POLITECNICO
MILANO 1863

Spacecraft Attitude Dynamics and Control

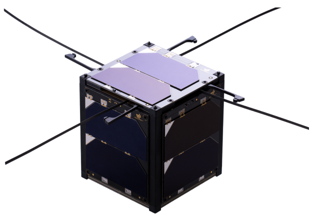
Alessandro Turano

Report
Space Engineering
Academic Year 2017-2018

Abstract

The aim of this report is to design and test an attitude and control system of a 1U dual-spin CubeSat.
The spacecraft has to fly in a Sun Synchronous orbit at 650 Km from the Earth ground.
The mission consist of a De-tumble and then Earth Pointing.
The actuators provided are 3 Reaction Wheels and 4 thrusters, while the sensors used are magnetometer and a gyro.

Figure 1: Picture of the CubeSat



Contents

1 Parameters **1**

2 Description of the spacecraft **2**
2.1 Mass, power and volume budget 2
2.2 Modelling parameters 3
2.3 ADCS architecture 3

3 Simulink model **4**
3.1 Kinematic and Dynamic 4
3.2 Disturbance models 5
3.2.1 Gravity gradient torque 5
3.2.2 Magnetic torque 6
3.2.3 Torque due to atmospheric drag 7
3.2.4 Solar radiation torque 9
3.3 Control algorithm 11
3.4 Sensors models 13
3.4.1 Magnetometer model 13
3.4.2 Gyro model 13
3.5 Determination algorithm 14
3.6 Actuators models 15
3.6.1 Thrusters 15
3.6.2 Reaction Wheels 16

4 Results **17**

5 Conclusion **20**

List of Figures

1 Picture of the CubeSat I

2.1 ADCS architecture 3

3.1 Gravity gradient torque 5
3.2 Magnetic torque 7
3.3 Torque due to atmospheric drag 8
3.4 Solar radiation torque 10
3.5 Response of the control algorithm 12
3.6 Response of the Thrusters 12

4.1 Angular Velocities 18
4.2 DCM Matrix 18
4.3 Pointing Error [deg] 19
4.4 Switching from Detumbling to pointing 19

5.1 Rendering of the simulation 20

List of Tables

1.1 Keplerian parameters of the Sun-synchronous orbit 1

2.1 Mass, power and volume budget 2
2.2 Moments of inertia 3

4.1 Initial values 17
4.2 Gain factors 17

Chapter 1

Parameters

The requirements given for the mission are:

- Use only magnetometer and gyro as sensors
- Use only 4 thrusters and 3 Reaction Wheels as actuators
- 1U dual-spin CubeSat spacecraft
- 650 Sun-synchronous orbit

The first step is to compute the Keplerian parameters of the Sun-synchronous orbit at the given altitude.

The mission consist of two different phases, the first one is de-tumbling and the second one Earth pointing.
To simulate correctly the results, models for Kinematics and Dynamics, disturbances, sensors and actuators are considered.
Descriptions and assumptions of these models can be found in the following chapters, while results and conclusion are in the last chapters.
Before analyzing these aspects, descriptions of the spacecraft and of the ADCS architecture are presented.

Table 1.1: Keplerian parameters of the Sun-synchronous orbit

Parameter	**Value**	**Unit of measure**
a	7021	$[km]$
e	0	\
i	96.7826	$[^\circ]$

Chapter 2

Description of the spacecraft

2.1 Mass, power and volume budget

The spacecraft considered is, as said before, a 1U dual-spin CubeSat.
The dimensions of this kind of spacecraft are fixed, the geometry is a rectangular parallelepiped of 11.35×10×10 cm (while the useful volume is a cube of 1 liter).
The mass budget is about 1.3 kg.

Table 2.1: Mass, power and volume budget

Component	**Number of units**	**Mass**	**Power**	**Volume**
Micro Propulsion System	1	0.3	0.1	0.0001
CubeWheel Medium	3	0.150	0.190	0.0000667
Cube Computer	1	0.060	0.15	0.0000864
Magnetometer	1	0.175	0.60	0.0000702
SCR1100-D02	3	0.001	0.15	0.000000771
Total	\	0.988	1.87	0.0004590
Total + 10%	\	1.087	2.057	0.0005049
Unit of measure	\	$[kg]$	$[W]$	$[m^3]$

2.2 Modelling parameters

The moments of inertia are estimated starting from the geometry.

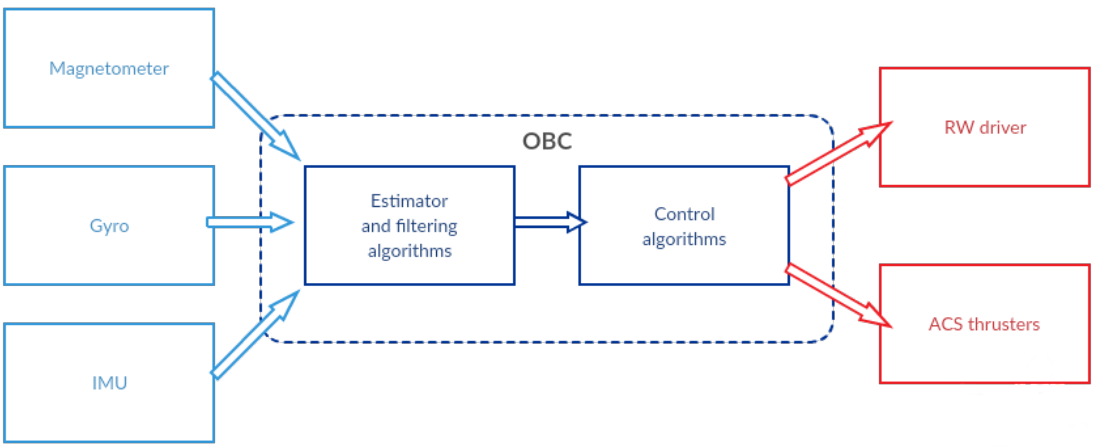
Table 2.2: Moments of inertia

	Nomenclature	**Value**	**Unit of measure**
X axis	I_1	0.0025	$[kg * m^2]$
Y axis	I_2	0.0022	$[kg * m^2]$
Z axis	I_3	0.0020	$[kg * m^2]$
Spinning wheel	I_r	0.0015	$[kg * m^2]$

The inertia used for the spinning wheel is computed starting from an estimation of the mass of the wheel and its angular speed.
The Spinning wheel is kept at constant at constant spinning of 10.5 rad/s.

2.3 ADCS architecture

Figure 2.1: ADCS architecture



Chapter 3

Simulink model

3.1 Kinematic and Dynamic

The kinematic representation is done using the Direction Cosines Matrix (DCM), in which each row of the matrix represents one axis:

$$A = \begin{bmatrix} u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \\ w_1 & w_2 & w_3 \end{bmatrix}$$

Thanks to this matrix it is possible to switch from one reference frame to another simply multiplying the vector by the Direction Cosines Matrix, for example:

$$r_b = A_{b/n} * r_n$$

The dynamic of the CubeSat has been modeled using the Euler equation, considering the architecture of a dual spin spacecraft:

$$\begin{cases} I_x\dot{\omega}_x + (I_z - I_y)\omega_z\omega_y + I_r\dot{\omega}_r = M_x \\ I_y\dot{\omega}_y + (I_x - I_z)\omega_x\omega_z + I_r\omega_r\omega_z = M_y \\ I_z\dot{\omega}_z + (I_y - I_x)\omega_y\omega_x - I_r\omega_r\omega_y = M_z \\ I_r\dot{\omega}_r = M_r \end{cases}$$

Where M_i is the torque acting on the i axis and is composed by the control torque plus the disturbances acting on the spacecraft.
Integrating this system of equations it is possible to find the angular velocities and orientation of the spacecraft at each time-step.
To speed up the computations, this integration is done using the quaternions representation.
After the integration, the result is converted back in the Direction Cosines Matrix representation to be more easy to use and understand.
As said before, constant ω_r is considered, so $\dot{\omega}_r$ and also M_r are equal to zero.

3.2 Disturbance models

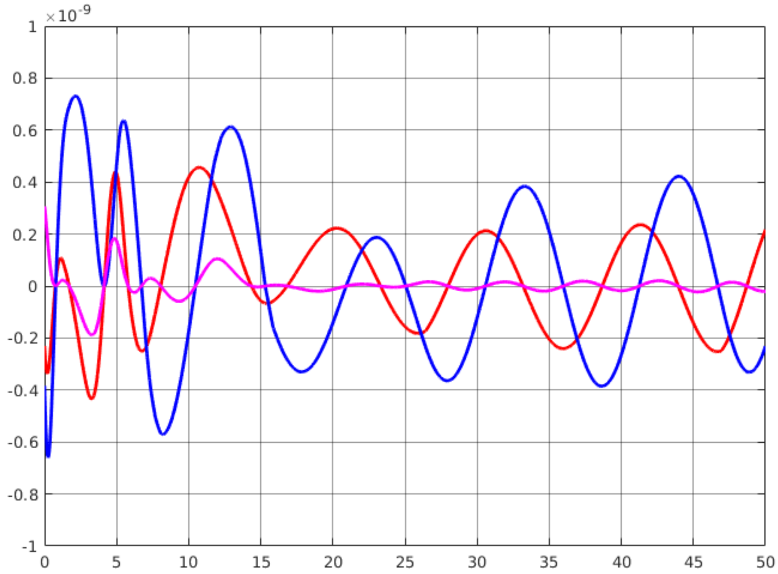
3.2.1 Gravity gradient torque

This disturbance is given by the non-uniformity of the gravity field.
It has to be said that in the case of a CubeSat, the disturbance torque is very small and can be neglected.
Anyway, to prove that, this model is integrated in our simulation, using the following formula:

$$\underline{M} = \frac{3Gm_t}{R^3} \begin{Bmatrix} (I_z - I_y)c_2c_3 \\ (I_x - I_z)c_1c_3 \\ (I_y - I_x)c_1c_2 \end{Bmatrix}$$

Where c_1,c_2,c_3 are the direction cosines of the radial direction in principal axes.
As it can be seen from the following plot, the order of magnitude of the torque is 10^{-10}.

Figure 3.1: Gravity gradient torque



3.2.2 Magnetic torque

Magnetic torque has been modeled as $\underline{M} = \underline{m} \otimes \underline{B}$, where $\underline{m}$ is the magnetic induction due to parasitic current in the satellite and $\underline{B}$ is the magnetic field of the Earth.
In the model proposed, $\underline{m}$ has been chosen arbitrarily as: $\underline{m} = \begin{bmatrix} 0.1 & 0.1 & 0.1 \end{bmatrix}'$.
The magnetic field $\underline{B}$ can be modeled as the gradient of a scalar potential V, that is $\underline{B} = -\nabla V$.
V is modeled as a series expansion of spherical harmonics:

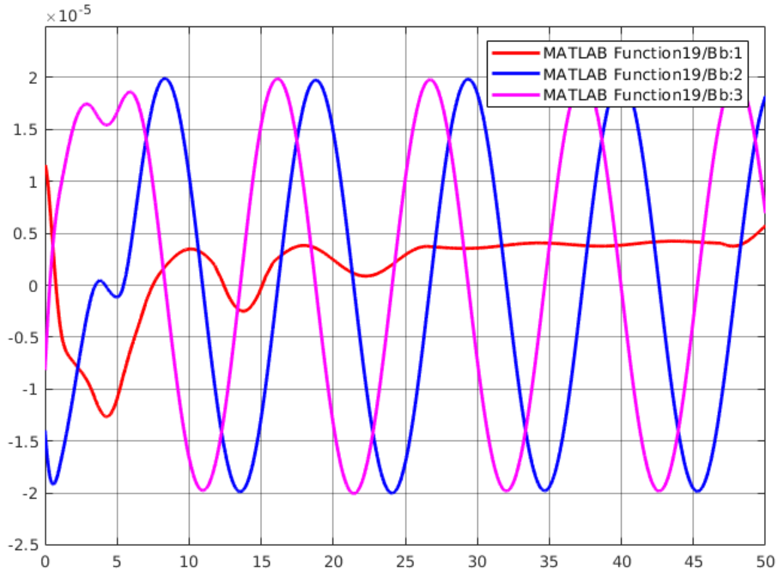
$$V(r,\theta,\phi) = R\sum_{n=1}^{k}(\frac{R}{r})^{n+1}\sum_{m=0}^{n}(g_n^m cos(m\phi) + h_n^m sin(m\phi)P_n^m(\theta)$$

Where R is the Earth's radius, g_n^m and h_n^m are called Gaussian and are valuated from experimental data (known up to order 13) and are subjected to time variation (IGRF 2000), r, θ and φ are the spherical coordinates of the position of the satellite (distance from the center of the Earth, colatitude and East longitude from Greenwich).

The choice of having a precise model of the magnetic field is due to the operating orbit of this mission that is below the 7000 Km height therefore a simpler dipole model cannot be adopted. To get polynomials P_n^m recursive formulas have been used. The components of magnetic field are:

$$\begin{cases} B_r = \sum_{n=1}^{k}(\frac{R}{r})^{n+2}(n+1) * \sum_{m=0}^{n}(g_n^m cos(m\phi) + h_n^m sin(m\phi))P_m^m(\theta) \\ B_\theta = -\sum_{n=1}^{k}(\frac{R}{n})^{n+2}\sum_{m=0}^{n}(g_n^m cos(m\phi) + h_n^m sin(m\phi))\frac{\partial P_m^m(\theta)}{\partial\theta} \\ B_\phi = -\frac{1}{sin(\theta)} * \sum_{n=1}^{k}(\frac{R}{n})^{n+2}\sum_{m=0}^{n}m(-g_n^m sin(m\phi) + h_n^m cos(m\phi))P_m^m(\theta) \end{cases}$$

Figure 3.2: Magnetic torque



The order of magnitude is 10^{-5}.

3.2.3 Torque due to atmospheric drag

Aerodynamic torque has been modeled using the following formula: $\underline{M} = \sum_{i=1}^{N} \underline{r_i} \otimes \underline{F_i}$
Where F_i is the aerodynamic force acting on a surface of area A:

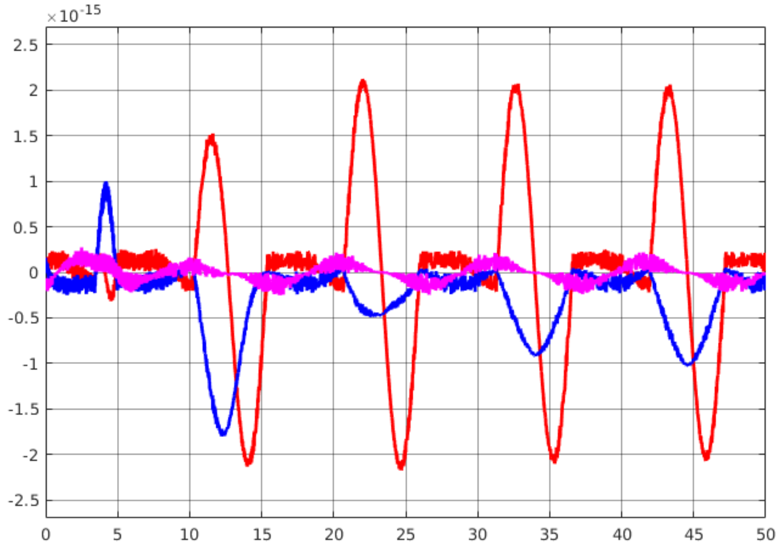
$$\underline{F_i} = \frac{1}{2}\rho C_d v^2 \sum_{i=1}^{n} \underline{r_i} \otimes (\hat{N}_i \bullet \hat{v}) A_i$$

The versor $\hat{N}$ is the normal of the face i, while the vector r_i is the position of the aerodynamic center of the face i.
An estimation for this point is done shifting it of a random distance between 0 and 10% of the length of one side, in a random direction from the geometric center.
The velocity of the CubeSat has been computed using a function that converts the Keplerian coordinates in the two vectors of position and speed.
The Keplerian parameter θ is computed integrating the angular speed.

The density of the air has been selected from appropriate tab, while the coefficient of drag has been selected from literature with a value of $C_d = 2.2$.
The Area A is 0.001 m^2 since the spacecraft considered is a 1U CubeSat.

The aerodynamic force is considered only for the surfaces where the condition $(\hat{N} \bullet \hat{v}) > 0$ is respected.

Figure 3.3: Torque due to atmospheric drag



The order of magnitude of this disturbance is very small, with a value of 10^{-15}.

3.2.4 Solar radiation torque

To compute this disturbance torque, the position of the Sun is needed.
As done before for the position of the Spacecraft, a transformation from Keplerian coordinates to position and speed vector is done.
The position of the CubeSat with respect to the Sun is computed with a sum of vector (preceded by a rotation, needed to align the two reference frames).

Solar radiation torque has been modeled using the following formula: $\underline{M} = \sum_{i=1}^{N} \underline{r_i} \otimes \underline{F_i}$, where F_i is now the resulting force due to the solar radiation pressure on a flat panel of area A:

$$\underline{F} = P * A * (\hat{S} \bullet \hat{N})[(1 - \rho_s)\hat{S} + (2\rho_s(\hat{S} \bullet \hat{N}) + \frac{2}{3}\rho_d)\hat{N}]$$

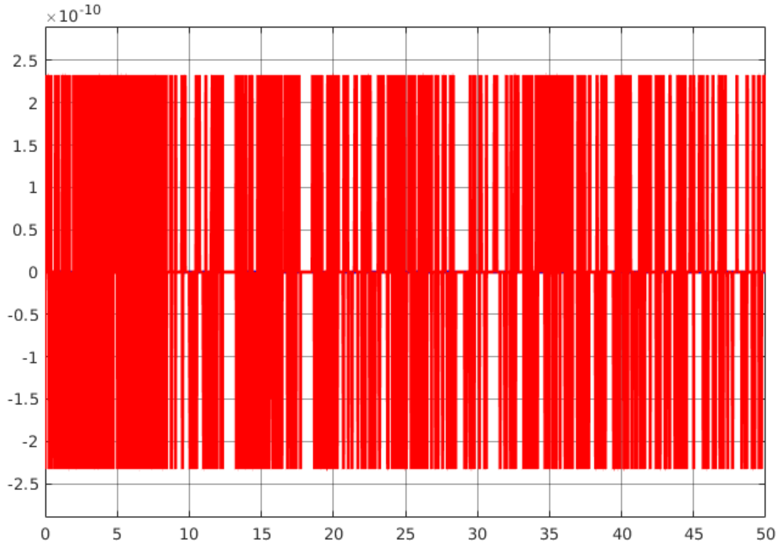
In this formula ρ_s and ρ_d are respectively the coefficient of diffuse and specular reflection and their values are: $\rho_s = 0.5$ and $\rho_d = 0.1$.
The angle θ is the angle between the surface orthogonal direction $\hat{N}$ and the unit direction of the satellite-Sun direction $\hat{S}$:

$$\theta = cos^{-1}(\hat{S} \bullet \hat{N})$$

An average radiation pressure of $P = 4.6 * 10^{-6} \frac{N}{m^2}$ has been used.

In the model is also considered that when the face is not illuminated there is no force acting on the surface of the CubeSat.
This result is achieved doing a test on θ, when it is greater than 90° the force F_i is set to zero
Another function compute if we are in the light or in the shadow of the sun.
During the transition from light to shadow a parameter λ changes its value from 1 to 0.
Multiplying the force $\underline{F}$ by the parameter λ, the correct solution is found.

Figure 3.4: Solar radiation torque



The order of magnitude in this case is 10^{-10}, also this disturbance can be neglected.

3.3 Control algorithm

Since it can be seen that the spacecraft Detumble quickly, both the errors of the two phases (Detumble and Earth-pointing) are computed with respect to frame Body/LVLH (however, this can be easily changed in the model using a “Manual Variant Source Block”).

A general linear control system has been adopted. It has the following form:

$$\underline{u} = \underline{k_1}[A_{b/l}^T - A_{b/l}] + \underline{k_2}\underline{\omega_{b/l}}$$

$\underline{k_1}$ and $\underline{k_2}$ can be set to increase or decrease the response of the control system to the two constituents of the error.
Since they are vector, the values can be different for different axes.
The first component of both $\underline{k_1}$ and $\underline{k_2}$ is 0 because only an Earth-pointing is needed, so the spacecraft can freely rotate around the pointing axis.

Both the de-tumbling and Earth-pointing are simulated with one single model.
To switch from one phase to the other, a “SwitchCase Block” controls two Case Subsystems.
The transition from one phase to the other is done only when the absolute value of all the components of the vector $\underline{u}$ are under a given value for a given amount of time.

Figure 3.5: Response of the control algorithm

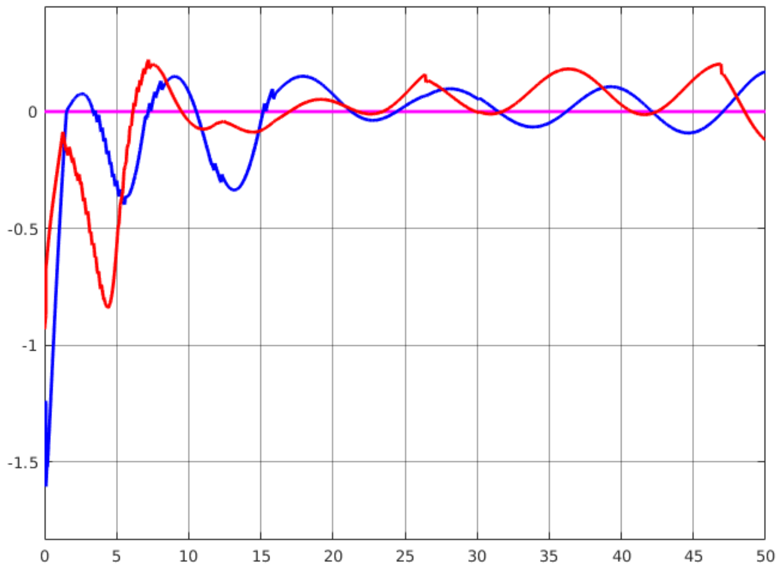
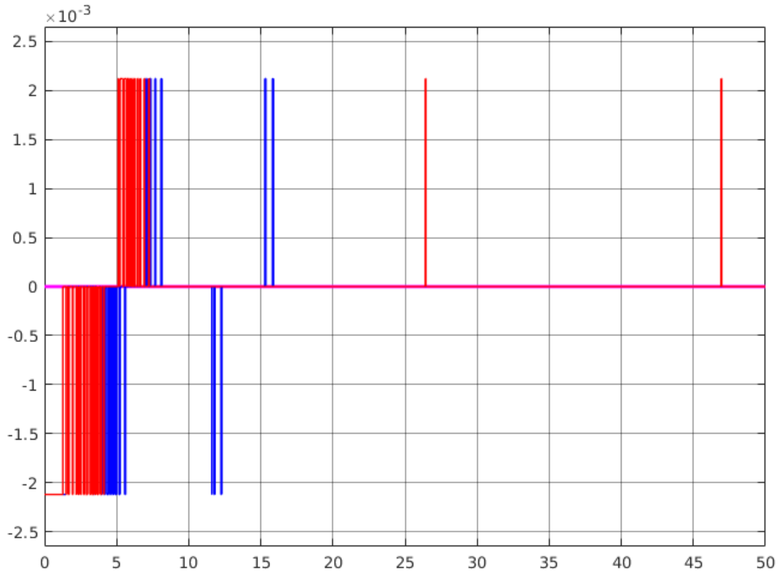


Figure 3.6: Response of the Thrusters



3.4 Sensors models

3.4.1 Magnetometer model

The magnetic field was already found in the Magnetic torque section, so to simulate the magnetometer signal, it is enough to multiply an error matrix A_ϵ for the ideal magnetic field:

$$\underline{B_{b_{magnetometer}}} = A_\epsilon \underline{B_b}$$

The matrix $A_e$is defined in the following way:

$$A_\epsilon = \begin{bmatrix} cos\epsilon * cos\epsilon & cos\epsilon * sin\epsilon * sin\epsilon & -cos\epsilon * sin\epsilon * cos\epsilon + sin\epsilon * sin\epsilon \\ -sin\epsilon * cos\epsilon & -sin\epsilon * sin\epsilon * sin\epsilon + cos\epsilon * cos\epsilon & sin\epsilon * sin\epsilon * cos\epsilon + cos\epsilon * sin\epsilon \\ sin\epsilon & -cos\epsilon * sin\epsilon & cos\epsilon * cos\epsilon \end{bmatrix}$$

The potential accuracy of the Magnetometer is 30 arc-minutes, thus $\epsilon = 0.0087\, rad$.

3.4.2 Gyro model

The Gyro models into account the potential accuracy of the sensor (using the $A_\epsilon$defined for the Magnetometer model) and three different type of noise.
The potential accuracy in this case is much higher with $\epsilon = 0.005\, rad$, but since the Gyro can only determine the angular velocities, integrating them, the error accumulates.
So the resulting $A_{b/n}$ matrix can be imprecise after a long time.

The noises are defined in that way:

$$\begin{cases} n_i = n_e + n_v + n_u \\ n_e = \sigma_\epsilon \xi_e \\ n_v = \sigma_v \sqrt{\delta t} \xi_v \\ n_u(t) = n_u(t - \delta t) + \sigma \sqrt{\delta t} \xi_u \end{cases}$$

ξ_e, ξ_v and ξ_u are three random numbers generated by the "Random Number Block", while σ_i, σ_v and σ_t are the square root of the variance.
So the measured angular velocity is:

$$\omega_g = A_e * \omega_{b/n} + n$$

3.5 Determination algorithm

To determine the attitude of the spacecraft, the first step is to find the starting $A_{b/n}$ matrix, $A_{b/n}(0)$.
Since both the vector computed by the magnetometer B_b and the one found by theoretic model B_n are known, and

$$\underline{B_b} = A_{b/n}\underline{B_n}$$

it is possible to find the matrix $A_{b/n}(0)$ solving a minimization problem.
A simple cost function has been chosen:

$$J(A_{b/n}) = \left\|\underline{B_b} - A_{b/n}\underline{B_n}\right\|^2$$

Using the integrated constrained minimization function **fmincon** it is possible to minimize the cost function and find the correct $A_{b/n}$ matrix.
To do this, we use as first guess the $A_{b/n}$ matrix computed at the last iteration. This grants that the research start near the optimal solution, so the local minimum found has excellent odds of being the global one.
For the first time-step $A_{b/n}(0)$ is used as guess. This problem also exists in the real mission.
A good starting guess (near the global minimum) is needed to assure that the algorithm works, otherwise a better optimization algorithm should be used (a metaheuristic one, for example Taboo search algorithm, which is also much more complex).

The evolution of the attitude is found using ω_g ($\omega_{b/n}$computed by the Gyro) in the Euler equations and integrating them.
To eliminate the accumulation of error, every 30 seconds the minimization problem is solved and the new $A_{b/n}(0)$ is found.

3.6 Actuators models

3.6.1 Thrusters

For the control of the thrusters, a "Schmitt trigger" nonlinear switch is selected. Thanks to this choice, the thrusters are activated on the basis of a variable $\epsilon = \theta + \tau\partial\theta$.
This type of control prevents the thrusters from being switched on and off too often as this would cause instability.

Since for the kind of thrusters selected it is impossible to change the thrust magnitude, the thrust value is computed as:

$$\underline{T} = T_{max} * sign(\underline{u})$$

Then the correct thrust is modeled using a "Zero-Hold Block", since there is a maximum frequency at which the thrusters can be activated and deactivated, a "Rate Limiter Block", since the real thrust does not reach the maximum value directly, and a "Transport Delay Block", to simulate the small latencies.

3.6.2 Reaction Wheels

The First thing to define is the orientation of the reaction wheels.
Since this is a pointing mission, only two reaction wheels are needed, but for redundancy it is possible to add another one.
The orientation of the first two is on the principal axes that are not pointing towards the Earth, while the third one is diagonally oriented between the three axes:

$$A = \begin{bmatrix} 0 & 0 & \frac{1}{\sqrt{3}} \\ 1 & 0 & \frac{1}{\sqrt{3}} \\ 0 & 1 & \frac{1}{\sqrt{3}} \end{bmatrix}$$

The resulting $\dot{\underline{h}}$ is computed with the following formula:

$$\dot{\underline{h}} = A^T * [(A * A^T)^{-1}] * [(A * \underline{h}) \otimes \underline{\omega_{b/n}} - \underline{u}]$$

Then to correctly simulate saturation on $\underline{h}$ and the limited torque effect on $\dot{\underline{h}}$, a "Saturation Block" is used.
The last step is to compute the real control torque with the following expression:

$$\underline{u} = -A * \dot{\underline{h}} - \underline{\omega_{b/n}} \otimes (A * \underline{h})$$

Chapter 4

Results

The simulations starts from the following conditions:

Table 4.1: Initial values

$\underline{\omega_{b/n_x}}(0)$	$0.5\ \frac{rad}{s}$
$\underline{\omega_{b/n_y}}(0)$	$1.75\ \frac{rad}{s}$
$\underline{\omega_{b/n_z}}(0)$	$1.25\ \frac{rad}{s}$

The gain factors of the PD controller are:

Table 4.2: Gain factors

$\underline{k_1}$	0.75
$\underline{k_2}$	0.75

Since the aim of detumbling is reducing the angular velocities and not pointing in the right direction, during this phase the gain factor k2 is reduced by a factor of 2.

The computation of the real thrust profile through the three blocks of "Zero-Hold", "Rate Limiter" and "Transport Delay" is disabled during the simulations to speed up the computations (otherwise the computational time while detumbling would increase by a factor of about 500).
The Reaction Wheels saturation model also increases dramatically the computational time, but disabling it would make the simulation meaningless.
For this reason the simulation time is very narrow. Below are the results of a 120 seconds simulation.

Figure 4.1: Angular Velocities

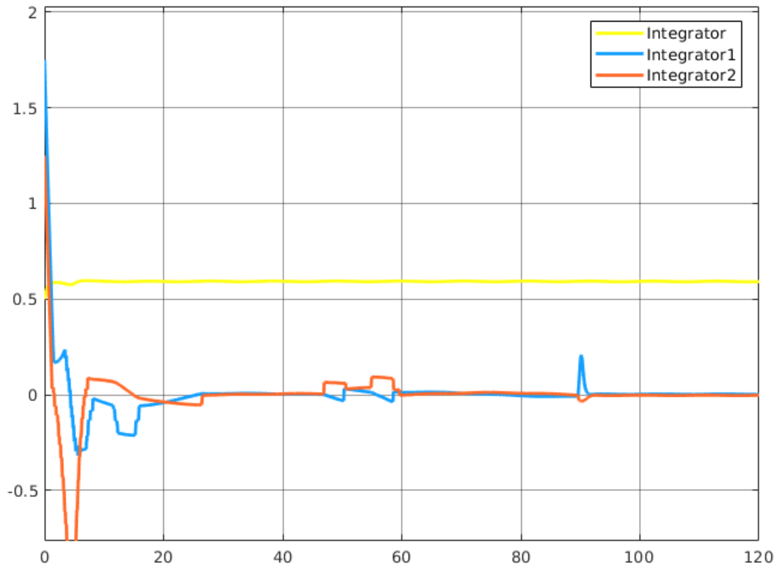


Figure 4.2: DCM Matrix

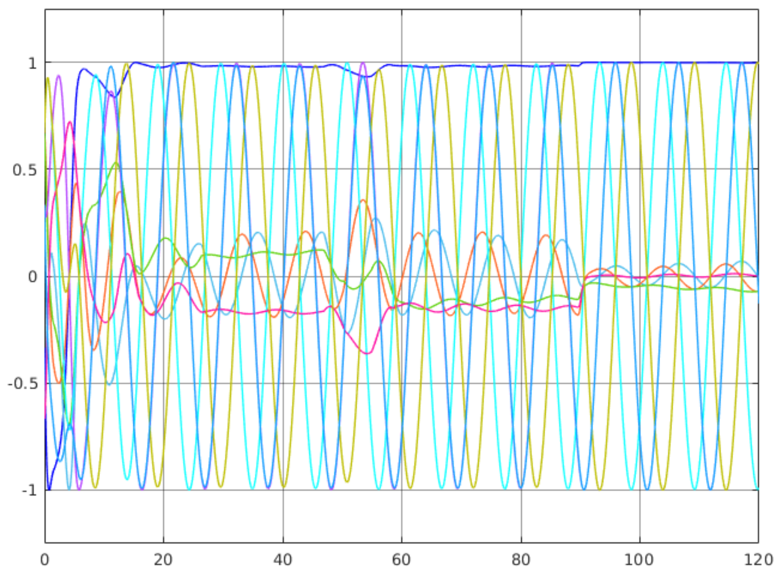


Figure 4.3: Pointing Error [deg]

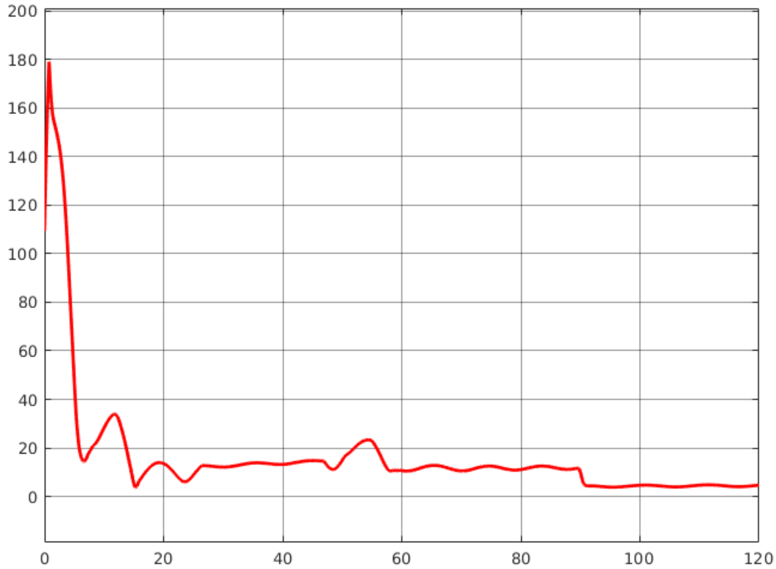
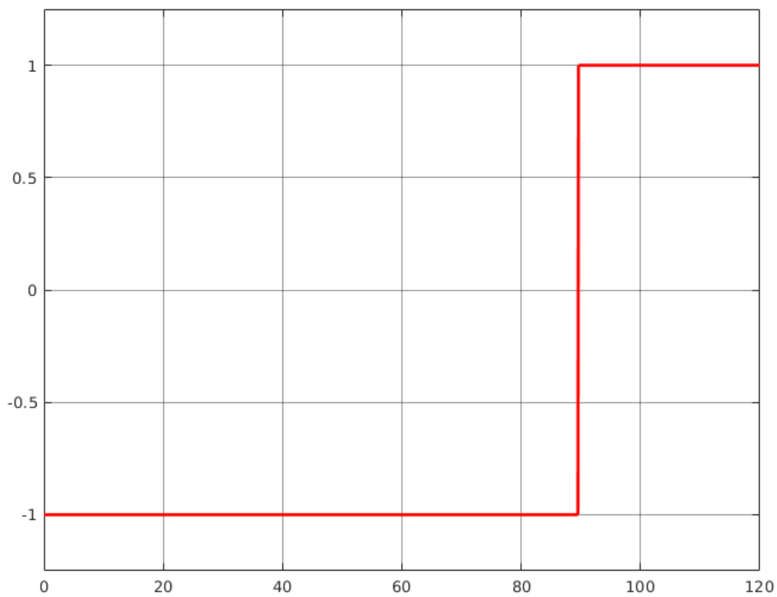


Figure 4.4: Switching from Detumbling to pointing



Chapter 5

Conclusion

As it can be easily seen, the designed controller systems works and Detumble the spacecraft in about 90 seconds.
After the detumbling, an accuracy of pointing under 6° is granted.
The momentum storage provided by the Reaction wheel is sufficient and desaturation is not needed.
The accuracy is not very small due to the fact that the attitude is determined in an imprecise way.
A better sensor combination should be considered.

During the switch from detumbling to pointing we can observe an instantaneous increase in the response %u.
This increment is given by the difference between the two error vectors used by detumbling and pointing phases.

On the other hand, the simulation is very slow to run and, to be useful in the real world, the actuators models should be computationally optimized.

Figure 5.1: Rendering of the simulation

The rendering of the simulation can be found at this link
or scanning the QR code above.
The spacecraft is scaled up by a factor of 10^7, to be more visible.