum can be expressed as:

$$\vec{L}_o = I_o \vec{\omega}_o \quad (25)$$

Therefore, the rate of change of angular momentum vector is broken up in two parts. Change in angular momentum due to angular acceleration $I\dot{\vec{\omega}}$ and change in angular momentum due to inertia tensor rotation (considering a generic rotation matrix $R = R(t)$) which using rotation matrix differential equation (5), $\dot{R} = RS(\omega)$, $S(\omega)R = \omega \times R$, results in : $\vec{\omega} \times (I\vec{\omega})$.

$$\vec{\tau} = \frac{d\vec{L}}{dt} = \frac{d(I\vec{\omega})}{dt} = I\dot{\vec{\omega}} + \frac{d(RIR^T)}{dt}\vec{\omega} = I\dot{\vec{\omega}} + (\dot{R}IR^T + R\dot{I}R^T)\vec{\omega} \quad (26)$$

Using anticommutativity of cross product $a \times b = -(b \times a)$ and a property of skew-symmetric operator ($S = -S^T$):

$$I\dot{\vec{\omega}} + (\dot{R}IR^T + R\dot{I}R^T)\vec{\omega} = I\dot{\vec{\omega}} + S(\vec{\omega})RIR^T\vec{\omega} + RIR^TS^T(\vec{\omega})\vec{\omega} = \quad (27)$$

$$= I\dot{\vec{\omega}} + \vec{\omega} \times (I\vec{\omega}) + I\vec{\omega} \times \vec{\omega} = I\dot{\vec{\omega}} + \vec{\omega} \times (I\vec{\omega}) \quad (28)$$

Having:

$$\vec{\tau} = I\dot{\vec{\omega}} + \vec{\omega} \times (I\vec{\omega}) \quad (29)$$

The satellite model is:

$$I\dot{\vec{\omega}}_{ib}^b + \vec{\omega}_{ib}^b \times (I\vec{\omega}_{ib}^b) = \vec{\tau}^b \quad (30)$$

where I is the moment of inertia, $\vec{\omega}_{ib}^b$ is the angular velocity from body to inertia decomposed in body frame and $\vec{\tau}^b$ are the torques acting on the satellite also decomposed in body frame (such as gravitational torque). Another way of representing the previous equation is :

$$I\dot{\vec{\omega}}_{ib}^b + S(\vec{\omega}_{ib}^b)I\vec{\omega}_{ib}^b = \vec{\tau}^b \quad (31)$$

The angular velocity of the satellite relative to the inertial frame is expressed in the body frame according to:

$$\vec{\omega}_{ib}^b = \vec{\omega}_{io}^b + \vec{\omega}_{ob}^b = R_o^b \vec{\omega}_{io}^o + \vec{\omega}_{ob}^b \quad (32)$$

Where $\vec{\omega}_{io}^b$ angular velocity from orbit to inertia decomposed in body frame.

3.3 Kinematics

The kinematics describes the satellite's orientation in space and is derived by integration of the angular velocity. First, is presented the most general case of a rotation matrix. A rotation matrix may also be referred to as a direction cosine matrix, because the elements of this matrix are the cosines of the unsigned angles between both reference frames.

Let a be a reference frame with unitary vectors $(\vec{e}_x, \vec{e}_y, \vec{e}_z)$ and a' other reference frame with unitary vectors $(\vec{e}_x', \vec{e}_y', \vec{e}_z')$. Then orientation from a to a' is completely determined by transformation matrix $R_a^{a'}$, this is the direction cosine matrix (DCM):

$$R_a^{a'} = \begin{pmatrix} \vec{e}_x' \cdot \vec{e}_x & \vec{e}_x' \cdot \vec{e}_y & \vec{e}_x' \cdot \vec{e}_z \\ \vec{e}_y' \cdot \vec{e}_x & \vec{e}_y' \cdot \vec{e}_y & \vec{e}_y' \cdot \vec{e}_z \\ \vec{e}_z' \cdot \vec{e}_x & \vec{e}_z' \cdot \vec{e}_y & \vec{e}_z' \cdot \vec{e}_z \end{pmatrix} \quad (33)$$

In terms of generic angles between each axes from one reference frame to the other, θ , the rotation matrix may be written:

$$R_a^{a'} = \begin{pmatrix} \cos(\theta_{x'x}) & \cos(\theta_{x'y}) & \cos(\theta_{x'z}) \\ \cos(\theta_{y'x}) & \cos(\theta_{y'y}) & \cos(\theta_{y'z}) \\ \cos(\theta_{z'x}) & \cos(\theta_{z'y}) & \cos(\theta_{z'z}) \end{pmatrix} \quad (34)$$

Euler's Theorem:

Any two independent orthonormal coordinate frames may be related by a minimum sequence of rotations (less than four) about coordinate axes, where no two successive rotations may be about the same axis.

Then, it is possible to bring a rigid body into an arbitrary orientation by performing three successive rotations. The composition of three rotations, one over each (x, y, z) axes, also suppose $\theta = \theta(t)$:

$$R(t) = \begin{pmatrix} \cos(\theta_3) & \sin(\theta_3) & 0 \\ -\sin(\theta_3) & \cos(\theta_3) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(\theta_2) & 0 & -\sin(\theta_2) \\ 0 & 1 & 0 \\ \sin(\theta_2) & 0 & \cos(\theta_2) \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_1) & \sin(\theta_1) \\ 0 & -\sin(\theta_1) & \cos(\theta_1) \end{pmatrix} \quad (35)$$

where θ_1 is a rotation over x -axis (roll usually represented by ϕ), θ_2 is a rotation over y -axis (pitch, θ) and θ_3 is a rotation over z -axis (yaw, ψ).

Finally, kinematics in this matrix form yields:

$$\frac{dR(t)}{dt} = R(t) \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix} \quad (36)$$

Euler angles representation of kinematics:

$$\begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{pmatrix} = \frac{1}{\cos(\theta_2)} \begin{pmatrix} \cos(\theta_3) & \sin(\theta_1)\sin(\theta_2) & \cos(\theta_1)\sin(\theta_2) \\ 0 & \cos(\theta_1)\cos(\theta_2) & -\sin(\theta_1)\cos(\theta_2) \\ 0 & \sin(\theta_1) & \cos(\theta_1) \end{pmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} \quad (37)$$

Euler angles representation is useful for small maneuvers, but has **singularities at $\cos(\theta_2)$** .

This is the reason we use quaternions, applying a conversion from Euler angles to quaternions:

$$\begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix} \quad (38)$$

Simplified:

$$\dot{q} = \begin{pmatrix} \dot{\eta} \\ \dot{\varepsilon} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} -\varepsilon^T \\ \eta I_{3 \times 3} + S(\varepsilon) \end{pmatrix} \omega_{ob}^b \quad (39)$$

To find the rotation velocity for the body frame relative to the orbit frame:

$$\vec{\omega}_{ob}^b = \vec{\omega}_{ib}^b - \vec{\omega}_o c_1^b = \vec{\omega}_{ib}^b - R_o^b \vec{\omega}_{io}^o \quad (40)$$

Where c are columns in $R_o^b = (c_1^b \ c_2^b \ c_3^b)$.

3.4 Environmental and Actuators Torques

3.4.1 Environmental Torques

In order to design the attitude control and prediction system, environmental disturbance torques acting on the spacecraft shall be modeled sufficiently. The torques must be modeled as a function of time, the spacecraft's position and attitude so that they can be integrated to Euler's equations and any other mathematical models.

The dominant sources of environmental disturbance torques on the spacecraft attitude are the solar radiation pressure, aerodynamic drag and Earth's gravitational and magnetic fields.

EXTERNAL DISTURBANCES

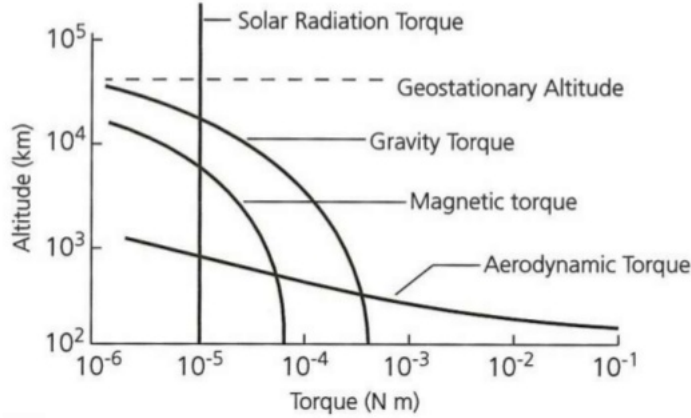


Figure 5: Environmental disturbance torques as a function of altitude. Diagrams like this are strongly dependent on the mass and geometry of the spacecraft, although proves to be useful for having an idea of the typical order of magnitude of these torques. The solar radiation pressure is effective on attitude of the satellite for altitudes higher than 1000 km. The gravity gradient disturbance are most significant below 1000 km. Aerodynamic perturbations are most effective below 500 km and negligible over 1000 km altitudes.

Gravity Gradient

There are many mathematical models for gravity gradient torque. The most common one can be derived (careful derivation is explained in [1] p. 530-533) assuming homogeneous mass distribution of the Earth, the gravity gradient is:

$$\vec{\tau}_{grav} = \frac{3\mu}{R_o^3} \vec{u}_e \times (I \vec{u}_e) \quad (41)$$

where $\mu=3.986 \cdot 10^{14} \text{m}^3 \cdot \text{s}^{-2}$ is the Earth's gravitational coefficient, R_o is the distance from Earth's center (m), I is the inertia tensor and finally, \vec{u}_e is the unit vector towards nadir, i.e., downward-facing viewing geometry, usually pointing the center of Earth (its opposite is the zenith).

For example, a spacecraft in a low earth orbit (LEO: altitude between 200 km and 1500 km) has $\frac{3\mu}{R_o^3} \cong 4 \cdot 10^{-6} \text{s}^{-2}$ with the moments of inertia of the Space

Shuttle are on the order of $10^6 \text{ kg}\cdot\text{m}^{-2}$, so the gravitational torques on this large vehicle are on the order of $1 \text{ N}\cdot\text{m}$.

In body frame:

$$\vec{\tau}_{grav} = 3\omega_o^2 \vec{c}_3^b \times (I\vec{c}_3^b) ; \omega_o^2 = \frac{\mu}{R_o^3} \quad (42)$$

where \vec{c}_3^b is, again, the third column of the rotation matrix, R_o^b , which transforms z_b into z_o , using quaternions:

$$c_3 = \begin{pmatrix} 2(\varepsilon_1\varepsilon_3 - \eta\varepsilon_2) \\ 2(\varepsilon_2\varepsilon_3 + \eta\varepsilon_1) \\ 1 - 2(\varepsilon_1^2 + \varepsilon_2^2) \end{pmatrix} \quad (43)$$

Thus:

$$\vec{\tau}_{grav}^b = 3\omega_o^2 \begin{pmatrix} 2(I_z - I_y)(\varepsilon_2\varepsilon_3 + \eta\varepsilon_1)(1 - 2(\varepsilon_1^2 + \varepsilon_2^2)) \\ 2(I_x - I_z)(\varepsilon_1\varepsilon_3 - \eta\varepsilon_2)(1 - 2(\varepsilon_1^2 + \varepsilon_2^2)) \\ 2(I_y - I_x)(\varepsilon_1\varepsilon_3 - \eta\varepsilon_2)(\varepsilon_2\varepsilon_3 + \eta\varepsilon_1) \end{pmatrix} \quad (44)$$

Solar Radiation Pressure

Radiation pressure is the pressure exerted upon any surface exposed to electromagnetic radiation.

For example, if the effects of the sun's radiation pressure on the spacecraft of the Viking program had been ignored, the spacecraft would have missed Mars orbit by about 15,000 kilometers being the average distance between Earth and Mars about 225 million km.

The intensity of the solar radiation varies over time, this makes the determination of its energy and frequencies difficult. Most analysis uses the solar radiation constant $SF^{[2]}$:

$$SF = 1353 \frac{\text{W}}{\text{m}^2} \quad (45)$$

the force of solar pressure per unit area is then given by:

$$p_{SR} = \frac{SF}{c} = 4.51 \cdot 10^{-6} \text{ N}\cdot\text{m}^{-2} \quad (46)$$

The expression of the force due to solar radiation on the satellite then becomes:

$$F_{SR} = -p_{SR} \cdot C_R \cdot A_{\odot} \cdot r_{\oplus\odot} \quad (47)$$

where A_{\odot} is the exposed area to the Sun. (\odot is the symbol for the sun, \oplus for Earth) The reflectivity, C_R , indicates how the satellite reflects incoming radiation, and its value is between 0.0 and 2.0. Because C_R is time variant and the constant change in orientation of the object to the sun^[6].

The resulting magnitude of the torque can be expressed as:

$$\tau_{SR} = F_{SR}(c_{pSR} - c_g) \quad (48)$$

where c_{pSR} is the center of solar radiation pressure and c_g the center of gravity. For more information on this topic, see [22, 4].

Aerodynamic Pressure

Satellites orbiting the Earth at low altitude will be influenced by the air density. This disturbance is most effective on satellites orbiting below 400-500 km. This may reduce the velocity of the satellite, and the result will be lower altitude for the satellite. The torque is written [12]:

$$\vec{\tau}_{aero} = \frac{1}{2}\rho V^2 C_d A_{inc} (\vec{u}_v \times ((c_p - c_g)\vec{u}_v)) \quad (49)$$

where ρ is the atmospheric density, C_d is the drag coefficient, A_{inc} is the area perpendicular to u_v , which is the unit vector in velocity, V , direction.

$$F_{aero} = \frac{1}{2}\rho_m V^2 C_d A_{inc} \quad (50)$$

this force is known as “lift force” and can be easily obtained from basic fluid dynamics.

Magnetic Disturbance

This torque is resulted from the interaction of Earth’s magnetic field and spacecraft’s residual magnetic field. If \vec{m} is the sum of all magnetic moments in the satellite, the torque acting on the satellite^[3]:

$$\vec{\tau} = \vec{m} \times \vec{B} \quad (51)$$

where \vec{B} is Earth’s magnetic field vector can be described using IGRF or Dipole Model, see [3, 4]. \vec{m} is caused by satellite-generated current loops, permanent magnets or induced magnets which should be computed.

3.4.2 Actuators Torques

Reaction wheels, momentum wheels, or magnetic torquers are devices used for the changing satellite’s angular momentum. They are simply used on spacecraft for several aims: to add stability against disturbance torques, to absorb cyclic torques, and to transfer momentum to the satellite body for slewing maneuvers.

Magnetic Torquers

Of special interest, as GranaSAT-I will use this type of torquer. Torque produced by the magnetic torquers in body frame is :

$$\vec{\tau}_m^b = \vec{m}^b \times \vec{B}^b \quad (52)$$

\vec{m}^b is the magnetic dipole moment generated by the torquer, \vec{B}^b is the local geomagnetic field vector, relative to the satellite.

It is interesting to mention that these kind of magnetic actuators can only create a torque within a plane (perpendicular to the local magnetic field), which may represent a limitation as for ADCS with these torquers.

Magnetic dipole moment is given by :

$$\vec{m}^b = \begin{pmatrix} N_x i_x A_x \\ N_y i_y A_y \\ N_z i_z A_z \end{pmatrix} = \begin{pmatrix} m_x^b \\ m_y^b \\ m_z^b \end{pmatrix} \quad (53)$$

where, N_k is the number of windings in the torquer, A_k is the span area of the coil, and i_k the torquerer current.

Using skew-symmetric operator :

$$\vec{\tau}_m^b = S(\vec{m}^b) \vec{B}^b = \begin{pmatrix} B_z^b m_y^b - B_y^b m_z^b \\ B_x^b m_z^b - B_z^b m_x^b \\ B_y^b m_x^b - B_x^b m_y^b \end{pmatrix} \quad (54)$$

4 Simulation Results

GranaSAT-I is going to be a nanosatellite (**10x10x10 cm cube**) with approximately **1 kg mass**, supposed to carry a camera as the payload, which should take pictures of the city of Granada.

It will also have a very powerful LED matrix that must be seen from Earth surface and follow a LEO (Low Earth Orbit), that is typically consider from 200 km to 1500 km.

As well as having an (diagonal) inertia matrix:

$$I_{xx} = 0.0018, I_{yy} = 0.0017, I_{zz} = 0.0015 \text{ (m}^2\cdot\text{kg)}$$

which is given by a Solidworks 3D model prototype.

This simulated satellite follows a LEO orbit, typical in cubesats, at $h = 750$ km with very small eccentricity $e = 0.005$ as well as an inclination of $i = 17.2^\circ$.

Magnetorquers physical properties in each axis are,

$$\left\{ \begin{array}{ll} \textit{Coil Resistances} & 50 \, \Omega \\ \textit{Coil Areas} & 88 \text{ cm}^2 \\ \textit{Intesity Limits} & 100 \text{ mA} \end{array} \right.$$

Inputs of the simulations, are essentially the quaternion q , and $\vec{\omega}_{ib}^b$, angular velocity from body to inertia decomposed in body frame.

Various initial values haven been tested, as we would not know them until the satellite in orbit starts measuring, they are chosen randomly, those represented in the following figures are the result of,

$$\left\{ \begin{array}{ll} \textit{Orientation : Quaternion} & q = [-0.001 \ 0.957 \ 0.0928 \ -0.275] \\ \textit{Orientation : Euler Angles} & [\varphi \ \theta \ \psi] = [176.688^\circ \ -31.769^\circ \ -12.020^\circ] \\ \textit{Angular Velocity} & \vec{\omega}_{ib}^b = [4 \ 2 \ 1] \text{ rad}\cdot\text{s}^{-1} \end{array} \right.$$

The different controllers are tested on the Simulink model of this project. Henceforth, a comparative and detailed study as well as a latter discussion is given in this section. All graphs represent time in orbits, in this scenario, $T_{orbit} = 5940$ s.

4.1 Angular Velocity Feedback Controller

The angular velocity feedback controller or Wisniewski controller [6],

$$\vec{m}^b = H\vec{\omega}_{ob}^b \times \vec{B}^b \quad (55)$$

with a gain $H = 500$.

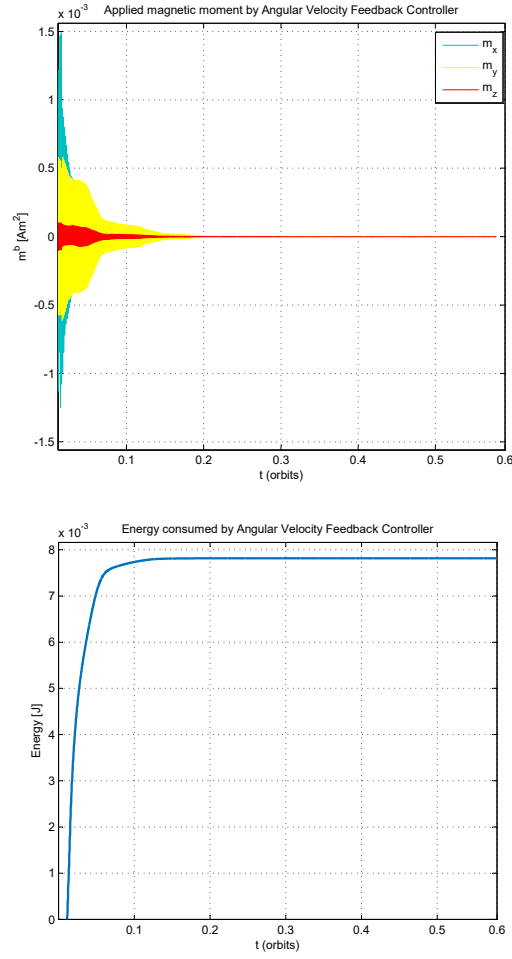


Figure 6: Angular Velocity Feedback Controller. Topmost, applied torque. Down, energy consumed.

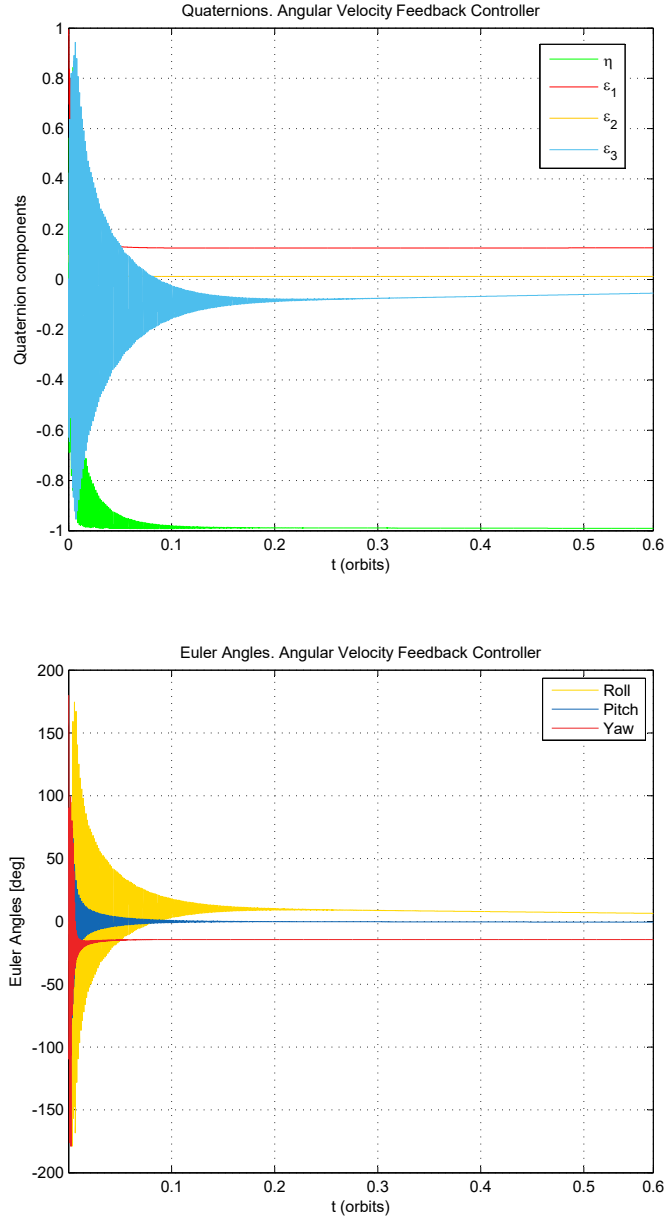


Figure 7: Angular Velocity Feedback Controller. Attitude control. Quaternions and Euler Angles are represented.

4.2 Attitude Feedback Controller

A known controller in bibliography [12] is the attitude feedback controller, similarly built to the previous one,

$$\vec{m}^b = H\vec{\omega}_{ob}^b \times \vec{B}^b - \alpha \cdot \vec{\varepsilon} \times \vec{B}^b \quad (56)$$

where is $\vec{\varepsilon} = [\varepsilon_1 \ \varepsilon_2 \ \varepsilon_3]$ is the vectorial part of the quaternion.

with a gain $H = 500$, $\alpha = 0.9$.

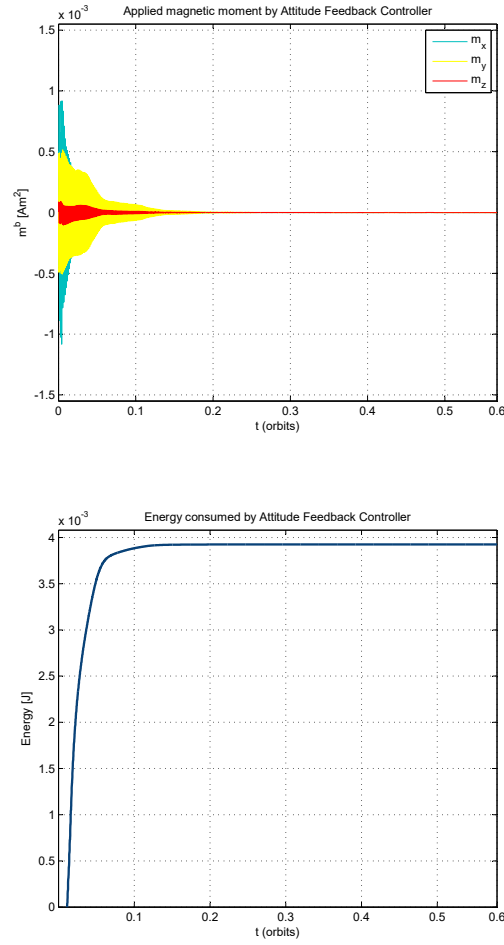


Figure 8: Attitude Feedback Controller. Topmost, applied torque. Down, energy consumed.

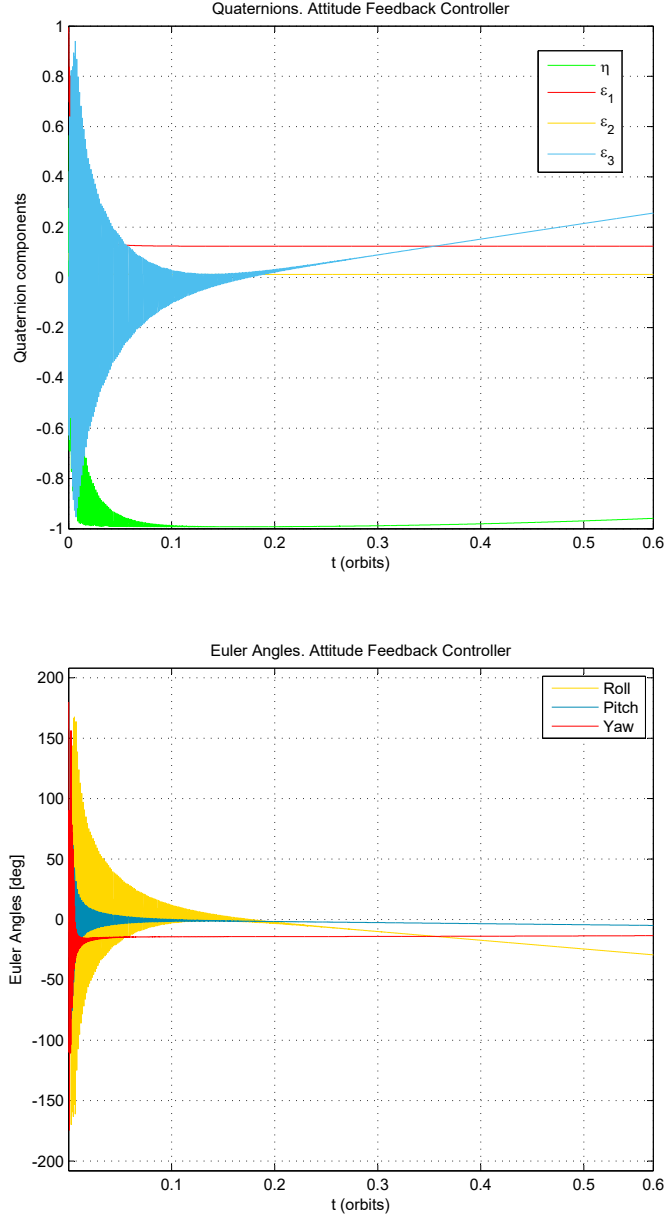


Figure 9: Attitude Feedback Controller. Attitude control. Quaternions and Euler Angles are represented. Important to notice how this controller slowly deviates from equilibrium after reaching it.

4.3 B dot Controller

Probably one of the most popular controllers is this one, due to its simplicity and robustness [18].

$$\vec{m}^b = -k \frac{\dot{\vec{B}}^b}{\|\vec{B}^b\|^2} \quad (57)$$

with a gain $k = 1.25 \cdot 10^{-8}$.

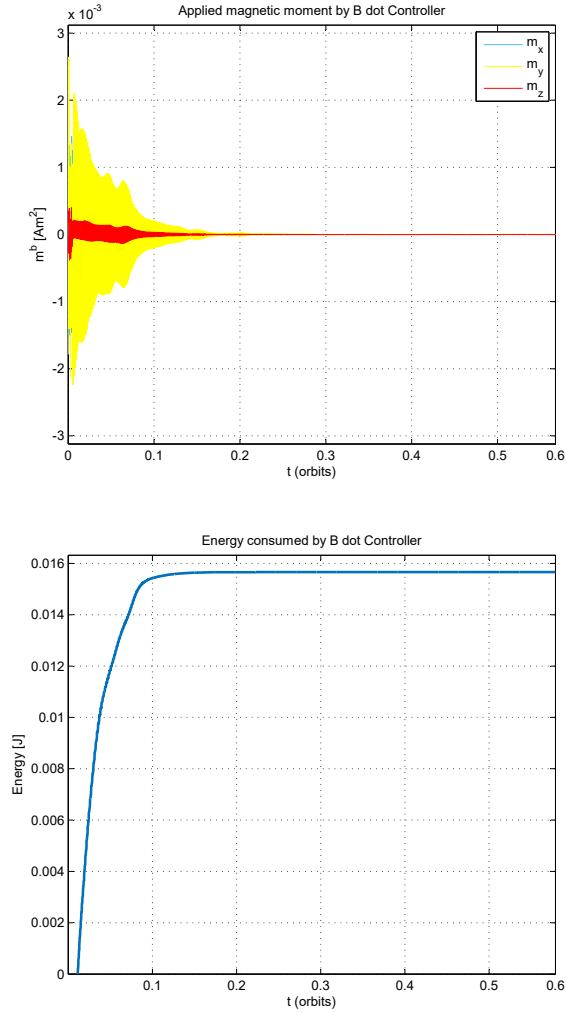


Figure 10: B dot Controller. Topmost, applied torque. Down, energy consumed. This controller has the highest energy consumption.

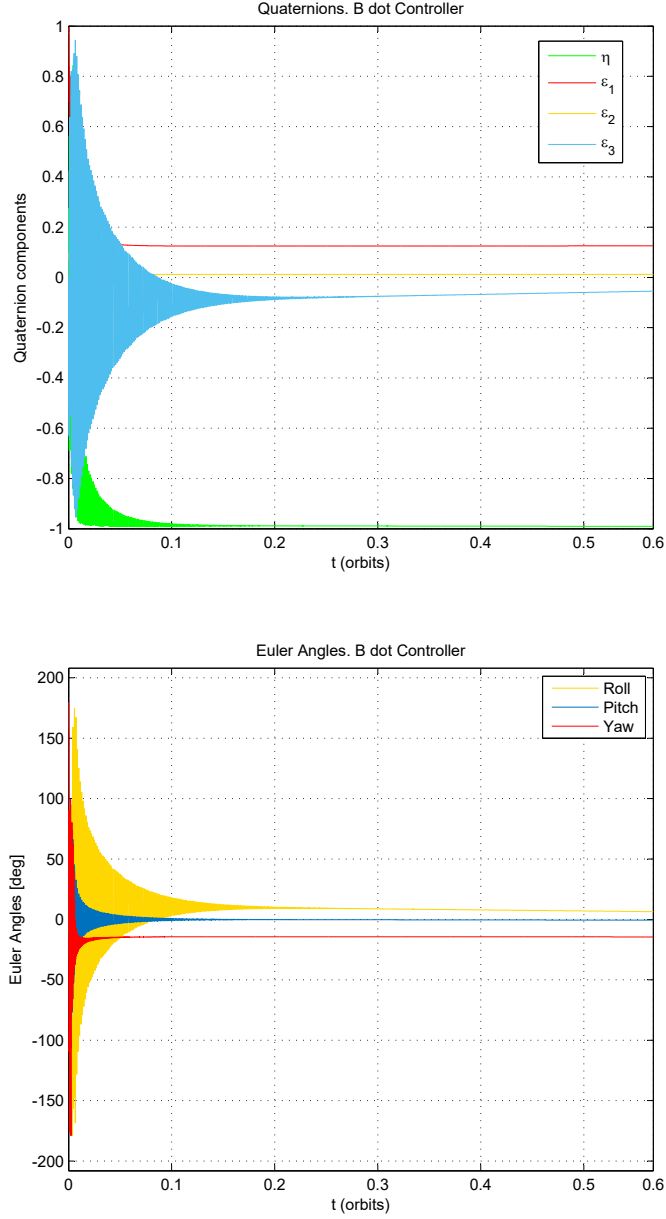


Figure 11: B dot Controller. Attitude control. Quaternions and Euler Angles are represented.

5 Conclusions

A Mathematical Formalism along with a Physical Model of the satellite GranaSAT-I has been implemented in Simulink in order to create a Attitude Control and Determination System. For this purpose, some previous considerations have taken place. For instance, this model only considers two main perturbations (Earth's magnetic field and gravity gradient[with WGS84]) all through the satellite's orbit, propagated with a simple version of SPG4 propagator. In this framework, and having in consideration the physical characteristics of this future spacecraft, simulations gave the previous section results, which at first glance, seem to be consistent and logical in accordance to the references of this work.

From the known physics problem of the two bodies attracted gravitationally, Kepler laws can be obtained and thus build an orbit in which GranaSAT-I would move. Also, a clear explanation of different reference frames is written.

Control Theory deserves an obvious important place in this work, as it provides the mathematical tools to ensure the system is stabilizable and can reach a stable state, being specially important, Lyapunov Stability. This part of the thesis has been of special hardship, as my initial knowledge of it was limited and lots of information, more than here presented, have been processed. In the end, a stable system and convergent results are obtained.

GranaSAT-I is considered as a rigid body thus Euler's equations (for rigid body dynamics) are applied and a first fundamental equation of this work is obtained, as it will allow us to get the angular velocity. The attitude or orientation control needs of quaternions for internal computation, despite being unintuitive, they are useful. Using quaternions we can figure out the orientation of the satellite and if it reaches a stable state (unitary quaternion $q = [\pm 1 0 0 0]$).

Applying all of this, a Simulink model was built. And different controllers were introduced. Results were presented and the following discussion is presented.

5.1 Energy

Energetic efficiency is of capital importance in a spacecraft.

These controllers consume generally, in the order of some mJ, not B dot controller, that presents a higher energy consumption, about one order of magnitude higher. It is a result that makes sense looking at the magnetic moment generated by the coils. Additionally, the angular velocity and attitude controller present similar results, as their mathematical expression are analogous. Nevertheless, the most efficient controller turns out to be the attitude feedback controller, but as it will be explained is not as reliable for stabilization than the others.

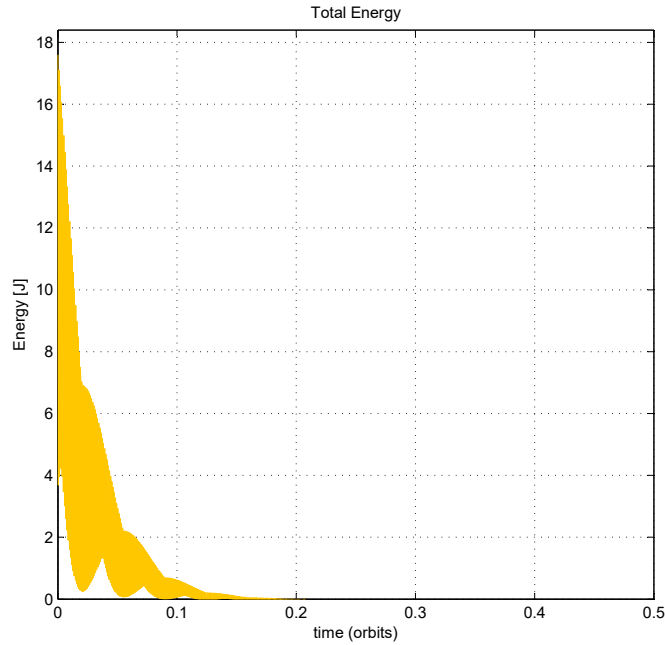


Figure 12: Total Energy (expressed in equation 18) of the satellite in orbit dissipates over time.

5.2 Magnetic Moment

The applied magnetic moment by the controller makes possible to obtain the desired satellite's orientation.

At first, torque is applied with higher intensity, as an initial impulse, is interesting to notice how it is particularly higher for the B dot. As well, as how it is applied in different axis to reach stabilization.

Magnetic moment is applied until the satellite is stable and with much less intensity when is stabilized, this allows the angular velocity to be aligned with Earth's local magnetic field.

Typical values shows it ranges from $[-1.5 \cdot 10^{-3}$ to $1.5 \cdot 10^{-3}] \text{ A} \cdot \text{m}^2$.

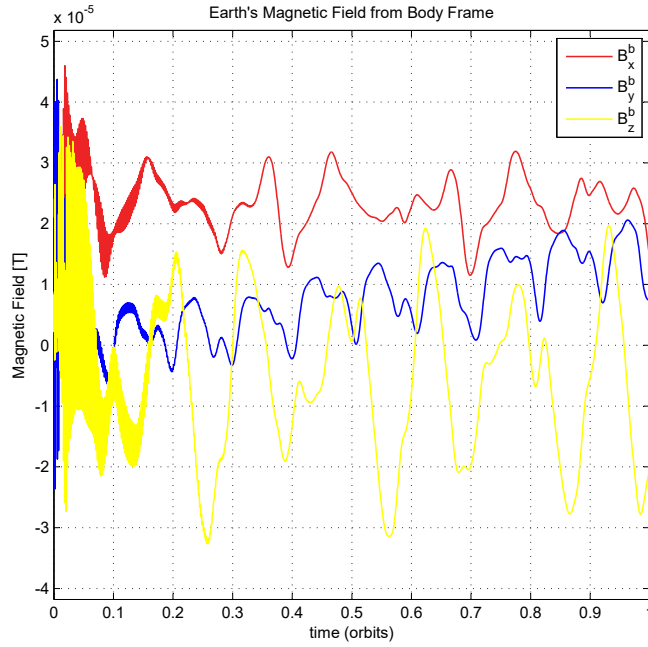


Figure 13: Earth Magnetic Field components as seen from body frame. Before stabilization and the mentioned alignment of the angular velocity, the satellite measures higher variations due to its own rotation.

5.3 Stability

A fundamental requirement for controllers is for them to be able to reduce angular speed and align body and orbital frames, that means $R_o^b = I_{3 \times 3}$. According to equation (32),

$$\vec{\omega}_{ib}^b = \vec{\omega}_{io}^b + \vec{\omega}_{ob}^b = R_o^b \vec{\omega}_{io}^o + \vec{\omega}_{ob}^b$$

Therefore,

$$\vec{\omega}_{ib}^b = \vec{\omega}_{io}^b + \vec{\omega}_{ob}^b = \vec{\omega}_{io}^o + \vec{\omega}_{ob}^b$$

with R_o^b deduced in [6],

$$R_o^b = 2 \begin{pmatrix} \frac{1}{2} & \varepsilon_3 & -\varepsilon_2 \\ -\varepsilon_3 & \frac{1}{2} & \varepsilon_1 \\ \varepsilon_2 & -\varepsilon_1 & \frac{1}{2} \end{pmatrix}$$

For stability to be studied, quaternions, or equivalently, Euler angles are shown. Euler angles, to reach stability, need to tend to $[0 \ 0 \ 0]$, simultaneously, quaternions to $q = [\pm 1 \ 0 \ 0 \ 0]$.

The three controllers give an acceptable result, angular speed is reduce to less than 0.001 rad/s. But total alignment is not fully reached. Not only that, but the attitude feedback controller tends to be unstable and after convergence to $q = [\pm 1 \ 0 \ 0 \ 0]$, starts to deviate slowly with small angular velocity. Therefore, this last controller is not recommended, in spite of its power efficiency.

After an orbit, and a slight difference, the most stable controller is the Wisniewski controller as it gives $q = [-0.9915 \ 0.1258 \ 0.0122 \ -0.03015]$ while B dot presents $q = [-0.9823 \ 0.1810 \ 0.0182 \ -0.0521]$.

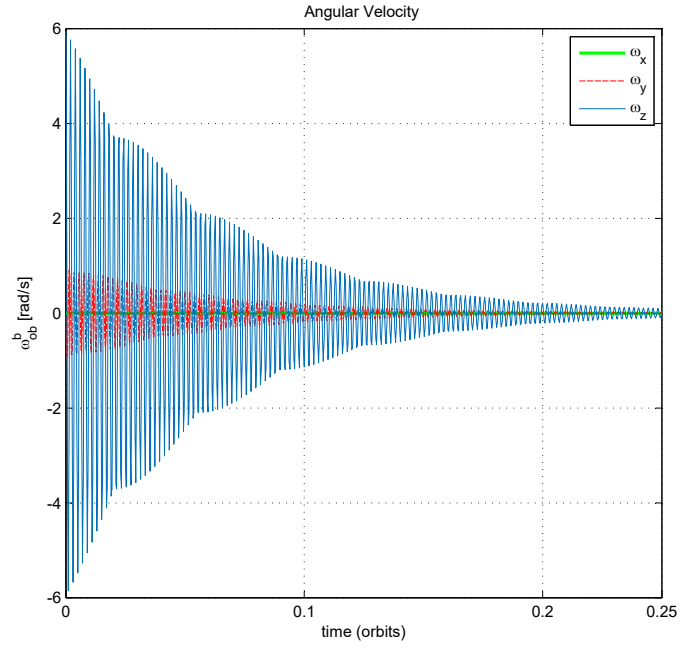


Figure 14: Convergence of components of $\vec{\omega}_{ob}^b$, angular velocity of the body frame with respect to orbit frame (expressed in body frame), to $\vec{\omega}_{ob}^b = [0 \ 0 \ 0]$ rad/s.

6 Future Work and Recommendations

This work is possible due to the maintained effort of the Aerospace GranaSAT Group carried out by my colleagues and, to some extent, to my previous collaborations to GranaSAT-I project, as in designing a 3D model in Solidworks of a prototype first small rocket (it measured around 50 cm high), or such as simple launching tests or study of groundtrack control and finally this ADCS. This is a first approximation of a attitude control system that will be some day controlling a real spacecraft.

Continuing this work means having in consideration all the results given and improvements that can be added.

A more accurate model can be created, for example, adding aerodynamic drag to the Simulink model, an easy implementable task. Or changing World Magnetic Model to IGRF. A more precise propagator, could be a good idea, although through references can be found not to be of paramount importance (providing a decent propagator). One practical idea would be to implement a function to read TLE (two-line element set) automatically so the propagator would not need to be introduced keplerian elements and more data, of course modifications to the Simulink model shall be added in that scenario, for example regarding aerodynamic drag, that strongly depends on altitude.

More complex enhancements, are the complete migration of the simulink code to C (for microcontroller) or HDVL (for FPGA), as the program is itself a feedback loop system and that can represent some complications, none the less, to encourage people that will follow this path, a migration to C of the propagator is given in the appendix. This, however, might be more doable, with less difficulty than implementing a linearized system with a linear quadratic estimator, like Kalman filter. Again, in the appendix, the linearized model is written.

ADCS Budget Estimation

Although a nanosatellite budget can range from \$2.000 to \$20.000, more affordable options are taking place due to cheaper high performance electronics, making possible to design a low cost ADCS.

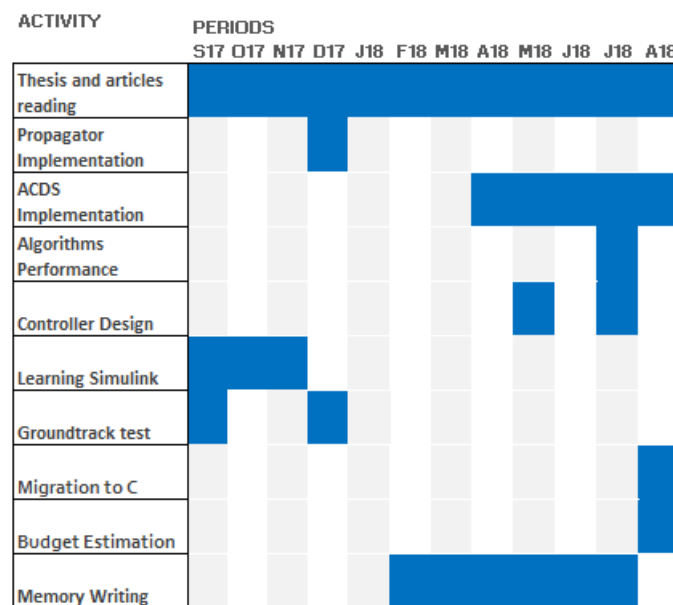
Therefore, based on [27, 25, 16, 17, 18], a *rough* estimation on building this attitude control system is given:

	Magnetorquers	Gyroscope	Coils	Microcontroller	GPS	Total
High estimate	\$9,256.95 ¹	\$71.87 ³	\$4,976.88 ⁵	\$112.20 ⁷	\$9,495 ⁹	\$23,912.90
Low estimate	\$22.44 ²	\$55.67 ⁴	\$12.45 ⁶	\$9.46 ⁸	\$99.00 ¹⁰	\$199.02

¹ ISIS Magnetorquer Board ² Honeywell HMC 1052L ³ ADXRS453BEYZ Analog Devices ⁴ ADXRS453BRGZ Analog Devices ⁵ CubeWheel Small Cube Sat Shop ⁶ MilliBird50 Didel ⁷ TE0887-03M Trenz Electronic GmbH ⁸ PIC24FJ256GA110 ⁹ OEM4-G2L NovAtel ¹⁰ Venus838FLPx-L Navspark

Further information of an overall cost at this type of mission can be found in [27] NASA Cost Symposium AMES.

Gantt Chart of this Project



References

- [1] James R. Wertz. Spacecraft Attitude Determination and Control. Kluwer Academic, 2002.
- [2] Fundamentals of Spacecraft Attitude Determination and Control. F. Landis Markley, John L. Crassidis. Springer, 2014.
- [3] Orbital Mechanics for Engineering Students. Howard D. Curtis. Elsevier, 2005.
- [4] Satellites Orbits and Missions. Michel Capderou. Springer, 2005.
- [5] Introduction to Simulink® with Engineering Applications. Orchard Publications. 2006.
- [6] Satellite Attitude Control Using Only Electromagnetic Actuation Ph.D. Thesis Rafał Wisniewski 1996.
- [7] Rotaciones, secuencia aeroespacial y cuaterniones. Una revisión de las relaciones fundamentales. Eduardo Serrano, Ricardo Oscar Sirne y Guillermo La Mura.
- [8] AstroNet-II: The Astrodynamics Network. Attitude dynamics and control. University of Roma Tor Vergata. James Biggs, 2013.
- [9] Mathematical Model of the Solar Radiation Force and Torques Acting on the Components of a Spacecraft. R. M. Georgevic. NASA, 1971.
- [10] Attitude Determination and Control Systems Scott R. Starin, NASA Goddard Space Flight Center John Eterno, Southwest Research Institute, 2011.
- [11] Attitude control for the Norwegian student satellite nCube. Master Thesis. Eli Jerpseth Øverby, 2004.
- [12] LEO Satellites: Attitude Determination and Control Components: Some Linear Attitude Control Techniques Graduate School of Natural and Applied Sciences Of Middle East Technical University. Ceren Kaplan, 2006.
- [13] Controls Algorithm For A Satellite Using Earth's Magnetic Field: Orbit Maneuvers And Attitude Positioning Karthik Ganesh University of Central Florida, 2007.
- [14] NASA Cost Symposium AMES Cost Model Ames Micro/Nanosatellites Cost Model, 2015.
- [15] Satellite Attitude Determination with Low-Cost Sensors by John C. Springmann, 2013.
- [16] Hardware Review of an On Board Controller for a Cubesat Norwegian University of Science and Technology Trondheim Magne Alver Normann, 2015.

- [17] Review of the hardware description of the camera module prototype for NTNU Test Satellite (NUTS) By Jon Kalevi Oltedal Fall Semester, 2015.
- [18] Attitude Determination and Control System design of Kufasat. Mohammed Chessab Mahdi and Abdal-Razak Shehab, 2014.
- [19] Useful to visualize Euler to Quaternions: <http://quaternions.online>
- [20] <http://www.kostasalexis.com/frame-rotations-and-representations.html>
- [21] Comunicaciones por Satélite Curso 2009/10 Comunicaciones por Satélite. ©Ramón Martínez, Miguel Calvo UPM. Parámetros orbitales Formato TLE de NORAD.
- [22] Spacecraft Attitude Dynamics and Control Christopher D. Hall January 12, 2003
- [23] Astronautica y Vehículos Espaciales. Rafael Vazquez Valenzuela Departamento de Ingeniería Aeroespacial Escuela Superior de Ingenieros, Universidad de Sevilla. 2012.
- [24] Satellite Dynamics and Control in a Quaternion Formulation - Lecture note for course 31365 Spacecraft Dynamics and Control at DTU. 2010.
- [25] Pose Estimation using Linearized Rotations and Quaternion Algebra. Timothy Barfoot, James R. Forbes, Paul T. Furgalea. University of Toronto Institute for Aerospace Studies.
- [26] Basic Navigational Mathematics, Reference Frames and the Earth's Geometry. Springer-Verlag Berlin Heidelberg, 2013.
- [27] Modeling and Simulation for Automatic Control Olav Egeland and Jan Tommy Gravdahl. Norwegian University of Science and Technology Trondheim, Norway. 2002.

APPENDIX

Linearization of the Satellite Model (for future implementation)

What Is Linearization? (Brief Explanation)

Linearization is a linear approximation of a nonlinear system that is valid in a small region around the operating point, using a Taylor series.

For example, suppose that the nonlinear function is $y = x^2$. Linearizing this nonlinear function about the operating point $x = 1, y = 1$ results in a linear function $y = 2x - 1$. Near the operating point, $y = 2x - 1$ is a good approximation to $y = x^2$. Away from the operating point, the approximation is poor. The actual region of validity depends on the nonlinear model.

Extending the concept of linearization to dynamic systems, we can write continuous-time nonlinear differential equations in this form :

$$\begin{aligned}\dot{x}(t) &= f(x(t), u(t), t) \\ y(t) &= g(x(t), u(t), t)\end{aligned}\tag{58}$$

In these equations, $x(t)$ represents the system states, $u(t)$ represents the inputs to the system, and $y(t)$ represents the outputs of the system. A linearized model of this system is valid in a small region around the operating point we linearize. To represent the linearized model, define new variables centered about the operating point :

$$\begin{aligned}\delta x(t) &= x(t) - x_o \\ \delta u(t) &= u(t) - u_o \\ \delta y(t) &= y(t) - y_o\end{aligned}\tag{59}$$

The linearized model in terms of $\delta x(t)$, $\delta u(t)$, and $\delta y(t)$ is valid when the values of these variables are small :

$$\begin{aligned}\delta \dot{x}(t) &= A\delta x(t) + B\delta u(t) \\ \delta y(t) &= C\delta x(t) + D\delta u(t)\end{aligned}\tag{60}$$

Kinematics

The kinematics of the satellite is already described :

$$\dot{q} = \begin{pmatrix} \dot{\eta} \\ \dot{\varepsilon} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} -\varepsilon^T \\ \eta I_{3 \times 3} + S(\varepsilon) \end{pmatrix} \omega_{ob}^b \quad (61)$$

We linearize the system around the points $\eta = 1$ and $\varepsilon = 0$ which results in the system^[3]:

$$\dot{q} = \begin{pmatrix} \dot{\eta} \\ \dot{\varepsilon} \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{1}{2}\omega_{ob}^b \end{pmatrix} ; \omega_{bo}^b = 2\dot{\varepsilon} \quad (62)$$

Rotation Matrix

The linearization of the rotation matrix between body and orbit frame around the points $\eta = 1$ and $\varepsilon = 0$ results in^[3]:

$$R_b^o = I_{3 \times 3} + 2S(\varepsilon) \quad (63)$$

Angular Velocity

First,

$$R_o^b = (R_b^o)^T \quad (64)$$

$$R_o^b = I_{3 \times 3} - 2S(\varepsilon) \quad (65)$$

$$R_o^b = 2 \begin{pmatrix} \frac{1}{2} & \varepsilon_3 & -\varepsilon_2 \\ -\varepsilon_3 & \frac{1}{2} & \varepsilon_1 \\ \varepsilon_2 & -\varepsilon_1 & \frac{1}{2} \end{pmatrix} \quad (66)$$

The angular velocity in body frame relative to inertial frame :

$$\omega_{ib}^b = \omega_{io}^b + \omega_{ob}^b = R_o^b \omega_{io}^o + \omega_{ob}^b \quad (67)$$

Using $\omega_{ob}^b = 2\dot{\varepsilon}$:

$$\omega_{ib}^b = \begin{pmatrix} 2(\dot{\varepsilon}_1 - \omega_o \varepsilon_3) \\ 2(\dot{\varepsilon}_2 + \frac{1}{2}\omega_o) \\ 2(\dot{\varepsilon}_3 + \omega_o \dot{\varepsilon}_1) \end{pmatrix} \quad (68)$$

the time derivative is :

$$\dot{\omega}_{ib}^b = \begin{pmatrix} 2(\ddot{\varepsilon}_1 - \omega_o \dot{\varepsilon}_3) \\ 2\ddot{\varepsilon}_2 \\ 2(\ddot{\varepsilon}_3 + \omega_o \ddot{\varepsilon}_1) \end{pmatrix} \quad (69)$$

Gravitational torque

In previous section, was derived :

$$\tau_{grav}^b = 3\omega_o^2 \begin{pmatrix} 2(I_z - I_y)(\varepsilon_2\varepsilon_3 + \eta\varepsilon_1)(1 - 2(\varepsilon_1^2 + \varepsilon_2^2)) \\ 2(I_x - I_z)(\varepsilon_1\varepsilon_3 - \eta\varepsilon_2)(1 - 2(\varepsilon_1^2 + \varepsilon_2^2)) \\ 2(I_y - I_x)(\varepsilon_1\varepsilon_3 - \eta\varepsilon_2)(\varepsilon_2\varepsilon_3 + \eta\varepsilon_1) \end{pmatrix} \quad (70)$$

Linearized around $\eta = 1$ and $\varepsilon = 0$,

$$\tau_{grav}^b = 3\omega_o^2 \begin{pmatrix} 2(I_z - I_y)\varepsilon_1 \\ 2(I_x - I_z)\varepsilon_2 \\ 0 \end{pmatrix} \quad (71)$$

Magnetic Torquer Linearization

The torque from magnetic torquer is given as :

$$\tau_m^b = S(m^b)B^b = S(m^b)R_b^o B^o = S(m^b)[I_{3 \times 3} - 2\eta S(\varepsilon) + 2S^2(\varepsilon)]B^o \quad (72)$$

Linearized around $\eta = 1$ and $\varepsilon = 0$,

$$\tau_m^b = S(m^b)B^b = \begin{pmatrix} B_z^o m_y^b - B_y^o m_z^b \\ B_x^o m_z^b - B_z^o m_x^b \\ B_y^o m_x^b - B_x^o m_y^b \end{pmatrix} \quad (73)$$

Linearization of the Satellite Mathematical Model with Magnetic Torquer as Actuator

This case has interest for GranaSAT-I as the spacecraft will use magnetic torquers, coupled with Earth's magnetic field, as actuators. Mathematical Linear model of the satellite can be obtained as :

$$I\dot{\omega}_{ib}^b = -\omega_{ib}^b \times (I\omega_{ib}^b) + S(m^b)B^b + \tau_{grav}^b \quad (74)$$

where all terms have been explained before. Insterting them in this model yields:

$$\begin{aligned} 2I_x(\ddot{\varepsilon}_1 - \omega_o\dot{\varepsilon}_3) &= (I_y - I_z)(2\omega_o\dot{\varepsilon}_3 + 8\omega_o^2\varepsilon_1) + (B_z^o m_y^b - B_y^o m_z^b) \\ I_y(\ddot{\varepsilon}_2) &= -6(I_x - I_z)\omega_o^2\varepsilon_2 + (B_x^o m_z^b - B_z^o m_x^b) \\ 2I_z(\ddot{\varepsilon}_3 - \omega_o\dot{\varepsilon}_1) &= (I_y - I_x)(2\omega_o\dot{\varepsilon}_3 + 2\omega_o^2\varepsilon_1) + (B_y^o m_x^b - B_x^o m_y^b) \end{aligned} \quad (75)$$

Then using :

$$\begin{aligned}
k_x &= \frac{I_y - I_z}{I_x} \\
k_y &= \frac{I_x - I_z}{I_y} \\
k_z &= \frac{I_y - I_x}{I_z}
\end{aligned} \tag{76}$$

Gives :

$$\begin{aligned}
\ddot{\varepsilon}_1 &= (1 - k_x)\omega_o\dot{\varepsilon}_3 - 4k_x\omega_o^2\varepsilon_1 + \frac{1}{2I_x}(B_z^om_y^b - B_y^om_z^b) \\
\ddot{\varepsilon}_2 &= -3k_y\omega_o^2\varepsilon_2 + \frac{1}{2I_y}(B_x^om_z^b - B_z^om_x^b) \\
\ddot{\varepsilon}_3 &= -(1 - k_z)\omega_o\dot{\varepsilon}_1 - k_z\omega_o^2\varepsilon_3 + \frac{1}{2I_z}(B_y^om_x^b - B_x^om_y^b)
\end{aligned} \tag{77}$$

The system can be represented by state-space representation in linear form given by :

$$\dot{x} = Ax(t) + B(t)u(t) \tag{78}$$

if we define the states vectors to be $x = [\varepsilon_1 \ \dot{\varepsilon}_1 \ \varepsilon_2 \ \dot{\varepsilon}_2 \ \varepsilon_3 \ \dot{\varepsilon}_3]^T$ and inputs $u = [m_x \ m_y \ m_z]^T$, then, A matrix can be written as,

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -4k_x\omega_o^2 & 0 & 0 & 0 & 0 & (1 - k_x)\omega_o \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -3k_y\omega_o^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -(1 - k_z)\omega_o & 0 & 0 & -k_z\omega_o^2 & 0 \end{pmatrix} \tag{79}$$

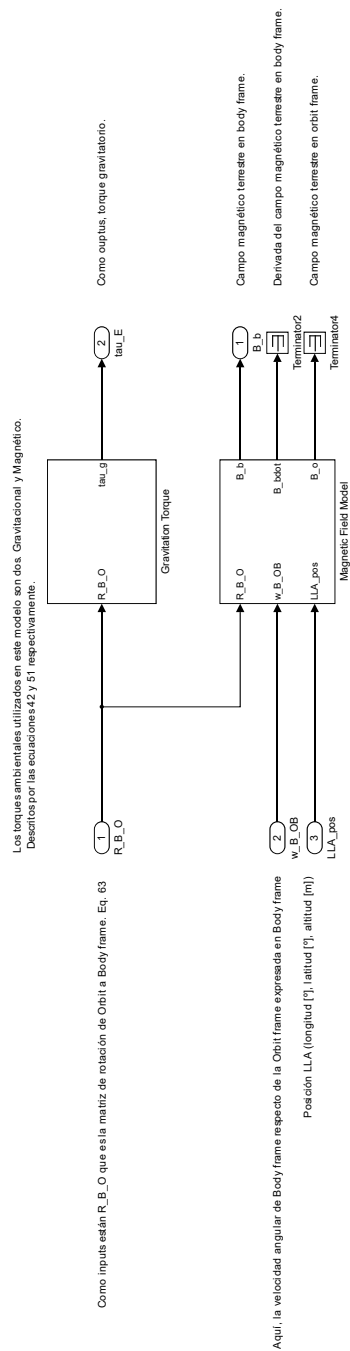
and

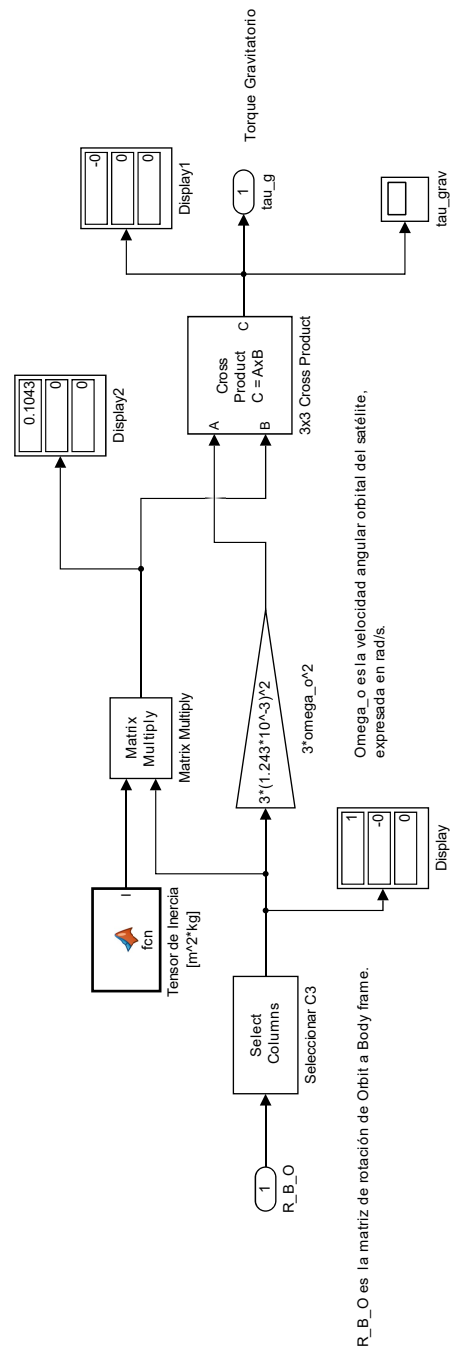
$$B(t) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{2I_x}B_z^o & -\frac{1}{2I_x}B_y^o \\ 0 & 0 & 0 \\ -\frac{1}{2I_y}B_z^o & 0 & \frac{1}{2I_y}B_x^o \\ 0 & 0 & 0 \\ \frac{1}{2I_z}B_y^o & -\frac{1}{2I_z}B_x^o & 0 \end{pmatrix} \tag{80}$$

Simulink Model (first iteration is represented)



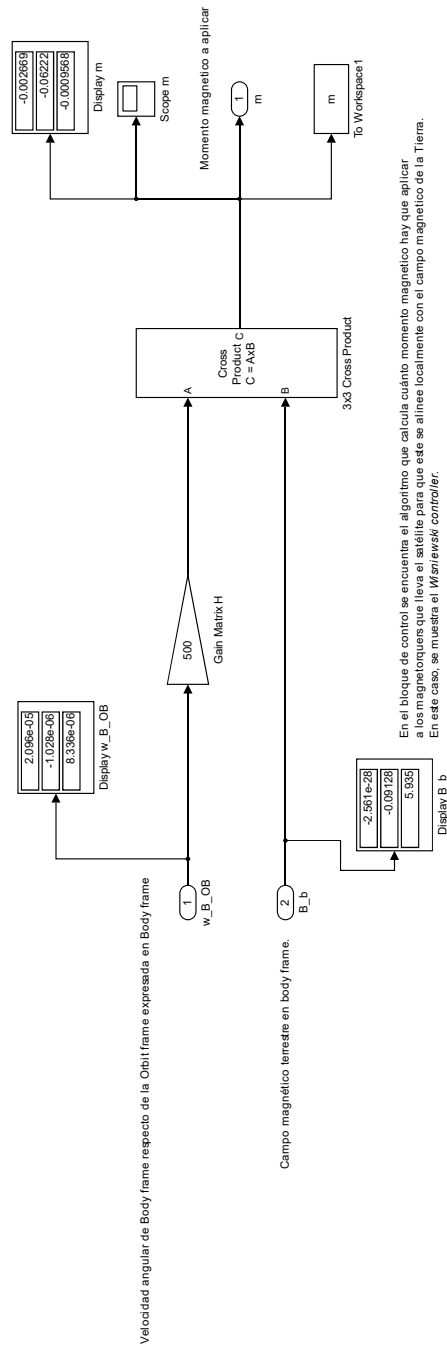


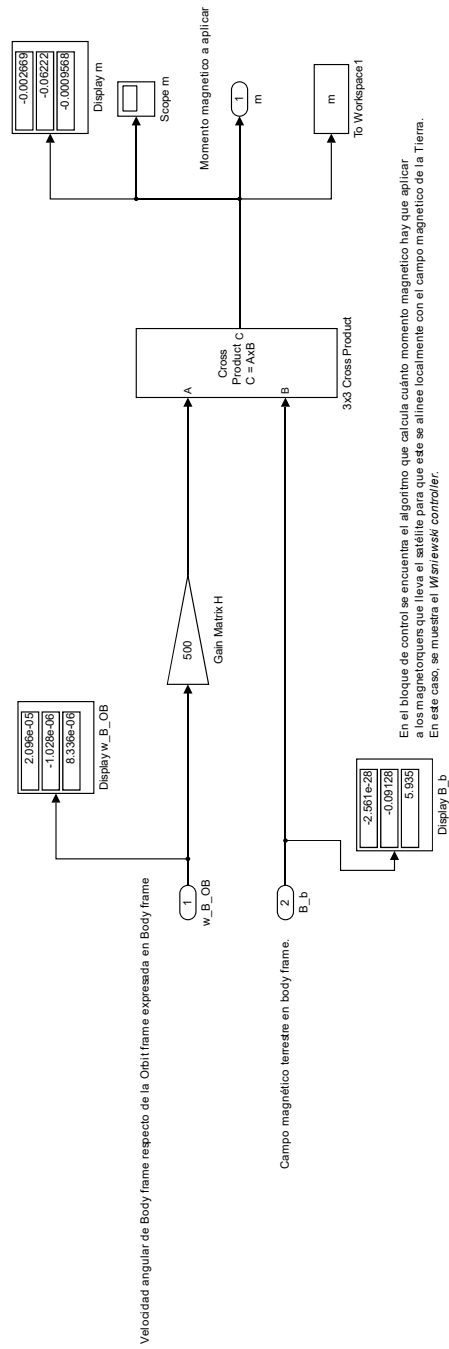


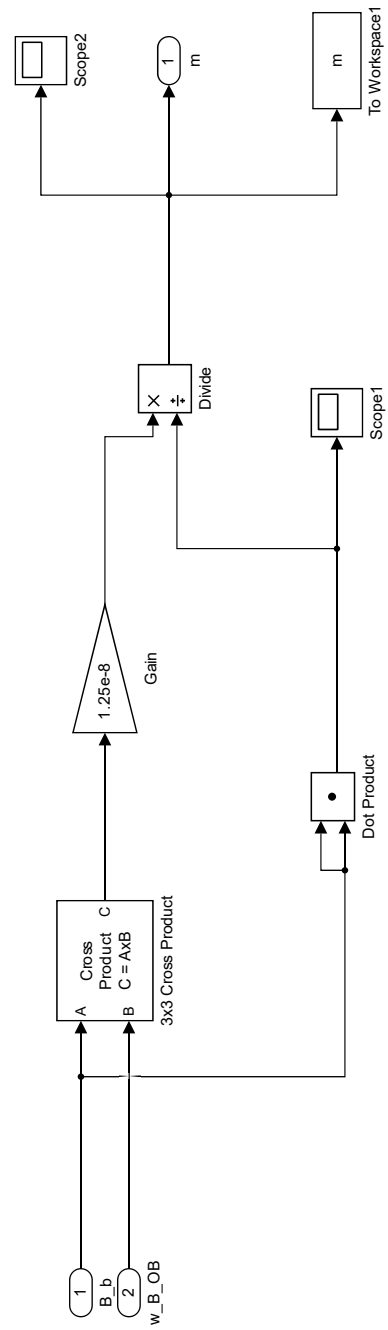


REFERENCES

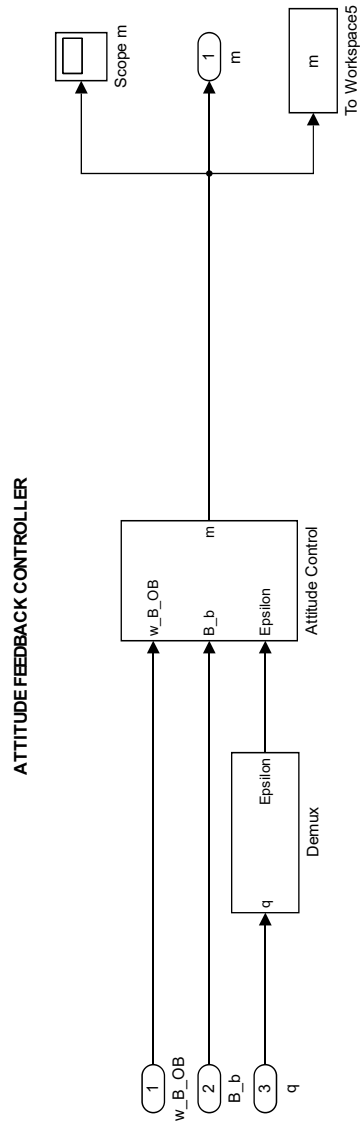


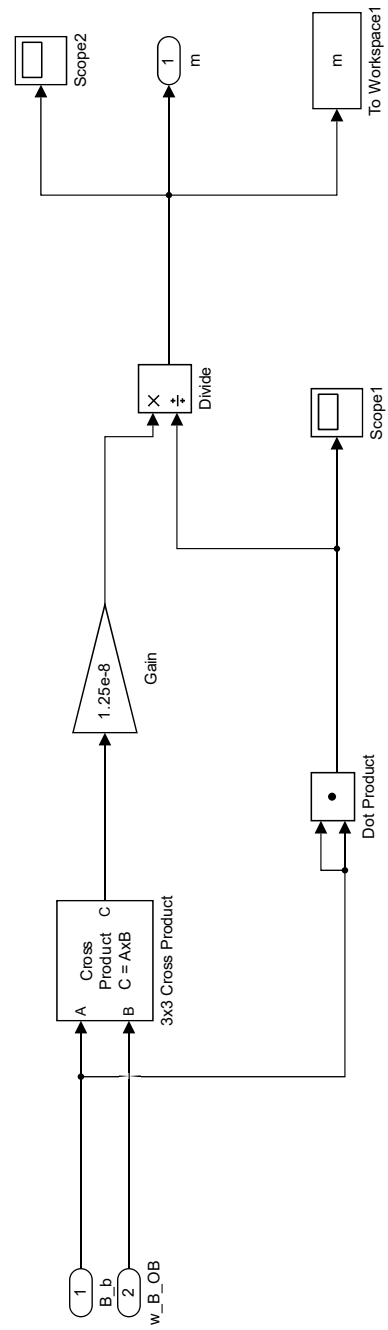




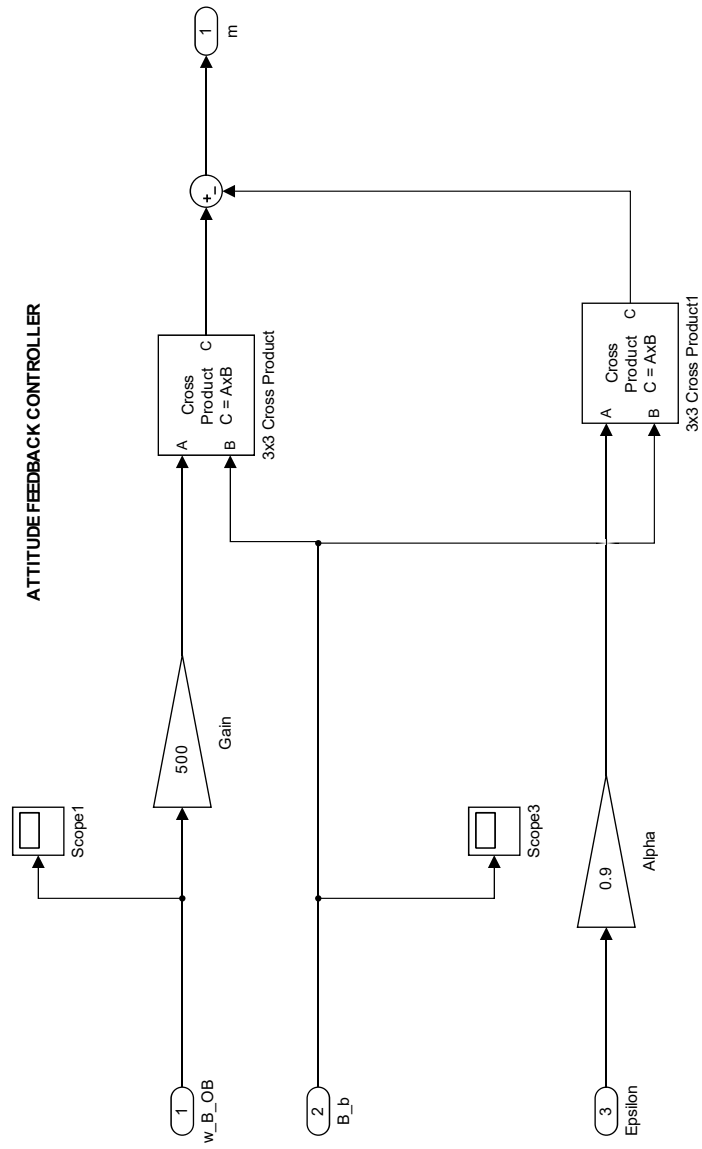


B DOT CONTROLLER

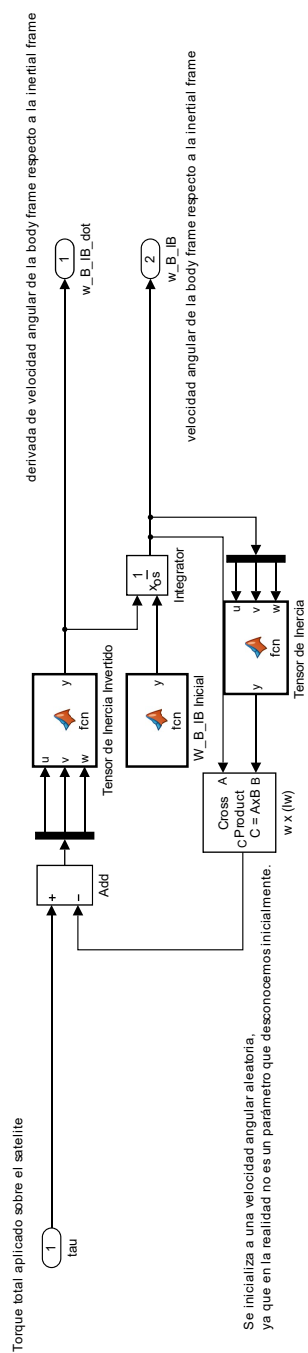


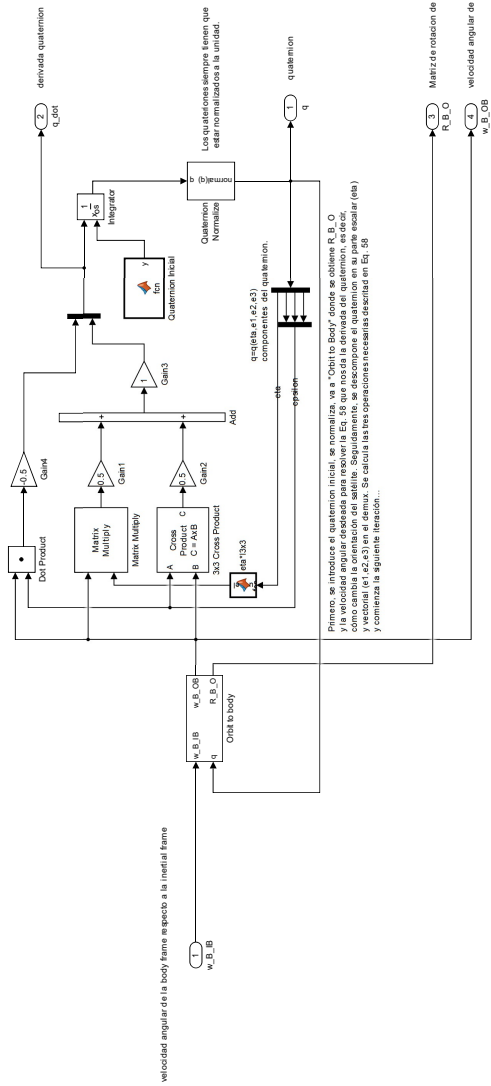


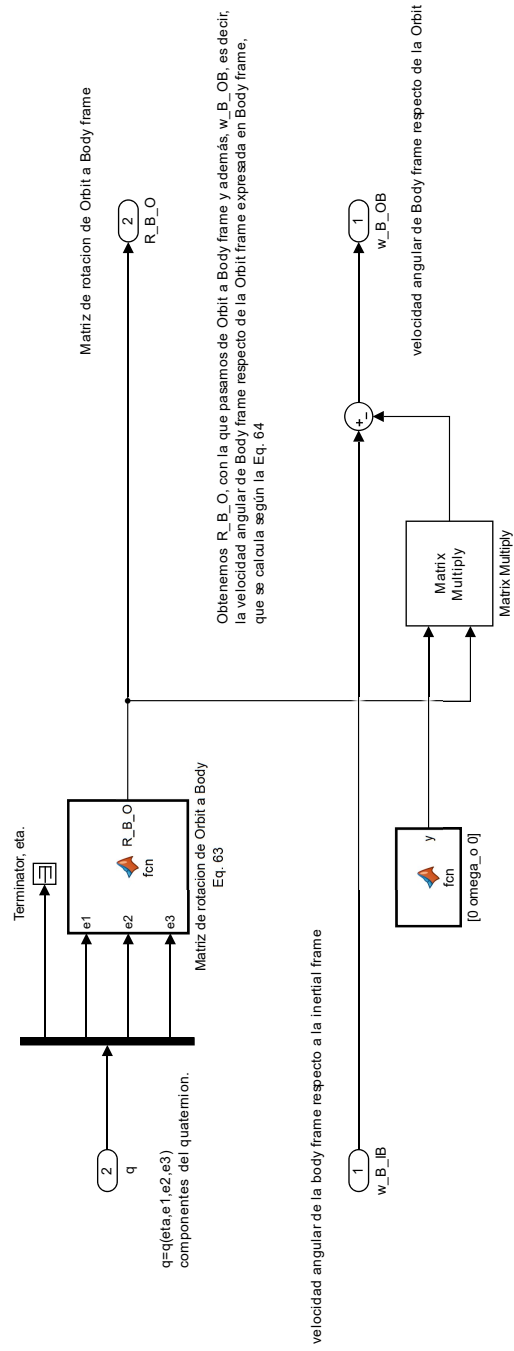
B DOT CONTROLLER

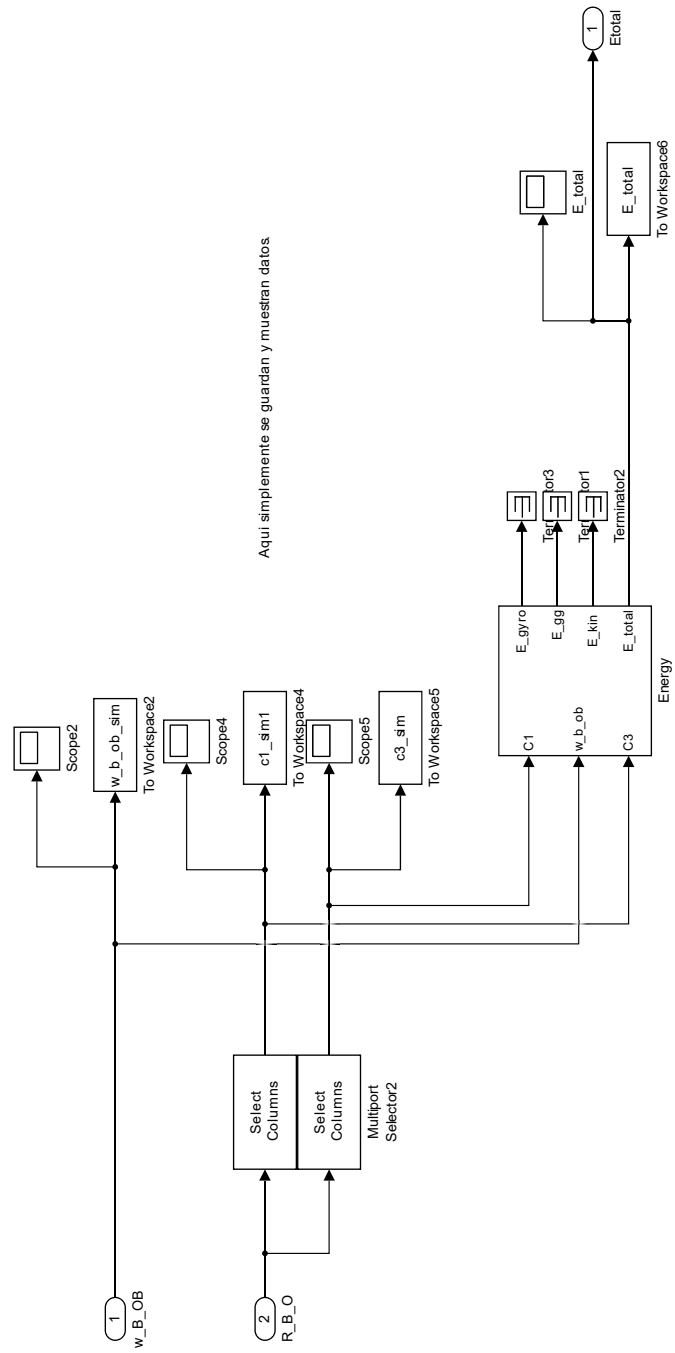


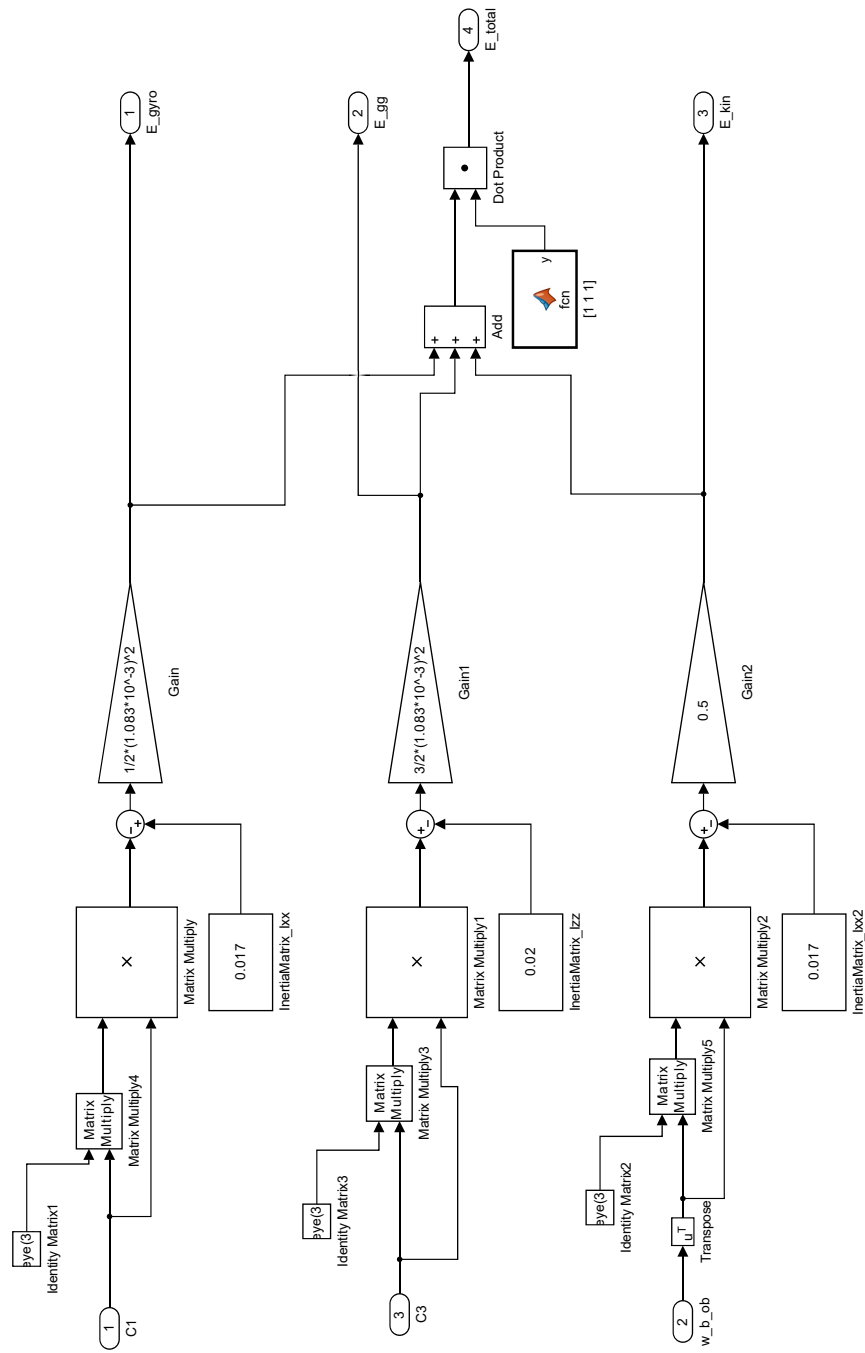
Este bloque resuelve la ecuación diferencial dada en Eq. 30, que modela la dinámica del satélite











Migration to C (for future implementation)

Only the propagator turns out to yield 2044 lines of code, without .h or additional functions.

First lines are shown in the figure below.

Files can be found at:

https://consigna.ugr.es/f/gMetADiIBYQicmfT/codigo_pdf.pdf

https://consigna.ugr.es/f/AxvXnqvlYOK5875L/sat_propagator_grt_rtw.zip

```

1  /*
2  * sat_propagator.c
3  *
4  * Code generation for model "sat_propagator".
5  *
6  * Model version : 1.30
7  * Simulink Coder version : 8.6 (R2014a) 27-Dec-2013
8  * C source code generated on : Tue Aug 21 21:50:38 2018
9  *
10 * Target selection: GRT-LEG
11 * Note: GRT includes extra infrastructure and instrumentation for prototyping
12 * Embedded hardware selection: 32-bit Generic
13 * Code generation objectives: Unspecified
14 * Validation result: Not run
15 */
16 #include "sat_propagator.h"
17 #include "sat_propagator_private.h"
18
19 /* Block signals (auto storage) */
20 B_sat_propagator_T sat_propagator_B;
21
22 /* Block states (auto storage) */
23 DW_sat_propagator_T sat_propagator_DW;
24
25 /* Real-time model */
26 RT_MODEL_sat_propagator_T sat_propagator_M;
27 RT_MODEL_sat_propagator_T *const sat_propagator_M = &sat_propagator_M;
28
29 /*
30 * Output and update for action system:
31 *   '<S12>/If Action Subsystem'
32 *   '<S20>/If Action Subsystem'
33 */
34 void sat_propagator>IfActionSubsystem(real_T rtu_yin, real_T rtu_min, real_T
35   'rtu_yout, real_T *rtu_mout, IfActionSubsystem_sat_propagator_T *localP)
36 {
37   /* Bias: '<S14>/Bias1' */
38   'rtu_yout = rtu_yin + localP->Bias1_Bias;
39
40   /* Bias: '<S14>/Bias' */
41   'rtu_mout = rtu_min + localP->Bias_Bias;
42 }
43
44 /*
45 * Output and update for action system:
46 *   '<S12>/If Action Subsystem1'
47 *   '<S20>/If Action Subsystem1'
48 */
49 void sat_propagator>IfActionSubsystem1(real_T rtu_yin, real_T rtu_min, real_T
50   'rtu_yout, real_T *rtu_mout)
51 {
52   /* Input: '<S15>/y in' */
53   'rtu_yout = rtu_yin;
54
55   /* Input: '<S15>/m in' */
56   'rtu_mout = rtu_min;
57 }
58
59 real_T rt_roundd_snf(real_T u)
60 {
61   real_T y;
62   if (fabs(u) < 4.503599627370496E+15) {
63     if (u >= 0.5) {
64       y = floor(u + 0.5);
65     } else if (u > -0.5) {
66       y = u * 0.0;
67     } else {
68       y = ceil(u - 0.5);
69     }
70   } else {
71     y = u;
72   }
73   return y;
74 }
75
76
77 real_T rt_moddd_snf(real_T u0, real_T ul)
78 {
79   real_T y;
80   real_T tmp;
81   if (ul == 0.0) {
82     y = u0;
83   } else if ((!(rtIsNaN(u0)) && !(rtIsInf(u0)) && !(rtIsNaN(ul)) && !(rtIsInf
84     (ul)))) {

```