https://lib.uliege.be https://matheo.uliege.be

Master Thesis: OUFTI-NEXT ADC Subsystem

Auteur : p214516
Promoteur(s) : Kerschen, Gaetan
Faculté : Faculté des Sciences appliquées
Diplôme : Cours supplémentaires destinés aux étudiants d'échange (Erasmus, ...)
Année académique : 2017-2018
URI/URL : http://hdl.handle.net/2268.2/8514

Avertissement à l'attention des usagers :

Tous les documents placés en accès ouvert sur le site le site MatheO sont protégés par le droit d'auteur. Conformément aux principes énoncés par la "Budapest Open Access Initiative"(BOAI, 2002), l'utilisateur du site peut lire, télécharger, copier, transmettre, imprimer, chercher ou faire un lien vers le texte intégral de ces documents, les disséquer pour les indexer, s'en servir de données pour un logiciel, ou s'en servir à toute autre fin légale (ou prévue par la réglementation relative au droit d'auteur). Toute utilisation du document à des fins commerciales est strictement interdite.

Par ailleurs, l'utilisateur s'engage à respecter les droits moraux de l'auteur, principalement le droit à l'intégrité de l'oeuvre et le droit de paternité et ce dans toute utilisation que l'utilisateur entreprend. Ainsi, à titre d'exemple, lorsqu'il reproduira un document par extrait ou dans son intégralité, l'utilisateur citera de manière complète les sources telles que mentionnées ci-dessus. Toute utilisation non explicitement autorisée ci-avant (telle que par exemple, la modification du document ou son résumé) nécessite l'autorisation préalable et expresse des auteurs ou de leurs ayants droit.

Politecnico di Milano & University of Liège - Faculty of Applied Sciences

SCHOOL OF INDUSTRIAL AND INFORMATION ENGINEERING
Master of Science in Space Engineering

OUFTI-Next ADC subsystem

Graduation Studies conducted for obtaining the Master′s degree in "Space Engineering"

Student:
Cristiano Contini

Professors:
Dr. G.Kerschen
Dr. C.Colombo

Academic Year 2017–2018

Acknowledgments

> *We are only as strong as we are united, as weak as we are divided.*
>
> Albus Dumbledore, J.K. Rowling

First of all, I wish to thank Prof. Gaëtan Kerschen and Dr. Xavier Werner who made me feel part of a big family since the first day I arrived. They transmitted me their enthusiasm and let me be part of the extraordinary project of OUFTI-Next. Their support and encouragement have been fundamental to me during these months.
The opportunity to design a machine that will fly in space is the dream of every Space Engineering student, continuously inspired by the human desire to go beyond the limits, try what nobody could do before and explore the unknown.
This work and the life experience correlated to it have been possible thanks to Professor Camilla Colombo, who first introduced me to the European dimension of the Space Engineering. She encouraged me to leave Italy and start a new and unexpected adventure, that made me know a wonderful country and meet amazing people.
A special thanks goes to my parents and brother, who trusted and supported me during this years of studies, efforts, difficulties and successes, you have always been the polar star of my life.
I wish to thank in particular my girlfriend and adventure companion Barbara, who makes me always see what is good on Earth with respect to space. You are the one who inspired me the desire to travel and see this world with great enthusiasm.
Thanks also to ASSYST team: Andrea P., Luca M., Andrea S., Adriano G., Francesca S., Paolo V., Giovanni B., Andrea G. and Marco G. who taught me how to work in team, being my family for almost a semester. Finally a special thanks goes to Simone, who supported and beared me, everyday during the years of our University career.

Cristiano Contini
August 2018

Abstract

> *Nec refert quibus adsistas regionibus eius, usque adeo quem quisque locum possedit, in omnìs tantundem partìs infinitum omnem relinquit. Praeterea si iam finitum constituatur omne quod est spatium, si quis procurrat ad oras ultimus extremas iaciatque volatile telum, id validis utrum contortum viribus ire quo fuerit missum mavis longeque volare, an prohibere aliquid censes obstareque posse?*
>
> Lucrezio, *De Rerum Natura*, I, 951/983

OUFTI-Next is the name of the CubeSat demonstrator the University of Liège is designing in order to demonstrate the feasibility of an ambitious project regarding Earth observation: a fleet of CubeSats shooting pictures every day of big agricultural fields with an Infra-Red camera, showing their hydraulic stress and providing fundamental informations about how much and where to irrigate, in order to preserve reserves of water, especially in countries suffering its scarcity. The Thesis is focused on the ADC subsystem of OUFTI-Next, with the aim to build a MATLAB simulator capable to provide results useful to understand the project feasibility, to size the subsystem and trade-off between COTS available on the market (*iADCS100*, KUL, etc...), to optimize the picture acquisition through dedicated slew maneuvers and to analyze the perturbations induced by the cryocooler on-board.

Contents

Chapter 1

Introduction

This Master Thesis is the first part of a deeper work about orbit and attitude dynamics coupled in ADCS of high surface-to-mass ratio satellites.
Nowadays spacecrafts are designed with complex shapes and large areas, in order to accomplish more and more peculiar mission objectives, maintaining a reduced mass to reduce the launch cost. As a result, for Earth observation satellites, aerodynamic drag and solar radiation pressure become the main actors, driving and coupling the orbital dynamics with the attitude one. This is the case of ZodiArt, [4], a new mission carried out by Politecnico di Milano and supervised by Prof. Camilla Colombo, whose aim is to launch a fleet of MicroSats each of them equipped with a reflective balloon, to promote space advertisement during night time and to enhance Earth observation during day time.
The work will be subdivided as follow:

- Period at ULiège on OUFTI-Next:
    - ADCS design and requirements based on the optical design and performance.
    - Optimization of the image acquisition through slewing maneuvers.
    - Development of an ADCS simulator.
    - Study of the perturbations due to the cryocooler.
    - Study of the COTS available on the market (*Hyperion*, KUL).
- Period at Politecnico di Milano:
    - Test of the simulator provided to ULiège with ZodiArt platform.
    - Development of the simulator including a united state model to couple the orbit and attitude dynamics.
    - Sizing of the ADC subsystem of ZodiArt platform in order exploiting perturbations to control the spacecraft.
    - Provide more accurate simulations also for OUFTI-Next.

This Master Thesis reports the results and the work done about OUFTI-Next project, during the period spent in ULiège. A second report will be presented in Politecnico di Milano once the whole Thesis will be completed.

1.1 OUFTI-Next mission

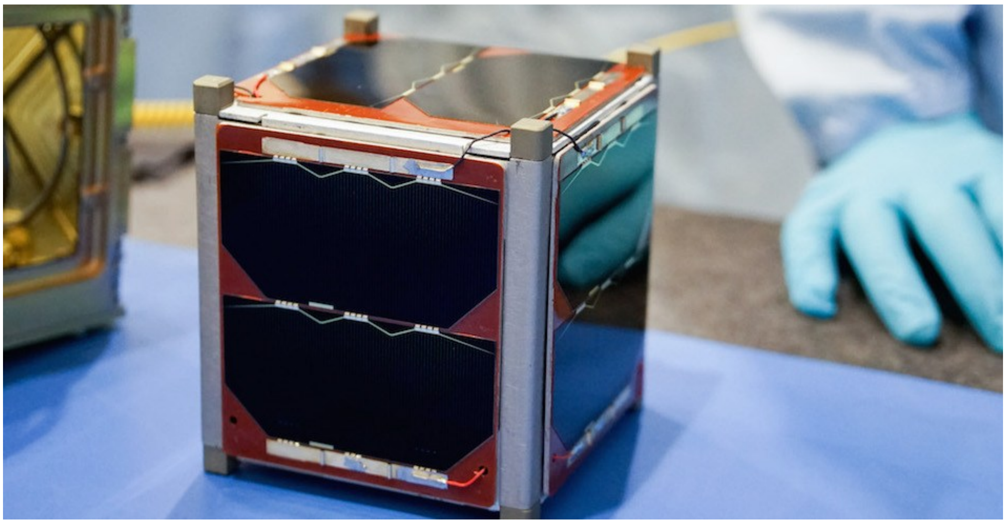


Figure 1.1 OUFTI-1 CubeSat.

The OUFTI project was born at University of Liege in November 2016, as a continuation of the first CubeSat ever launched in Belgium, OUFTI-1 in April 2016, when Bernard Tychon proposed to build a CubeSat capable to detect hydraulic stress in agricultural fields and to provide useful informations about their correct irrigation.
The phase A of the mission has coming to its end and each subsystem has been preliminary sized.

1.1.1 ADC subsystem

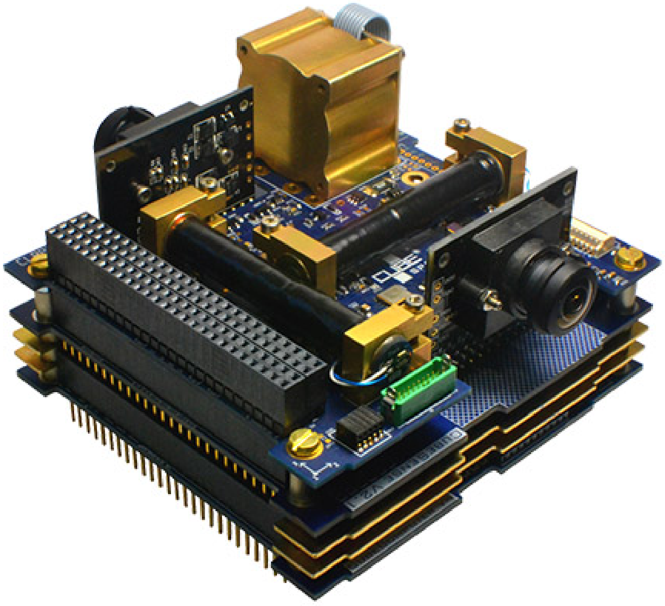


Figure 1.2 Example of ADC subsystem for CubeSats.

The ADC subsystem has the aim to control the spacecraft, providing the requested control torque to perform each mission phase, and determine its position and velocity (the state) in order to close the feedback loop shown in figure 5.1 on page 23.

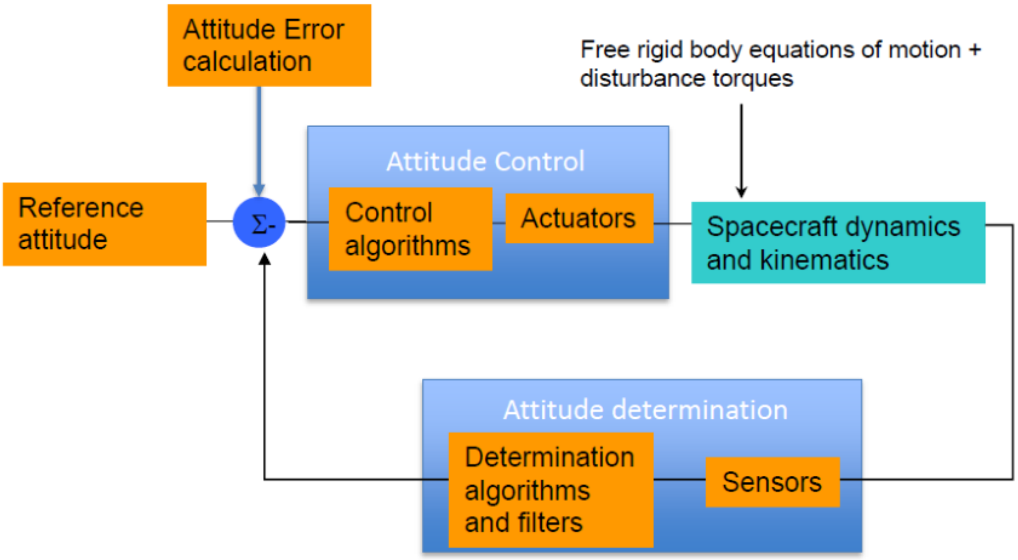


Figure 1.3 ADC subsystem feedback loop.

Generally an ADCS simulator is composed by:

- Spacecraft dynamics and kinematics model.
- Control algorithm.
- Actuators models.
- Sensors models.
- Determination algorithms.
- Reference attitude.
- External and internal disturbances models.
- Orbital model or ephemeris of the spacecraft.

At this stage of the Thesis, the orbital dynamics is still not coupled with the attitude one, so, as it will be explained in the following chapters, OUFTI-Next ephemeris are provided as an input to the simulator, instead of integrating the orbital model.

1.1.2 Report structure

This report will be subdivided in chapters describing first the requirements for the ADCS and how the OUFTI ephemeris has been computed, then how each element of the simulator has been implemented, the mission profiles selected for the simulations, the optimization process, the general results obtained, the cryocooler disturbance implementation, the trade-off analysis between two different COTS, the measurements uncertainties mitigation and emergency strategies.

Chapter 2

Nomenclature and reference frames

In this Thesis three main reference systems will be used: the *inertial reference frame*, the noninertial *body-fixed reference frame* and *LVLH reference frame*. The first one has its origin in the center of the Earth, its X-axis is oriented towards the vernal direction and the Z-axis is pointed towards the North Pole. The second one instead is a noninertial frame, characterized by the X-axis directed in the opposite direction with respect to the detector pointing vector, while the Z and Y-axis directed towards the surfaces with solar panels, figure 2.1. The S-band antenna is considered to be on the -Z-axis.

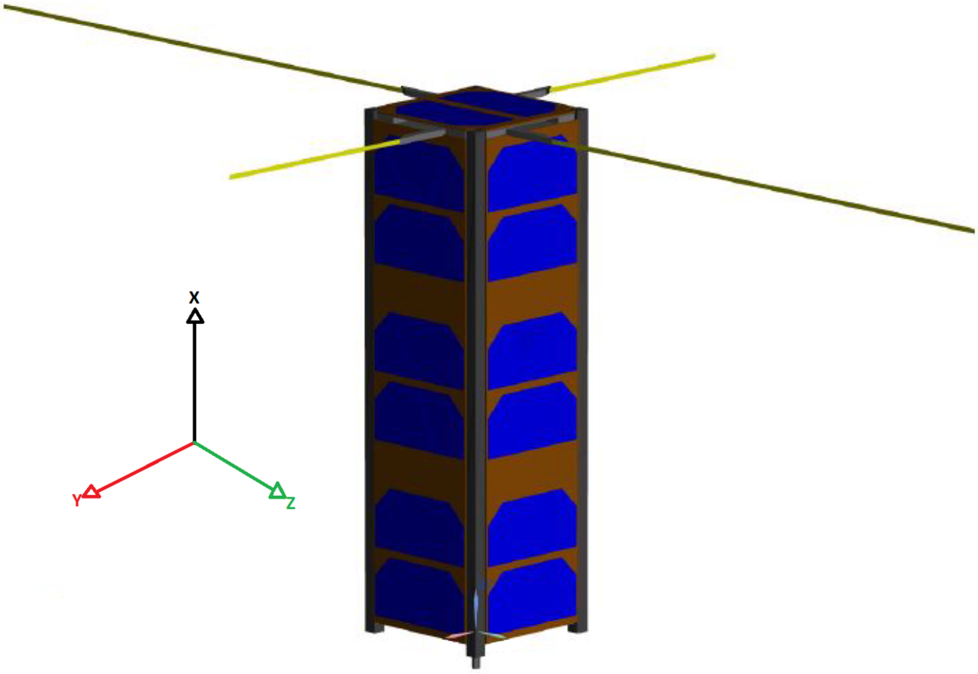


Figure 2.1 The reference frame used for the design of OUFTI-Next ADCS.

The last one is also a noninertial reference frame whose orthogonal unit vectors are $\hat{\mathbf{r}}$, $\hat{\mathbf{s}}$ and $\hat{\mathbf{w}}$, where $\hat{\mathbf{r}}$ is the position vector direction. $\hat{\mathbf{w}}$, instead, is the unit vector normal to the osculating orbital plane, in the direction of angular momentum vector $\mathbf{h}$, the transverse vector $\hat{\mathbf{s}}$ is normal to both $\mathbf{r}$ and $\hat{\mathbf{w}}$ and it therefore points in the direction of the orbiting body's local horizon, figure 2.2 on the following page.

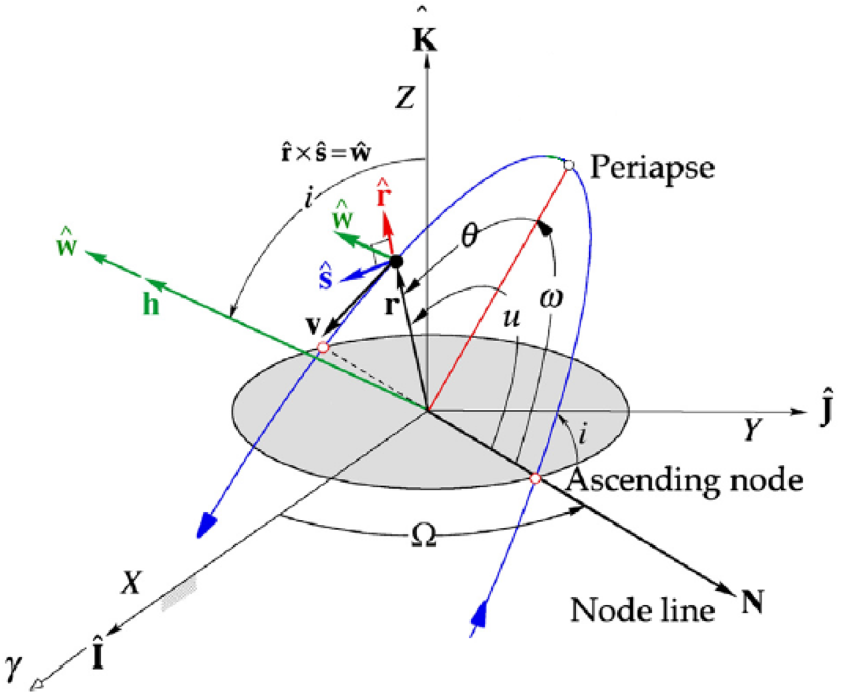


Figure 2.2 The local vertical–local horizon rsw frame. u is the argument of latitude, [5].

The subscript $(\cdot)_{b/n}$, means a rotation between the *body-fixed reference frame*, indicated by b, from the *inertial reference frame*, indicated by n. A matrix in *LVLH frame*, $\boldsymbol{A_{LVLH}}$, is actually a rotation with respect to the *inertial frame* and can be indicated as $(\cdot)_{l/n}$. As a result the following matrix multiplication can generate a rotation from *LVLH frame* to *body-fixed reference frame*, $\boldsymbol{A_{b/l}}$:

$$\boldsymbol{A_{b/l}} = \boldsymbol{A_{b/n}}\boldsymbol{A_{l/n}}^T \tag{2.1}$$

where $(\cdot)^T$ is the transpose of the matrix.

The attitude has been implemented exploiting the quaternions, 4-dimensional numbers representing a single rotation in 3D space. This is due to the *Euler's rotation theorem*: "Any single rotation can be represented by a vector, $\hat{\boldsymbol{e}}$, (eigenvector) that remains fixed during that rotation and a simple rotation around that vector by an angle, θ (eigenangle)", figure 2.3.

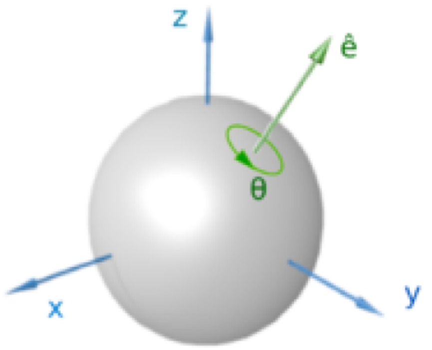


Figure 2.3 Quaternion representing a single rotation of a 3D-body.

where:

$$\begin{cases} q_1 = e_1 sin\frac{\theta}{2} \\ q_2 = e_2 sin\frac{\theta}{2} \\ q_3 = e_3 sin\frac{\theta}{2} \\ q_4 = cos\frac{\theta}{2} \end{cases} \tag{2.2}$$

Other kind of implementations are: *direct cosine matrices*, *Euler angles* and *Rodrigez vectors*. The reasons why it has been decided to use quaternions instead of the other kind of representation are the following:

- Only 4 parameters are needed to be integrated. This is a good trade-off between the *direct cosine matrices* (9) and the *Euler angles/Rodrigez vectors* (3).
- They are globally defined, meaning that no singularities are present. Singularities affect both *Euler angles* and *Rodrigez vectors*.

The disadvantages are the fact that they are not unique, meaning that more than one quaternion can indicate the same rotation, they need a quaternion normalization at each integration step and they are not intuitive.

The attitude matrices are direct cosine matrices indicated with $\boldsymbol{A}$ and they will be transformed in quaternion, $\boldsymbol{q}$, using the following relations:

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

$$\begin{cases} q_1 = \frac{1}{4q_4}(A_{23} - A_{32}) \\ q_2 = \frac{1}{4q_4}(A_{31} - A_{13}) \\ q_3 = \frac{1}{4q_4}(A_{12} - A_{21}) \\ q_4 = \pm\frac{1}{2}(1 + A_{11} + A_{22} + A_{33})^{\frac{1}{2}} \end{cases} \quad \begin{matrix} q_1^2 = \pm\frac{1}{2}\sqrt{1 + A_{11} - A_{22} - A_{33}} \\ q_2^2 = \frac{1}{4q_1^2}(A_{12} + A_{21}) \\ q_3^2 = \frac{1}{4q_1^2}(A_{13} + A_{31}) \\ q_4^2 = \frac{1}{4q_1^2}(A_{23} - A_{32}) \end{matrix} \quad \begin{matrix} q_2^3 = \pm\frac{1}{2}\sqrt{1 - A_{11} + A_{22} - A_{33}} \\ q_1^3 = \frac{1}{4q_2^3}(A_{12} + A_{21}) \\ q_3^3 = \frac{1}{4q_2^3}(A_{23} + A_{32}) \\ q_4^3 = \frac{1}{4q_2^3}(A_{31} - A_{13}) \end{matrix} \quad \begin{matrix} q_3^4 = \pm\frac{1}{2}\sqrt{1 - A_{11} - A_{22} + A_{33}} \\ q_1^4 = \frac{1}{4q_3^4}(A_{13} + A_{31}) \\ q_2^4 = \frac{1}{4q_3^4}(A_{23} + A_{32}) \\ q_4^4 = \frac{1}{4q_3^4}(A_{12} - A_{21}) \end{matrix}$$

Figure 2.4 Direct cosine matrix to quaternion transformation and viceversa.

Finally, the cross product between 2 3-dimensional vectors will be also performed through the skewsymetric operator, $S(\cdot)$, defined in [9] as following:

$$\boldsymbol{a} \times \boldsymbol{b} = \boldsymbol{S}(\boldsymbol{a})\boldsymbol{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \boldsymbol{b} \tag{2.3}$$

Chapter 3

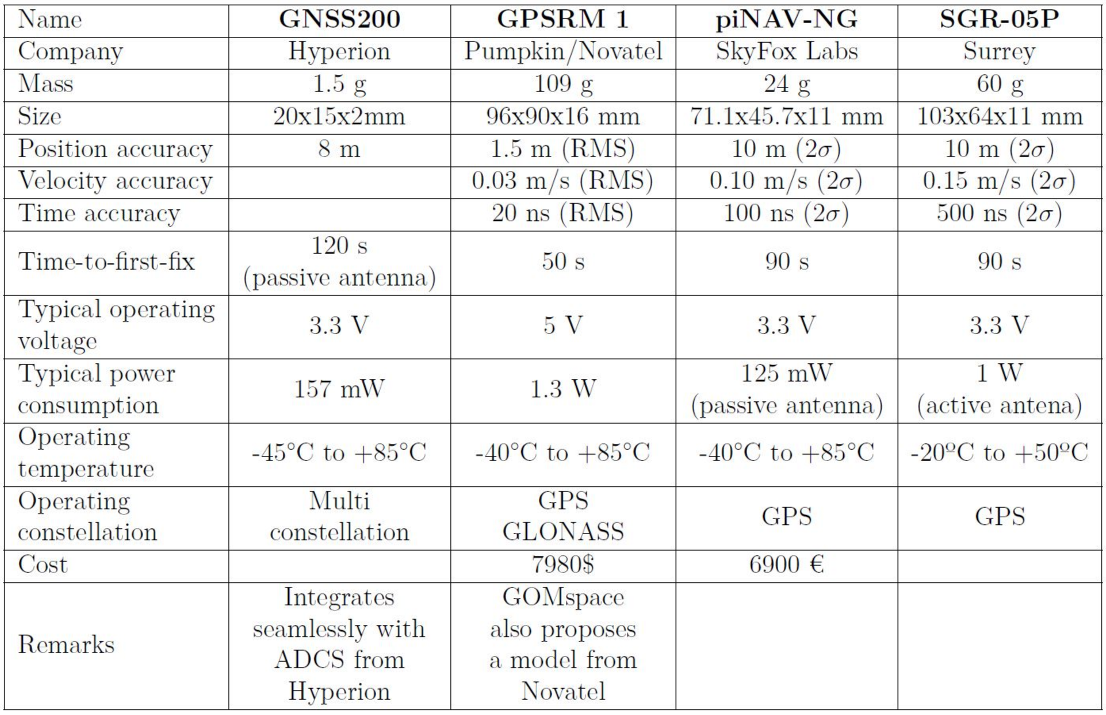
ADCS requirements and other subsystems main results

In this chapter the high level requirements for the ADC subsystem are listed, as well as preliminary considerations about the subsystem:

REQ. ID	Importance	Verification	Requirement	Note
ON - P - F - ADCS - 0010	M		The ADCS s\s shall be able to detumble the spacecraft after detachment from the launcher.	
ON - P - F - ADCS - 0020	M		The ADCS s\s shall provide sun-pointing with the faces covered by solar panels, during "Charging Mode".	
ON - P - F - ADCS - 0030	M		The ADCS s\s shall provide ground station tracking with the face where the S-band antenna is body-mounted, during "Downloading Mode".	
ON - P - F - ADCS - 0040	M		The ADCS s\s shall provide a "Target following" slew maneuver, during "Acquisition Mode".	
ON - P - P - ADCS - 0050	M		The ADCS s\s shall be able to determine the attitude within +/- 1 arcsec in each axis (TBC).	
ON - P - F - ADCS - 0060	M		The ADCS s\s shall provide a de-saturation strategy.	
ON - P - F - ADCS - 0070	M		The ADCS s\s shall counteract environmental disturbances.	e.g.: SRP, Atmospheric Drag, etc.
ON - P - F - ADCS - 0080	M		The ADCS s\s shall counteract internal disturbances.	e.g.: Cryocooler vibrations
ON - P - F - ADCS - 0090	M		The ADCS pack shall be chosen among the ones availale on the market.	e.g.: BCT, KUL, etc.
ON - P - F - ADCS - 0100	NH		The ADCS pack should be chosen among the ones availale on the national (Belgium) market.	e.g.: KUL
ON - P - P - ADCS - 0110	M		The ADCS s\s shall be able to provide a pointing accuracy of < 0.1 deg and a control stability of < 0.01 deg (0,5 arcmin) during "Acquisition Mode".	

Figure 3.1 OUFTI-NEXT project ADCS preliminary requirements

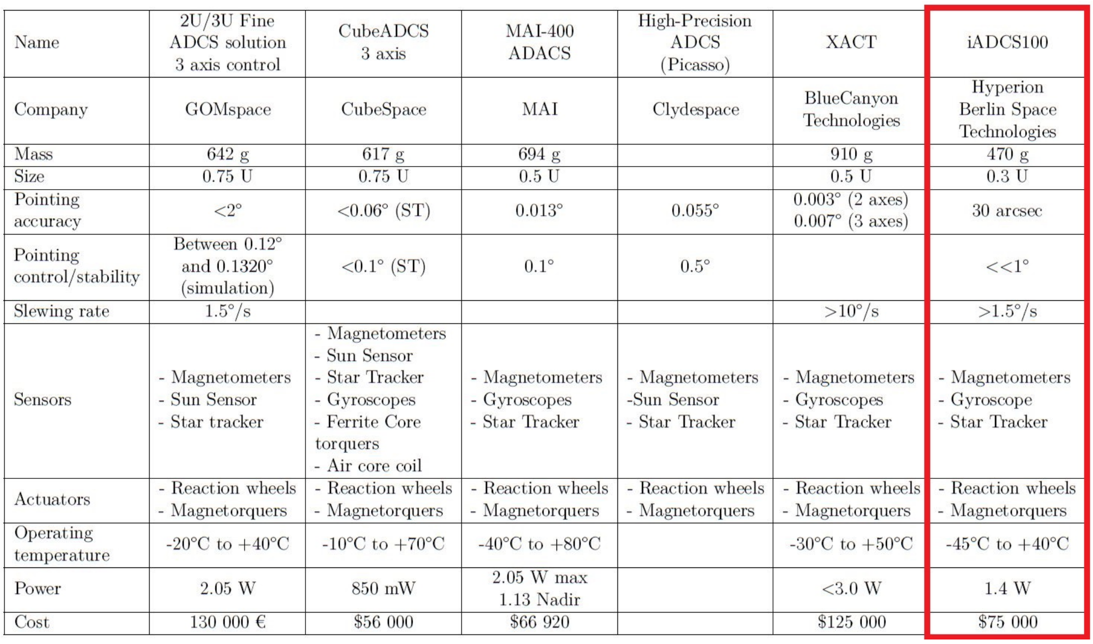
Here are reported the already considered COTS models for GNSS units, figure 3.2 on the following page [6].



Name	**GNSS200**	**GPSRM 1**	**piNAV-NG**	**SGR-05P**
Company	Hyperion	Pumpkin/Novatel	SkyFox Labs	Surrey
Mass	1.5 g	109 g	24 g	60 g
Size	20x15x2mm	96x90x16 mm	71.1x45.7x11 mm	103x64x11 mm
Position accuracy	8 m	1.5 m (RMS)	10 m (2σ)	10 m (2σ)
Velocity accuracy		0.03 m/s (RMS)	0.10 m/s (2σ)	0.15 m/s (2σ)
Time accuracy		20 ns (RMS)	100 ns (2σ)	500 ns (2σ)
Time-to-first-fix	120 s (passive antenna)	50 s	90 s	90 s
Typical operating voltage	3.3 V	5 V	3.3 V	3.3 V
Typical power consumption	157 mW	1.3 W	125 mW (passive antenna)	1 W (active antena)
Operating temperature	-45°C to +85°C	-40°C to +85°C	-40°C to +85°C	-20ºC to +50ºC
Operating constellation	Multi constellation	GPS GLONASS	GPS	GPS
Cost		7980$	6900 €	
Remarks	Integrates seamlessly with ADCS from Hyperion	GOMspace also proposes a model from Novatel		

Figure 3.2 GNSS COTS models

For what concerns ADCS units, a preliminary trade-off has already been done in [6], providing as the best solution, especially for its reduced volume, the *iADCS100*, whose characteristics are presented in figure 3.3.



Name	2U/3U Fine ADCS solution 3 axis control	CubeADCS 3 axis	MAI-400 ADACS	High-Precision ADCS (Picasso)	XACT	iADCS100
Company	GOMspace	CubeSpace	MAI	Clydespace	BlueCanyon Technologies	Hyperion Berlin Space Technologies
Mass	642 g	617 g	694 g		910 g	470 g
Size	0.75 U	0.75 U	0.5 U		0.5 U	0.3 U
Pointing accuracy	<2°	<0.06° (ST)	0.013°	0.055°	0.003° (2 axes) 0.007° (3 axes)	30 arcsec
Pointing control/stability	Between 0.12° and 0.1320° (simulation)	<0.1° (ST)	0.1°	0.5°		<<1°
Slewing rate	1.5°/s				>10°/s	>1.5°/s
Sensors	- Magnetometers - Sun Sensor - Star tracker	- Magnetometers - Sun Sensor - Star Tracker - Gyroscopes - Ferrite Core torquers - Air core coil	- Magnetometers - Gyroscopes - Star Tracker	- Magnetometers -Sun Sensor - Star Tracker	- Magnetometers - Gyroscopes - Star Tracker	- Magnetometers - Gyroscope - Star Tracker
Actuators	- Reaction wheels - Magnetorquers	- Reaction wheels - Magnetorquers	- Reaction wheels - Magnetorquers	- Reaction wheels - Magnetorquers	- Reaction wheels - Magnetorquers	- Reaction wheels - Magnetorquers
Operating temperature	-20°C to +40°C	-10°C to +70°C	-40°C to +80°C		-30°C to +50°C	-45°C to +40°C
Power	2.05 W	850 mW	2.05 W max 1.13 Nadir		<3.0 W	1.4 W
Cost	130 000 €	$56 000	$66 920		$125 000	$75 000

Figure 3.3 *iADCS100* unit characteristics and performances

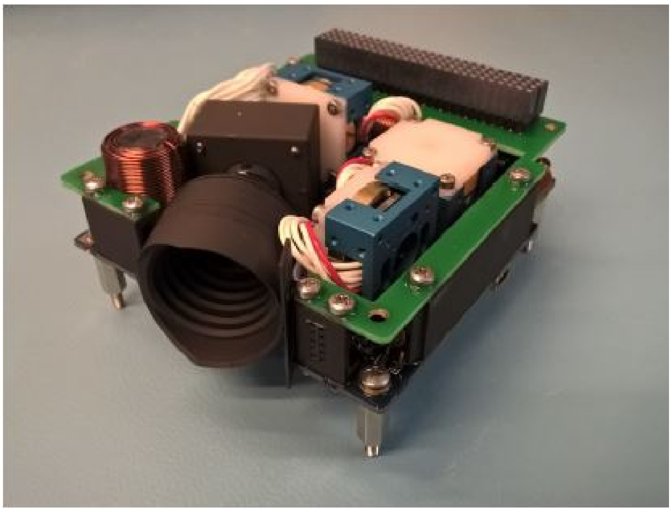


Figure 3.4 *iADCS100* unit

Modes definition The modes of the mission can be summarized in figure 3.5.

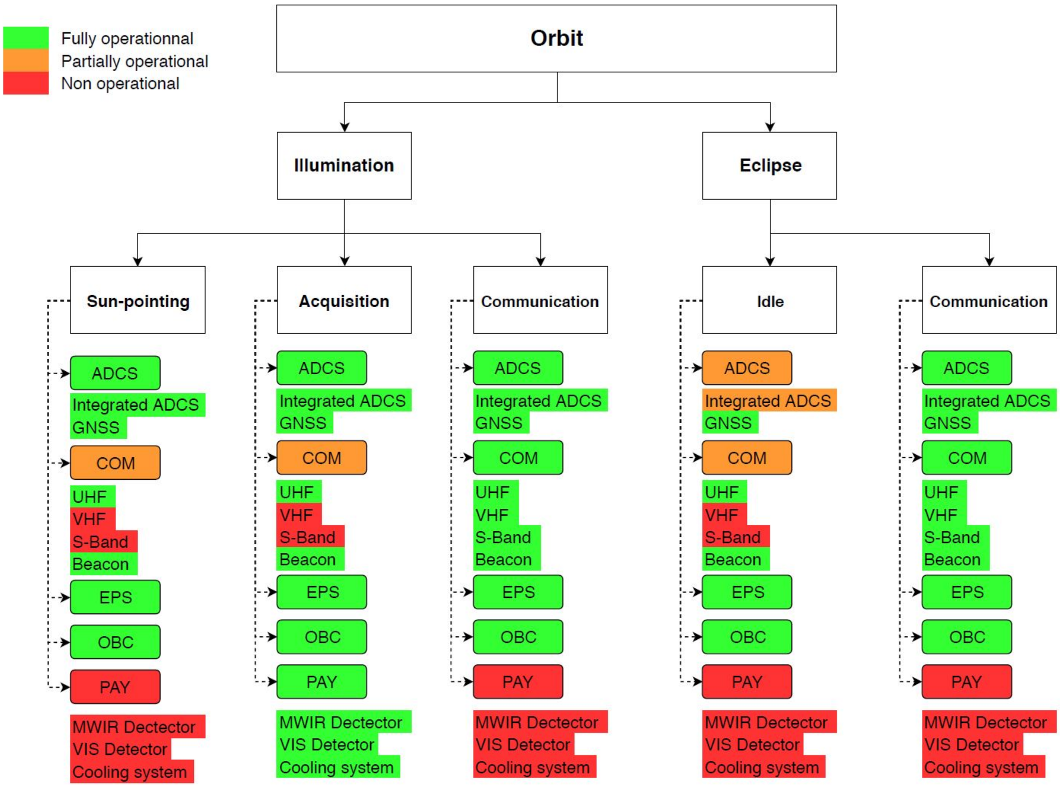


Figure 3.5 Mission modes

Here are also reported important results from the other subsystems, relevant from the ADCS point of view: *Mission Analysis, Electric Power Generation, Telemetry & Telecommunication, Configuration* and *Payload*. These results were obtained in [6].

3.1 Mission Analysis

The most important design choice has been to shoot pictures while being over the area of interest between 12:00 and 14:00 in solar time. It is also known as the Local Mean Time (LMT).

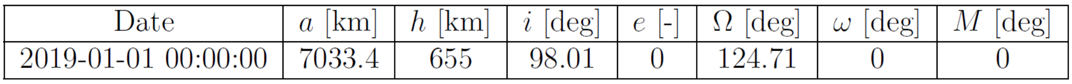
In order to satisfy this constraint, two orbits have been investigated: ISS (400km) and SSO (650km). Each orbit will be studied with its relative attitude.

ISS orbit The orbital parameters are provided in table 3.1.

Table 3.1 ISS orbital parameters

Semi-major axis (km)	about 400
Eccentricity (-)	0.0003830
Inclination (deg)	51.6416
RAAN (deg)	321, 9140 or user defined
Argument of perigee (deg)	239.4439 or user defined
True anomaly (deg)	user defined

The ground tracks repetition is about 59.5 days, meaning that the Local Mean Time LMT repeats at the equator each 59.5 days. It is remarkable that for 1-3 days the satellite will see mostly oceans in the southern hemisphere, independently from epoch, RAAN, argument of perigee and true anomaly, since the passages are cyclic. The lifetime on this orbit is about 5.5 months, [6]. Similar results have been obtained also for Cygnus Cargo orbit at 500 km altitude. Nevertheless this solution can increase the lifetime of about 2 years.



Date	a [km]	h [km]	i [deg]	e [-]	Ω [deg]	ω [deg]	M [deg]
2019-01-01 00:00:00	7033.4	655	98.01	0	124.71	0	0

Figure 3.6 SSO (655 km) with $\tau_{AN} = 13:30$. Epoch: January 1, 2019, [6].

SSO orbit First of all it must be stated that the orbit will not be maintained Sun-Synchronous, since no propulsion is provided on *OUFTI-Next*. Then, in order to define a Sun-Synchronous orbit, it is fundamental to provide the LMT at the ascending node, τ_{AN}, since it will remain constant due to the orbit properties. After some analysis, [6], it has been decided to have a $\tau_{AN} = 13:30$, in order to provide between 12:00 and 14:00 LMT a quite wide range of altitudes, figure 3.7 on the facing page.

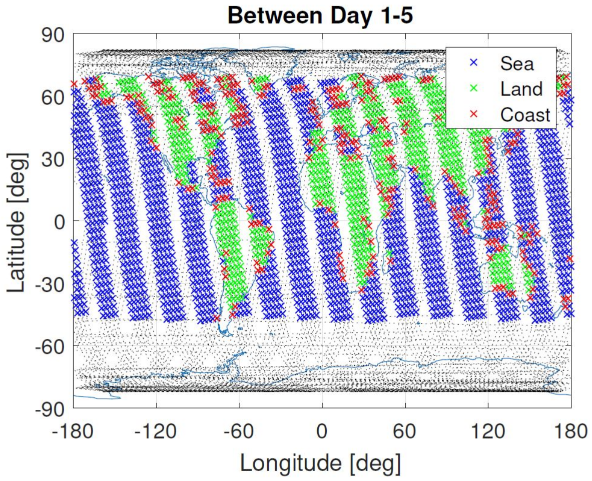


Figure 3.7 SSO (600 km) with $\tau_{AN} = 13:30$. Epoch: January 1, 2019, [6].

It seems that an even higher in altitude SSO is the best solution for *OUFTI-Next* mission, but the problem is the necessity to de-orbit without propulsion, since the lifetime will be longer than 25 years, as requested by ECSS. Moreover it is important to notice that, if images are acquired close to 12:00 LMT, sun glint could be a problem.
As stated before, all the three orbits, will be analysed from the point of view of the ADCS subsystem. Anyway some important conclusions can be listed:

- On the one hand, ISS orbit has too long and variable (due to drag) repeated ground tracks and, as a result, it cannot be exploited for smart irrigation experiments. On the other hand it provides a lifetime compatible with the ECSS requirements.

- SSO orbits provide instead almost constant (low perturbation level) recursions of even 10 days, but the disposal phase remains irresolvable without propulsion.

3.2 Electric Power Generation

From EPS point of view, a preliminary analysis has been carried out, considering only 1 lateral surface covered by solar panels. This choice is not the best; the optimal configuration is the following: "2 × 3U panels on a long face as well as on 1U panel on +Z. Indeed, this configuration can increase the power generated up to 10 W, but the satellite needs to be oriented correctly to the sun. The optimum is a rotation of -13° around X or Y to see the top face and 45° around Z to present the corner between two long faces", [6]. In order to understand better, a useful image from [6] can be used, showing *OUFTI-Next* from the point of view of the Sun, figure 3.8 on the next page.

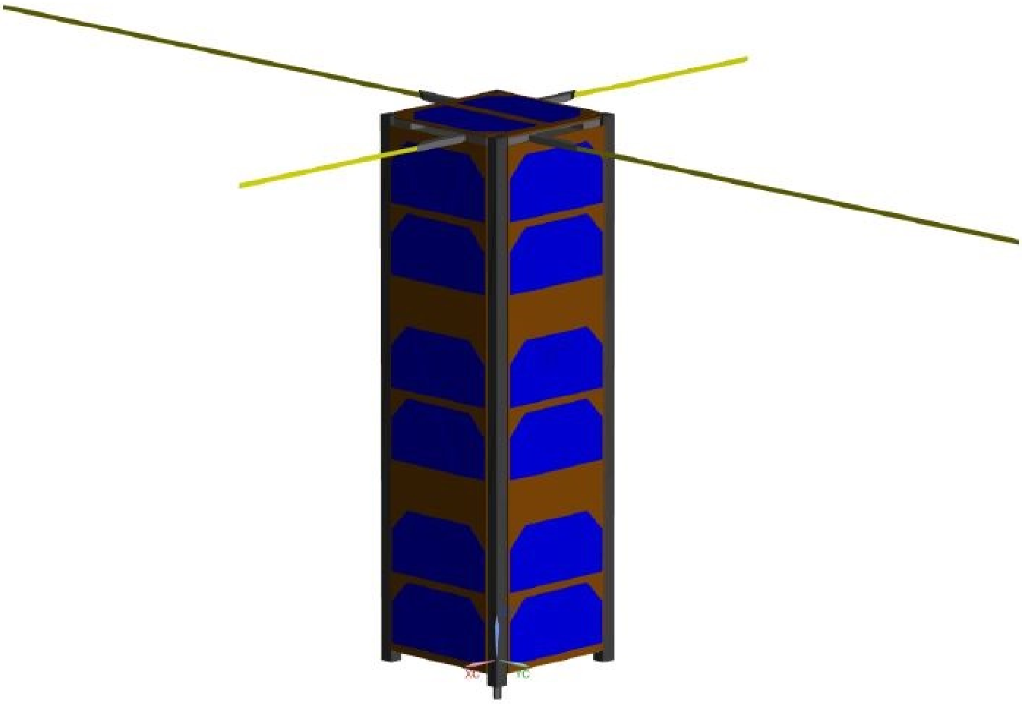
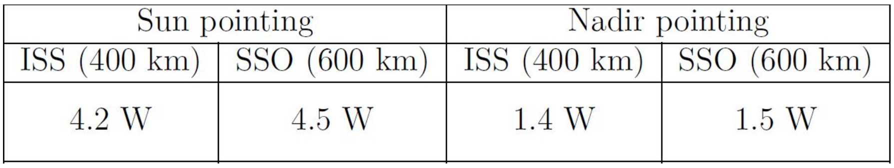


Figure 3.8 Optimum orientation of the platform for the EPS subsystem. View from the sun to the satellite. The reference frame indicated is different from the one used in this Thesis.

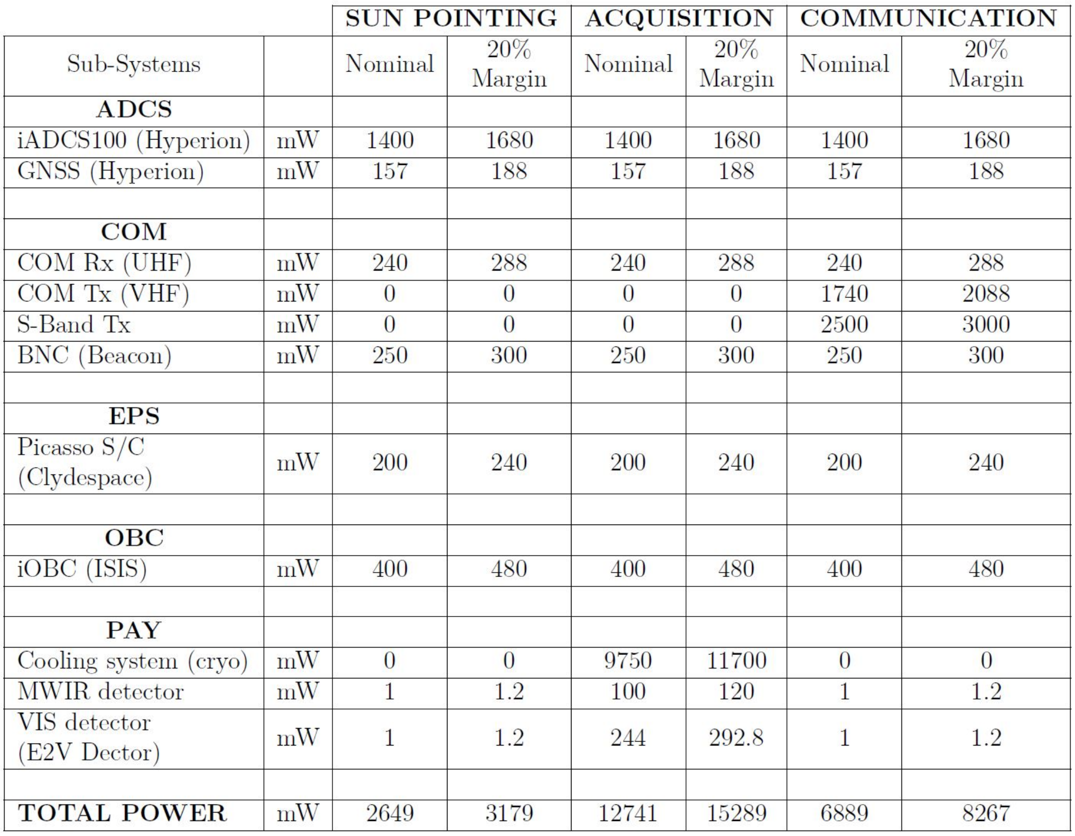
The problem is that Sun-sensors integrated inside the solar panels are quite course: ≈ 10 deg. Some companies integrates finer Sun-sensors inside the ADCS unit, but this is not the case for *iADCS100*. Since the ADCS unit has not been selected yet, both the 1 X 3U and 2 X 3U + 1 X 1U strategies will be studied. In figure 3.9, the power budget in the case of 1 X 3U strategy is presented, [6]:



Sun pointing		Nadir pointing	
ISS (400 km)	SSO (600 km)	ISS (400 km)	SSO (600 km)
4.2 W	4.5 W	1.4 W	1.5 W

Figure 3.9 1 X 3U power budget.

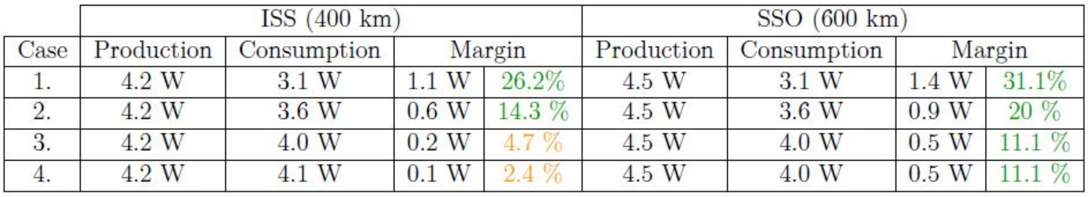
These data have been compared with the power consumption levels, figure 3.10 on the next page.



		SUN POINTING		ACQUISITION		COMMUNICATION	
Sub-Systems		Nominal	20% Margin	Nominal	20% Margin	Nominal	20% Margin
ADCS							
iADCS100 (Hyperion)	mW	1400	1680	1400	1680	1400	1680
GNSS (Hyperion)	mW	157	188	157	188	157	188
COM							
COM Rx (UHF)	mW	240	288	240	288	240	288
COM Tx (VHF)	mW	0	0	0	0	1740	2088
S-Band Tx	mW	0	0	0	0	2500	3000
BNC (Beacon)	mW	250	300	250	300	250	300
EPS							
Picasso S/C (Clydespace)	mW	200	240	200	240	200	240
OBC							
iOBC (ISIS)	mW	400	480	400	480	400	480
PAY							
Cooling system (cryo)	mW	0	0	9750	11700	0	0
MWIR detector	mW	1	1.2	100	120	1	1.2
VIS detector (E2V Dector)	mW	1	1.2	244	292.8	1	1.2
TOTAL POWER	mW	2649	3179	12741	15289	6889	8267

Figure 3.10 Power consumption levels.

4 scenarios are reported in figure 3.11: case 1 is not an interesting one since nothing happens in contrary to case 2. For this one, the margin is almost 15 %. Moreover, a 20 % margin was already considered for the consumption. It means that one image per orbit is clearly feasible. One orbit can be fully dedicated to download data but it can also be done on the same orbit (case 3).

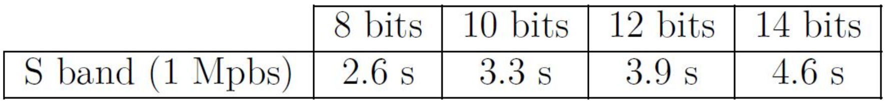


	ISS (400 km)				SSO (600 km)			
Case	Production	Consumption	Margin		Production	Consumption	Margin	
1.	4.2 W	3.1 W	1.1 W	26.2%	4.5 W	3.1 W	1.4 W	31.1%
2.	4.2 W	3.6 W	0.6 W	14.3 %	4.5 W	3.6 W	0.9 W	20 %
3.	4.2 W	4.0 W	0.2 W	4.7 %	4.5 W	4.0 W	0.5 W	11.1 %
4.	4.2 W	4.1 W	0.1 W	2.4 %	4.5 W	4.0 W	0.5 W	11.1 %

Figure 3.11 Power budget (cryocooler) on two orbits (mean power and 20 % margin on the consumption). Two orbits considered: SSO at 600 km with $\tau_{AN} = 13:30$ and the ISS orbit. Scenarios: 1: Sun pointing during full illumination and idle in eclipse. Scenario 2: Sun pointing + acquisition (4 min) during full illumination and idle in eclipse. Scenario 3: Sun pointing + acquisition (4 min) + communication (8 min) during full illumination and idle in eclipse. Scenario 4: Sun pointing + acquisition (4 min) during full illumination and idle + communication (8 min) in eclipse, [6].

3.3 Telemetry & Telecommunication subsystem

From the TMTC equipment point of view, there is a S-band antenna body mounted on one of the lateral surfaces not covered by solar panels. Being the platform symmetric, it has been decided to place the antenna on the -Z face. It has been demonstrated that the visibility time during a passage above Liege is on the average (among the 3 orbits studied) about 6 min, meaning that with a simple nadir pointing with the -Z face we can download more than 1 image per passage, being the time requested to download an image, [6]:



	8 bits	10 bits	12 bits	14 bits
S band (1 Mpbs)	2.6 s	3.3 s	3.9 s	4.6 s

Figure 3.12 Time to download 1 image in S-band, depending on the bit depth.

In order to receive the signal, a passive antenna has been considered to be sufficient [6].

3.4 Configuration

The CubeSat is made by 3 standard units, each of them is typically 10 cm x 10 cm x 10 cm. Nevertheless a 3U CubeSat is usually a little bit bigger; for simplicity the overall volume will be kept 10 cm x 10 cm x 30 cm.

Table 3.2 ADCS subsystem inputs

Mass (kg)	2-4
Platform dimension (CubeSat Units)	3
Payload volume (CubeSat Units)	1.5

A CAD drawing is presented in figure 3.13 .

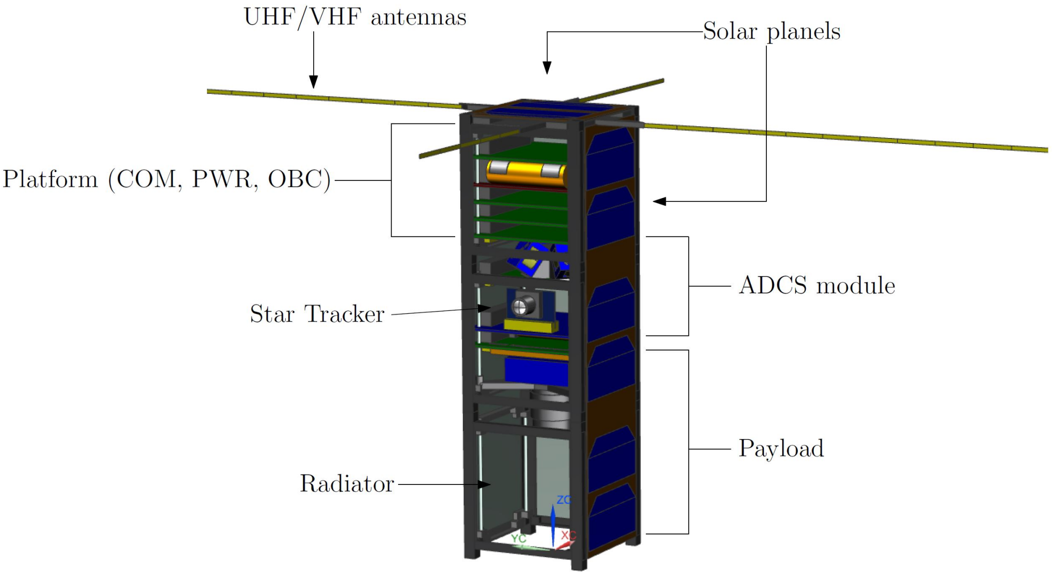
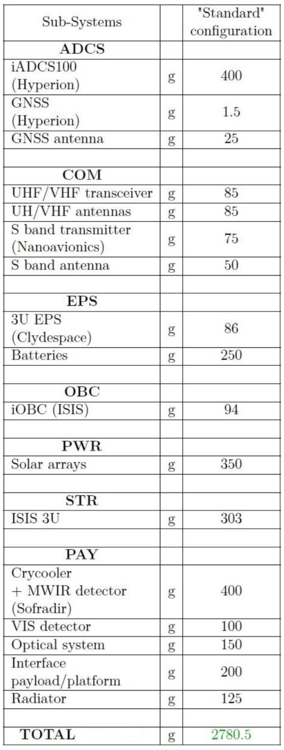


Figure 3.13 OUFTI-NEXT CAD model. The reference frame indicated is different from the one used in this Thesis.

The center of mass is estimated to be +1.6 cm from the geometric center of the CubeSat. A preliminary mass budget is presented in figure 3.14 on the facing page.



Sub-Systems		"Standard" configuration
ADCS		
iADCS100 (Hyperion)	g	400
GNSS (Hyperion)	g	1.5
GNSS antenna	g	25
COM		
UHF/VHF transceiver	g	85
UH/VHF antennas	g	85
S band transmitter (Nanoavionics)	g	75
S band antenna	g	50
EPS		
3U EPS (Clydespace)	g	86
Batteries	g	250
OBC		
iOBC (ISIS)	g	94
PWR		
Solar arrays	g	350
STR		
ISIS 3U	g	303
PAY		
Crycooler + MWIR detector (Sofradir)	g	400
VIS detector	g	100
Optical system	g	150
Interface payload/platform	g	200
Radiator	g	125
TOTAL	g	2780.5

Figure 3.14 Preliminary mass budget

3.5 Payload

The payload is an Infrared camera, seen as a "black box" of 1.5U with an output oriented in the -X face and located on the bottom of the platform. The remaining part of the satellite is dedicated to the other subsystems. ADCS unit will be located almost in the middle of the platform.

From the point of view of the pointing budget, in [6], a preliminary estimation has been carried out, table 3.3, based on the work of other two master thesis, [15] and [3].

Table 3.3 Preliminary pointing budget

Accuracy	0.1 deg (6 arcmin)
Stability	0.01 deg (0.5 arcmin)

Chapter 4

Ephemeris Generation

The simulations of OUFTI-Next are provided with a model considering the orbit and attitude dynamics decoupled. In further developments of this Thesis for the ZodiArt mission, a fully coupled model will be implemented, exploiting a unified states model.
In order to obtain the ephemeris of OUFTI, an orbital propagator has been implemented in Matlab™. It is based on the restricted 2-body problem perturbed equation:

$$\frac{d^2\mathbf{r}}{dt} = -\frac{\mu}{r^3}\mathbf{r} + \mathbf{p} \tag{4.1}$$

where $\mathbf{r}$ is the position vector, μ the Earth planetary constant and $\mathbf{p}$ the perturbation vector. This last vector has been obtained considering the perturbations induced by:

- Earth oblateness.
- Earth zonal harmonics.
- Sun and Moon third body perturbations.
- Geomagnetic field.
- Solar radiation pressure.
- Atmospheric drag.

considering the spacecraft as a point mass with the maximum cross-area and the actual mass of OUFTI-Next.

All the previous perturbations have been implemented following the equations reported in [5]. The Matlab class is able to compute both the ephemerides and the osculating orbital elements through Gauss planetary equations, once the initial conditions and $\mathbf{p}$ are provided, working in parallel on 4 physical CPU cores:

$$\begin{cases} \frac{dh}{dt} = rp_s \\ \frac{de}{dt} = \frac{h}{\mu}sin\theta p_r + \frac{1}{\mu h}[(h^2 + \mu r)cos\theta + \mu er]p_s \\ \frac{d\theta}{dt} = \frac{h}{r^2} + \frac{1}{eh}[\frac{h^2}{\mu}cos\theta p_r - (r + \frac{h^2}{\mu})sin\theta p_s] \\ \frac{d\Omega}{dt} = \frac{r}{hsin\ i}sin(\omega + \theta)p_w \\ \frac{di}{dt} = \frac{r}{h}cos(\omega + \theta)p_w \\ \frac{d\omega}{dt} = -\frac{1}{eh}[\frac{h^2}{\mu}cos\theta p_r - (r + \frac{h^2}{\mu})sin\theta p_s] - \frac{rsin(\omega+\theta)}{htan\ i}p_w \end{cases} \tag{4.2}$$

where:

- $r = \frac{h^2}{\mu(1+ecos\theta)}$
- h: Angular momentum.
- e: Eccentricity.
- Ω: Right ascension of the ascending node.
- i: Orbit inclination.
- ω: Anomaly of the perigee.
- θ: True anomaly.

In order to exploit this set of equations, vector **p** must be decomposed along the non-inertial LVLH frame.

Here the ISS and SSO orbits are represented respectively after 1 day, 10 days and 1 month, without station-keeping.

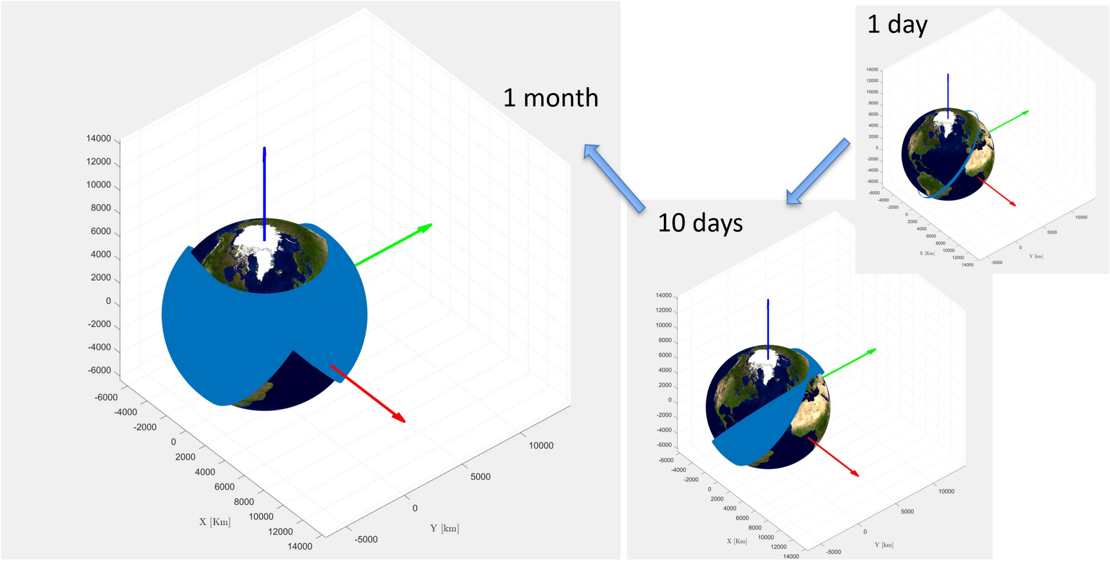


Figure 4.1 ISS orbit perturbed, initial epoch: 01/01/2019 at 12:00:00 GMT/UTC+0.

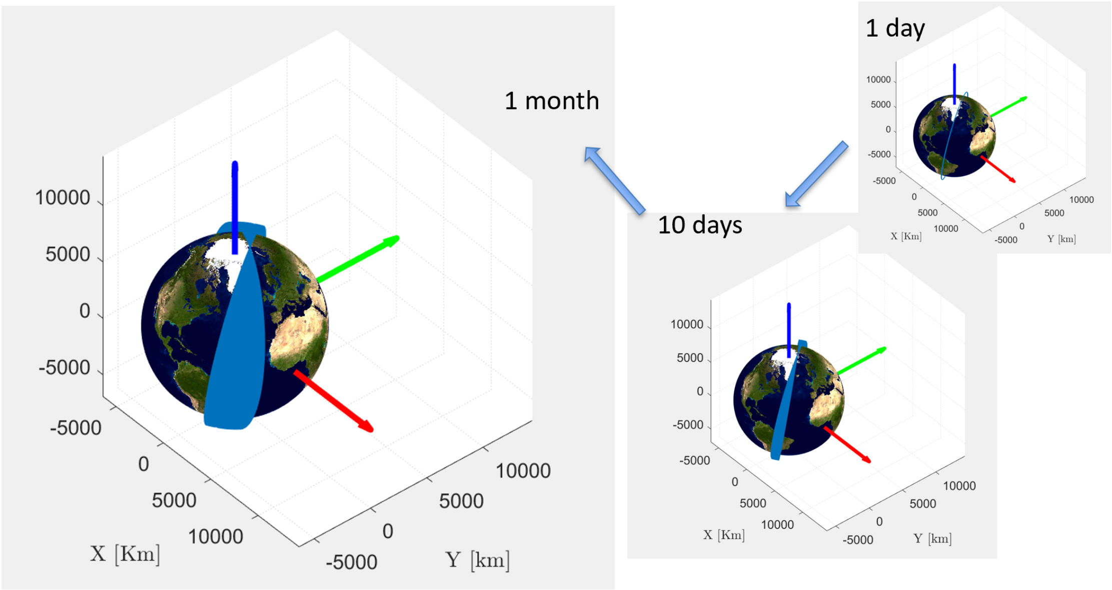


Figure 4.2 SSO orbit perturbed, initial epoch: 01/01/2019 at 12:00:00 GMT/UTC+0.

Chapter 5

Simulator

5.1 Choice of the software

One of the most used softwares for system control design is Simulink™ and for this reason, a preliminar and simplified analysis on the feasibility of some maneuvers, eventually implemented for this Thesis, has been performed with it. Nevertheless, Simulink™ is not the best option when more complex systems have to be coded and optimizations algorithms have to be run in parallel. As a result, it has been decided to use Matlab™. The only exception is the de-tumbling phase, for which it has been used a Simulink™ model, as it will be explained in the next sections.

In order to make easer to follow the development of the simulator, a Simulink™ model will be used to graphically help the reader. This model is a simplified version of the one implemented in Matlab™ and has been developed at the beginning of this work.

The ADCS loop shown in figure 5.1 is coded in the Simulink™ model represented in figure 5.2 on the next page.

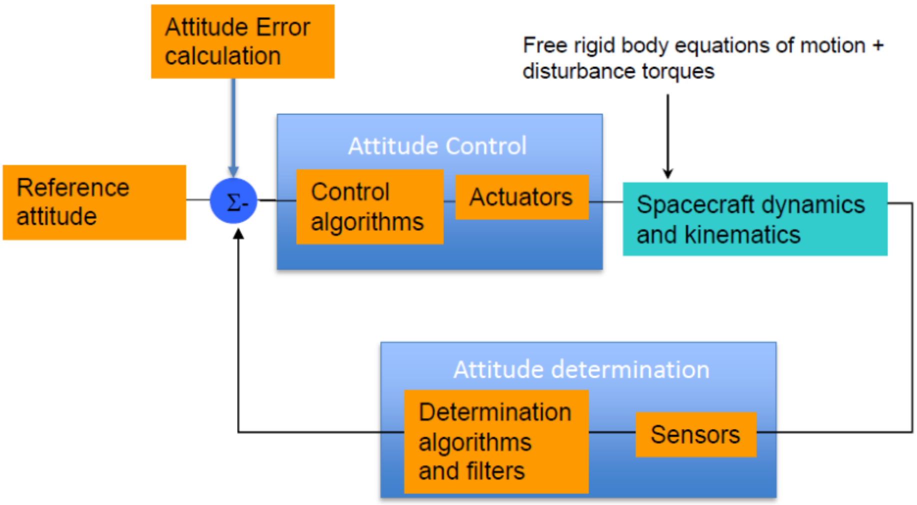


Figure 5.1 ADC subsystem feedback loop.

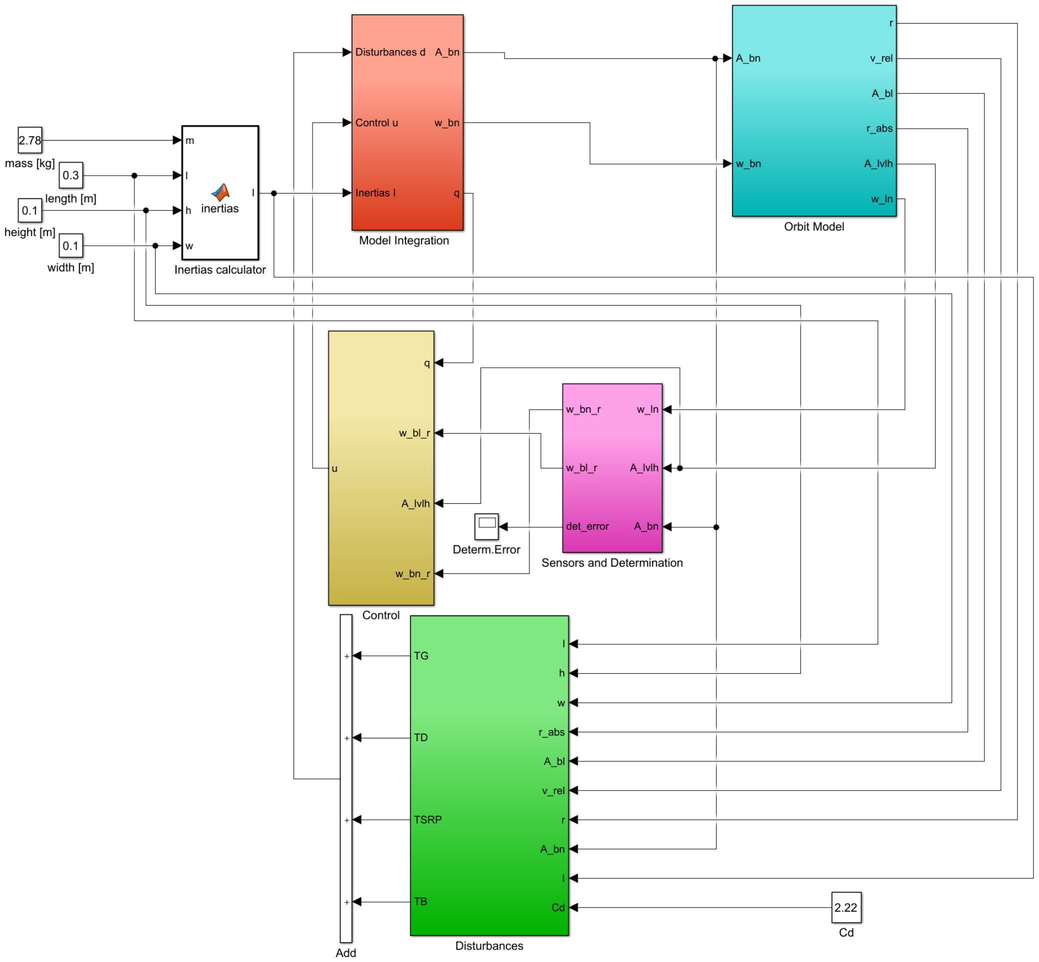


Figure 5.2 ADC subsystem coded in Simulink™.

This chapter will be subdivided in sections, describing each element of the loop and the logic behind their implementations.

5.2 ADCS main input data

The preliminary design of the ADC subsystem has been based on the *iADCS100* unit characteristics and then, only after having performed all the simulations with these data, a trade-off will be done with respect to the KUL unit. Here are reported the fundamental data obtained from *iADCS100* datasheet [10], and contacts with *Hyperion*, the company supplying this device:

- Maximum/minimum dipole moment of magnetic torquers: ± 0.4 Am^2.
- Maximum/minimum reaction wheels torque: ± 0.1 mNm.
- Maximum/minimum reaction wheels momentum storable: ± 6 mNms.
- Gyros drift ramp: ≈ 18.5 deg/h.
- Number of observable stars by the star tracker: 6.
- Star tracker accuracy: 30 arcsec.
- Star tracker update frequency: 5 Hz.

Then the remaining parameters were obtained from literature:

- ρ_a, the surface absorption coefficient, assumed ≈ 0.9203, [13].
- ρ_d, the surface diffusive coefficient, assumed ≈ 0.03, [13].
- ρ_s, the surface scattering coefficient, assumed ≈ 0.0727, [13].
- $\boldsymbol{m} = [5e^{-4},\ 5e^{-4},\ 5e^{-4}]$ Am^2, the residual dipole moment, from [19].

It is important to remark that the following relation must be respected when selecting the proper optical coefficients for the CubeSats surfaces:

$$\rho_a + \rho_s + \rho_d = 1 \tag{5.1}$$

In the case of OUFTI-Next, it has been assumed to have the coefficient referred to the solar panels also referred to all the faces, since the longest mission phases are the Sun-pointing ones and the values are easier to be found, since the optical properties of the surfaces without solar panels are not known yet.

5.3 Kinematics

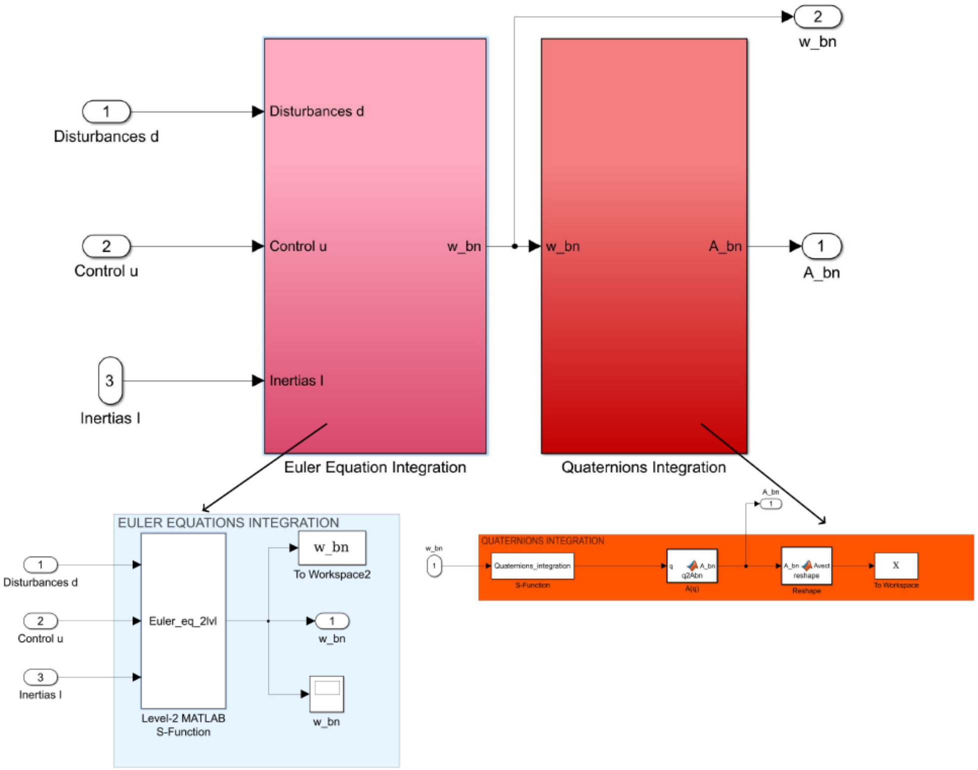


Figure 5.3 Kinematics blocks in Simulink™.

The dynamics has been modeled using Euler equations:

$$\begin{cases} \dot{\omega}_1 = \frac{I_2 - I_3}{I_1}\omega_2\omega_3 + \frac{u_1 + d_1}{I_1} \\ \\ \dot{\omega}_2 = \frac{I_3 - I_1}{I_2}\omega_1\omega_3 + \frac{u_2 + d_1}{I_2} \\ \\ \dot{\omega}_3 = \frac{I_1 - I_2}{I_3}\omega_2\omega_1 + \frac{u_3 + d_1}{I_3} \end{cases} \tag{5.2}$$

where I_i are the inertias of the spacecraft, u_i the components of torque control vector and d_i the components of the disturbances torques. From the angular velocities, the 4 components of the quaternion representing the attitude of the spacecraft are obtained integrating:

$$\dot{\boldsymbol{q}} = \frac{1}{2}\begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix}\boldsymbol{q} \tag{5.3}$$

The initial velocities depend on the mission phase:

- De-tumbling will be initiated with random initial conditions distribution centered in 0.5 rad/s
- If a simulation starts with a phase different from de-tumbling (like during nominal phases of the mission after having de-tumbled the spacecraft), it will be initialized with random initial condition distribution centered in $1e^{-3}$ rad/s.
- All the other phases will start after a previous one and, as a result, their initial conditions will be the final state of the previous phase.

Once the Euler equations have been integrated, the angular velocities in body frame, $\boldsymbol{\omega_{b/n}}$, are obtained and used to integrate the quaternion with the following initial condition: $\boldsymbol{q_0} = [1/\sqrt{3},\ 1/\sqrt{3},\ 1/\sqrt{3},\ 0]$ (unit quaternion). The quaternion obtained is then transformed in the body-fixed frame representing the actual attitude, $\boldsymbol{A_{b/n}}$.

5.4 Orbital model

At this stage of the Thesis development, the orbital motion is fully decoupled from the attitude dynamics. This means that the model will exploit a spline interpolation of the OUFTI ephemeris and Sun-satellite vector computed before the simulation. *Spline* interpolation has been preferred to the *linear* one, since it avoids discontinuities in interpolated vectors, which would affect strongly the numerical integration. In this block the following quantities are evaluated:

- Position vector, through interpolation.
- Velocity vector, through interpolation.
- Sun-satellite vector, through interpolation.

then the angular momentum vector is obtained through:

$$\mathbf{h} = \mathbf{r} \times \mathbf{v} \tag{5.4}$$

After having obtained this quantities, the reference body-fixed frames are evaluated depending on the mission phase. These matrices represent the desired attitude and will be discussed in the sections dedicated to each phase. Each raw of these matrices is a desired direction of the body-fixed frame and since they are referred to the LVLH frame, they are indicated as $\boldsymbol{A_{LVLH}}$.

5.5 Disturbances model

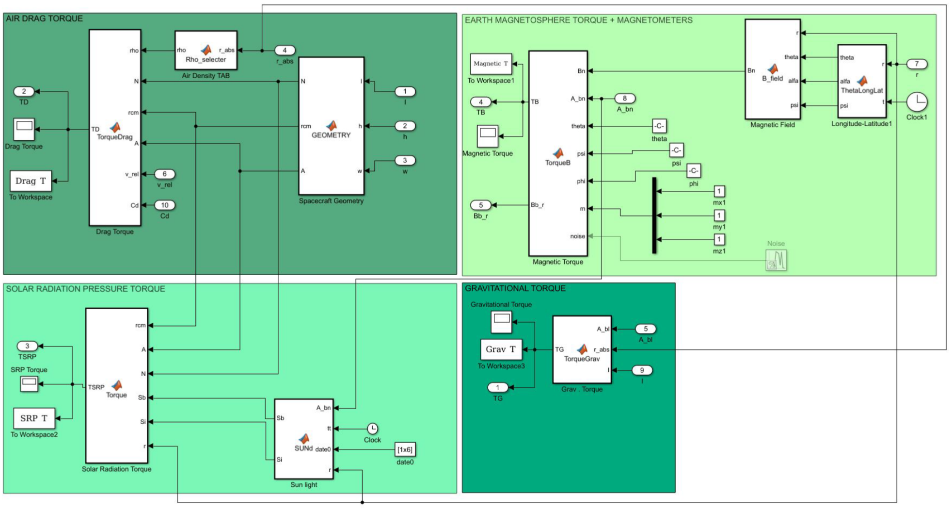


Figure 5.4 Disturbances blocks in Simulink™.

The disturbances are torque vectors, $\boldsymbol{d}$, that will be inserted in the 3D rigid body dynamics of the spacecraft. The following sections will analyze in detail the 4 main sources of disturbances: *atmospheric drag*, *solar radiation pressure*, *geomagnetic field interaction* and *gravity gradient*.

5.5.1 Atmospheric drag

The atmospheric drag is:

$$\boldsymbol{F_{drag}} = -\frac{1}{2}\rho(h)C_d A_{cross} v_{rel}^2 \frac{\boldsymbol{v_{rel}}}{||\boldsymbol{v_{rel}}||} \tag{5.5}$$

where:

- $\rho(h)$ is the density model dependent on the altitude h.
- $\boldsymbol{v_{rel}}$ is the relative velocity between the CubeSat and the Earth (in body frame).
- C_d is the drag coefficient (here assumed to be 2.22).
- A_{cross} are the cross sectional areas perpendicular to $\boldsymbol{v_{rel}}$.

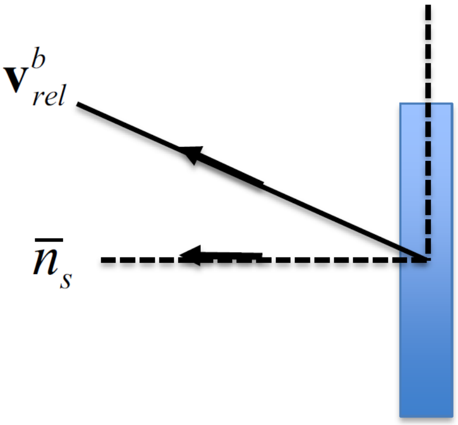


Figure 5.5 Spacecraft surface with respect to relative velocity.

The model used for atmospheric density is a simple exponential model:

$$\rho = \rho_0 exp\left[-\frac{h-h_0}{H}\right] \tag{5.6}$$

where:

- ρ_0 is reference density.
- h_0 is the reference altitude.
- H is the scale height

H varies from 7.25km at the Earth's surface to 268 km at 1000 km altitude. H is the altitude over which the density changes by the value of $e = 2.72$. The reference density varies from 1.225 kg/m^3 at the surface to $3.019 \times 1e{-}15$ kg/m^3 at 1000 km altitude [20]. The source for the previous parameters, depending on the height, is Wertz, 1978, 820, which uses the *U.S. Standard Atmosphere*(1976) for 0 km, CIRA-72 for $25 - 500$ km and CIRA-72 with $T_{\text{inf}} = 1000$ K for $500 - 1000$ km. The scale heights have been adjusted to maintain a piecewise-continuous formulation of the density.

In order to obtain the overall torque vector acting on the CubeSat, the force vector is multiplied by the arm between the point where the force is applied and the assumed center of mass, $\boldsymbol{r}$. Since its position is uncertain or dependent on time, a 10% error has been considered. Obviously, if the product between the surface's normal and the relative velocity is less than 0, the resulting torque is null. In index form:

$$\boldsymbol{T_{drag}} = \begin{cases} -\frac{1}{2}\rho(h)C_d v_{rel}^2 \frac{\boldsymbol{v_{rel}}}{||\boldsymbol{v_{rel}}||} \sum\limits_{i=1}^{n} \boldsymbol{r_i} \times (\boldsymbol{n_i} \frac{\boldsymbol{v_{rel}}}{||\boldsymbol{v_{rel}}||})A_i & (\boldsymbol{n_i} \frac{\boldsymbol{v_{rel}}}{||\boldsymbol{v_{rel}}||}) > 0 \\ 0 & (\boldsymbol{n_i} \frac{\boldsymbol{v_{rel}}}{||\boldsymbol{v_{rel}}||}) < 0 \end{cases} \tag{5.7}$$

5.5.2 Solar radiation pressure

The solar radiation pressure can be modeled in a similar way to the drag torque, starting from the force expression, [2]:

$$\boldsymbol{F_{SRP}} = P\boldsymbol{A_{cross}}\left[\rho_a(\boldsymbol{S}\cdot\boldsymbol{N}) + 2\rho_s(\boldsymbol{S}\cdot\boldsymbol{N})^2\cdot\boldsymbol{N} + \frac{2}{3}\rho_d(\boldsymbol{S}\cdot\boldsymbol{N})\right] \tag{5.8}$$

where:

- P is the average pressure due to radiation, $P = 4.563 \times 10^{-6}$ N/m^2.
- $\boldsymbol{S}$ is the unit vector pointing from the Sun to the surface as shown in figure 5.6.
- $\boldsymbol{N}$ is the surface normal vector.
- ρ_a is the surface absorption coefficient, assumed ≈ 0.9203, [13].
- ρ_d is the surface diffusive coefficient, assumed ≈ 0.03, [13].
- ρ_s is the surface scattering coefficient, assumed ≈ 0.0727, [13].

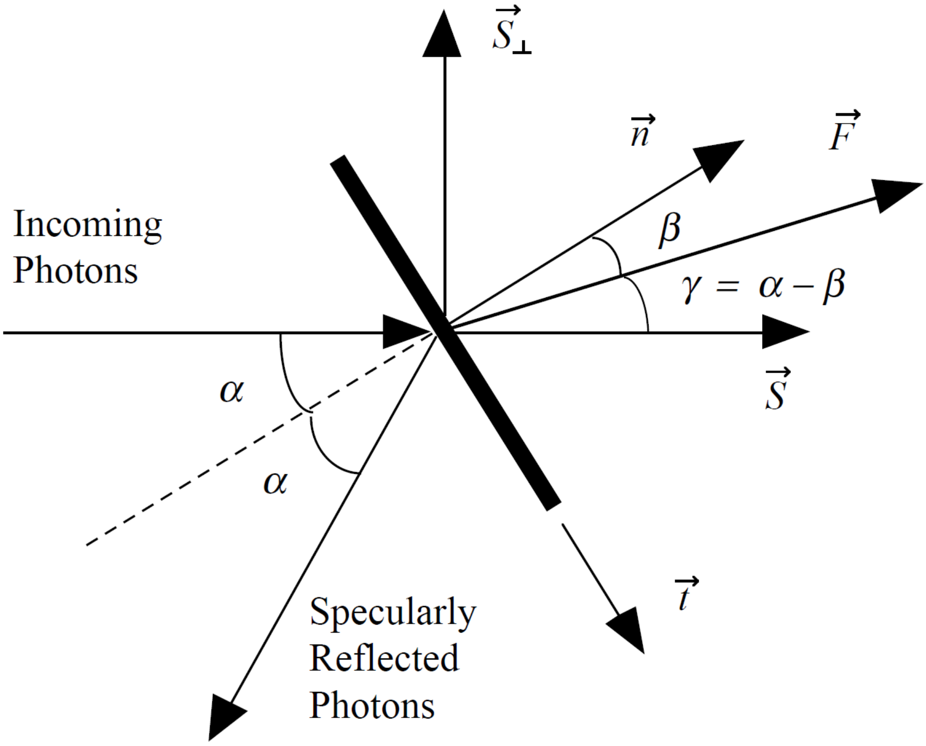


Figure 5.6 Incoming photons impacting on a flat surface.

As for the drag, in order to obtain the overall torque, a multiplication times the arm is required and, as before, a 10% error has been considered in the position of the centers of pressure on OUFTI-Next surfaces:

$$\boldsymbol{T_{drag}} = \begin{cases} \sum_{i=1}^{n} \boldsymbol{r_{cp,\ i}} \times \boldsymbol{T_{drag_i}} & (\boldsymbol{S_b} \times \boldsymbol{N}) > 0 \\ 0 & (\boldsymbol{S_b} \times \boldsymbol{N}) < 0 \end{cases} \tag{5.9}$$

As we can notice from the formula above, if the Sun-Earth vector projection on the surface normal is negative, there is no SRP torque.

In order to simulate more precisely this effect, an eclipse model has been implemented. The algorithm was proposed by Vallado in [20]:

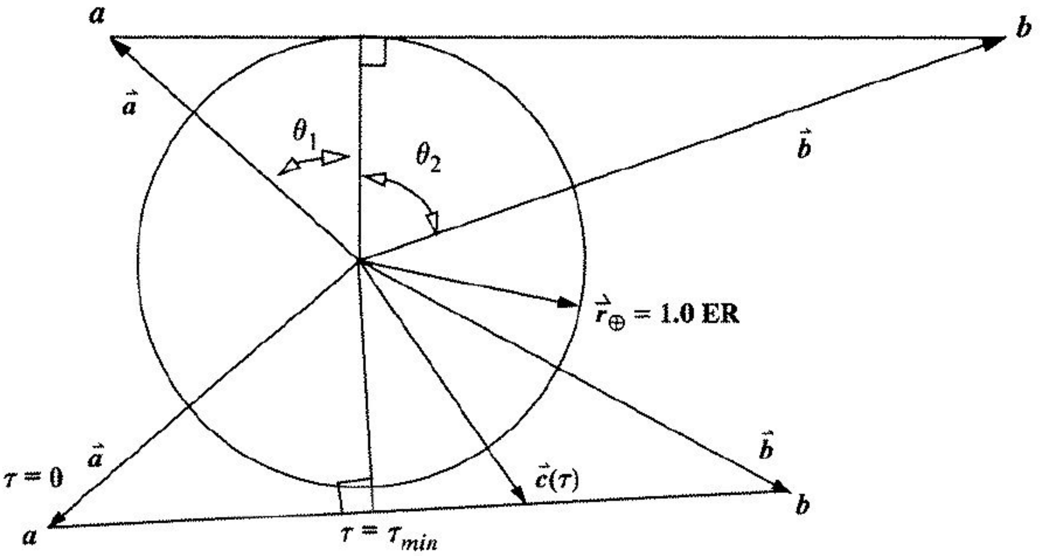


Figure 5.7 Vallado's light-shadow model

All the vectors are normalized with respect to the Earth radius and a and b represent respectively the Sun vector[20](assumed mission day-one: January the 1^{st} 2019 at 12:00 A.M.) and the CubeSat position vector. $\boldsymbol{c}$, instead, can be obtained as function of τ_{min} as follows:

$$|\boldsymbol{c}(\tau_{min})|^2 = (1 - \tau_{min})|\boldsymbol{a}|^2 + (\boldsymbol{a} \cdot \boldsymbol{b})\tau_{min} \tag{5.10}$$

and

$$\tau_{min} = \frac{\boldsymbol{a} \cdot (\boldsymbol{a} - \boldsymbol{b})}{|\boldsymbol{a} - \boldsymbol{b}|} \tag{5.11}$$

In this way, once found τ_{min}, "if $\tau_{min} < 0$ or $\tau_{min} > 0$, there is line-of-sight between the two vectors. From fig.5.7, this is reasonable because both vectors are in the same quadrant with respect to the attracting body. If τ_{min} falls between 0 and 1, substitute its value into eq.5.10 to determine the square of the magnitude at the minimum location. If this squared magnitude is larger than (or equal) to 1, line-of-sight exists. In all remaining cases, it does not."[20]

5.5.3 Geomagnetic field

The Earth magnetic field has been modeled with the Matlab™ build-in function *wrldmagm.m*, [23], which evaluates the Earth magnetic field components in the years between 2015 and 2020, requiring as inputs:

- Height in meters.
- A scalar geodetic latitude, in degrees, where north latitude is positive, and south latitude is negative.
- A scalar geodetic longitude, in degrees, where east longitude is positive, and west longitude is negative.
- A scalar decimal year. Decimal year is the desired year in a decimal format to include any fraction of the year that has already passed.

The function output, once obtained in *inertial reference frame*, is then transformed in body-frame coordinates through:

$$\boldsymbol{B_b} = \boldsymbol{A_{b/n}} \boldsymbol{B_n} \tag{5.12}$$

Finally the magnetic torque can be computed, considering as CubeSat internal induction, $\boldsymbol{m}$:

$$\boldsymbol{T_{magnetic}} = \boldsymbol{m} \times \boldsymbol{B_b} \tag{5.13}$$

It is important to underline the limitations of this model, as reported in [23]: "The WMM specification produces data that is reliable five years after the epoch of the model, which begins January 1 of the model year selected. The WMM specification describes only the long-wavelength spatial magnetic fluctuations due to the Earth's core. Intermediate and short-wavelength fluctuations, contributed from the crustal field (the mantle and crust), are not included. Also, the substantial fluctuations of the geomagnetic field, which occur constantly during magnetic storms and almost constantly in the disturbance field (auroral zones), are not included."

In order to reduce the computational time of de-tumbling phase, an approximated Earth magnetic field model, based on the "Schmidt normalized associated Legendre Polynomial" has been implemented and it will be described in the section dedicated to de-tumbling phase.

5.5.4 Gravity gradient

The gravitational disturbance is modeled as follows:

$$\boldsymbol{T_{gravit}} = \frac{3\mu}{r^5} \int\limits_B (\boldsymbol{r_b} \cdot \boldsymbol{r})(\boldsymbol{r_b} \times \boldsymbol{r}) dm \tag{5.14}$$

where $\boldsymbol{r}$ is the CubeSat position vector and $\boldsymbol{r_b}$ is the vector connecting its center of mass to the infinitesimal cube of mass *dm*. In order to get an easier expression, the torque can be rearranged in this way:

$$\boldsymbol{T_{gravit}} = \int\limits_B \begin{bmatrix} (y^2 - z^2)c_2c_3 \\ (z^2 - x^2)c_1c_3 \\ (x^2 - y^2)c_1c_2 \end{bmatrix} dm = \frac{3\mu}{R^3} \begin{bmatrix} (I_3 - I_2)c_2c_3 \\ (I_1 - I_3)c_1c_3 \\ (I_2 - I_1)c_1c_2 \end{bmatrix} \tag{5.15}$$

where, being $\boldsymbol{\hat{r}}$ the normalized position vector:

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = nadir\ direction = - \begin{bmatrix} \hat{r}_x \\ \hat{r}_y \\ \hat{r}_z \end{bmatrix} \tag{5.16}$$

.

5.6 Sensors and determination

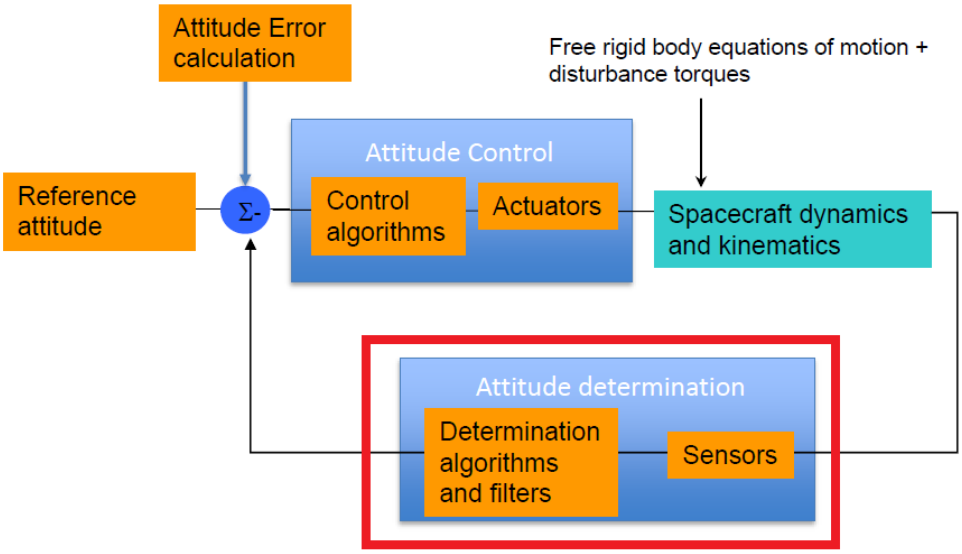


Figure 5.8 ADCS feedback loop with *sensors & determination* block highlighted.

The aim of the *sensors & determination* block is to introduce noises and measurement errors inside the integrated states, in order to simulate the sensors accuracy effect on the system and then reconstruct the state, thanks to the determination algorithm. In order to do that, first the models of sensors have been analyzed and then two different determination strategies have been adopted.

It is important to remark, since the beginning of this section, that in the final simulation no white noises nor sensors update frequencies have been simulated, since they would slow down the simulations too much. Indeed, they affects quite heavily the computational time, since the mission profiles to be simulated are characterized by long time intervals to be integrated. During the work, the presence of noises and update rates, has been implemented on simplified and shorter simulations, providing positive results in terms of accuracy of the determination even in presence of this kind of effects. In chapter 9 the simulation of the final and complete models for short time intervals, considering also this sources of measurements errors to show the effects on the determination, will be provided.

In this section the models of the main sensors (star tracker and gyros) are described, as well as the determination algorithms used.

5.6.1 Gyros

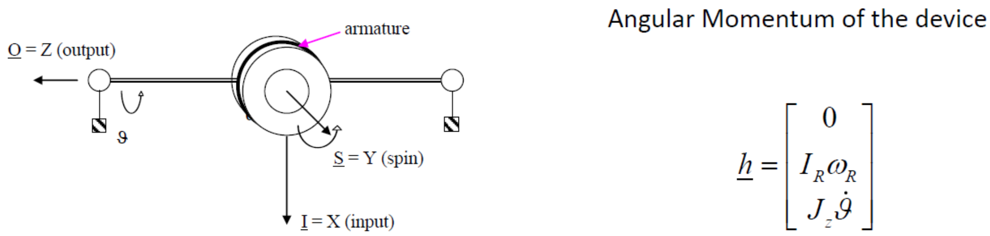


Figure 5.9 Gyros scheme.

In order to model the gyros, the angular velocities in body fixed frame, $\boldsymbol{\omega_{b/n}}$, should be perturbed by the accuracy matrix presented in eq.11.12, where the argument of trigonometric functions are the accuracy on each axis and by a dynamic Gaussian noise expressed as follows:

$$\boldsymbol{A_\epsilon} = \begin{bmatrix} cos(\psi)cos(\theta) & cos(\psi)sin(\theta)sin(\phi) + sin(\psi)cos(\phi) & -cos(\psi)sin(\theta)cos(\phi) + sin(\psi)sin(\phi) \\ -sin(\psi)cos(\theta) & -sin(\psi)sin(\theta)sin(\psi) + cos(\psi)cos(\theta) & sin(\psi)sin(\theta)cos(\psi) + cos(\psi)sin(\theta) \\ sin(\theta) & -cos(\theta)sin(\phi) & cos(\theta)cos(\phi) \end{bmatrix} \tag{5.17}$$

$$\boldsymbol{\omega_{b/n,real}} = \boldsymbol{A_\epsilon A_{b/n} w_{b/n}} + \boldsymbol{noise} \tag{5.18}$$

where:

$$\begin{cases} noise = n_i + n_u & \\ n_e = \sigma_\epsilon \zeta_e & Mesurement\ noise \\ n_u = n_u(t-\delta t) + \sigma_u\sqrt{\delta t}\zeta_u & Drift\ rate\ ramp - Float\ torque\ derivative\ noise \end{cases} \tag{5.19}$$

- ζ_i is a normally distributed random variable with 0 mean value and standard deviation 1.
- σ is the standard deviation (according to *Hyperion*, $\sigma_\epsilon \approx 10^{-4}$, while the drift rate ramp is equal to $\approx 9e^{-5}$ rad/s).
- δt is the sampling time (usually around $\frac{1}{400\ \text{Hz}} = 0.0025$ s).

The rate ramp of the gyros causes a continue decrease in accuracy with time, but this noise source is set to zero 5 times per second, when the star tracker provides an estimation of the state (as it will be reported later, the update frequency of the star tracker is 5 Hz).
According to *Hyperion*, the accuracy of the gyros is higher to the one characterizing the star tracker, which is the highest among sensors for ADCS, and for this reason no multiplication times $\boldsymbol{A_\epsilon}$ is requested. The model above for the noise, will be used in following dedicated simulations to test Gaussian noise effects on the complete model.

5.6.2 Star tracker

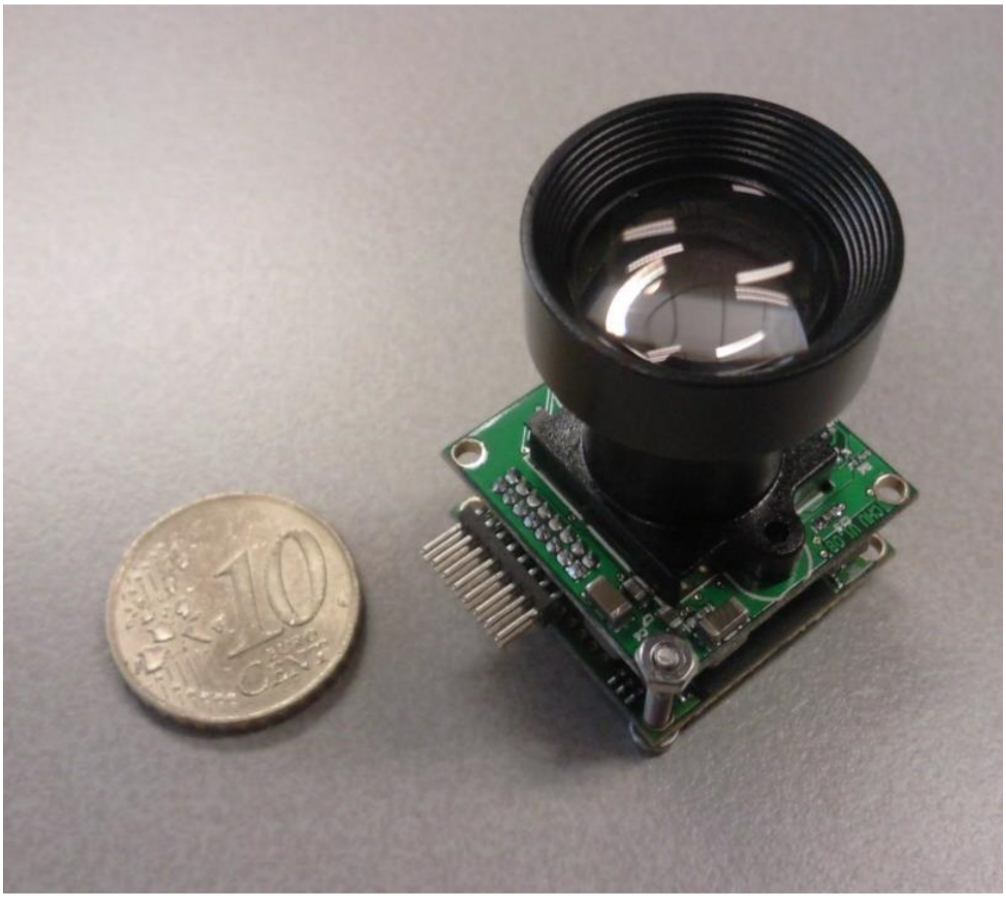


Figure 5.10 *ST-200* star tracker by *iADCS100*.

The star tracker used in *iADCS100* is the *ST-200*, figure 5.10, whose characteristics are the following:

- Number of observable stars: 6.
- Accuracy: 30 arcsec.
- Update frequency: 5 Hz.

Usually a star tracker is not equally accurate along each axis: it is less accurate around the optical axis (when the star field rotates) and it is more accurate along the two axes perpendicular to it (when the star field translates). Nevertheless, according to *Hyperion* flyer [10], the accuracy declared is equal along the 3 axes.

The star tracker is a sensor capable to obtain directly the attitude of the satellite, the matrix $\boldsymbol{A_{b/n}}$, through an algebric law, that will be presented afterwards. In order to model its output, $\boldsymbol{A_{b/n}}$, should be perturbed by the accuracy matrix presented in eq.11.12, where the argument of trigonometric functions are the accuracies on each axis (30 arcsec) and then, once transformed in quaternion, a random Gaussian noise will be added.

$$\boldsymbol{A_{b/n,real}} = \boldsymbol{A_{\epsilon}} \boldsymbol{A_{b/n}} \tag{5.20}$$

$$\boldsymbol{A_{b/n,real}} \rightarrow \boldsymbol{q} \tag{5.21}$$

$$q_{real} = q + noise \tag{5.22}$$

5.6.3 Kalman filter

Theory The Kalman filter is an algorithm capable of obtaining an estimation of the state, with the additional knowledge of the covariance matrix of the state error, P, representing the deviation with respect to the actual state, starting from noisy measurements. It is the best way to estimate the state starting from measurements subjected to Gaussian noise with 0 mean value, like the ones provided by spacecrafts' sensors and at the same time have an estimation of the determination error, through P. In this paragraph a short theoretical description is provided, based on [21].

The filter assumes that the state variables are random and with a Gaussian distribution, figure 5.11.

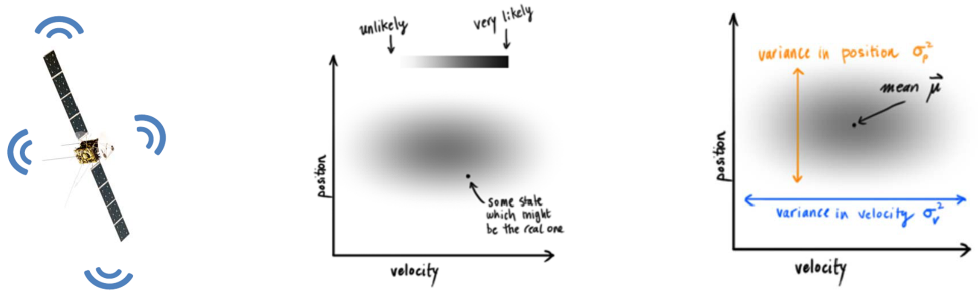
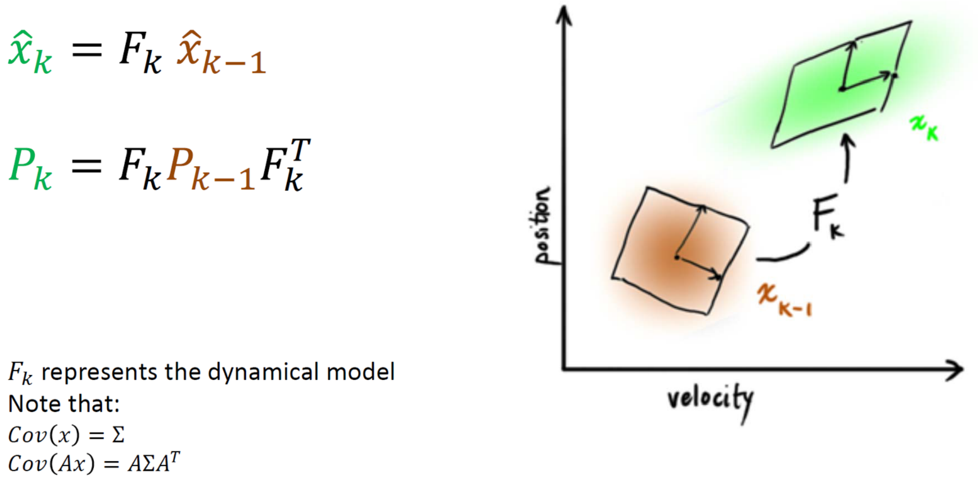


Figure 5.11 Kalman filter and the random state with Gaussian distribution, [21].

Then looking at the current state, $\hat{x}_{k-1}$, it predicts the next state, $\hat{x}_k$, according to the dynamical model, F_k. In a similar fashion it is possible to obtain also the predicted covariance matrix, P_{k-1}, figure 5.12



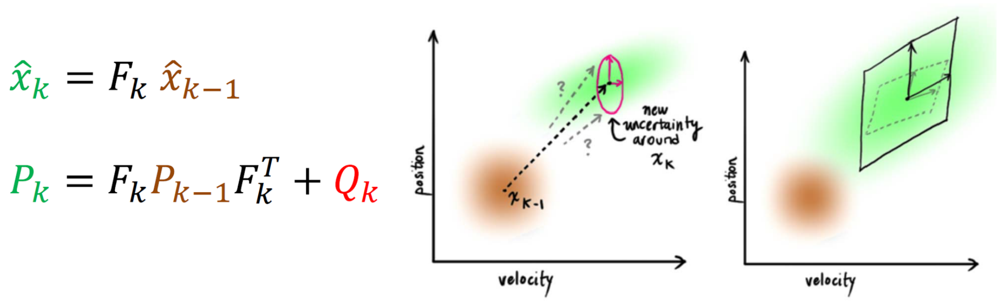
$$\hat{x}_k = F_k \hat{x}_{k-1}$$

$$P_k = F_k P_{k-1} F_k^T$$

F_k represents the dynamical model
Note that:
$Cov(x) = \Sigma$
$Cov(Ax) = A\Sigma A^T$

Figure 5.12 State and covariance matrix prediction, [21].

In order to be robust against some unmodelled dynamics or external disturbances, an additional uncertainty, Q_k, is applied to the predicted covariance matrix, figure 5.13 on the following page.

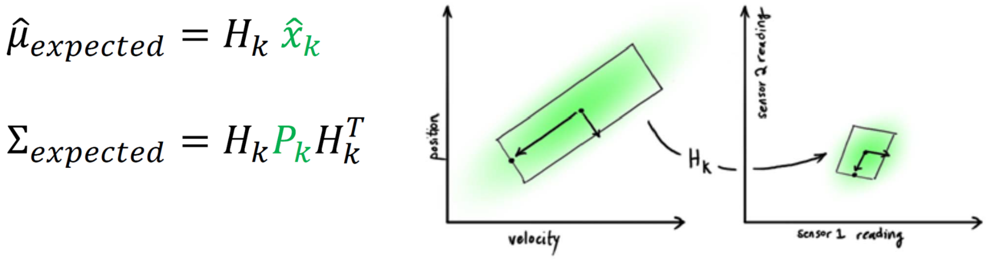


$$\hat{x}_k = F_k \hat{x}_{k-1}$$

$$P_k = F_k P_{k-1} F_k^T + Q_k$$

Figure 5.13 Additional uncertainty addition to predicted covariance, [21].

The predicted value has to be refined with measurements from sensors. As for the state, it is possible to obtain the expected measurements and their covariance matrix from the current state and through the measurement model, H_k, figure 5.14.



$$\hat{\mu}_{expected} = H_k \hat{x}_k$$

$$\Sigma_{expected} = H_k P_k H_k^T$$

Figure 5.14 Measurements and their covariance matrix prediction, [21].

Sensors instead produce given measurements z_k with an associate uncertainty, R_k. Multiplying the estimated state obtained through the measurement model and the obtained measurements together with their Gaussian distributions, it is possible to obtain the Kalman update factor, $z_k - H_k\hat{x}_k$, figure 5.15.

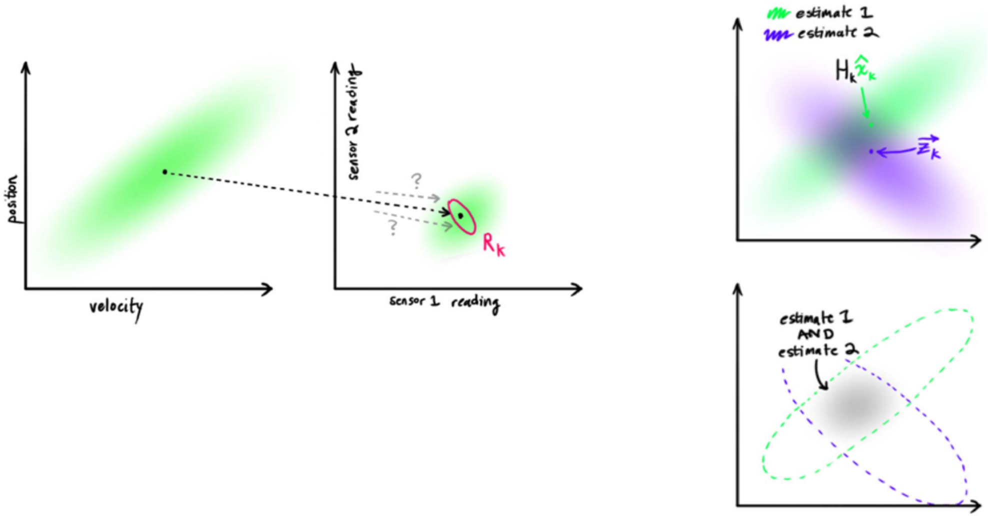


Figure 5.15 Update factor evaluation, [21].

Finally, the complete formulation is here reported:
Prediction

$$\hat{x}_k = F_k \hat{x}_{k-1} \tag{5.23}$$

$$P_k = F_k P_{k-1} F_k^T + Q_k \tag{5.24}$$

Update

$$\hat{x}'_k = \hat{x}_k + K'(z_k - H_k \hat{x}_k) \tag{5.25}$$

$$P'_k = P_k - K' H_k P_k \tag{5.26}$$

$$K' = P_k H_k^T (H_k P_k H_k^T + R_k)^{-1} \tag{5.27}$$

Kalman filters main kinds There are three main kinds of Kalman filters:

- *Linear Kalman filter*, as the one presented in the previous paragraph.
- *Extended Kalman filter*, used for non linear models and based on their linearization.
- *Unscented Kalman filter*, used for non linear models.

The *extended kalman filter* has a similar formulation with respect to the linear one:
Prediction

$$\hat{x}_k = \boldsymbol{f}(\hat{x}_{k-1}) \tag{5.28}$$

$$P_k = F_k P_{k-1} F_k^T + Q_k \tag{5.29}$$

Update

$$\hat{x}'_k = \hat{x}_k + K'(z_k - \boldsymbol{h}(\hat{x}_k)) \tag{5.30}$$

$$P'_k = P_k - K' H_k P_k \tag{5.31}$$

$$K' = P_k H_k^T (H_k P_k H_k^T + R_k)^{-1} \tag{5.32}$$

where $\boldsymbol{f}$ and $\boldsymbol{h}$ are the non linear models of the dynamics and measurements.

In this case F_k and H_k are the Jacobians of the models:

$$F_k = \frac{\partial f}{\partial x} \tag{5.33}$$

$$H_k = \frac{\partial h}{\partial x} \tag{5.34}$$

The limitation of this kind of filter is that they do not guarantee good accuracy if the system non linearities are severe, because it relies on linearization for mean and covariance update.

For this reason the filter selected for OUFTI-Next ADC subsystem is the *unscented Kalman filter*, which is not based on linearization.

Unscented Kalman filter The *unscented kalman filter* is based on *unscented transformations*:

1. Sample points (*sigma points*) are used to approximate the state distribution.
2. They are popagated through the non linear dynamical model.
3. Their propagation can be used to reconstruct the posterior state mean and covariance.

In this kind of filter no Jacobian nor linearization are requested. In figure 5.16 it is possible to understand graphically the differences among the three kinds of Kalman filters.

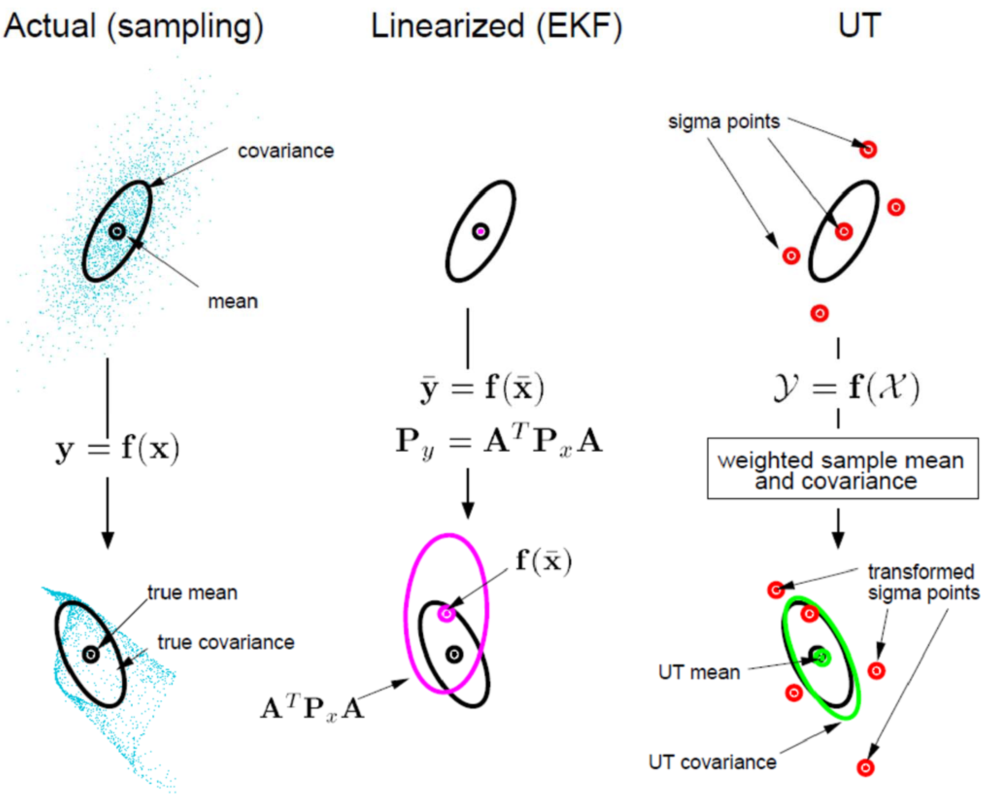


Figure 5.16 Three main kinds of Kalman filters compared, [8].

The procedure to implement it in the simulator is here schematically reported and it is based on, [24]:

1. The input of the filter are:
    - L, the number of states to estimate, in the case of OUFTI they are 7 (3 angular velocities and 4 quaternion components).
    - m, the number of measurements available, in the case of OUFTI they are 7 (3 angular velocities and 4 quaternion components).
    - f, the dynamical model.
    - h, the measurement model (it has been decided to be just the output of the integration for simplicity).
    - q, standard deviation of the process.
    - r, standard deviation of the measurements.
    - P, the initial state error covariance matrix, commonly initialized as $\boldsymbol{P} = \boldsymbol{I}$.

    In the case of OUFTI, the standard deviation of the measurements is $1.5e^{-4}$ for the star tracker, according to its accuracy, and $1e^{-4}$ for the gyros according to the typical measurements noise level. Then $\boldsymbol{Q}$ and $\boldsymbol{R}$ will be the covariance matrices respectively of process and measurements (diagonal matrices filled by the square of the standard deviations of the process and measurements). In order to find the proper value for the standard deviation of the process, some simulations have been performed and after having compared the determination errors, the value selected has been 1. Moreover the covariance matrix of the process, representing the uncertainties on the model (equations), has been defined as equal to the identity I , also because in this way we are accounting for dynamics not modeled, “smearing” artificially the next-step normal distribution, increasing the uncertainty of the prediction and raising, as a consequence, the probabilities of finding a good estimation of the state vector.

2. The state and measurements are perturbed with a white noise with 0 mean value and standard deviation defined above, but, as explained before, no noises have been implemented for the moment and, as a consequence, they are perturbed by a fixed value corresponding to the standard deviations (worst case).

3. Some parameters are defined, according to [11]:
   - $k = 0$
   - $\alpha = \sqrt{3}$
   - $\beta = 2$
   - $\lambda = \alpha^2(L + k) - L$
   - $c = L + \lambda$

4. Then the weights vectors (of length 2L + 1) are evaluated:
   - $W_m = \left[\frac{\lambda}{c}, \left[\frac{0.5}{c}, \frac{0.5}{c}, ...\right]\right]$
   - $W_c = W_m$
   - $W_c(1) = W_c(1) + (1 - \alpha^2 + \beta)$

5. At this point the state *sigma points* are generated around the state, $\boldsymbol{x}$:

$$\boldsymbol{A} = \sqrt{c}\ chol(\boldsymbol{P})^T \tag{5.35}$$

where “chol” means: Cholesky decomposition.

$$\boldsymbol{X} = \begin{bmatrix}\boldsymbol{x}\boldsymbol{A} + \boldsymbol{Y} & \boldsymbol{Y} - \boldsymbol{A}\end{bmatrix} \tag{5.36}$$

where $\boldsymbol{Y}$ is an $L \times L$ matrix full of ones and $\boldsymbol{X}$ dimension is $L \times 2L + 1$, containing the 2L+1 *sigma points* of the state.

6. The state *sigma point*, k, is then propagated through the dynamical model by an unscented transformation:

$$\boldsymbol{Y(:,k)} = f(\boldsymbol{X(:,k)}) \tag{5.37}$$

as well as the mean value vector, $\boldsymbol{y}$:

$$\boldsymbol{y} = \boldsymbol{W_m}(k)\boldsymbol{Y}(:,k) \tag{5.38}$$

providing finally the transformed deviation from the mean value, $\boldsymbol{Y_1}$, subtracting to each column i of $\boldsymbol{Y}$ the vector $\boldsymbol{y}$:

$$\boldsymbol{Y_1} = \boldsymbol{Y(:,i)} - \boldsymbol{y} \tag{5.39}$$

and the transformed covariance of the state error, $\boldsymbol{P_1}$:

$$\boldsymbol{P_1} = \boldsymbol{Y_1}\begin{bmatrix} W_{c,1} & 0 & 0 & 0 & ... & 0 \\ 0 & W_{c,2} & 0 & 0 & ... & 0 \\ 0 & 0 & W_{c,3} & 0 & ... & 0 \\ 0 & 0 & 0 & W_{c,4} & ... & 0 \\ ... & ... & ... & ... & ... & 0 \\ 0 & 0 & 0 & 0 & 0 & W_{c,2L+1} \end{bmatrix}\boldsymbol{Y_1}^T + \boldsymbol{R} \tag{5.40}$$

7. Same procedure must be followed for the measurements unscented transformation.

8. Then the transformation of the cross-covariance, $\boldsymbol{P_{12}}$:

$$\boldsymbol{P_{12}} = \boldsymbol{P_{1,s}} \begin{bmatrix} W_{c,1} & 0 & 0 & 0 & ... & 0 \\ 0 & W_{c,2} & 0 & 0 & ... & 0 \\ 0 & 0 & W_{c,3} & 0 & ... & 0 \\ 0 & 0 & 0 & W_{c,4} & ... & 0 \\ ... & ... & ... & ... & ... & 0 \\ 0 & 0 & 0 & 0 & 0 & W_{c,2L+1} \end{bmatrix} \boldsymbol{P_{1,m}}^T \tag{5.41}$$

   where $\boldsymbol{P_{1,s}}$ is the covariance of the state transformed and $\boldsymbol{P_{1,m}}$ is the one of the measurements.

9. The Kalman matrix $\boldsymbol{K}$ is evaluated:

$$\boldsymbol{K} = \boldsymbol{P_{12}} \boldsymbol{Y_1}^{-1} \tag{5.42}$$

10. The state is updated:

$$\hat{\boldsymbol{x}} = \boldsymbol{y} + \boldsymbol{K}(\boldsymbol{z} - \boldsymbol{v}) \tag{5.43}$$

    where $\boldsymbol{v}$ is the measurements mean value as $\boldsymbol{y}$ is the mean value of the state.

11. Finally the covariance of state error is updated:

$$\boldsymbol{P} = \boldsymbol{Y_1} - \boldsymbol{K}(\boldsymbol{P_{12}}^T) \tag{5.44}$$

    It is important to remark that, since the *unscented Kalman filter* is based on a weighted sum, it is not guaranteed that the estimated quaternion is unit. As a consequence the quaternion must be normalized after each estimation.

Numerical instability issue Unfortunately, the addition of such a non linear determination algorithm makes the simulation to slow dramatically down when a steady state condition is reached and maintained for long time, as commonly happens in each mission profile that will be presented. Each solver of Matlab™ tries to reduce the time step in order to drive the system eigenvalues inside the regions of numerical stability, without success, resulting in a numerical instability. A future development of this work could be the evaluation of the system eigenvalues linearizing the model around different reference points and verifying the numerical stability for each solver. In any case, even trying different solvers, the issue could not be solved and it has been decided to maintain the *unscented Kalman filter* only for the most delicate phase, the picture acquisition, also because it is the shortest in time.
For all the other phases, an algebric algorithm has been implemented, based on the star tracker characteristics.

5.6.4 Algebric algorithm

The star tracker provides multiple vectors of observations in the *body-fixed frame*, $\hat{\boldsymbol{O}}_{\boldsymbol{1}}$, $\hat{\boldsymbol{O}}_{\boldsymbol{2}}$, $\hat{\boldsymbol{O}}_{\boldsymbol{3}}$, ..., $\hat{\boldsymbol{O}}_{\boldsymbol{i}}$, and has in its memory a star catalogue with stars position vectors in *inertial frame*, $\boldsymbol{S_1}$, $\boldsymbol{S_2}$, $\boldsymbol{S_3}$, ..., $\boldsymbol{S_i}$. From this informations it is possible to retrieve the body-fixed frame matrix, $\boldsymbol{A_{b/n}}$, using the following expression:

$$\hat{\boldsymbol{O}}_{\boldsymbol{i}} = \boldsymbol{A_{b/n}} \boldsymbol{S_i} \tag{5.45}$$

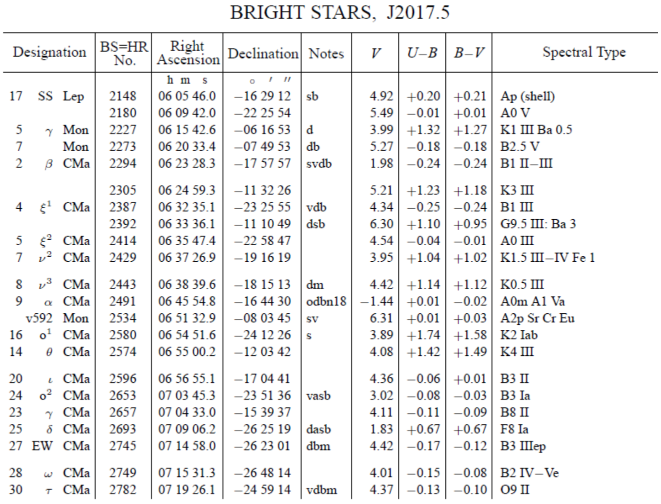
If we consider the 6 stars observable by the star tracker ST-200, the equations will be:

$$\begin{bmatrix} \hat{\boldsymbol{O}}_{\boldsymbol{1}} \; \hat{\boldsymbol{O}}_{\boldsymbol{2}} \; \hat{\boldsymbol{O}}_{\boldsymbol{3}} \; \hat{\boldsymbol{O}}_{\boldsymbol{4}} \; \hat{\boldsymbol{O}}_{\boldsymbol{5}} \; \hat{\boldsymbol{O}}_{\boldsymbol{6}} \end{bmatrix} = \boldsymbol{A_{b/n}} \begin{bmatrix} \hat{\boldsymbol{S}}_{\boldsymbol{1}} \; \hat{\boldsymbol{S}}_{\boldsymbol{2}} \; \hat{\boldsymbol{S}}_{\boldsymbol{3}} \; \hat{\boldsymbol{S}}_{\boldsymbol{4}} \; \hat{\boldsymbol{S}}_{\boldsymbol{5}} \; \hat{\boldsymbol{S}}_{\boldsymbol{6}} \end{bmatrix} \tag{5.46}$$

$$\boldsymbol{A_{b/n}} = \left[\hat{\boldsymbol{O}}_1\ \hat{\boldsymbol{O}}_2\ \hat{\boldsymbol{O}}_3\ \hat{\boldsymbol{O}}_4\ \hat{\boldsymbol{O}}_5\ \hat{\boldsymbol{O}}_6\right]\left[\hat{\boldsymbol{S}}_1\ \hat{\boldsymbol{S}}_2\ \hat{\boldsymbol{S}}_3\ \hat{\boldsymbol{S}}_4\ \hat{\boldsymbol{S}}_5\ \hat{\boldsymbol{S}}_6\right]^* \tag{5.47}$$

where “ * ” indicates the Moore-Penrose pseudo-inverse operation (a generalization of the inverse operator for rectangular matrices), since the matrices are not square.

An example of star catalogue, in particular the one used in these simulations, is the following, [22]:



BRIGHT STARS, J2017.5

Designation	BS=HR No.	Right Ascension	Declination	Notes	V	$U-B$	$B-V$	Spectral Type
		h m s	° ′ ″					
17 SS Lep	2148	06 05 46.0	−16 29 12	sb	4.92	+0.20	+0.21	Ap (shell)
	2180	06 09 42.0	−22 25 54		5.49	−0.01	+0.01	A0 V
5 γ Mon	2227	06 15 42.6	−06 16 53	d	3.99	+1.32	+1.27	K1 III Ba 0.5
7 Mon	2273	06 20 33.4	−07 49 53	db	5.27	−0.18	−0.18	B2.5 V
2 β CMa	2294	06 23 28.3	−17 57 57	svdb	1.98	−0.24	−0.24	B1 II–III
	2305	06 24 59.3	−11 32 26		5.21	+1.23	+1.18	K3 III
4 ξ^1 CMa	2387	06 32 35.1	−23 25 55	vdb	4.34	−0.25	−0.24	B1 III
	2392	06 33 36.1	−11 10 49	dsb	6.30	+1.10	+0.95	G9.5 III: Ba 3
5 ξ^2 CMa	2414	06 35 47.4	−22 58 47		4.54	−0.04	−0.01	A0 III
7 ν^2 CMa	2429	06 37 26.9	−19 16 19		3.95	+1.04	+1.02	K1.5 III–IV Fe 1
8 ν^3 CMa	2443	06 38 39.6	−18 15 13	dm	4.42	+1.14	+1.12	K0.5 III
9 α CMa	2491	06 45 54.8	−16 44 30	odbn18	−1.44	+0.01	−0.02	A0m A1 Va
v592 Mon	2534	06 51 32.9	−08 03 45	sv	6.31	+0.01	+0.03	A2p Sr Cr Eu
16 o^1 CMa	2580	06 54 51.6	−24 12 26	s	3.89	+1.74	+1.58	K2 Iab
14 θ CMa	2574	06 55 00.2	−12 03 42		4.08	+1.42	+1.49	K4 III
20 ι CMa	2596	06 56 55.1	−17 04 41		4.36	−0.06	+0.01	B3 II
24 o^2 CMa	2653	07 03 45.3	−23 51 36	vasb	3.02	−0.08	−0.03	B3 Ia
23 γ CMa	2657	07 04 33.0	−15 39 37		4.11	−0.11	−0.09	B8 II
25 δ CMa	2693	07 09 06.2	−26 25 19	dasb	1.83	+0.67	+0.67	F8 Ia
27 EW CMa	2745	07 14 58.0	−26 23 01	dbm	4.42	−0.17	−0.12	B3 IIIep
28 ω CMa	2749	07 15 31.3	−26 48 14		4.01	−0.15	−0.08	B2 IV–Ve
30 τ CMa	2782	07 19 26.1	−24 59 14	vdbm	4.37	−0.13	−0.10	O9 II

Figure 5.17 Example of star catalogue used in this Thesis, [22].

Then as explained in the star tracker model description, a bias matrix, $\boldsymbol{A_\epsilon}$, is multiplied to the obtained body-fixed frame matrix, $\boldsymbol{A_{b/n}}$, in order to take into consideration also the star tracker accuracy.

$$\boldsymbol{A_{b/n,real}} = \boldsymbol{A_\epsilon A_{b/n}} \tag{5.48}$$

Graphically this algorithm can shown as in figure 5.18.

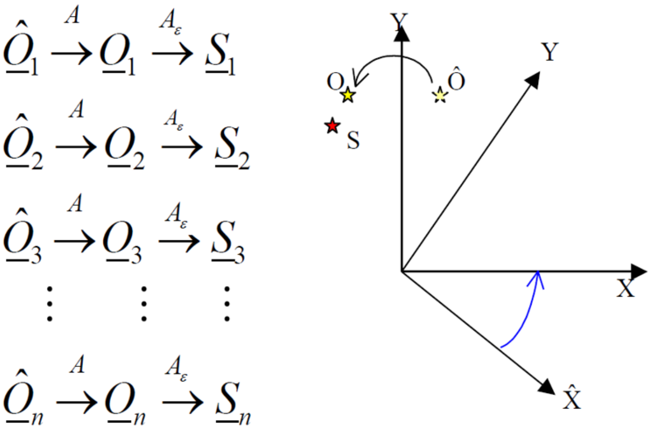


Figure 5.18 Algebric determination algorithm scheme.

In this way it has been possible to simulate without any numerical instability the longer phases of the mission, figure 5.19 on the next page.

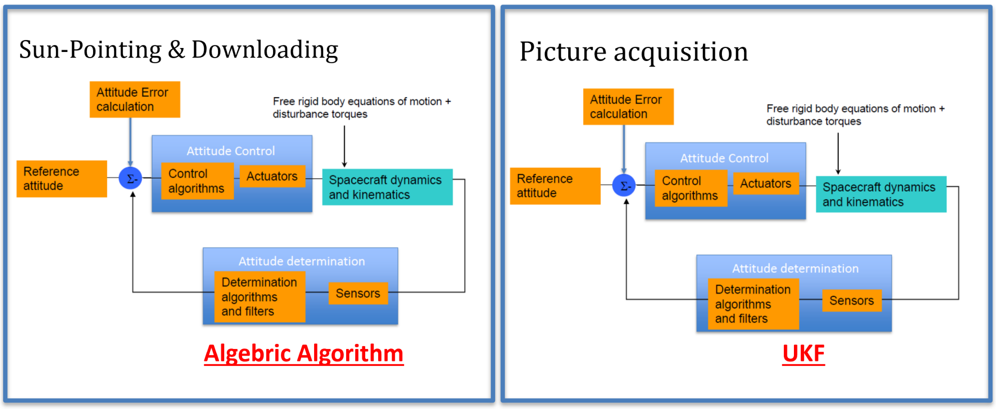


Figure 5.19 Determination strategy depending on the mission phase.

5.7 Control

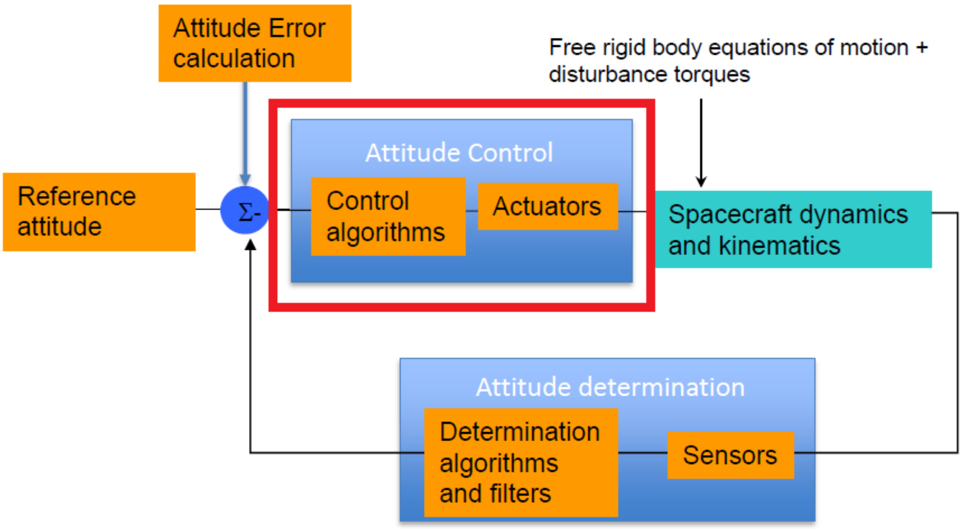


Figure 5.20 ADCS feedback loop with *control* block highlighted.

According to the stability theory of non linear systems, the control law of a PD controller shall be based on a Lyapunov function, $H(q)$, dependent on the state, such that for asymptotic stability:

- $H(x) > 0$, $\forall$ x $\neq x_{eq}$
- $H(x) = 0$, for $x = x_{eq}$
- $\dot{H}(x) < 0$

and for Lyapunov stability:

- $H(x) > 0$, $\forall$ x $\neq x_{eq}$
- $H(x) = 0$, for $x = x_{eq}$

- $\dot{H}(x) \leq 0$

For the simulator implemented, the method to represent the reference frame are the quaternions and, as a consequence, it has been selected a Lyapunov function based on the scalar number of the error quaternion, $q_{e,4}$:

$$H(q_{e,4}) = 1 - q_{e,4}^2 \tag{5.49}$$

This particular law has been selected since it is immune to *unwinding*, which is a typical issue, when quaternions are exploited. It is due to the not uncertainty about the scalar number sign while transforming the quaternion to direct cosine matrices and viceversa. The square in the law has the role to make the Lyapunov function independent from the sign of $q_{e,4}$

Once the control law is selected, the control torque will be:

$$u = K_p \frac{\partial H}{\partial q_{e,4}} \boldsymbol{q(1,2,3)} + K_d(\boldsymbol{\omega_d} - \boldsymbol{\omega}) \tag{5.50}$$

$$u = -2K_p \boldsymbol{q_{e,4}} \boldsymbol{q_e(1,2,3)} + K_d(\boldsymbol{\omega_d} - \boldsymbol{\omega}) \tag{5.51}$$

where:

- K_p and K_d are respectively the proportional and derivative constant, both positive.
- $\boldsymbol{\omega_d}$ are the desired angular velocities.
- $\boldsymbol{q_e}$ is the error quaternion between the actual attitude matrix, $\boldsymbol{A_{b/n}}$, and the desired one, $\boldsymbol{A_{LVLH}}$, transformed in quaternions.

In order to find $\boldsymbol{q_e}$, the following steps are requested:

1. Transform the $\boldsymbol{A_{LVLH}}$ matrix in control quaternion: $\boldsymbol{q_c}$.
2. Compute the error between the actual quaternion $\boldsymbol{q}$ and the control one $\boldsymbol{q_c}$, exploiting the quaternion multiplication:

   $$\boldsymbol{q_e} = (\boldsymbol{q_c})^{-1} \otimes \boldsymbol{q} \tag{5.52}$$

   Meaning, in matrix form:

   $$\boldsymbol{q_e} = \begin{bmatrix} q_{4c} & q_{3c} & -q_{2c} & -q_{1c} \\ -q_{3c} & q_{4c} & q_{1c} & -q_{2c} \\ q_{2c} & -q_{1c} & q_{4c} & -q_{3c} \\ q_{1c} & q_{2c} & q_{3c} & q_{4c} \end{bmatrix} \boldsymbol{q} \tag{5.53}$$

   Where, if $\boldsymbol{q_c} = [0\ 0\ 0\ 1]$, it means that $\boldsymbol{q_e} = \boldsymbol{q}$.

5.8 Actuators

iADCS100 is equipped by the common actuators available for CubeSats: *magnetic torquers* and *reaction wheels*. In this section their main characteristics will be described, as well as their implementation in the simulator.

5.8.1 Magnetic torquers

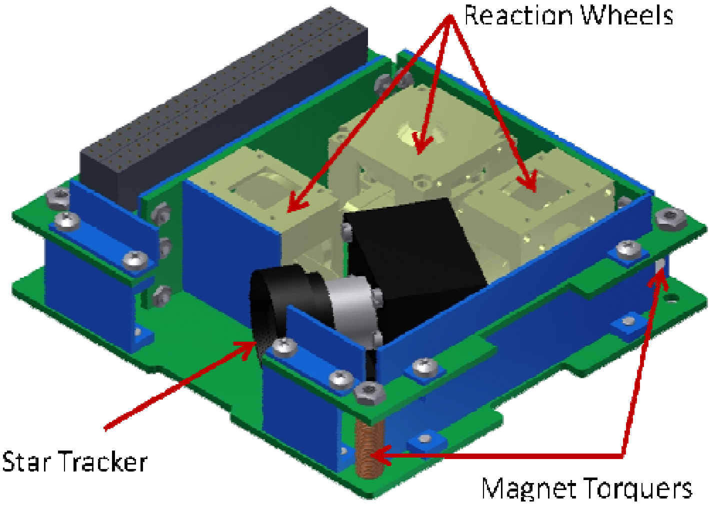


Figure 5.21 2 of the 3 magnetic torquers of *iADCS100*.

Magnetic torquers are devices producing a dipole moment capable to interact with the Earth magnetic sphere and turn the spacecraft. In order to command them it is possible to change the input current flowing in the armatures to change the dipole moment. They will be used during de-tumbling phase with a control strategy that will be presented afterwards in the section dedicated to de-tumbling phase. In order to model them, the ideal magnetic dipole requested is compared with the maximum available from the magnetic torquers and the output will be eventually saturated. The common configuration is based on 3 of them directed as the spacecraft axes of symmetry.

5.8.2 Reaction wheels

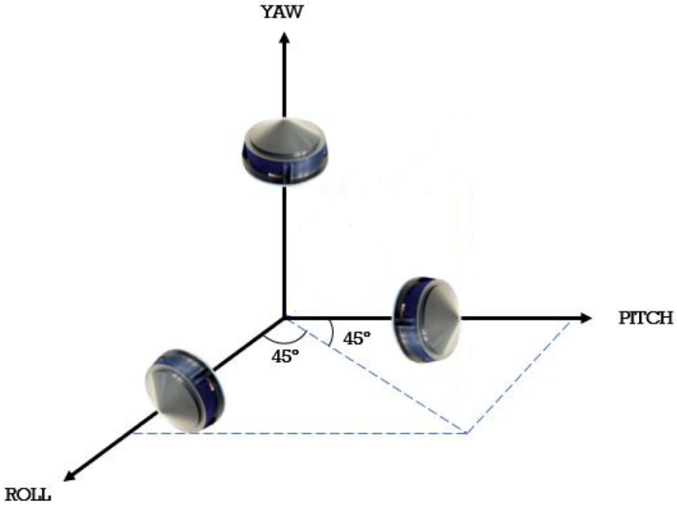


Figure 5.22 3 reaction wheels common configuration.

Reaction wheels are rotating devices with a fixed position inside the CubeSat, commonly, as in the case of *iADCS100*, they are three and they are directed as the yaw, pitch and roll axis, figure 5.22 on page 44. They are able to produce a variable torque on the satellite thanks to their angular acceleration. Their torque equation is:

$$\boldsymbol{u} = \boldsymbol{A}\boldsymbol{h_r} \times \boldsymbol{\omega} - \boldsymbol{A}\dot{\boldsymbol{h}}_{\boldsymbol{r}} \tag{5.54}$$

where:

- $\boldsymbol{A}$ is the constant reaction wheels configuration matrix, in this case, the unit matrix I.
- $\boldsymbol{h}$ is the reaction wheels momenta vector.

In order to model their dynamics, from the control required, $\boldsymbol{u_d}$, the momentum required is integrated, using the following equation:

$$\dot{\boldsymbol{h}}_{\boldsymbol{r,d}} = \boldsymbol{A}^{-1}(\boldsymbol{A}\boldsymbol{h_{r,d}} \times \boldsymbol{\omega} - \boldsymbol{u_d}) \tag{5.55}$$

and then $\boldsymbol{h_{r,d}}$ and $\dot{\boldsymbol{h}}_{\boldsymbol{r,d}}$ are compared with the maximum and minimum ones provided by the wheels and eventually are saturated, if they are higher or lower.

At this point the effective torque is evaluated:

$$\boldsymbol{u} = \boldsymbol{A}\boldsymbol{h_r} \times \boldsymbol{\omega} - \boldsymbol{A}\dot{\boldsymbol{h}}_{\boldsymbol{r}} \tag{5.56}$$

Reaction wheels are characterized by a peculiar issue: during long controlled phases, the wheels are continuously increasing their momentum in order to counteract the external disturbances. Once the maximum value of momentum is reached, the wheels can not accelerate and the control is lost. Once the maximum momentum storable is reached, a de-saturation phase is needed. In the case of OUFTI-Next a de-saturation phase has been simulated and it will be described in a following section.

5.9 Mission phases and control strategies

The following three representative missions have been implemented both for ISS orbit and SSO, with Nadir and target following picture acquisition strategies. All of them has as initial epoch: 01/01/2019 at 12:00:00.

Mission 1

1. De-Tumbling.
2. Sun-Pointing for 24 hours.
3. Sun-Pointing till 30° from target.
4. De-Saturation phase.
5. Picture acquisition phase.
6. Sun-Pointing phase till 10° from Liège.
7. De-saturation phase.
8. Downloading phase (Nadir).

Mission 2

1. *NO DE-TUMBLING PHASE.*
2. Sun-Pointing for 24 hours.
3. Sun-Pointing till 30° from target.
4. De-Saturation phase.
5. Picture acquisition phase.
6. Sun-Pointing phase till 10° from Liège.
7. De-saturation phase.
8. Downloading phase (Nadir).

Mission 3

1. De-Tumbling.
2. Sun-Pointing till saturation of the reaction wheels.
3. De-Saturation phase for 1 day.

Mission 1 was proposed in order to simulate a typical first operative mission after de-tumbling, while, instead, mission 2 was aimed to replicate a nominal mission scenario some time after the de-tumbling phase. Finally, mission 3 was intended to be exploited in order to obtain the maximum time in Sun-pointing before saturating the wheels and then to simulate what would happen during longer de-saturation phase, in terms of angular velocities: the behavior of the CubeSat can be useful to understand more in detail how much energy can be acquired by the solar panels even during de-saturation. Nevertheless, this point will be investigated in future design steps of OUFTI project.

5.9.1 De-tumbling

Right after being released by the launcher, the CubeSat is *tumbling*, meaning that it is spinning very fast around its axes. The first phase of a spacecraft operational life is the *de-tumbling phase*. During this phase the only actuators exploited are the magnetic torquers commanded by a control law, function of the Earth magnetic field.

The most common and Lyapunov stable law is called *B-dot law*, since it is based on the derivative of the magnetic field felt by the CubeSat in body-frame, $\boldsymbol{B_b}$:

$$\boldsymbol{m} = -k\dot{\boldsymbol{B}}_{\boldsymbol{b}} \tag{5.57}$$

$$\boldsymbol{u} = \boldsymbol{m} \times \boldsymbol{B_b} \tag{5.58}$$

where k is a positive constant and $\boldsymbol{m}$ the dipole moment requested, which will be compared to the available one and eventually saturated if it will exceed it.

Nevertheless, this law is based on the derivative of the magnetic field in body frame, $\dot{\boldsymbol{B}}_{\boldsymbol{b}}$, which is a quite expensive quantity to be evaluated computationally speaking. In order to obtain it, the previous step $\boldsymbol{A_{b/n}}$ is requested, in order to find an approximate value of $\dot{\boldsymbol{B}}_{\boldsymbol{b}}$, using finite differences:

$$\boldsymbol{B_b} = \boldsymbol{A_{b/n}}\dot{\boldsymbol{B}}_{\boldsymbol{b}} \tag{5.59}$$

$$\dot{\boldsymbol{B}}_{\boldsymbol{b}} = \frac{\boldsymbol{B_b}(t) - \boldsymbol{B_b}(t-\tau)}{\tau} \tag{5.60}$$

This procedure is really difficult to be implemented in Matlab™ without increasing dramatically the computational time. For this reason it has been decided to implement the de-tumbling phase in Simulink™, where it is possible to exploit *delay blocks* in order to retrieve previous values for $\boldsymbol{B_b}$. The model scheme is represented in figure 5.23.

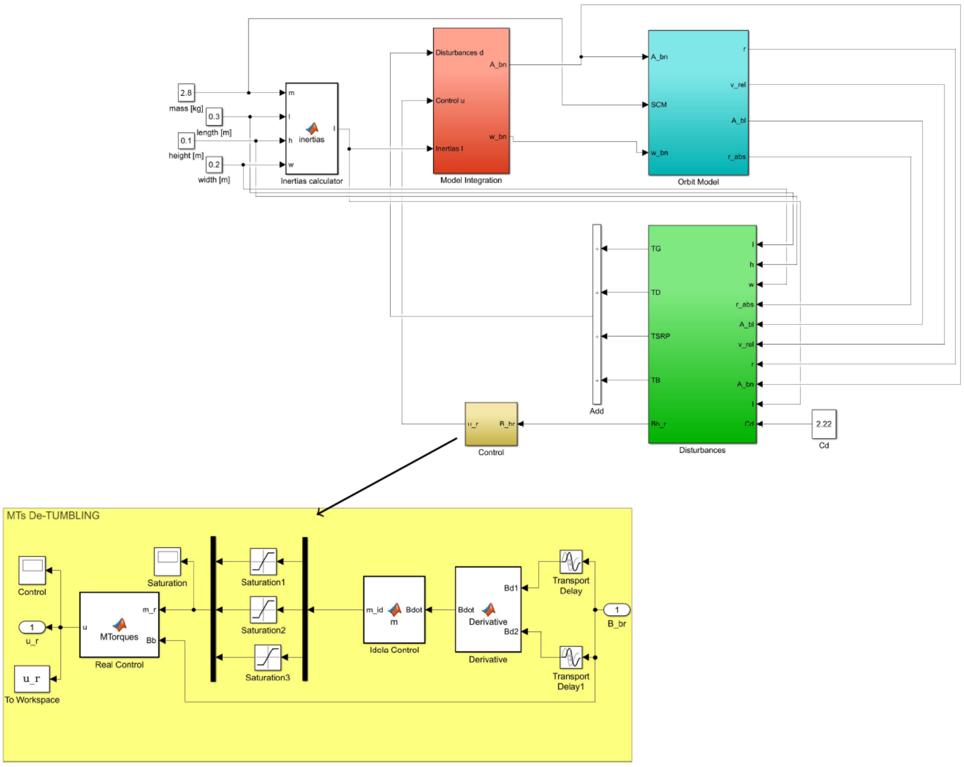


Figure 5.23 De-tumbling Simulink™ model with *control block* highlighted.

The constant k can be changed in order to modify the de-tumbling behavior and it can be also optimized in order to minimize the de-tumbling time. Nevertheless, the computational time requested to perform an optimization procedure is too long and a possible development of this analysis could be the simplification of the algorithm, allowing the constant to be optimized in reasonable time. All the simulations presented in this Thesis have been performed with $k = 10^5$.

For what concerns the de-tumbled condition, fundamental to stop this phase and start the following one, initially it was thought to use an angular velocity threshold: if the OUFTI maximum angular rate was below this value for a certain amount of time, the CubeSat was supposed to be de-tumbled. Then, looking at different de-tumbling simulations, it appeared clear that the spacecraft could in any case de-tumble in less than 5000 s and then maintains an almost periodic behavior. For this reason, all the de-tumbling simulations will end at $T_{final} = 5000$ s, around 1h40min.

During this phase no determination algorithm nor sensors models have been implemented, relaying only on the integrated state, since the spacecraft will rotate too fast for the star tracker to be used and the magnetometers/gyros models would slow down the simulation too much. The results are quite robust with respect to the unknown initial tumbling velocity, since the simulation has been initiated with random initial condition distribution centered in 0.5 rad/s, but it is also deeply dependent on:

- Residual dipole moment.
- Atmosphere condition at launch epoch.

- Solar activity at launch epoch.
- Season at launch epoch.

For this reason all the results presented about this phase are to be intended qualitatively more than quantitatively.

Magnetic field approximated model In order to reduce the computational time of this simulation, an approximated model for the Earth magnetic field has been implemented, based on the "Schmidt normalized associated Legendre Polynomial":

$$\boldsymbol{B} = -\nabla V \tag{5.61}$$

with:

$$V(r,\theta,\phi) = R_e \sum_{n=1}^{k} \left(\frac{R_e}{r}\right)^{n+1} \sum_{m=0}^{n} (g_n^m cos(m\phi) + h_n^m sin(m\phi)) P_n^m(\theta) \tag{5.62}$$

$$B_r = \sum_{n=1}^{k} \left(\frac{R_e}{r}\right)^{n+2} (n+1) \sum_{m=0}^{n} (g_n^m cos(m\phi) + h_n^m sin(m\phi)) \frac{\partial P_n^m(\theta)}{\partial \theta} \tag{5.63}$$

$$B_\theta = -\sum_{n=1}^{k} \left(\frac{R_e}{r}\right)^{n+2} \sum_{m=0}^{n} (g_n^m cos(m\phi) + h_n^m sin(m\phi)) P_n^m(\theta) \tag{5.64}$$

$$B_\phi = -\frac{1}{sin(\theta)} \sum_{n=1}^{k} \left(\frac{R_e}{r}\right)^{n+2} (n+1) \sum_{m=0}^{n} (-g_n^m sin(m\phi) + h_n^m cos(m\phi)) P_n^m(\theta) \tag{5.65}$$

where:

- R_e is the Earth mean radius.
- r is the CubeSat position vector.
- θ is the latitude measured in degrees positive from the equator (input from orbit model).
- ϕ is the Earth fixed longitude measured in degrees positive from the equator (input from orbit model).
- g_n^m and h_n^m are the tabulated Gaussian coefficients in $[T]$.
- P_n^m is the Schmidt quasi-normalized associated Legendre Polynomial.
- The model is expanded till order 9 ($n,\ m = 1, \ldots 9$).

Once obtained B_r, B_θ and B_ϕ, $\boldsymbol{B_n}$ in inertial frame can be evaluated:

$$B_{n1} = (B_r cos(\theta) + B_\theta sin(\theta))cos(\alpha) - B_\phi sin(\alpha) \tag{5.66}$$

$$B_{n2} = (B_r cos(\theta) + B_\theta sin(\theta))sin(\alpha) + B_\phi cos(\alpha) \tag{5.67}$$

$$B_{n3} = (B_r sin(\theta) - B_\theta cos(\theta)) \tag{5.68}$$

where α is the longitude measured in degrees positive from the equator. Finally the Earth magnetic field vector must be transported in body frame:

$$\boldsymbol{B_b} = \boldsymbol{A_{b/n} B_n} \tag{5.69}$$

5.9.2 Pointing phases

Every pointing phase is characterized by the evaluation of a desired reference frame matrix, $\boldsymbol{A_{LVLH}}$, that the CubeSat requires to follow, which is different in each phase.
Since no precise target has been identified yet, it has been decided to simulate picture acquisition of the Brasilia National Park, because it is near to the possible zone of interest and is characterized by similar environmental characteristics.

Sun-pointing During Sun-pointing phase the CubeSat has been decided to be taken at 45° inclination with respect to the Sun with the two surfaces covered by solar panels (Z and Y, according to the ADCS reference frame, figure 2.1 on page 5) and not also tilted in order to have also the top surface in light, because it was preferred to maintain the desired attitude simple.
The actuators used during this phase are only the 3 reaction wheels, with the following desired attitude matrix:

$$\boldsymbol{A_{LVLH}} = \begin{bmatrix} \boldsymbol{\hat{x}_p} \\ \boldsymbol{\hat{y}_p} \\ \boldsymbol{\hat{z}_p} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\hat{y}_p} \times \boldsymbol{\hat{r}_{Sun}} \\ \boldsymbol{R\hat{r}_{Sun}} \\ \boldsymbol{\hat{x}_p} \times \boldsymbol{\hat{y}_p} \end{bmatrix} \tag{5.70}$$

where $\boldsymbol{r_{Sun}}$ is the OUFTI-Sun direction and $\boldsymbol{R}$ is a rotational matrix:

$$\boldsymbol{R} = \begin{bmatrix} cos(-\frac{\pi}{4}) & -sin(-\frac{\pi}{4}) & 0 \\ sin(-\frac{\pi}{4}) & cos(-\frac{\pi}{4}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.71}$$

In this way the Sun direction is rotated of 45° around the X-axis, allowing both the surfaces covered by solar cells to be in light.

The desired angular velocity, $\boldsymbol{\omega_d}$, is null, since the CubeSat-Sun direction is almost fixed for the time intervals considered in these simulations.
As it will be shown later, the pointing accuracy reached during this phase is less than 0.1°. Nevertheless, the Sun direction unit vector is assumed to be known from ephemeris, previously evaluated; in a real scenario, this datum would be obtained from the Sun sensors. According to *Hyperion*, their unit (*iADCS100*) is not equipped with any of them, but it has interfaces for up to six of these devices. As a result, in the case it will be decided to use their unit, there will be two options: Sun sensors should be bought separately from the ADCS unit, or the ADCS will have to relay on the coarser Sun sensors integrated on the solar panels. In this last case, the accuracy is approximately ±10°, meaning that the final pointing error will be less than 1°, which is an acceptable value, according to the mission requirements.

Nadir pointing picture acquisition The easiest way to shoot pictures to the target is to pass over it pointing the Nadir. This is not the best option, since it is mandatory to have the orbital plane normal almost orthogonal to the target location in ECI frame, meaning that the spacecraft is actually orbiting right above the spot to be observed. This limits the number of pictures it is possible to shoot over the same place per unit time, especially for the ISS orbit, which, as explained before, as a long revisit time.
The actuators used during this phase are only the 3 reaction wheels, with the following desired attitude matrix:

$$\boldsymbol{A_{LVLH}} = \begin{bmatrix} \boldsymbol{\hat{x}_p} \\ \boldsymbol{\hat{y}_p} \\ \boldsymbol{\hat{z}_p} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\hat{r}} \\ \boldsymbol{\hat{h}} \times \boldsymbol{\hat{x}_p} \\ \boldsymbol{\hat{x}_p} \times \boldsymbol{\hat{y}_p} \end{bmatrix} \tag{5.72}$$

where: $\boldsymbol{\hat{r}}$ is the position unit vector and $\boldsymbol{\hat{h}}$ the angular momentum unit vector. In this way the spacecraft is pointing with the $-X$-axis to the ground, while the Y-axis is pointed in the same direction of the velocity vector.

The desired angular velocity is $\boldsymbol{\omega_d} = [0 \;\; 0 \;\; \omega_{orbit}]$ since the CubeSat rotates around its Z-axis with the angular rate it has on its orbit.

Target following picture acquisition The most particular phase is the target following picture acquisition, based on a slew maneuver performed in order to follow the target during the passage, to increase the exposition time and allow the orbital momentum vector to be slightly less or more 90° with respect to the target position in ECI reference frame. This decreases the revisit time, even if for the ISS orbit, the effect is almost negligible, [6].
The actuators used during this phase are only the three reaction wheels, with the following desired attitude matrix:

$$\boldsymbol{A_{LVLH}} = \begin{bmatrix} \boldsymbol{\hat{x}_p} \\ \boldsymbol{\hat{y}_p} \\ \boldsymbol{\hat{z}_p} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\hat{y}_p} \times \boldsymbol{\hat{z}_p} \\ \boldsymbol{\hat{z}_p} \times \boldsymbol{\hat{h}} \\ \boldsymbol{\hat{r}} \end{bmatrix} \tag{5.73}$$

where: $\boldsymbol{\hat{r}}$ is the position unit vector and $\boldsymbol{\hat{h}}$ the angular momentum unit vector. In this way the spacecraft is pointing with the -Z-axis to the ground, while the Y-axis is oriented along the velocity vector.

The desired angular velocity is $\boldsymbol{\omega_d} = [\omega_{orbit} \;\; 0 \;\; 0]$ since the CubeSat rotates around its X-axis with the angular rate it has on its orbit.

Downloading This phase is very similar to the Nadir pointing picture acquisition one, since it is based on a Nadir pointing as well. The only difference is that the surface Nadir pointing is the one with normal $-Z$. The S-band antenna field of view has not been selected yet, but it is supposed to be reasonably large enough to be compatible with a nadir pointing downloading strategy.
The actuators used during this phase are only the three reaction wheels, with the following desired attitude matrix:

$$\boldsymbol{A_{LVLH}} = \begin{bmatrix} \boldsymbol{\hat{x}_p} \\ \boldsymbol{\hat{y}_p} \\ \boldsymbol{\hat{z}_p} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\hat{r}_l C} \\ \boldsymbol{\hat{z}_p} \times \boldsymbol{\hat{x}_p} \\ \boldsymbol{\hat{x}_p} \times \boldsymbol{\hat{v}} \end{bmatrix} \tag{5.74}$$

where: $\boldsymbol{\hat{v}}$ is the velocity unit vector and $\boldsymbol{\hat{r}_{diff}}$ the unit vector obtained from the difference between the position unit vector and the target location unit vector in ECI reference frame:

$$\boldsymbol{\hat{r}_{diff}} = \frac{\boldsymbol{r} - \boldsymbol{r_{l,ECI}}}{||\boldsymbol{r} - \boldsymbol{r_{l,ECI}}||} \tag{5.75}$$

The desired angular velocity is obtained solving the expression of the desired reference frame dynamics with respect to $\boldsymbol{\omega_d}$:

$$\dot{\boldsymbol{A}}_{\boldsymbol{LVLH}} = \boldsymbol{S}(\boldsymbol{\omega_d})\boldsymbol{A_{LVLH}} \tag{5.76}$$

In order to differentiate $\dot{\boldsymbol{A}}_{\boldsymbol{LVLH}}$, the finite difference technique has been exploited:

$$\dot{\boldsymbol{A}}_{\boldsymbol{LVLH}} = \frac{\boldsymbol{A_{LVLH}}(t) - \boldsymbol{A_{LVLH}}(t-\tau)}{\tau} \tag{5.77}$$

with $\tau = 0.001$ s.

5.9.3 De-saturation

De-saturation phase is required when the reaction wheels have reached their maximum momentum storable. During this phase they are switched off and the magnetic torquers are activated with a less precise and computationally demanding law with respect to the de-tumbling one, whose Lyapunov stability is proven in [14]:

$$\boldsymbol{m} = -\frac{1}{||\boldsymbol{B_b}||^2}\boldsymbol{S}(\boldsymbol{B_b})\boldsymbol{\omega} \tag{5.78}$$

$$\boldsymbol{u} = \boldsymbol{m} \times \boldsymbol{B_b} \tag{5.79}$$

where $\boldsymbol{m}$ is the dipole moment requested, function of the cross product between $\boldsymbol{B_b}$ and $\boldsymbol{\omega}$ which will be compared to the available one and eventually saturated if it will exceed it.
In this way the computation of $\dot{\boldsymbol{B}}_{\boldsymbol{b}}$ is avoided and the law assumes a less complex expression.

5.10 Simulation and solvers

Once the model is implemented it is important to choose the right solver among the ones available on Matlab/Simulink™. The solver available are:

- *ODE45*: based on an explicit Runge-Kutta (4,5) formula, it is a one-step solver. [7][18]
- *ODE23*: an implementation of an explicit Runge-Kutta (2,3) pair of Bogacki and Shampine. It may be more efficient than ode45 at crude tolerances and in the presence of moderate stiffness. *Ode23* is a single-step solver. [1][18]
- *ODE 113*: a variable-step, variable-order (VSVO) Adams-Bashforth-Moulton PECE solver of orders 1 to 13. [16][18]
- *ODE15s*: a variable and multi-step, variable-order (VSVO) solver based on the numerical differentiation formulas (NDFs) of orders 1 to 5, used for stiff problems. [17][18]
- *ODE23s*: based on a modified Rosenbrock formula of order 2. Because it is a single-step solver, it may be more efficient than *ode15s* at solving problems that permit crude tolerances or problems with solutions that change rapidly. It can solve some kind of stiff problems for which *ode15s* is not effective. [18]

The simulations have been solved with each of these solvers, providing the same results with different computational times. Since the simulations results are equal to each other it is possible to state that the model is numerically stable, even if a more detailed and precise demonstration of stability should be done. The proof of numerical stability, performed linearizing the model around different reference conditions and comparing the eigenvalues location with the different solvers region of stability, will be done in future developments of this Thesis, since this was out of the scope of the work at ULg.
The solver chosen is *ODE15s*, since it is the fastest among the previous ones. Here is reported a table comparing the times requested by each solvers to simulate a Sun-pointing phase of 200 s with solver absolute and relative tolerances set to $1e^{-6}$.

Processor used: Intel® Core ™ i7-7700HQ CPU @2.80 GHz RAM 16GB

Solver	**CPU time (s)**
ODE45	651.89
ODE23	340.37
ODE113	365.14
ODE15s	8.42
ODE23s	48.87

Chapter 6

Optimizations

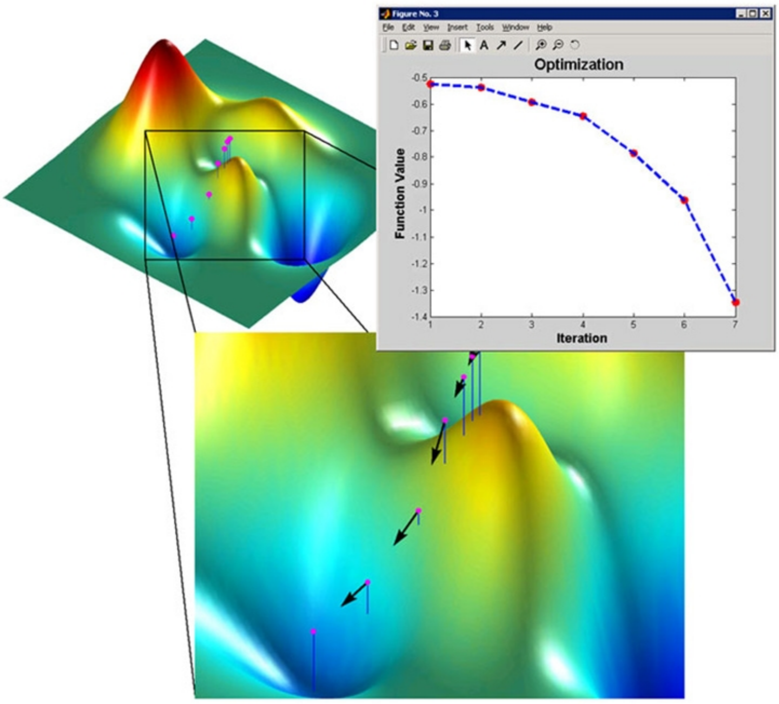


Figure 6.1 Optimization procedure in Matlab™ of a 3D function. In particular this figure represents the gradient path towards the minimum of the function.

The decision to implement the simulator mostly in Matlab™ was based on the fact that it is easier to optimize each phase (but the de-tumbling one, due to high computational time) using a global optimizer with the build-in Parallel Computin Toolbox™, such as *ga.m*, the genetic algorithm.
The optimized parameters are the 2 constants of the PD controller, K_p and K_d. The first step has been to separate between picture acquisition phases and the other ones, since the cost functions are based on different logics:

6.1 Picture acquisition phases

During this phases the pointing error shall be minimized, without taking care about the momentum stored inside the wheels. For this reason a single objective global optimization, using *ga.m* has been performed, with the following cost function:

$$J = \int_{t_0}^{t_f} (pointing\ error - 0.1)^2 \,\mathrm{d}t. \tag{6.1}$$

which is based on the pointing accuracy requested, 0.1° and squared to smooth the function. It is important to remark that only the last 30% of the period between t_0 and t_f has been considered inside J, in order not to consider the transient.

The optimization tolerance has been set to $1e^{-3}$, while the absolute and relative tolerances of the model has been set to $1e^{-6}$ and the initial conditions are generated with a random distribution around $1e^{-3}$ rad/s, in order to increase the generality.

All the optimization algorithms are run in parallel on 4 physical cores of the CPU and the optimization time is strictly related to the integration initial condition (random, with Gaussian distribution). For this reason the values reported in the following tables, concerning the computational time, can vary depending on them.

In table 6.1 the results obtained are summarized:

Table 6.1 Optimized control parameters for all picture acquisition phases.

Processor used: Intel® Core ™ i7-7700HQ CPU @2.80 GHz RAM 16GB

Solver	K_p	K_d	**CPU time (hours)**
-ISS- Nadir pointing	0.1330	0.7762	0.67
-ISS- Target following	0.0441	0.3878	9.8
-SSO- Nadir pointing	0.1420	0.9065	0.5
-SSO- Target following	0.0458	0.4074	12

6.2 Sun-pointing and downloading phases

The remaining phases are characterized by the necessity of reducing the momentum stored by the reaction wheel, maintaining the pointing error less than 1°. For this reason a multi-objective optimization is requested. In order to perform it, the Matlab™ algorithm *gamultiobj.m*, based on a Pareto front optimization has been exploited. The output of this function is the Pareto front, a set of optimum solutions plotted on a graph with x-axis and y-axis characterized respectively by the first and the second cost functions. For what concerns them, two sets of two cost functions have been used and then the solutions obtained have been compared to establish which one is the best:

The first set:

$$\begin{cases} J1 = \int_{t_0}^{t_f} (pointing\ error - 0.1)^2 \,\mathrm{d}t. \\ J2 = \int_{t_0}^{t_f} (momentum\ stored)^2 \,\mathrm{d}t. \end{cases} \tag{6.2}$$

Providing only one solution.

And the second set:

$$\begin{cases} J1 = \int_{t_0}^{t_f} (pointing\ error - 0.1)^2 \,\mathrm{d}t. \\ J2 = \frac{(|momentum\ stored_{f,1}| + |momentum\ stored_{f,2}| + |momentum\ stored_{f,3}|)}{||momentum\ stored_f||} \end{cases} \tag{6.3}$$

based on the final value of the stored momentum instead of its integral, providing a classical Pareto front.

Once the solutions are obtained, the one closest to the bisector of the Pareto front plot is selected as the best one, minimizing almost equally both cost functions.

Finally this solution is compared to the one of the first set and the best one will be inserted inside

the model.
All these optimizations have been limited to last 1 hour as maximum, while the picture acquisition ones did not have such a limitation, since they are the most critical.

6.2.1 Sun-pointing phase

The results for the Sun-pointing phases are the following:

- First set optimum solutions:

Table 6.2 Optimized control parameters for all Sun-pointing phases obtained with cost functions set 1.

Processor used: Intel® Core ™ i7-7700HQ CPU @2.80 GHz RAM 16GB			
Solver	K_p	K_d	**CPU time** (h)
ISS	0.0319	0.3517	1
SSO	0.0345	0.3805	1

- Second set optimum solutions on Pareto front: the best solution is considered the one nearest the bisector, once the axes have been normalized.

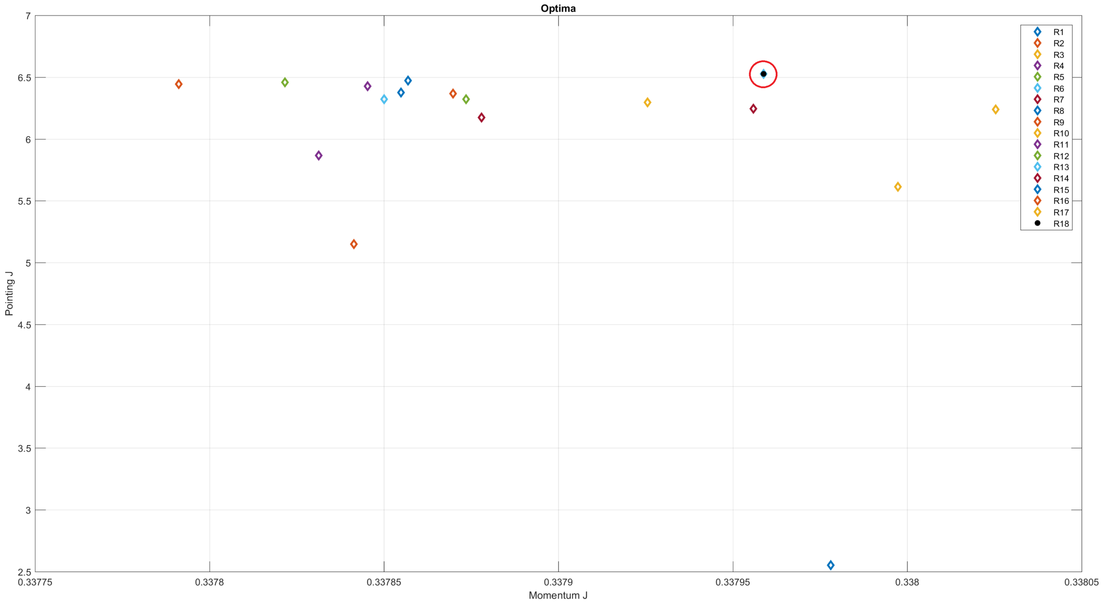


Figure 6.2 Pareto front for Sun-pointing phase in ISS orbit. The point closest to the bisector has been selected as the best and it is circled.

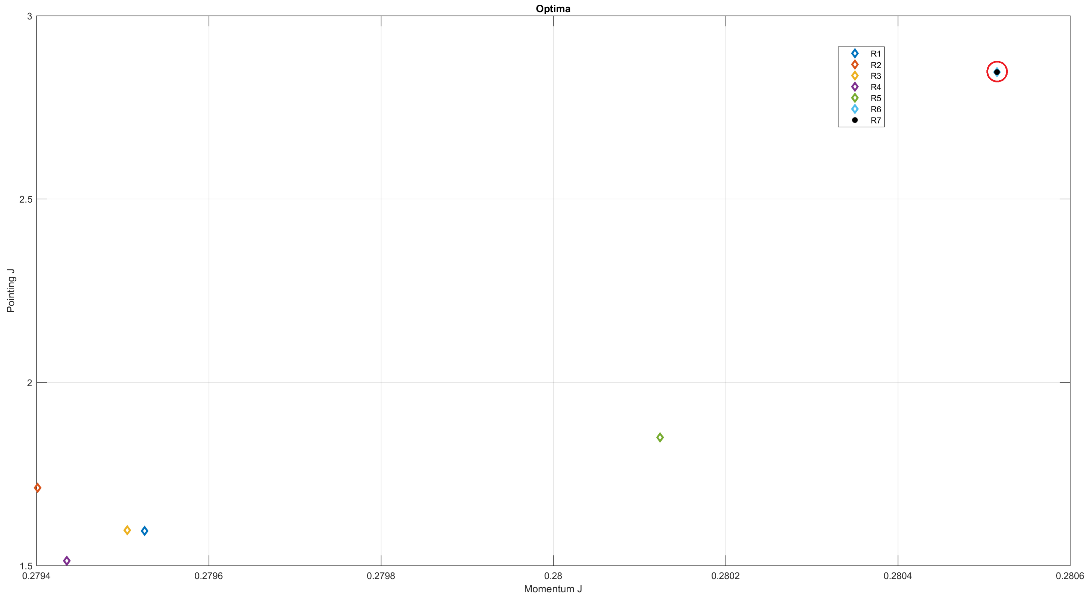


Figure 6.3 Pareto front for Sun-pointing phase in SSO orbit. The point closest to the bisector has been selected as the best and it is circled.

- Comparison between best solution from set 2 and the one from set 1: the best behavior on ISS orbit is obtained with the optimized parameters from set 1, since the momentum stored during the transient phase is reduced and the pointing accuracy is the same. On SSO, the trend is almost the same and, as a consequence, it would be a good choice to use the parameters coming from set 1 as well, but since the pointing accuracy is slightly better in the case of set 2 parameters, it has been decided to use them. In any case the difference between the two sets evidenced in simulation results is almost negligible.

In table 6.3 the optima are reported:

Table 6.3 Optimized control parameters for all Sun-pointing phases.

Processor used: Intel® Core ™ i7-7700HQ CPU @2.80 GHz RAM 16GB			
Solver	K_p	K_d	**CPU time** (min)
ISS	0.0319	0.3517	-
SSO	0.0701	0.5278	-

6.2.2 Download phase

The results for the downloading phases are the following:

- First set optimum solutions, equal for both of the orbits:

Table 6.4 Optimized control parameters for all downloading phases obtained with cost functions set 1.

Processor used: Intel® Core ™ i7-7700HQ CPU @2.80 GHz RAM 16GB			
Solver	K_p	K_d	**CPU time** (h)
ISS	0.0345	0.3805	1
SSO	0.0345	0.3805	1

- Second set optimum solutions on Pareto front: the best solution is considered the one nearest the bisector, once the axes have been normalized.

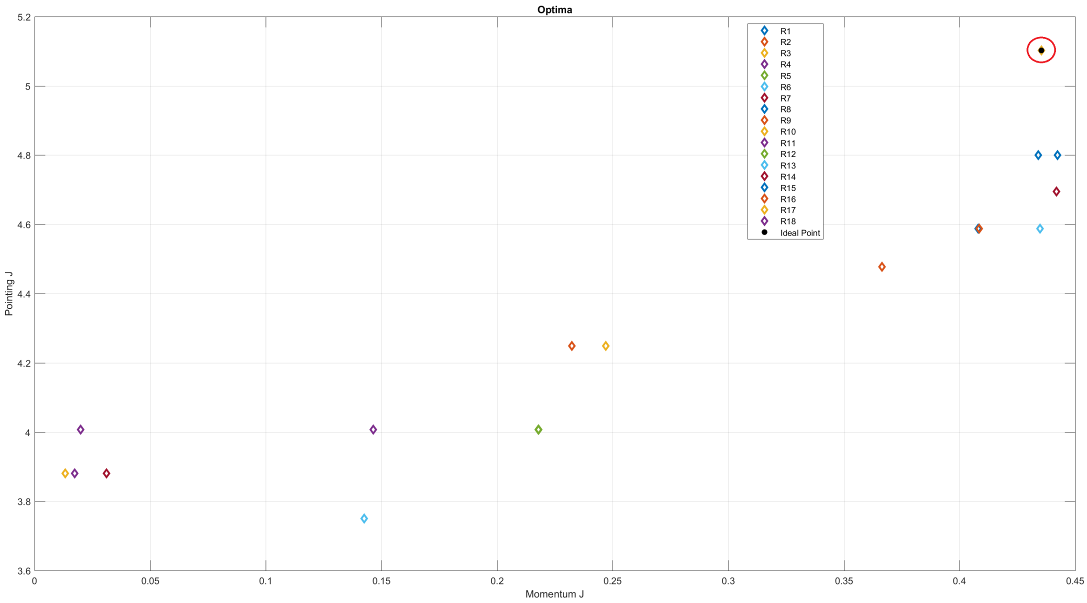


Figure 6.4 Pareto front for Sun-pointing phase in ISS orbit. The point closest to the bisector has been selected as the best and it is circled.

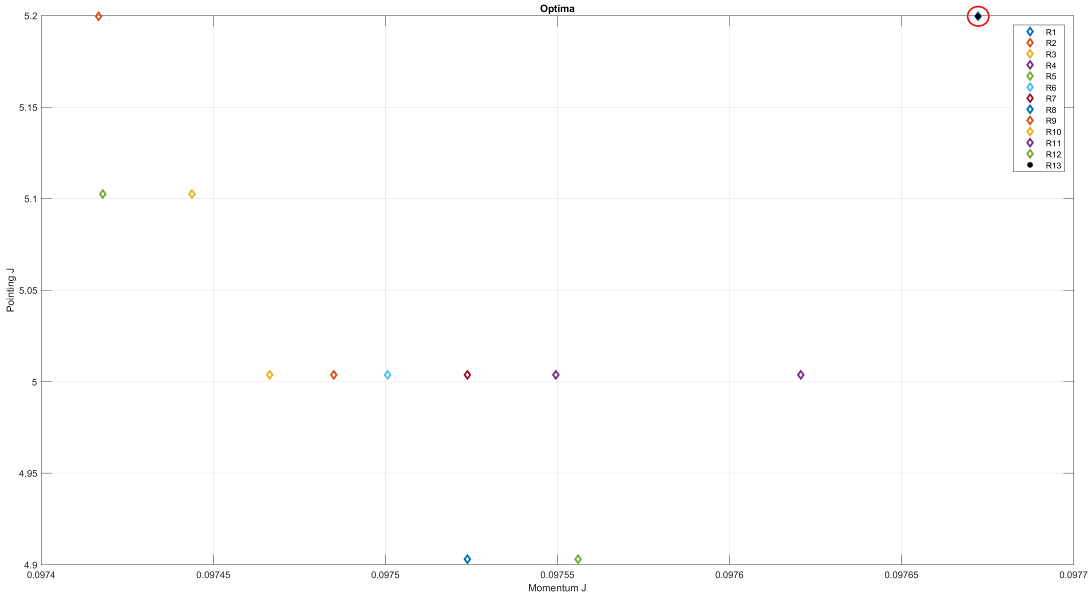


Figure 6.5 Pareto front for Sun-pointing phase in SSO orbit. The point closest to the bisector has been selected as the best and it is circled.

- Comparison between best solution from set 2 and the one from set 1: the best behavior on ISS orbit is obtained with the optimized parameters from set 1, while on SSO, parameters from set 2 are the ones providing the best results in terms of momentum stored and pointing accuracy, even if the behavior are really similar.

In table 6.5 the optima are reported:

Table 6.5 Optimized control parameters for all downloading phases.

Processor used: Intel® Core ™ i7-7700HQ CPU @2.80 GHz RAM 16GB

Solver	K_p	K_d	**CPU time** (min)
ISS	0.0345	0.3805	-
SSO	0.0350	0.3461	-

Chapter 7

Results

In this chapter the main results, subdivided per orbit, mission and phases will be presented and commented. Mission 2 results will be presented only in the appendix for both kind of orbit (ISS and SSO), in target following pointing, since they are always very similar to the ones of mission 1. Moreover, the results on ISS orbit will be more commented than the ones on SSO, since they are very similar, with the exception of the disturbances order of magnitude.
It has been decided to report the majority of the results, instead of selecting only some phases, just to let the reader and who will carry on this project to have a collection of results representing the nominal behavior for complete missions, both in ISS and SSO, which can be used as benchmark or for comparison with future results.
However it is important to remark that during long Sun-pointing phases in mission 1 and 2, it has been decided not to report all the desaturation phases requested every time the momentum stored reaches its maximum value. As a result, the phases have been run as no saturation occurs and then, just before the pointing and downloading phases, a de-saturation phase is shown.

7.1 ISS

7.1.1 Mission 1 -Nadir pointing-

Computational time usually requested: around 3 minutes, depending on the initial conditions.

De-tumbling After the de-tumbling phase, the angular velocities reached are:

$$\begin{cases} \omega_{b/n,1}(5000) = -5.868e^{-3} \text{ rad/s} \\ \omega_{b/n,2}(5000) = 4.506e^{-3} \text{ rad/s} \\ \omega_{b/n,3}(5000) = 1.663e^{-3} \text{ rad/s} \end{cases} \tag{7.1}$$

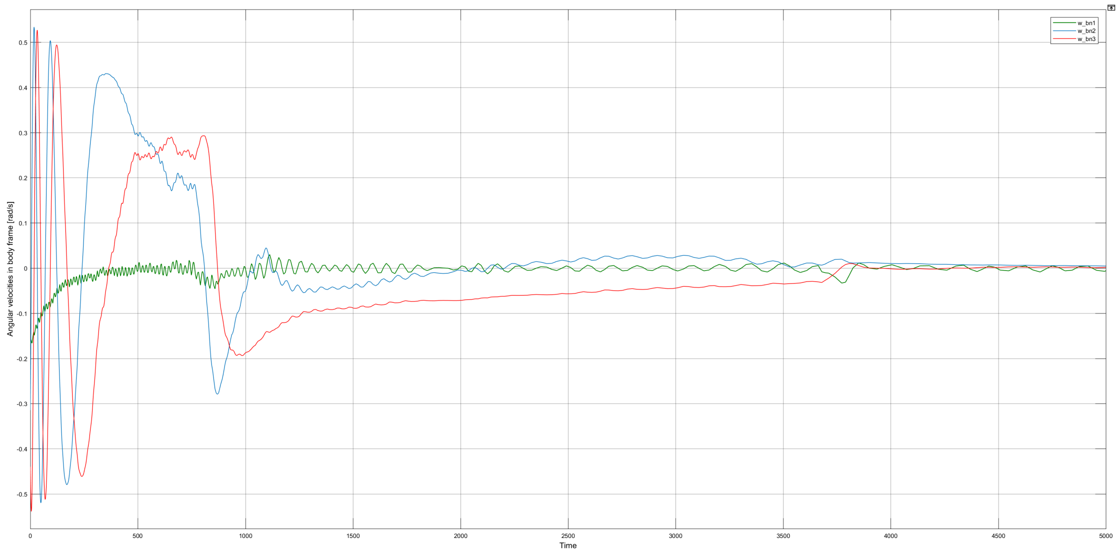


Figure 7.1 Angular velocities reduction during de-tumbling phase, in mission 1, on ISS orbit.

The disturbances felt by OUFTI-Next are instead represented in figure 7.2.

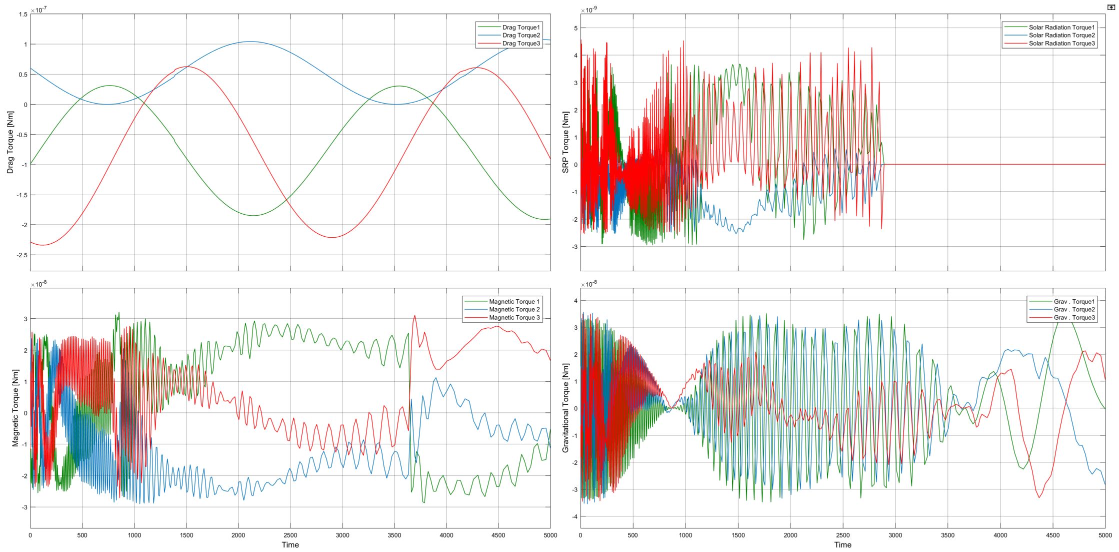


Figure 7.2 Main disturbances during de-tumbling phase, in mission 1, on ISS orbit.

Comparing this values with figure 7.3 on the facing page, representing the common disturbances torques for a satellite, it is possible to state that for a significantly smaller platform, like a 3U CubeSat, the values obtained are absolutely reasonable.

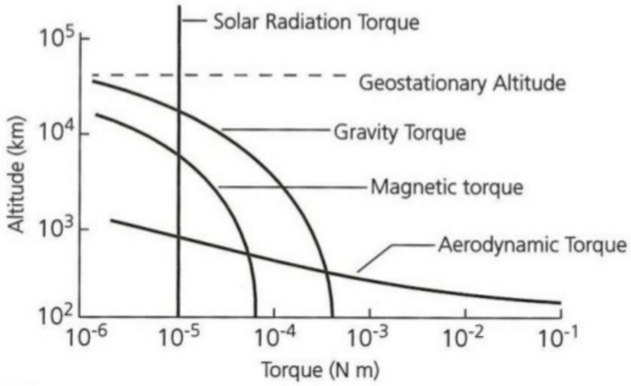


Figure 7.3 Common environmental disturbances for satellites orbiting around the Earth.

The solver relative tolerances have been set to $1e^{-3}$ just to speed up the simulation, since no lack of accuracy in the solution was evidenced.

Sun-pointing 1 From figure 7.4 on the next page it possible to appreciate that the star tracker based algebric determination algorithm works very well in estimating the attitude.
The length of this phase is 24 hours plus the time spent in orbit before arriving 30° from the target: the National Park of Brasilia.
The phase lasts around 74 hours in total and the solver absolute and relative tolerances have been set to $1e^{-6}$ both. The steady state is reached after around 100 s.
There is a peculiar behavior of the quaternions, evidenced in figure 7.4 on the following page: the quaternions estimated by the determination algorithm are the opposite of the actual ones. This is due to the quaternion normalization procedure, during state estimation, providing in some simulations the opposite of the integrated quaternion, but this is not altering the dynamics, since quaternions are not unique and even if they are the opposite, they represent the same rotation in space. Moreover, using the control law reported in the previous chapter, this does not affect at all the control action. This behavior will be present also in following simulations.

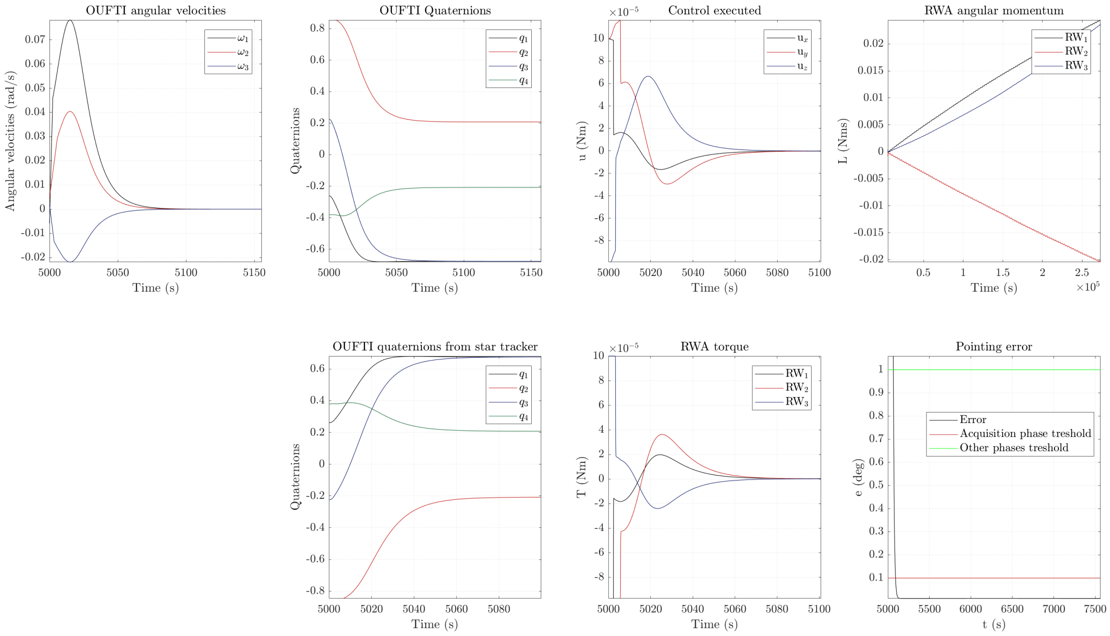


Figure 7.4 Most important results during Sun-pointing phase 1, in mission 1, on ISS orbit. The plots have been zoomed in order to catch the most interesting behaviors.

From figure 7.4 it is also possible to notice that the control executed and the torque provided by the reaction wheels is not the same and the behavior is almost the opposite. This is due to the fact that the actual control is provided only partially by the derivative of the reaction wheel momentum, $\dot{\boldsymbol{h}}$, represented in figure 7.4. The complete formulation is here recalled:

$$\boldsymbol{u} = \boldsymbol{A}\boldsymbol{h_r} \times \boldsymbol{\omega} - \boldsymbol{A}\dot{\boldsymbol{h}}_{\boldsymbol{r}} \tag{7.2}$$

and, as a consequence of the minus sign, the two behaviors are the opposite.

In figure 7.5 on the next page is possible to observe that the results in terms of disturbances are consistent with respect to the one obtained with Simulink™, figure 7.2 on page 60.

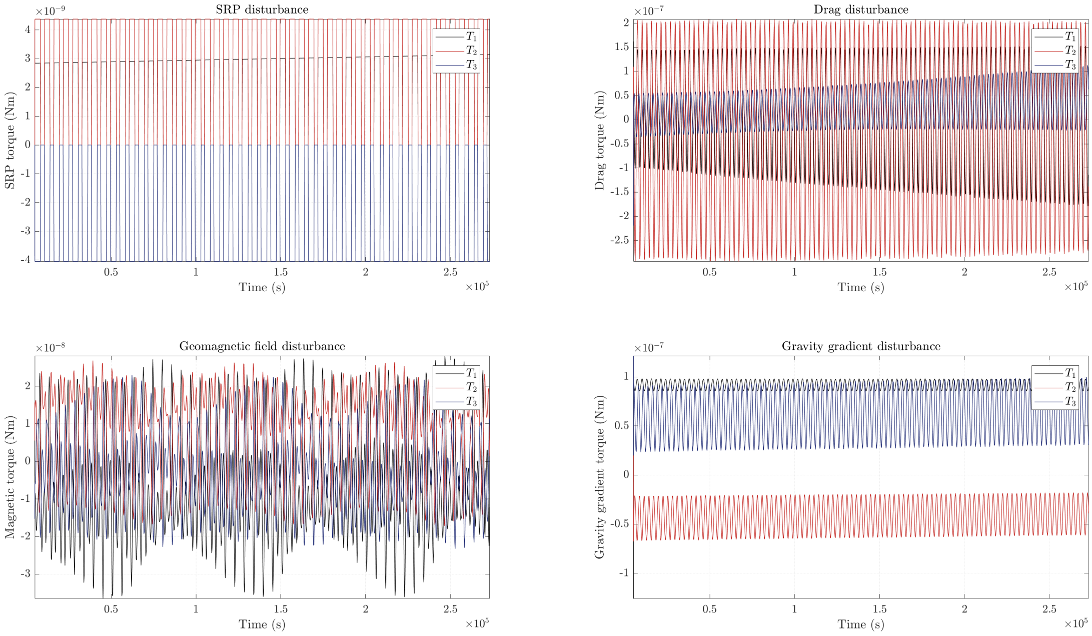


Figure 7.5 Main disturbances torques during Sun-pointing phase 1, in mission 1, on ISS orbit.

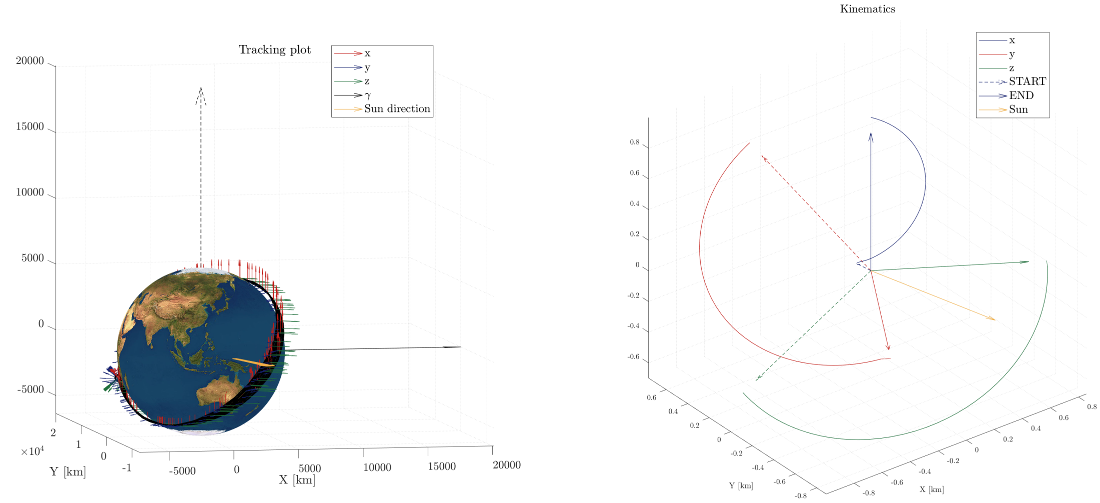


Figure 7.6 Body-fixed frame while orbiting around the earth during Sun-pointing phase 1, in mission 1, on ISS orbit.

De-saturation 1 From figure 7.7 on the next page it is clear that the angular velocities during de-saturation phase do not spread so much, making the spacecraft rotate very slowly.
The phase lasts 1 minute and the solver absolute and relative tolerances have been set to $1e^{-6}$ both.

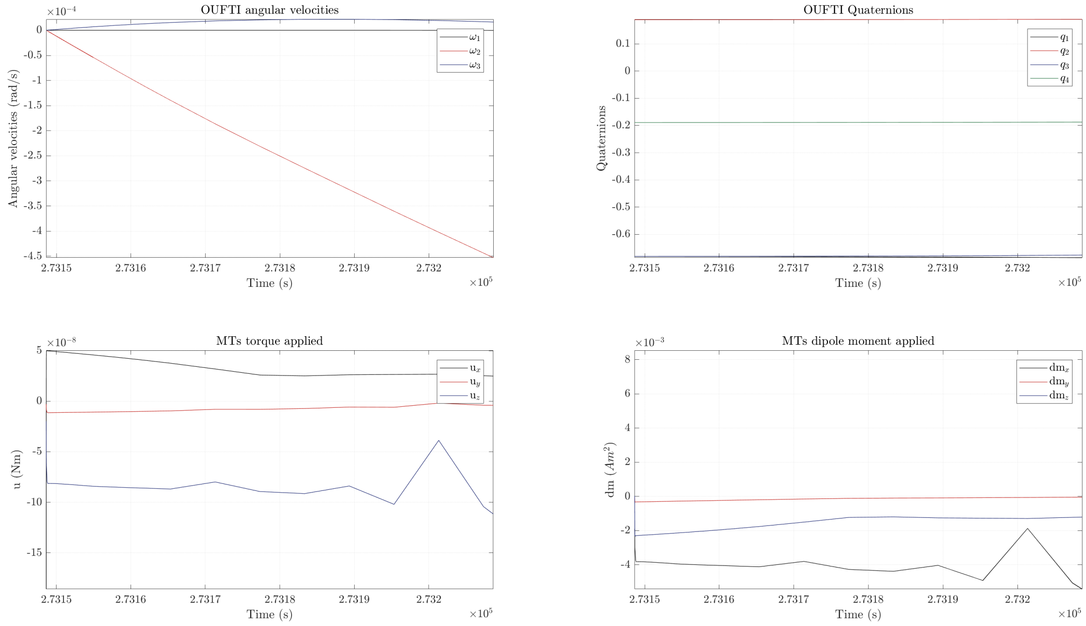


Figure 7.7 Most important results during de-saturation phase 1, in mission 1, on ISS orbit.

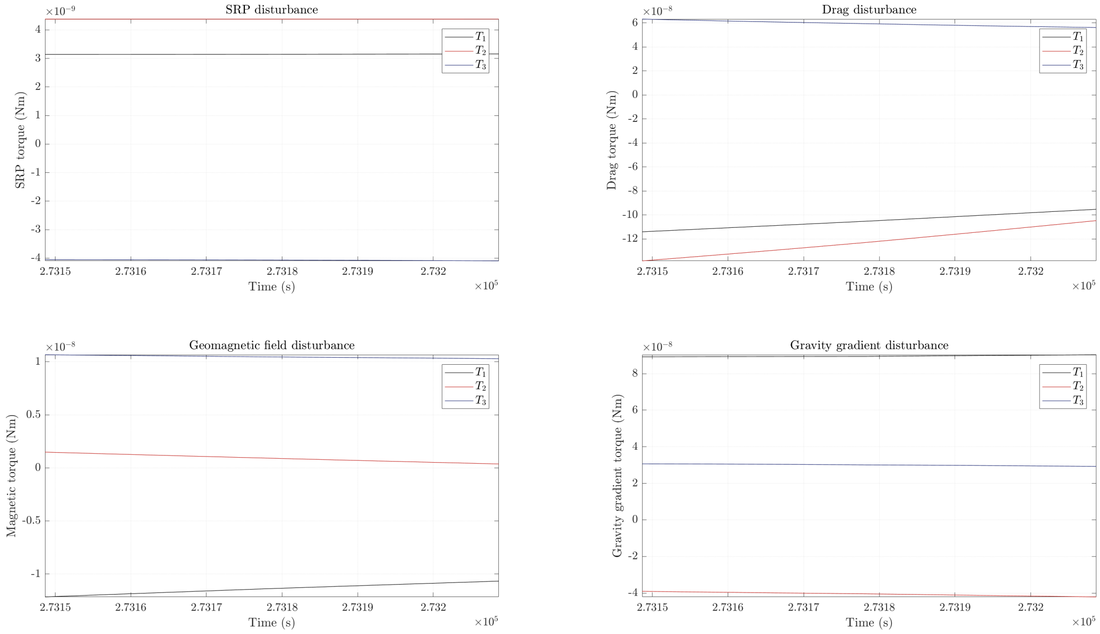


Figure 7.8 Main disturbances torques during de-saturation phase 1, in mission 1, on ISS orbit.

Picture acquisition The picture acquisition phase is the one with most stringent pointing requirements, but, as shown in figure 7.9 on the facing page, the control action is able to maintain the pointing error below the 0.1° threshold, thanks to the optimization of control parameters. The phase lasts around 15 minutes and the solver absolute and relative tolerances have been set to $1e^{-6}$ both. The steady state is reached in less than 100 s.

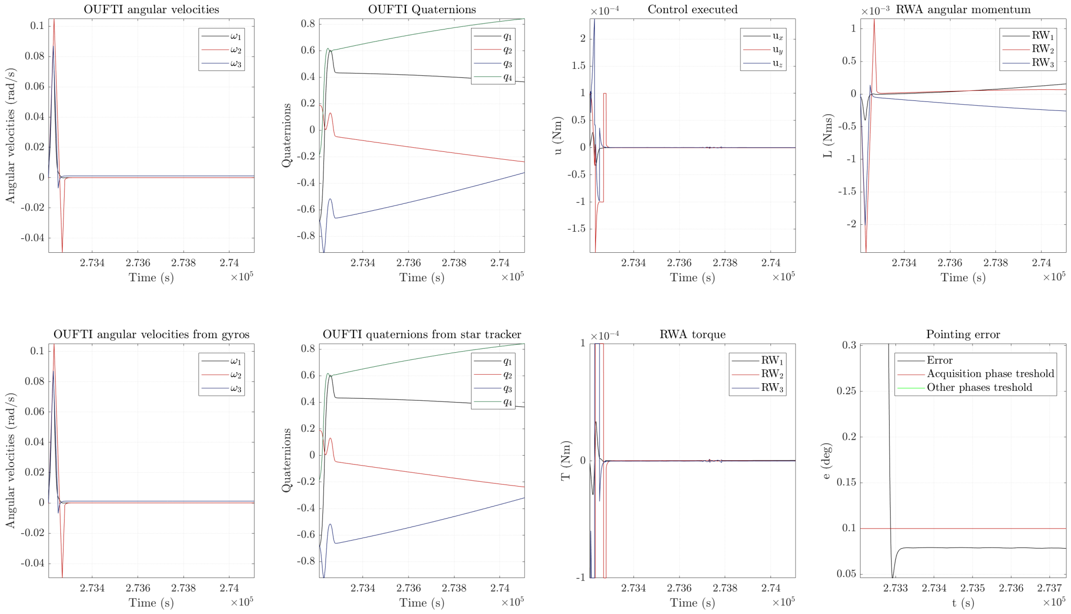


Figure 7.9 Most important results during Nadir pointing picture acquisition phase, in mission 1, on ISS orbit. The plots have been zoomed in order to catch the most interesting behaviors.

From figure 7.9 it is also possible to appreciate how well *the unscented Kalman filter* estimates the state, comparing the integrated angular velocities and quaternion plots with the ones provided by the gyros/star trackers and estimated by the filter.

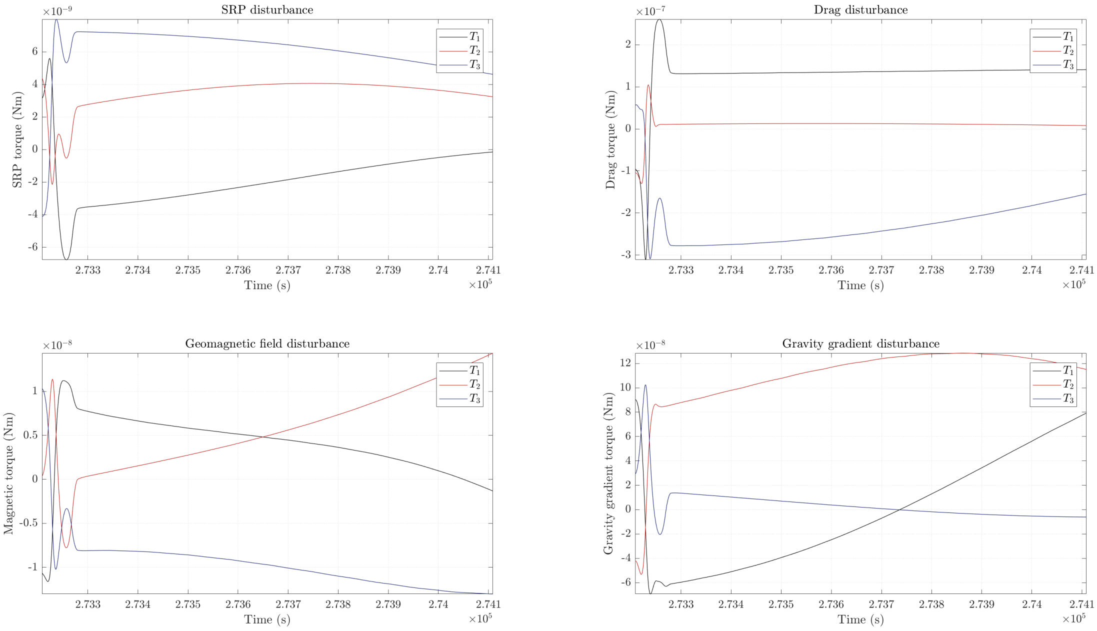


Figure 7.10 Main disturbances torques during Nadir pointing picture acquisition phase, in mission 1, on ISS orbit.

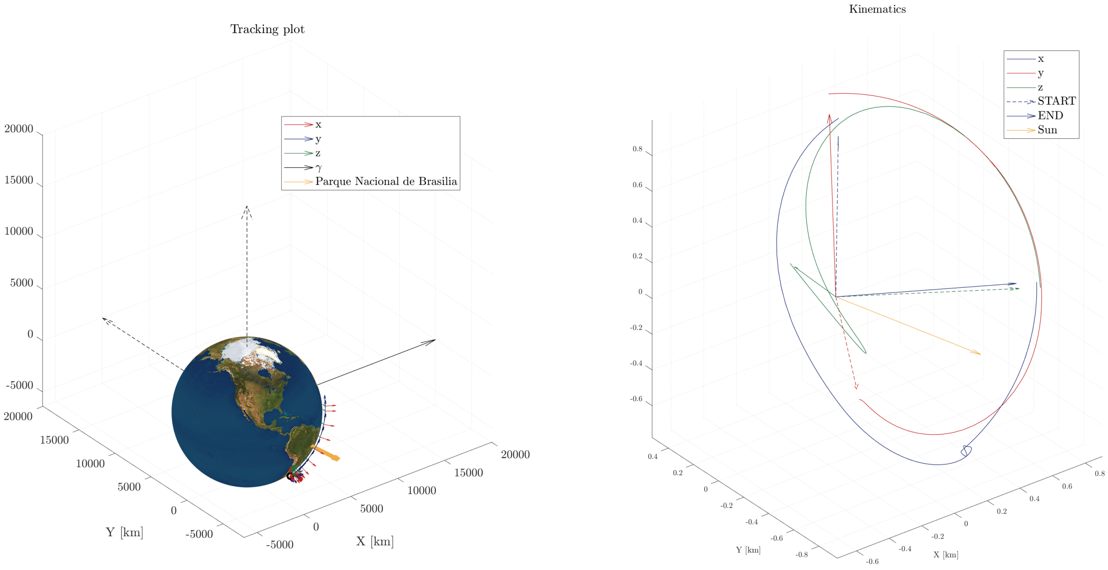


Figure 7.11 Body-fixed frame while orbiting around the earth during Nadir pointing picture acquisition phase, in mission 1, on ISS orbit.

Sun-pointing 2 This phase is quite similar to the previous one in Sun-pointing. The phase lasts less than 2 hours, the time needed to reach 10° from Liège, and the solver absolute and relative tolerances have been set to $1e^{-6}$ both. The steady state is reached after around 90 s.

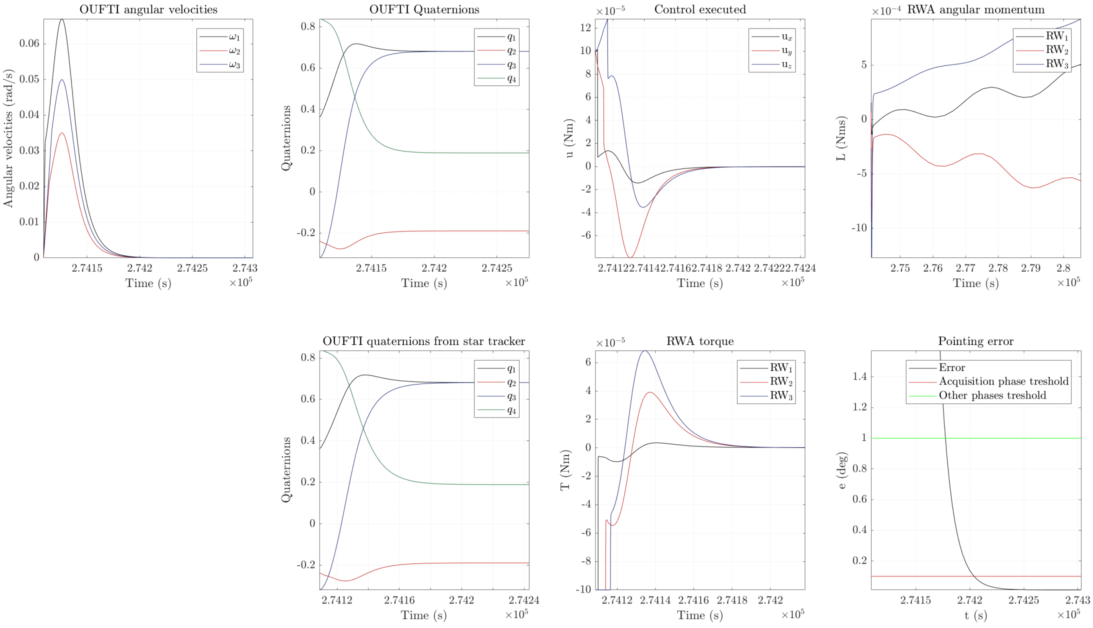


Figure 7.12 Most important results during Sun-pointing phase 2, in mission 1, on ISS orbit. The plots have been zoomed in order to catch the most interesting behaviors.

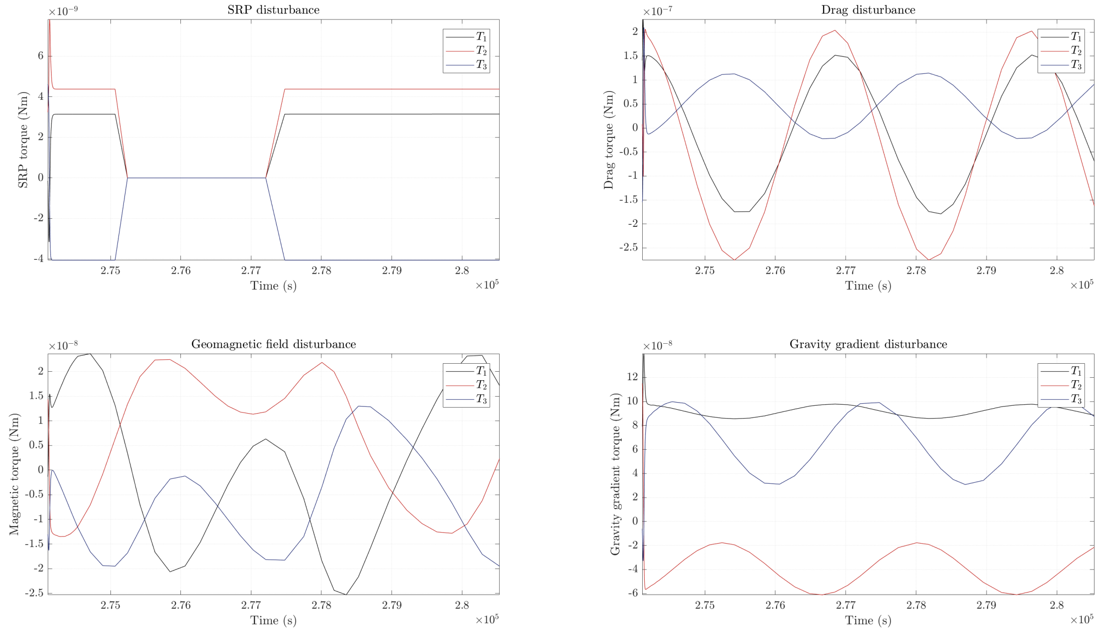


Figure 7.13 Main disturbances torques during Sun-pointing phase 2, in mission 1, on ISS orbit.

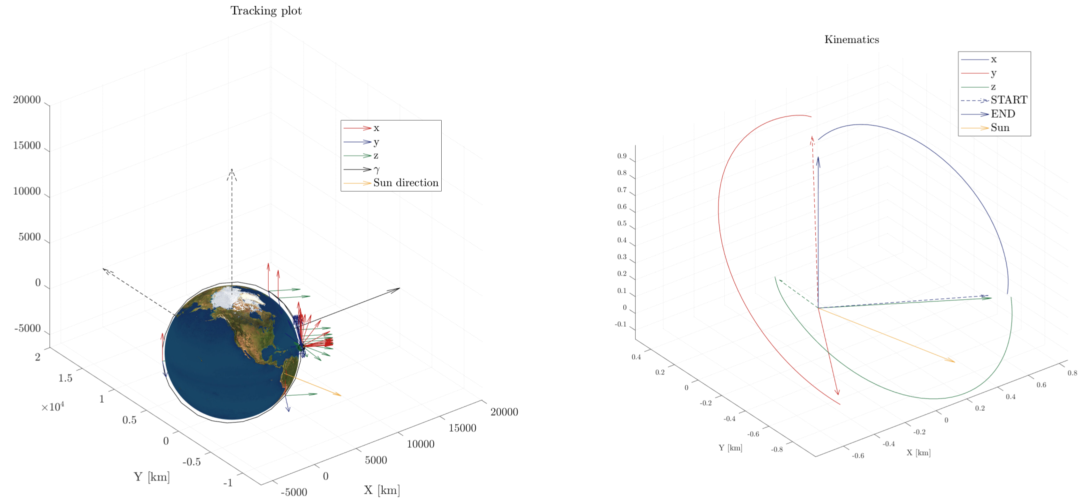


Figure 7.14 Body-fixed frame while orbiting around the earth during Sun-pointing phase 2, in mission 1, on ISS orbit.

De-saturation 2 This results are quite similar to the ones of the previous de-saturation phase and confirm that the spacecraft rotates very slowly and does not tumble. The phase lasts 1 minute and the solver absolute and relative tolerances have been set to $1e^{-6}$ both.

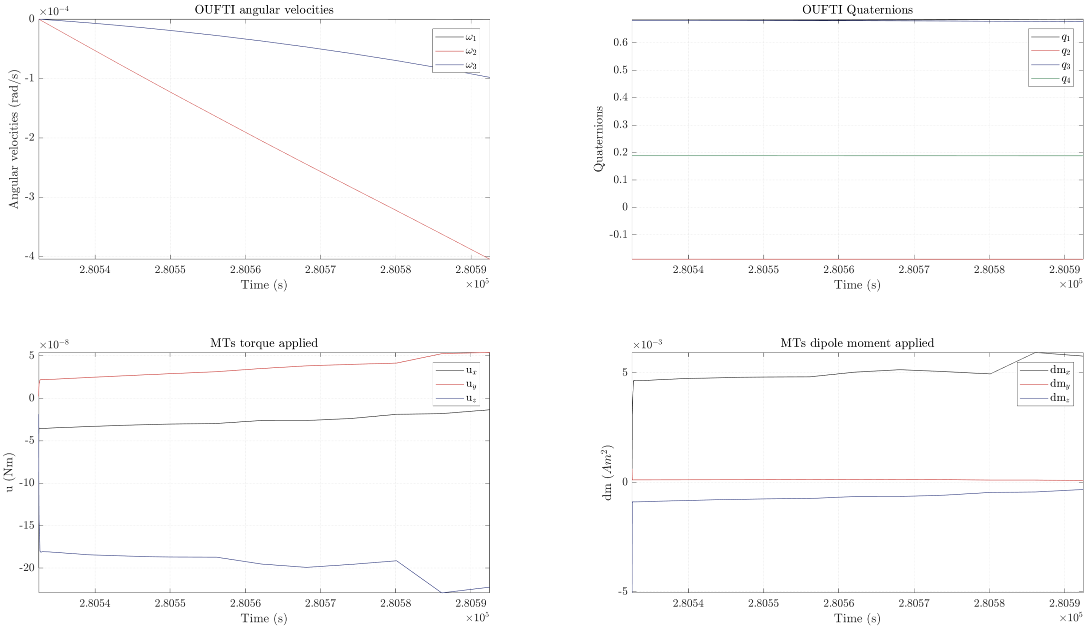


Figure 7.15 Most important results during de-saturation phase 2, in mission 1, on ISS orbit.

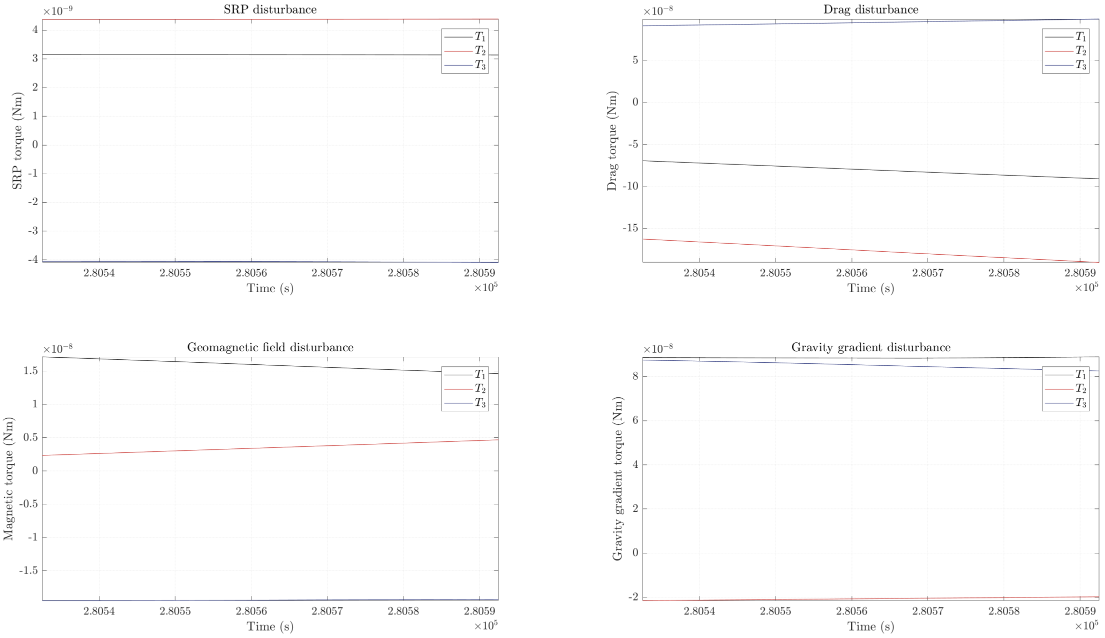


Figure 7.16 Main disturbances torques during de-saturation phase 2, in mission 1, on ISS orbit.

Downloading Downloading phase above Liège lasts 5 minutes and the solver absolute and relative tolerances have been set to $1e^{-6}$ both. The steady state is reached after around 150 s. From figure 7.17 on the facing page it is evident the peculiar behavior of the estimated quaternions, already mentioned in Sun-pointing phase 1.

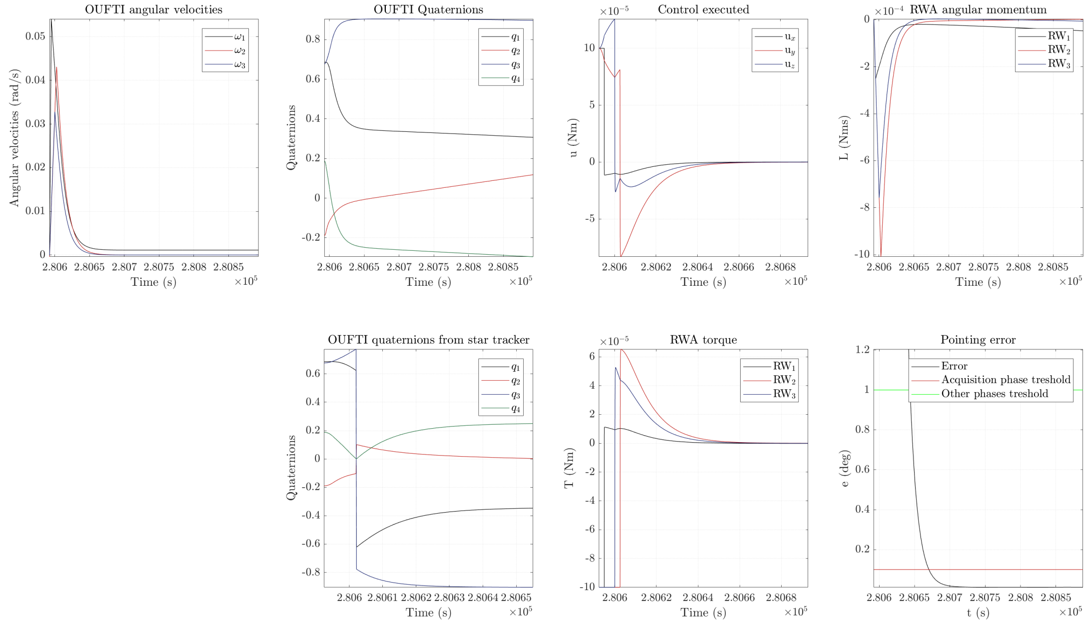


Figure 7.17 Most important results during downloading phase, in mission 1, on ISS orbit. The plots have been zoomed in order to catch the most interesting behaviors.

In the middle of downloading phase, the CubeSat enters in eclipse, as can be seen from solar radiation plot in figure 7.18.

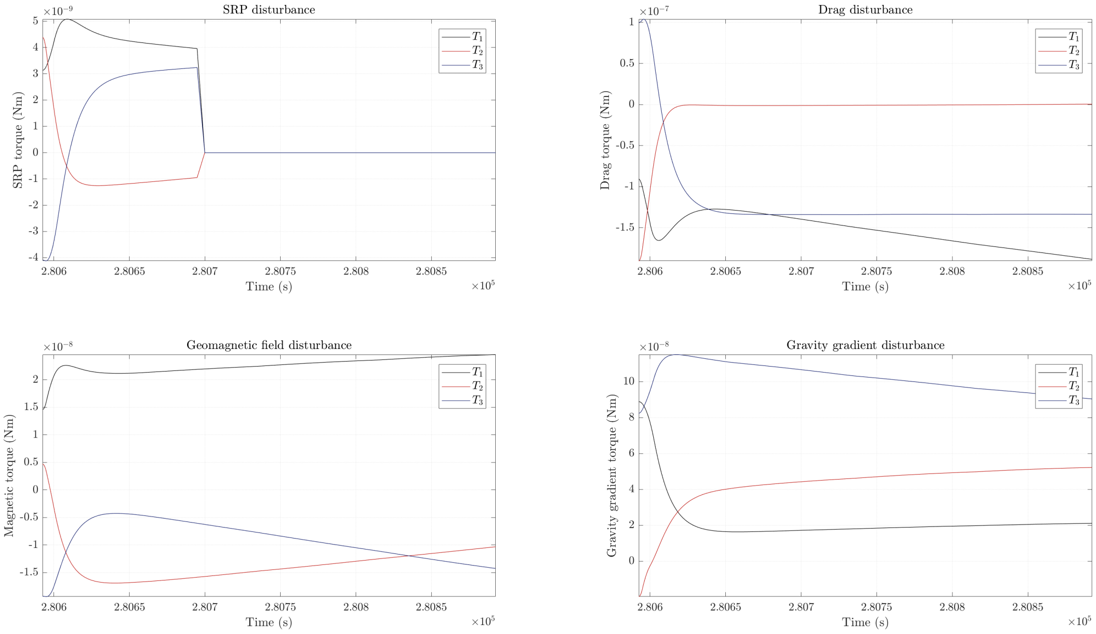


Figure 7.18 Main disturbances torques during downloading phase, in mission 1, on ISS orbit. The plots have been zoomed in order to catch the most interesting behaviors.

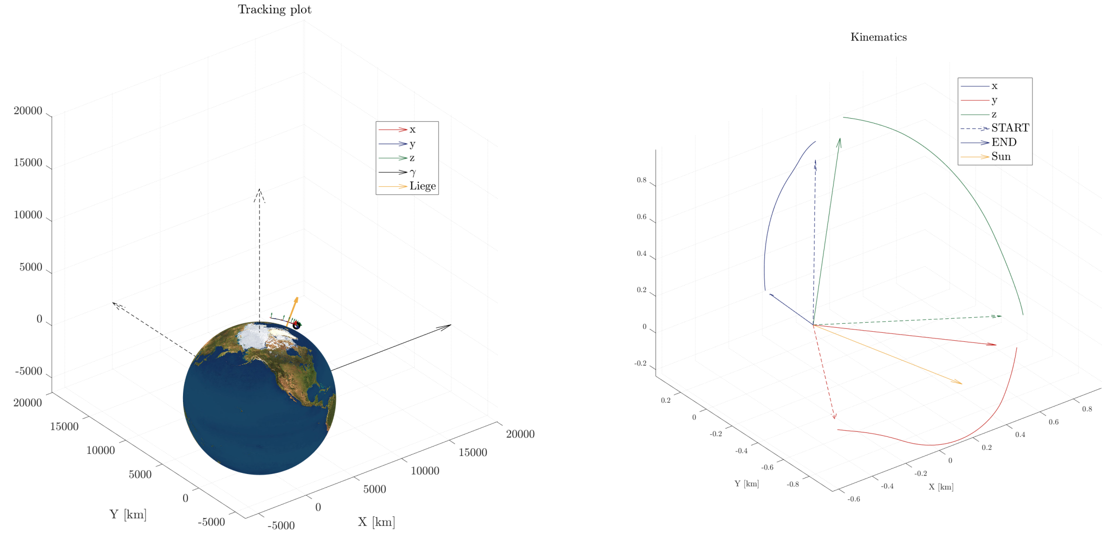


Figure 7.19 Body-fixed frame while orbiting around the earth during downloading phase, in mission 1, on ISS orbit.

7.1.2 Mission 3

Computational time usually requested: around 1 hour and 15 minutes, depending on the initial conditions.

De-tumbling After the de-tumbling phase, the angular velocities reached are:

$$\begin{cases} \omega_{b/n,1}(5000) = -8.140e^{-3} \text{ rad/s} \\ \omega_{b/n,2}(5000) = -9.977e^{-4} \text{ rad/s} \\ \omega_{b/n,3}(5000) = 3.509e^{-3} \text{ rad/s} \end{cases} \tag{7.3}$$

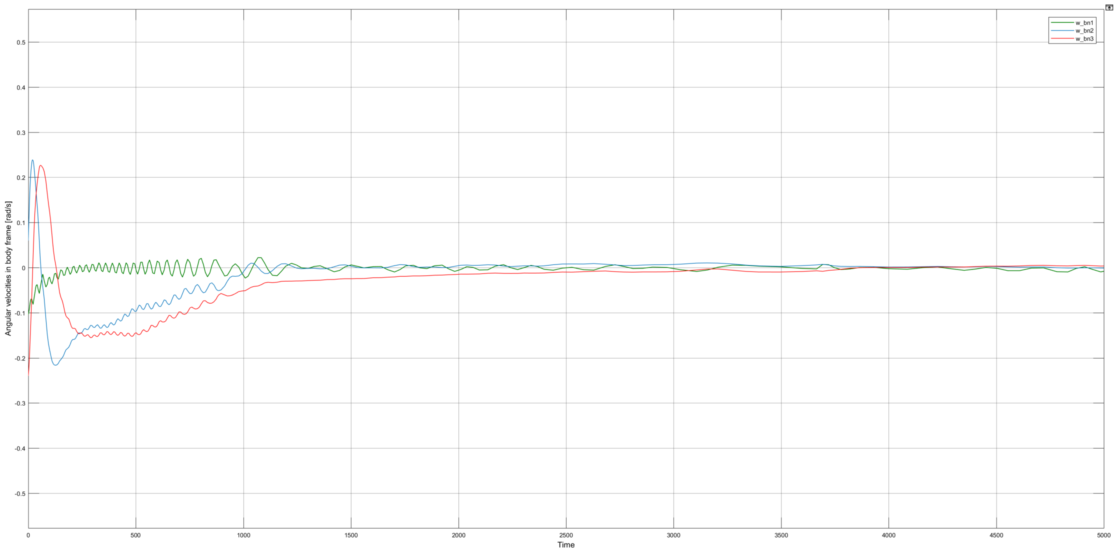


Figure 7.20 Angular velocities reduction during de-tumbling phase, in mission 3, on ISS orbit.

The disturbances felt by OUFTI-Next are instead represented in figure 7.21.

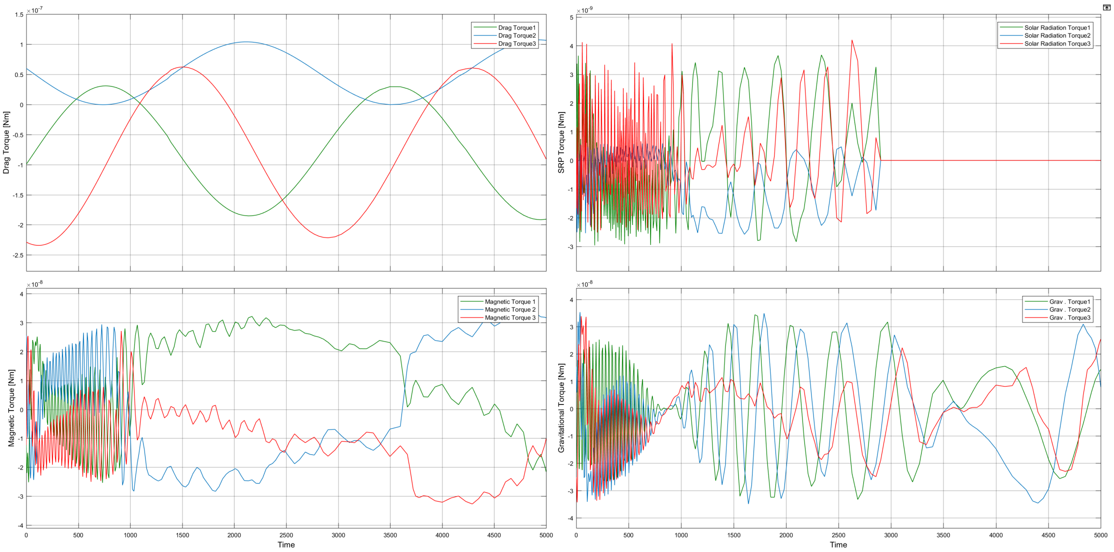


Figure 7.21 Main disturbances during de-tumbling phase, in mission 3, on ISS orbit.

Sun-pointing The Sun-pointing phase was set to last till the momentum stored by the wheels will reach the maximum value available.
The simulation lasts around 17 hours, the maximum time before saturation, with solver absolute and relative tolerances set to $1e^{-6}$ both. The steady state is reached after around 100 s.

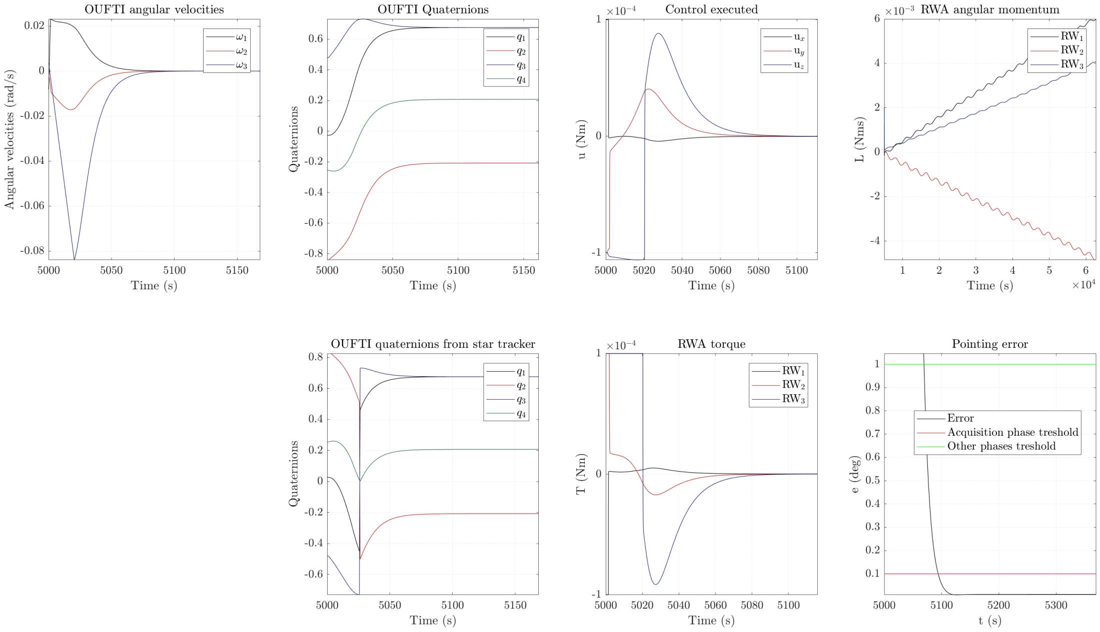


Figure 7.22 Most important results during Sun-pointing phase in mission 3, on ISS orbit. The plots have been zoomed in order to catch the most interesting behaviors.

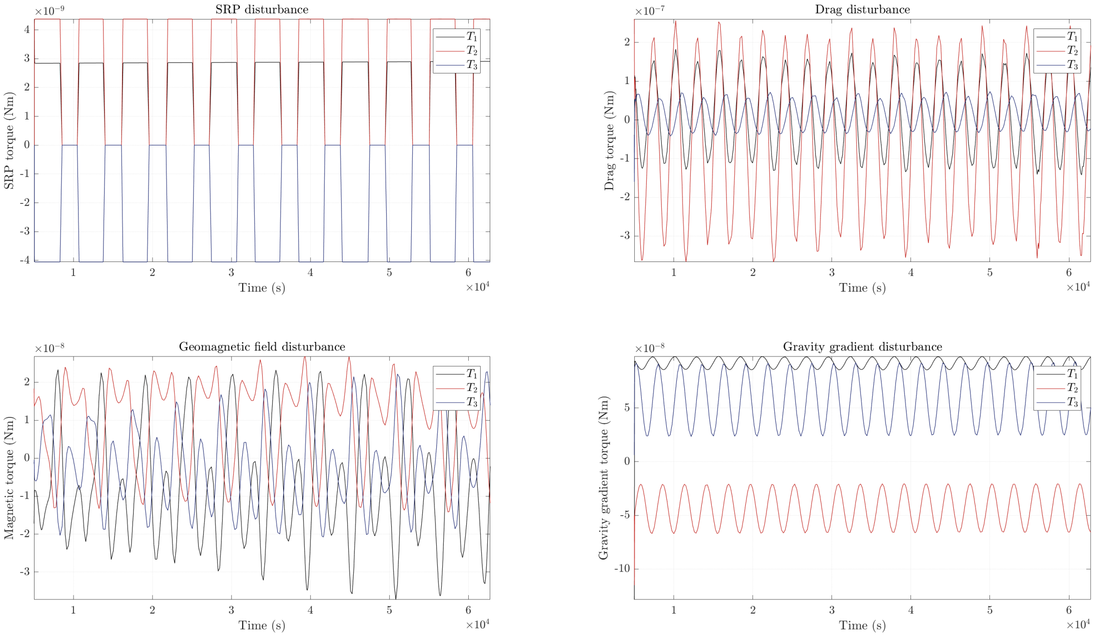


Figure 7.23 Main disturbances torques during Sun-pointing phase in mission 3, on ISS orbit.

De-saturation De-saturation phase lasts 1 day, just to show the behavior of the spacecraft during a longer de-saturation phase.

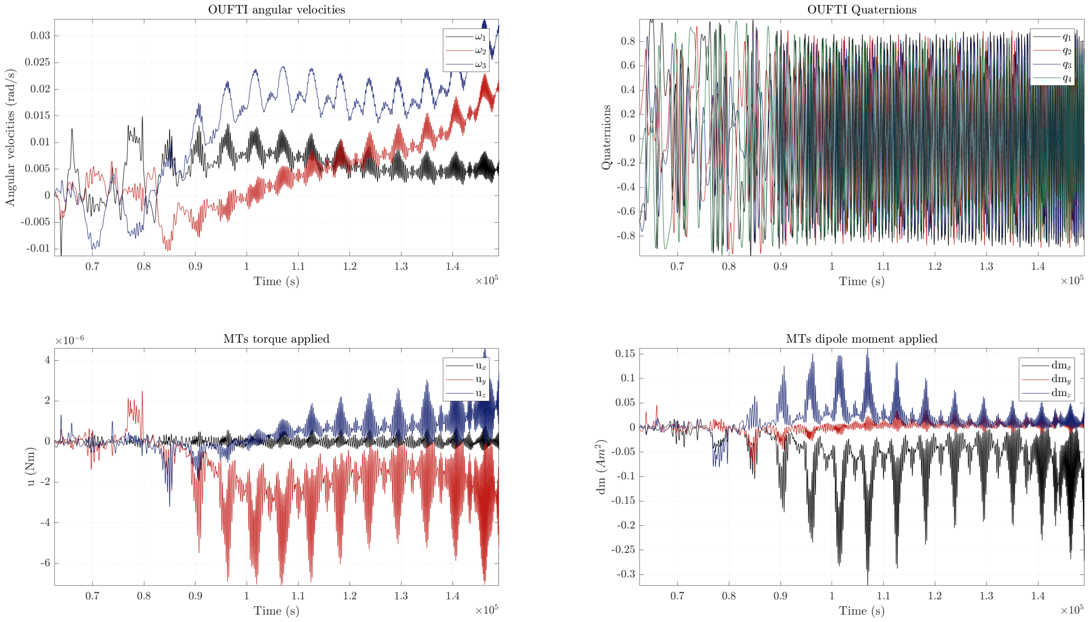


Figure 7.24 Most important results during de-saturation phase in mission 3, on ISS orbit.

From figure 7.24 it can be shown that the angular velocities of the platform increase very slowly, meaning that it is still possible to store solar energy during shorter de-saturation phases.

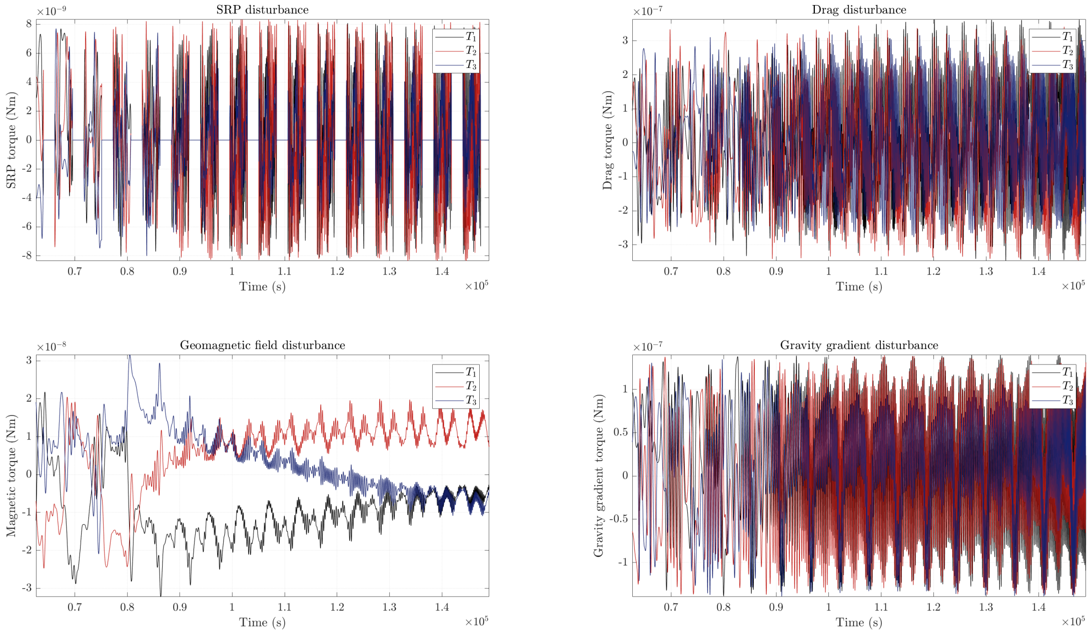


Figure 7.25 Main disturbances torques during de-saturation phase, in mission 3, on ISS orbit.

7.1.3 Mission 1 -Target following pointing-

For what concerns mission 1 performed with target following pointing, it has been thought to be tedious to present the results of all the phases, since they are similar to the ones already presented in mission 1 for Nadir pointing. As a result only the picture acquisition phase will be presented. In any case in appendix it is reported the complete mission 2 with target following pointing, if the reader wants to go into the details of the whole mission.
The computational time requested to simulate the whole mission is usually around 10 minutes, depending on the initial conditions.

Picture acquisition From figure 7.26 on the following page it is possible to appreciate the angular velocity around Z-axis increasing and decreasing while OUFTI is passing over the target, rotating in order to follow the National Park of Brasilia. A similar behavior is shown in the control torque and reaction wheels torque plots. The solver absolute and relative tolerances are set to $1e^{-6}$.

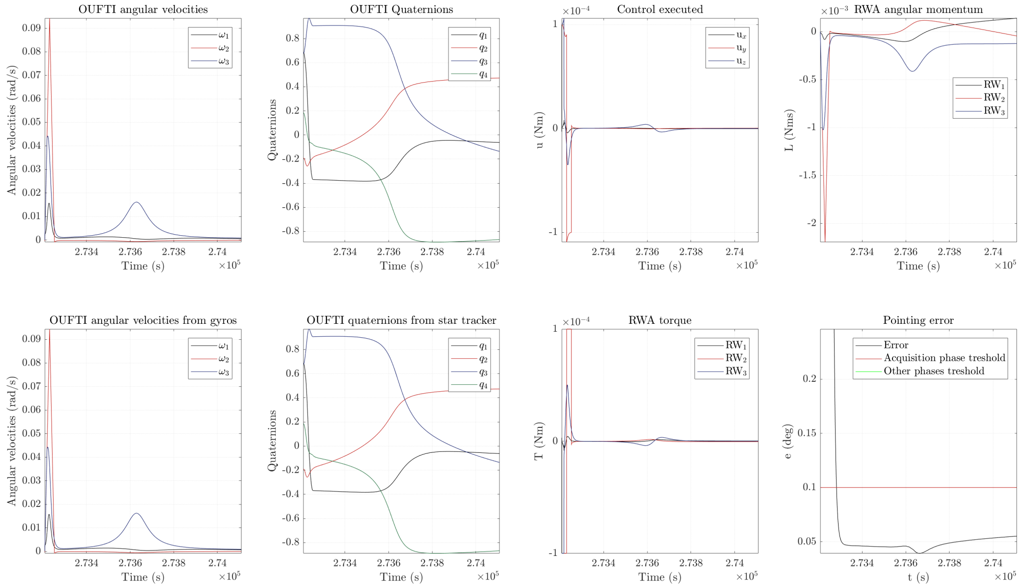


Figure 7.26 Most important results during target following picture acquisition phase, in mission 1, on ISS orbit. The plots have been zoomed in order to catch the most interesting behaviors.

Another peculiar trend is the one evidenced in figure 7.26, where the pointing error drops smoothly in almost 1 minute, exactly when OUFTI is passing above the Brazilian National Park. This behavior can be explained looking at figure 7.27, representing the disturbances level during the acquisition phase: when the CubeSat is approaching the nadir position with respect to the target, the disturbances torques are increasing in magnitude, reaching their maximum when OUFTI is right above the Brazilian National Park and then start decreasing, following exactly the behavior of the pointing error plot.

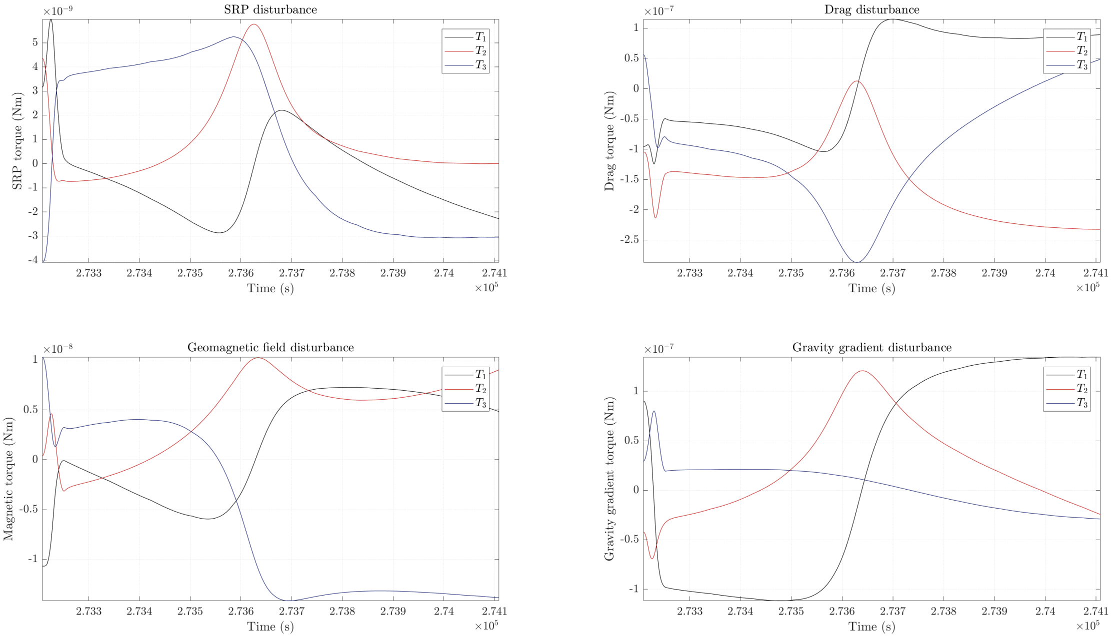


Figure 7.27 Main disturbances torques during target following picture acquisition phase, in mission 1, on ISS orbit.

From figure 7.28 instead, it is possible to see the red vector (the X-axis of the *body-fixed reference frame*) following the National Park of Brasilia.

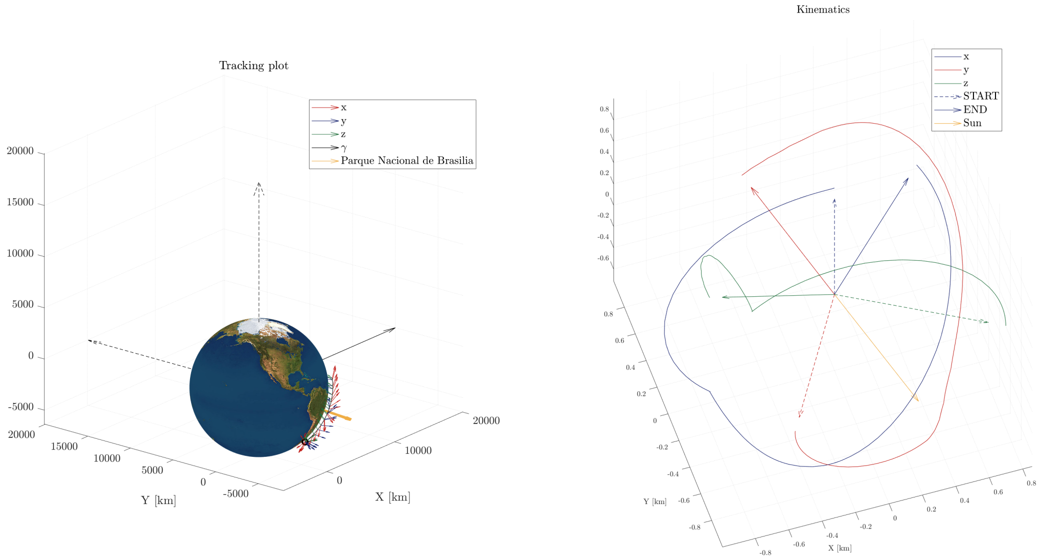


Figure 7.28 Body-fixed frame while orbiting around the earth during target following picture acquisition phase, in mission 1, on ISS orbit.

7.2 SSO

7.2.1 Mission 1 -Nadir pointing-

Computational time usually requested: around 3 minutes, depending on the initial conditions.

De-tumbling After the de-tumbling phase, the angular velocities reached are:

$$\begin{cases} \omega_{b/n,1}(5000) = 8.499e^{-4} \text{ rad/s} \\ \omega_{b/n,2}(5000) = 8.312e^{-5} \text{ rad/s} \\ \omega_{b/n,3}(5000) = -7.310e^{-3} \text{ rad/s} \end{cases} \tag{7.4}$$

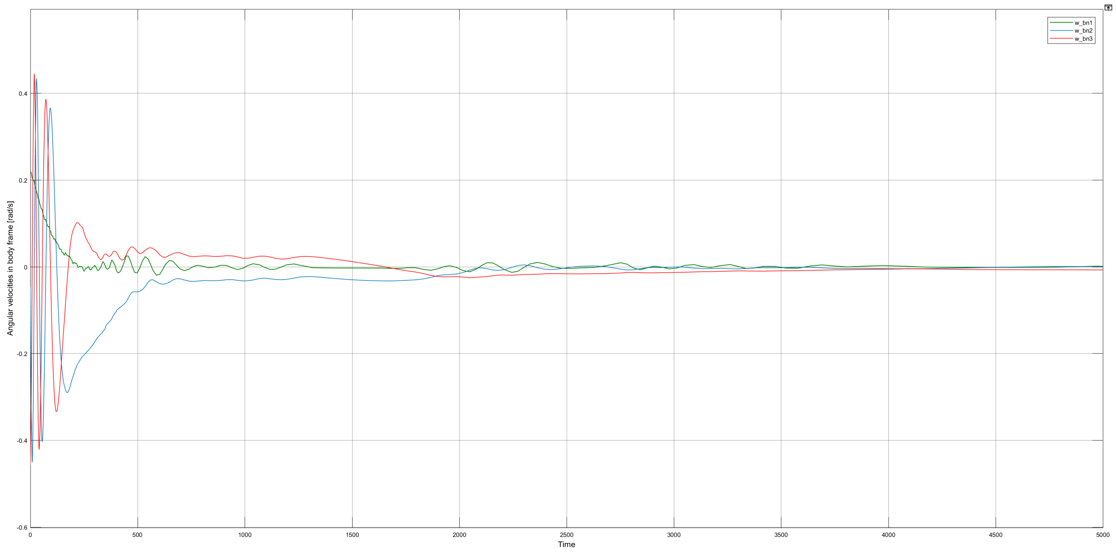


Figure 7.29 Angular velocities reduction during de-tumbling phase, in mission 1, on SSO.

The disturbances felt by OUFTI-Next are instead represented in figure 7.30.

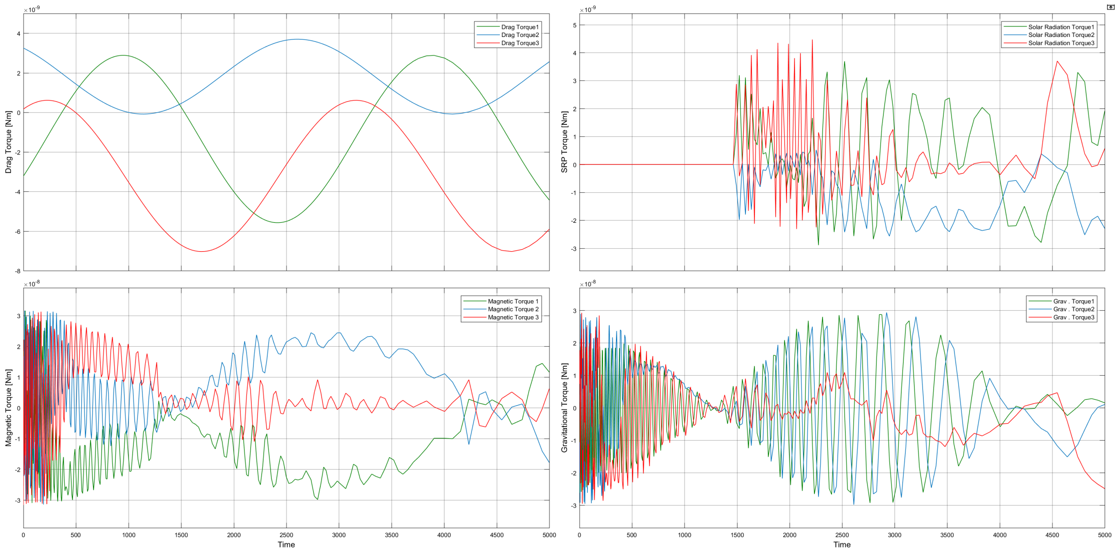


Figure 7.30 Main disturbances during de-tumbling phase, in mission 1, on SSO.

Comparing this values with figure 7.3 on page 61, representing the common disturbances torques for a satellite, it is possible to state that for a significantly smaller platform, like a 3U CubeSat, the values obtained are absolutely reasonable.
The solver relative tolerances have been set to $1e^{-3}$ just to speed up the simulation, since no lack of accuracy in the solution was evidenced.

Sun-pointing 1 The length of this phase is 24 hours plus the time spent in orbit before arriving 30° from the target: the National Park of Brasilia.
The phase lasts around 26.5 hours in total and the solver absolute and relative tolerances have been set to $1e^{-6}$ both. The steady state is reached after around 100 s.

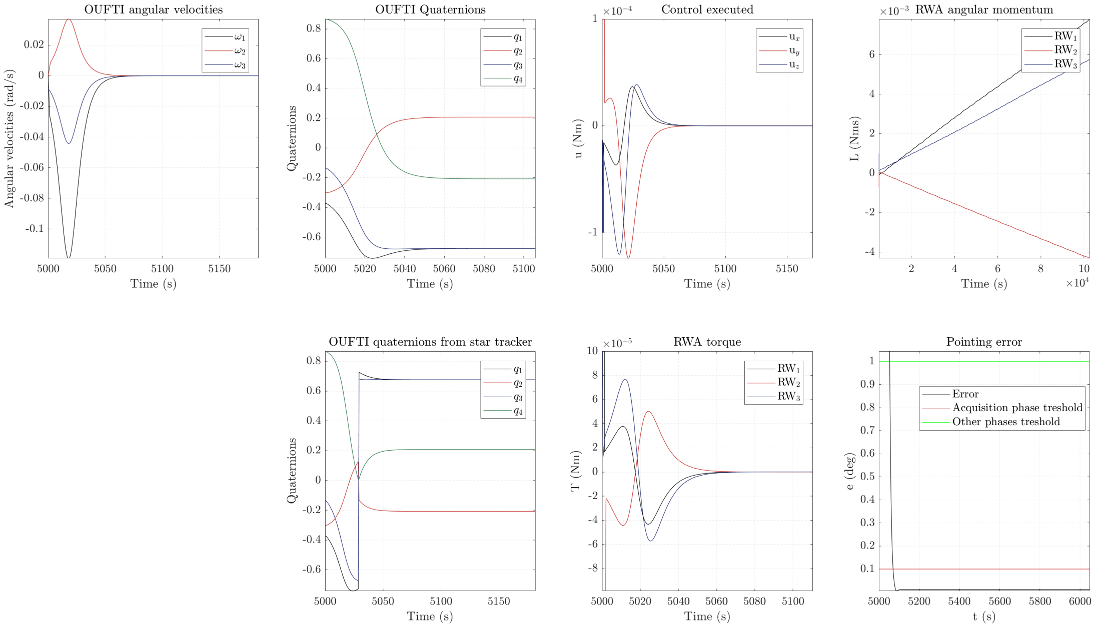


Figure 7.31 Most important results during Sun-pointing phase 1, in mission 1, on SSO. The plots have been zoomed in order to catch the most interesting behaviors.

From figure 7.31 it is possible to observe the change of sign of the quaternion obtained by the determination algorithm, a behavior similar to the one already inspected for ISS orbit.

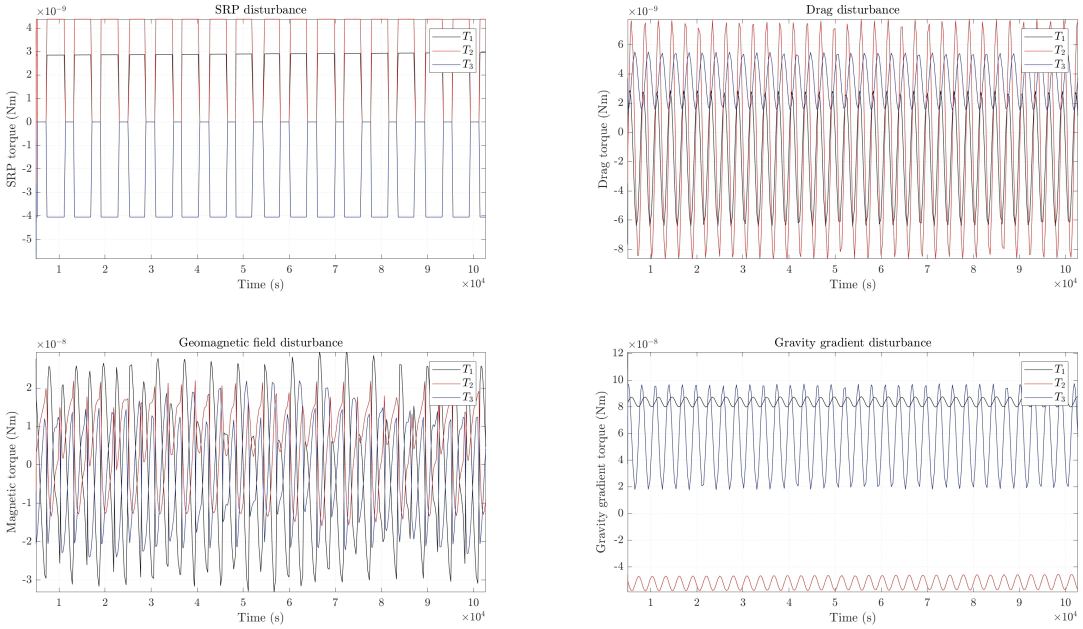


Figure 7.32 Main disturbances torques during Sun-pointing phase 1, in mission 1, on SSO.

In figure 7.32 it is possible to observe that the results in terms of disturbances are consistent with respect to the one obtained with Simulink™, figure 7.30 on page 76. The values of solar radiation pressure torque remain the same, exactly as it should be according to figure 7.3 on page 61. Magnetic and gravity gradient disturbances slightly change, in accordance with the previously mentioned figure, where it is possible to see that their curves do not change so much going from 400 km to 655 km of altitude. Finally the drag torque is the one changing the most, decreasing its value of two orders of magnitude.

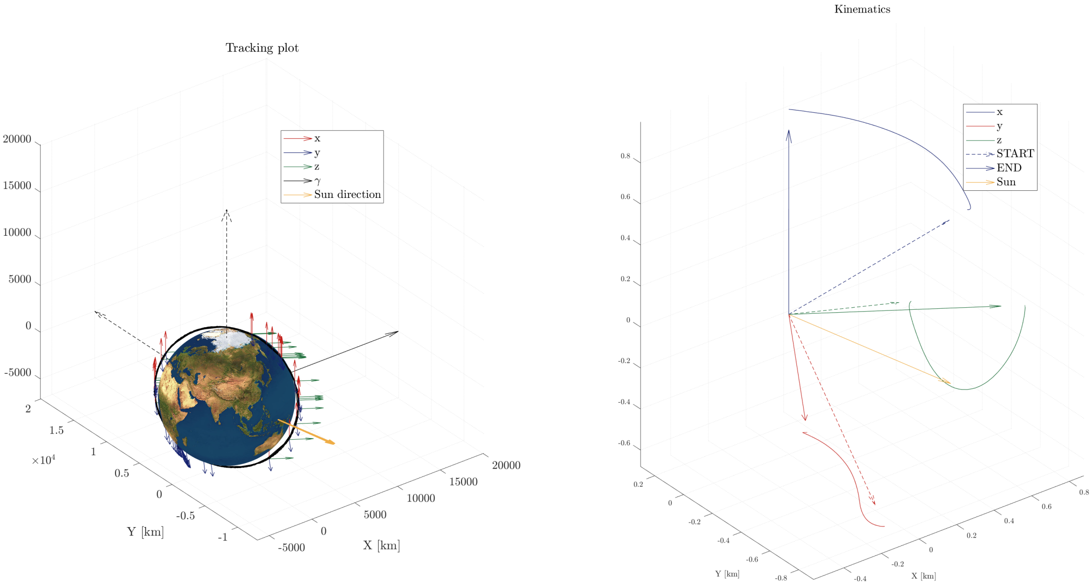


Figure 7.33 Body-fixed frame while orbiting around the earth during Sun-pointing phase 1, in mission 1, on SSO.

De-saturation 1 The phase lasts 1 minute and the solver absolute and relative tolerances have been set to $1e^{-6}$ both. It is interesting to compare these results with the ones of de-saturation phase 2, when the CubeSat is in eclipse, figure 7.43 on page 83. Figure 7.34 proves that the platform will rotate very slowly independently from solar radiation pressure effect.

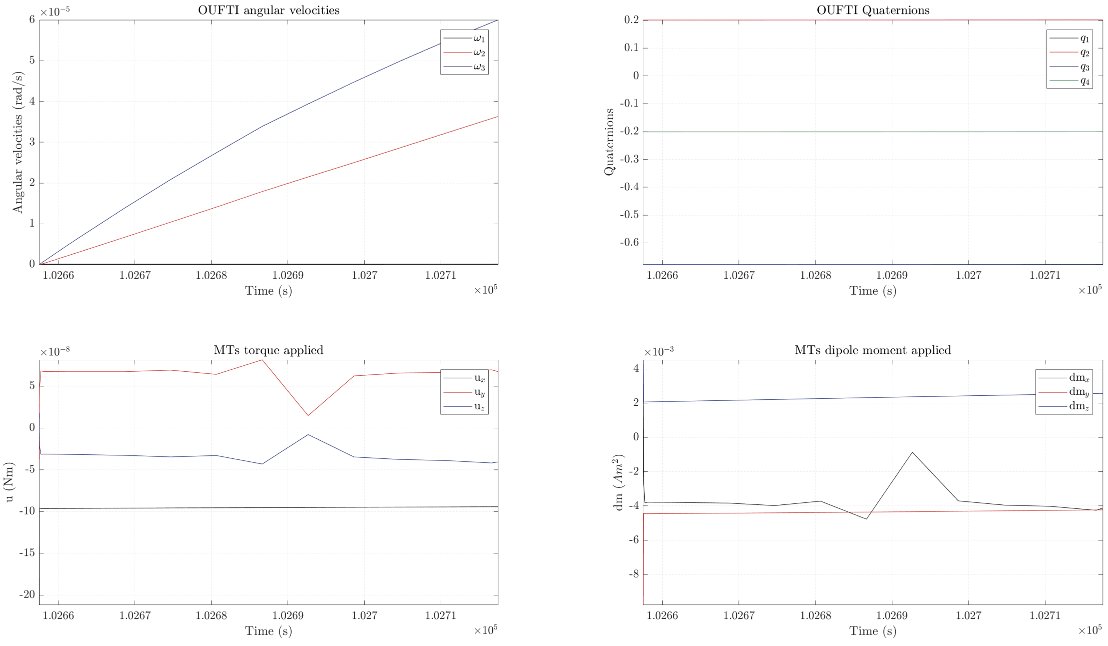


Figure 7.34 Most important results during de-saturation phase 1, in mission 1, on SSO.

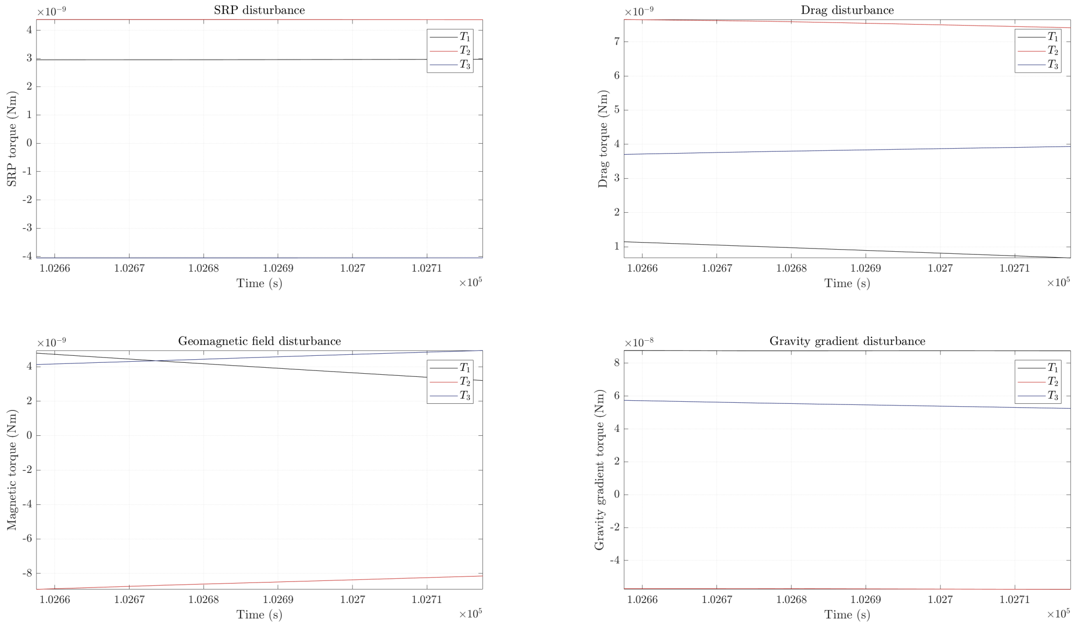


Figure 7.35 Main disturbances torques during de-saturation phase 1, in mission 1, on SSO.

Picture acquisition The picture acquisition phase in SSO requires the same pointing accuracy than the one in ISS orbit and, as shown in figure 7.36, the control action is able to maintain the pointing error below the 0.1° threshold, thanks to the control parameters optimization.
The phase lasts around 15 minutes and the solver absolute and relative tolerances have been set to $1e^{-6}$ both. The steady state is reached less than 90 s.

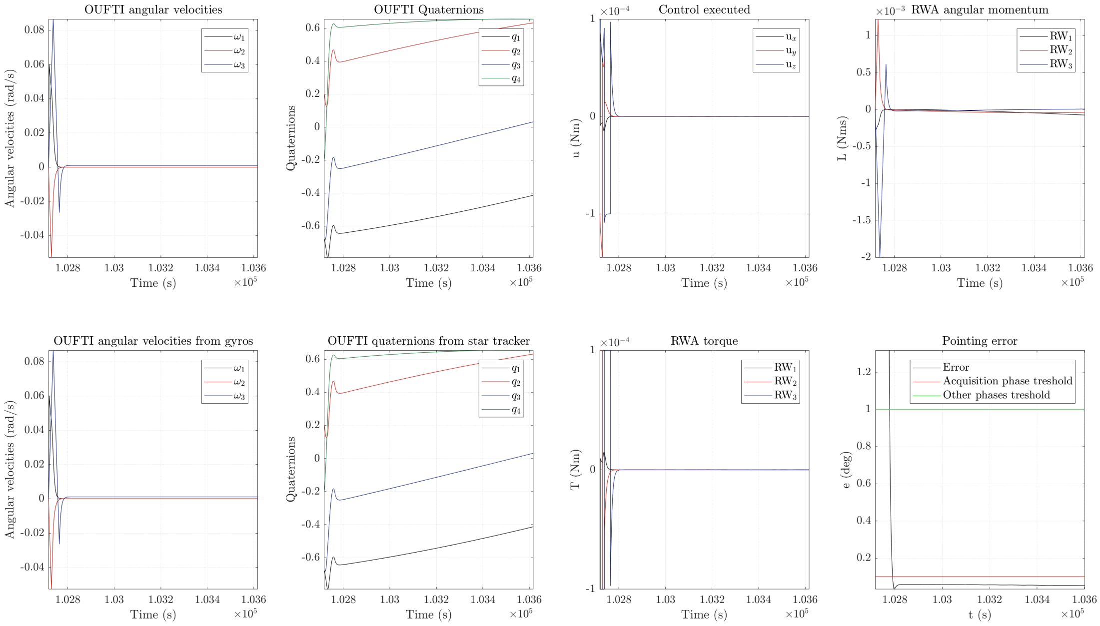


Figure 7.36 Most important results during Nadir pointing picture acquisition phase, in mission 1, on SSO. The plots have been zoomed in order to catch the most interesting behaviors.

From figure 7.36 it is also possible to appreciate how well the *unscented Kalman filter* estimates the state, comparing the integrated angular velocities and quaternion plots with the ones provided by the gyros/star trackers and estimated by the filter.

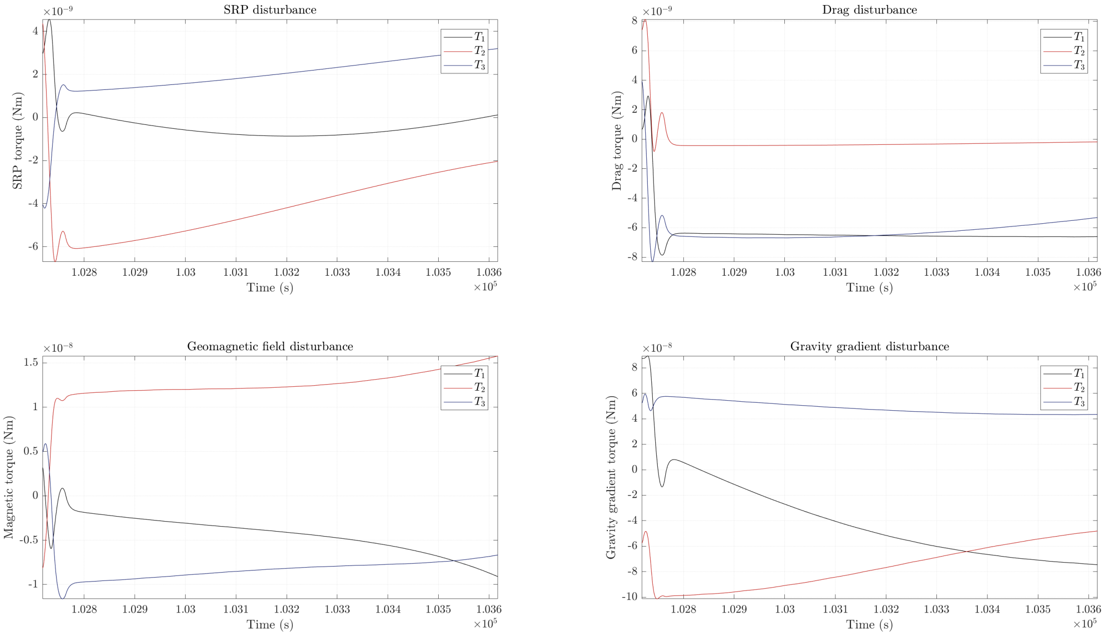


Figure 7.37 Main disturbances torques during Nadir pointing picture acquisition phase, in mission 1, on SSO.

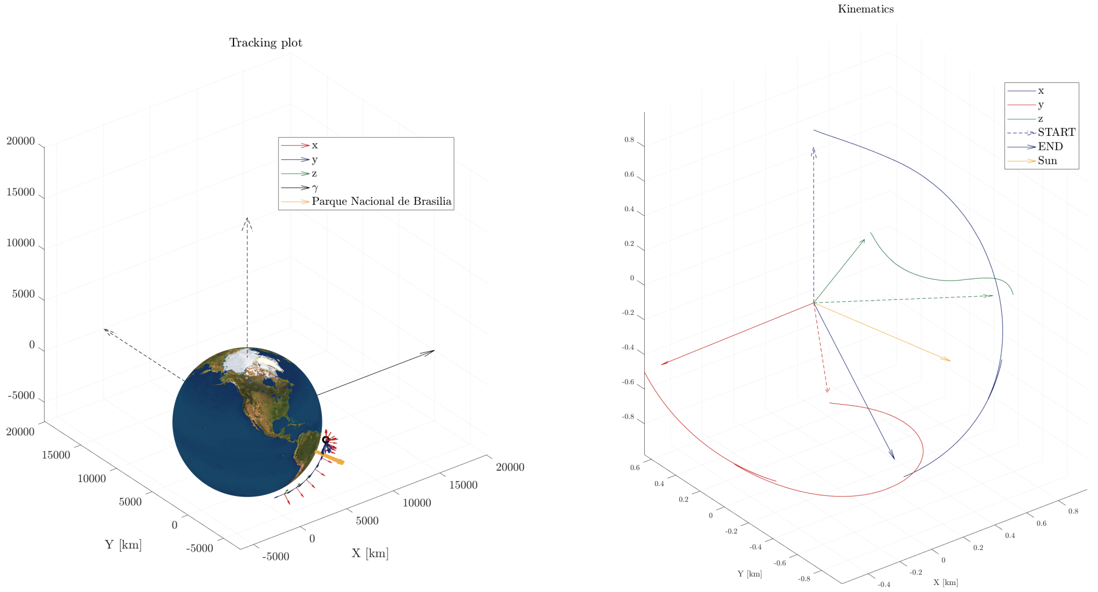


Figure 7.38 Body-fixed frame while orbiting around the earth during Nadir pointing picture acquisition phase, in mission 1, on SSO.

Sun-pointing 2 This phase is quite similar to the previous ones in Sun-pointing. The phase lasts around 31.6 hours, the time needed to reach 10° from Liège, and the solver absolute and relative tolerances have been set to $1e^{-6}$ both. The steady state is reached after around 80 s.

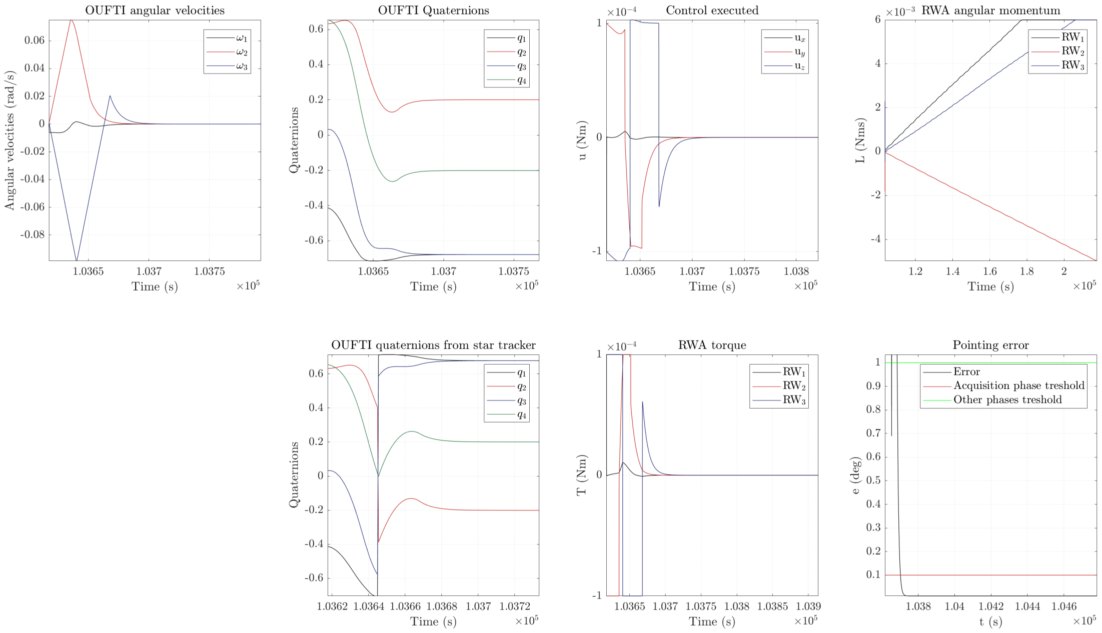


Figure 7.39 Most important results during Sun-pointing phase 2, in mission 1, on SSO. The plots have been zoomed in order to catch the most interesting behaviors.

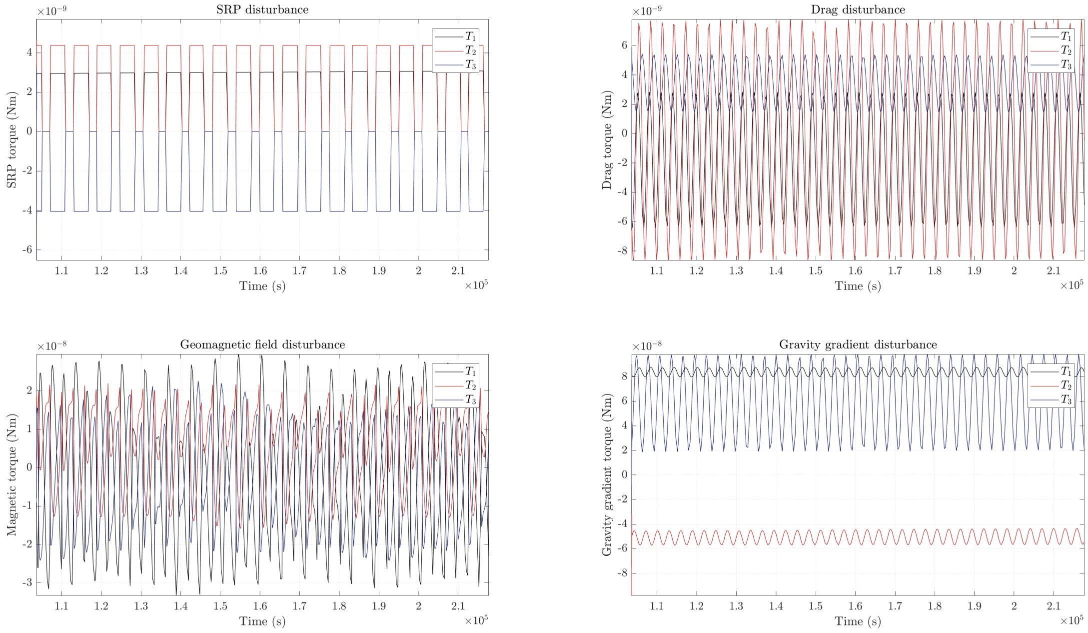


Figure 7.40 Main disturbances torques during Sun-pointing phase 2, in mission 1, on SSO.

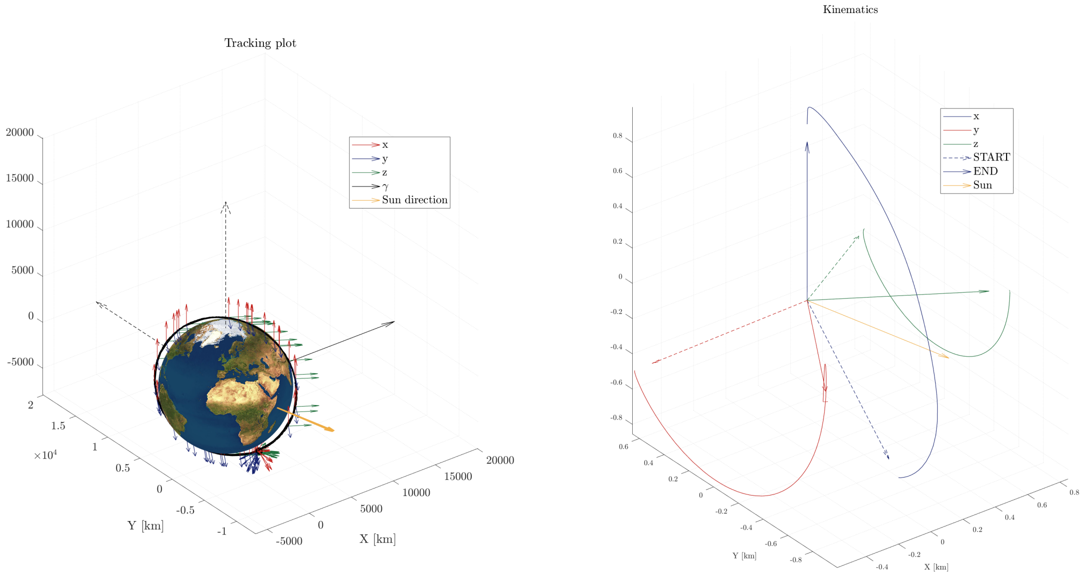


Figure 7.41 Body-fixed frame while orbiting around the earth during Sun-pointing phase 2, in mission 1, on SSO.

De-saturation 2 These results are quite similar to the ones of the previous de-saturation phase and confirm that the spacecraft rotates very slowly.
The phase lasts 1 minute and the solver absolute and relative tolerances have been set to $1e^{-6}$ both.

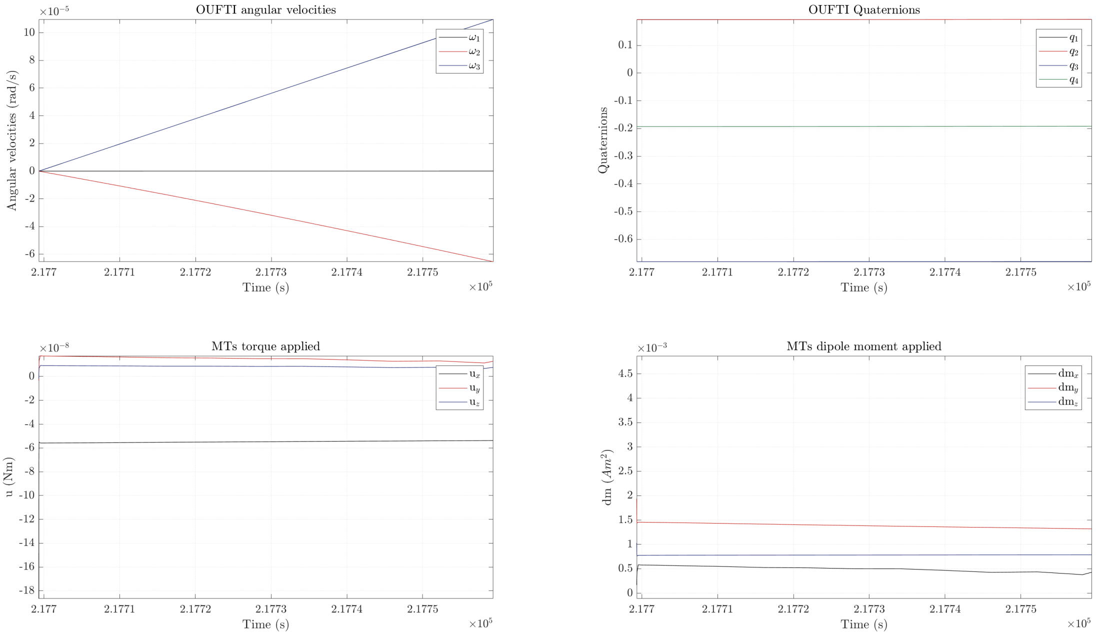


Figure 7.42 Most important results during de-saturation phase 2, in mission 1, on SSO.

The peculiarity with respect to de-saturation phase 1 is that it occurs in eclipses, as it can be seen from the solar radiation pressure plot in figure 7.43 on the facing page.

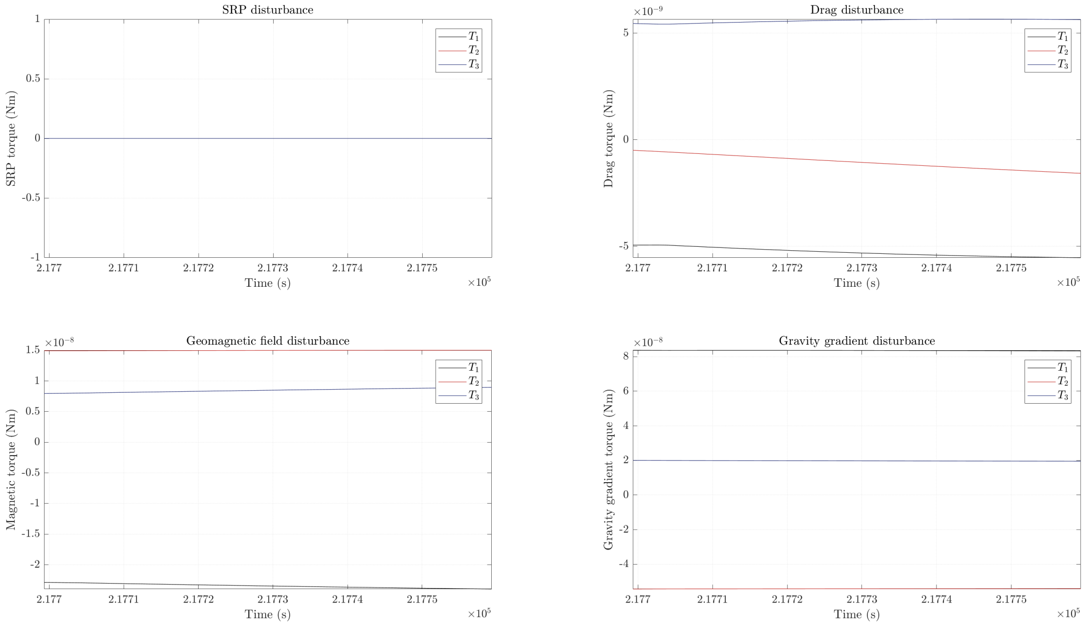


Figure 7.43 Main disturbances torques during de-saturation phase 2, in mission 1, on SSO.

Downloading Downloading phase above Liège lasts 5 minutes and the solver absolute and relative tolerances have been set to $1e^{-6}$ both. The steady state is reached after around 90 s.

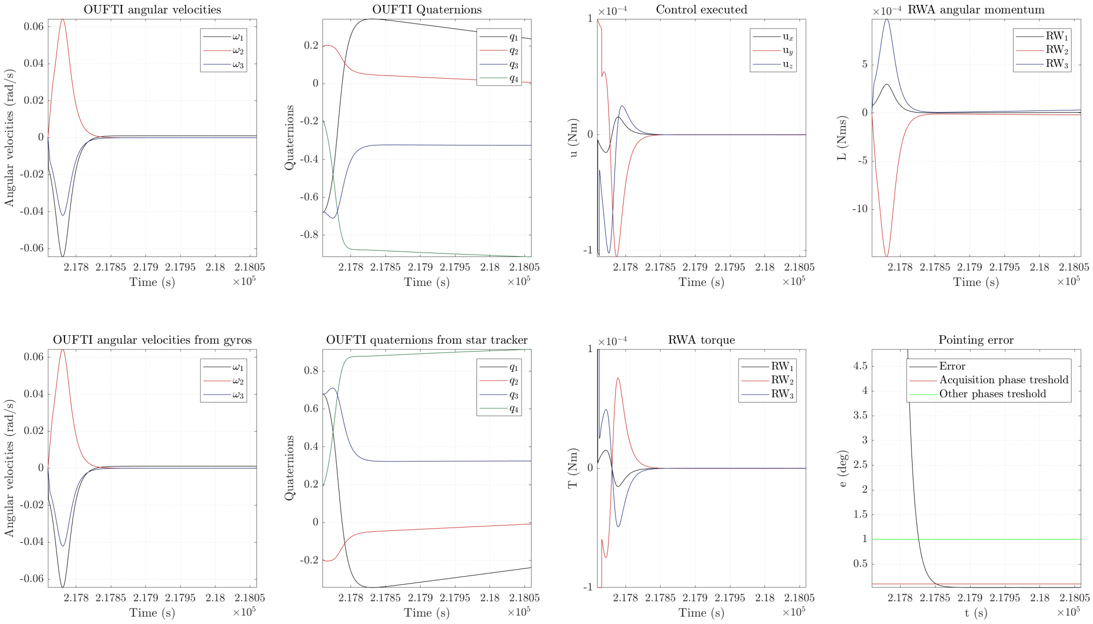


Figure 7.44 Most important results during downloading phase, in mission 1, on SSO. The plots have been zoomed in order to catch the most interesting behaviors.

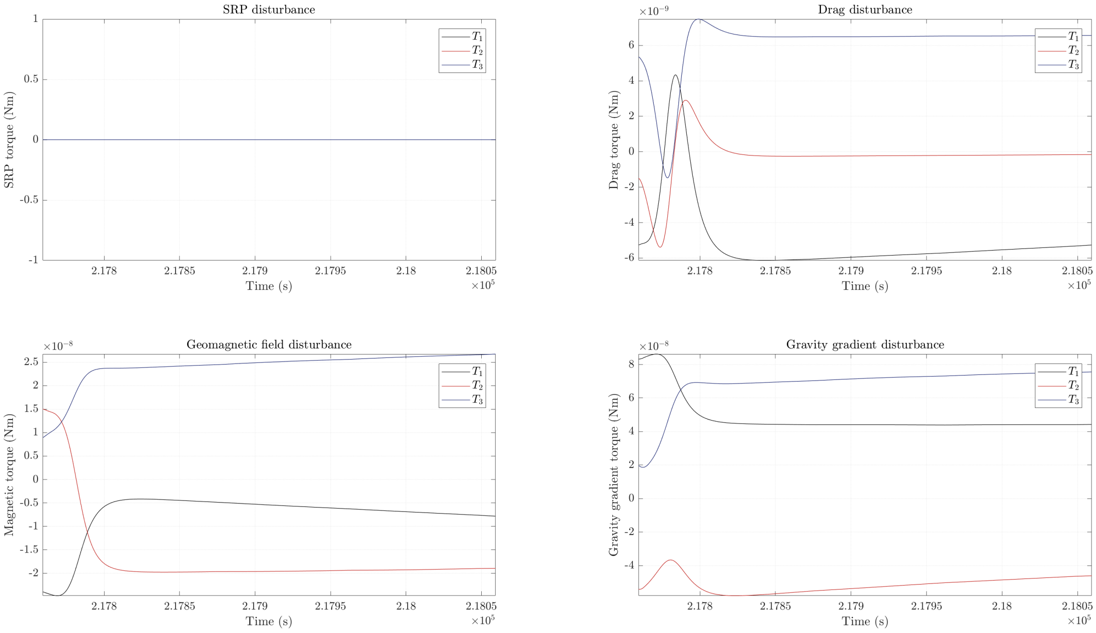


Figure 7.45 Main disturbances torques during downloading phase, in mission 1, on SSO. The plots have been zoomed in order to catch the most interesting behaviors.

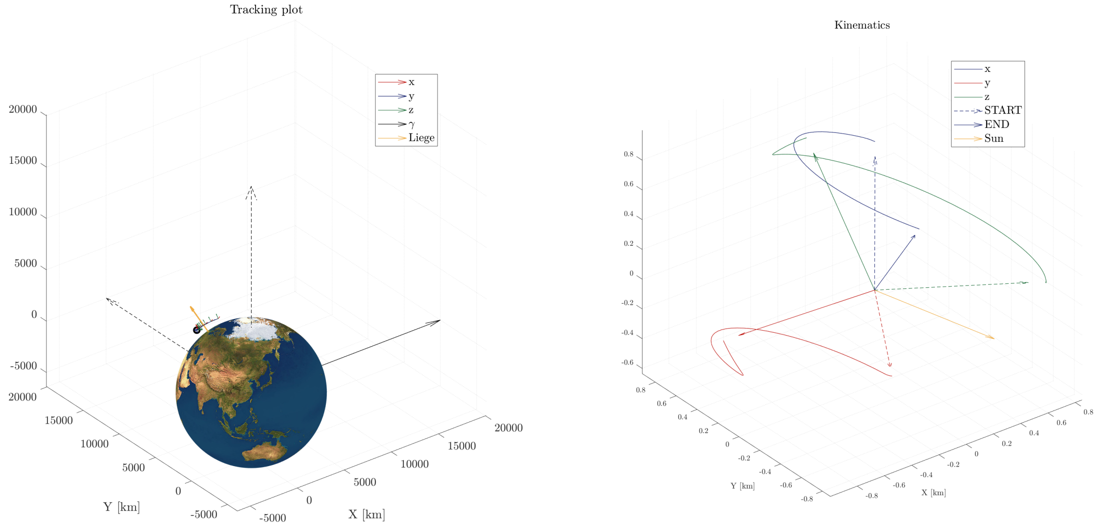


Figure 7.46 Body-fixed frame while orbiting around the earth during downloading phase, in mission 1, on SSO.

7.2.2 Mission 3

Computational time usually requested: around 50 minutes, depending on the initial conditions.

De-tumbling After the de-tumbling phase, the angular velocities reached are:

$$\begin{cases} \omega_{b/n,1}(5000) = 5.455e^{-3} \text{ rad/s} \\ \omega_{b/n,2}(5000) = 5.598e^{-3} \text{ rad/s} \\ \omega_{b/n,3}(5000) = 1.013e^{-2} \text{ rad/s} \end{cases} \tag{7.5}$$

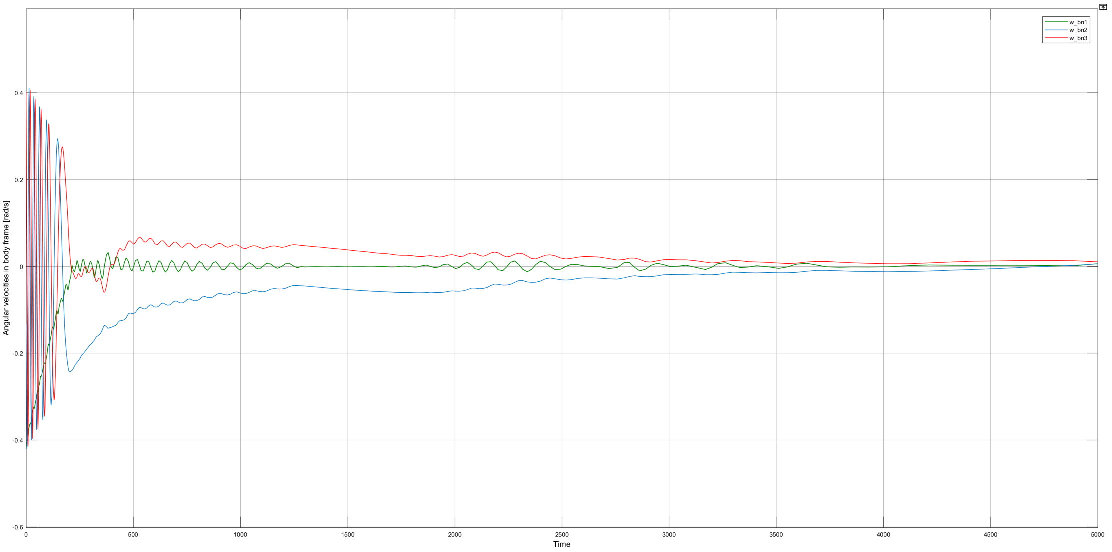


Figure 7.47 Angular velocities reduction during de-tumbling phase, in mission 3, on SSO orbit.

The disturbances felt by OUFTI-Next are instead represented in figure 7.48.

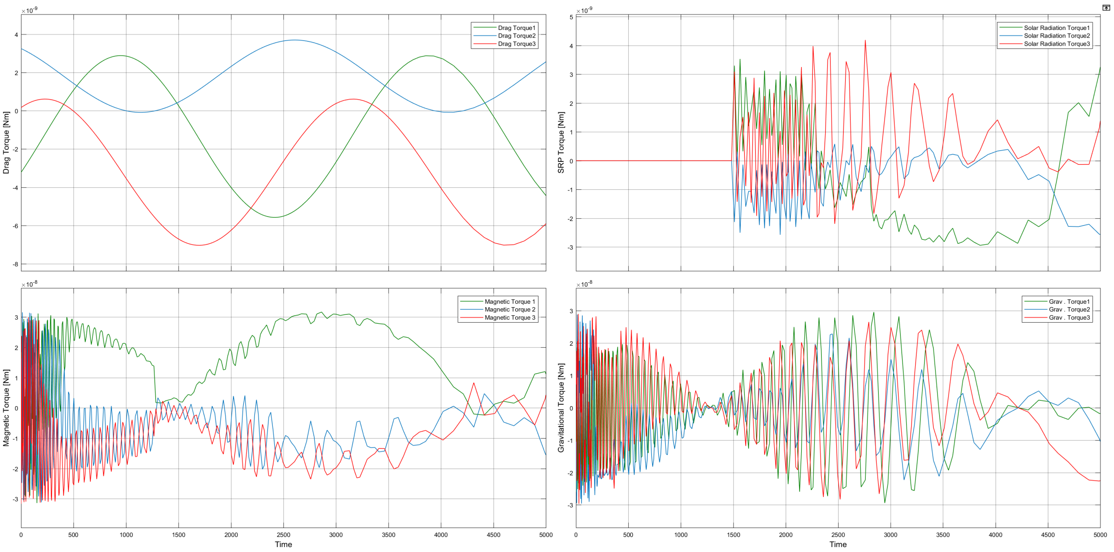


Figure 7.48 Main disturbances during de-tumbling phase, in mission 3, on SSO orbit.

Sun-pointing The simulation lasts around 22 hours, the maximum time interval before saturation in Sun-pointing, with solver absolute and relative tolerances set to $1e^{-6}$ both. The steady state is reached in less than 100 s.

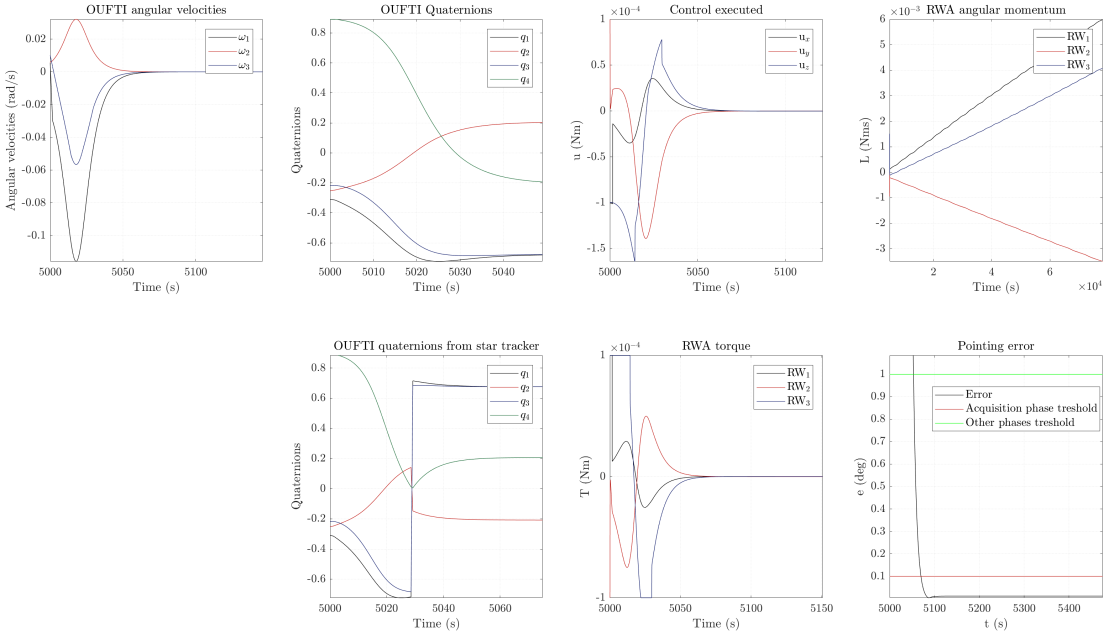


Figure 7.49 Most important results during Sun-pointing phase in mission 3, on SSO. The plots have been zoomed in order to catch the most interesting behaviors.

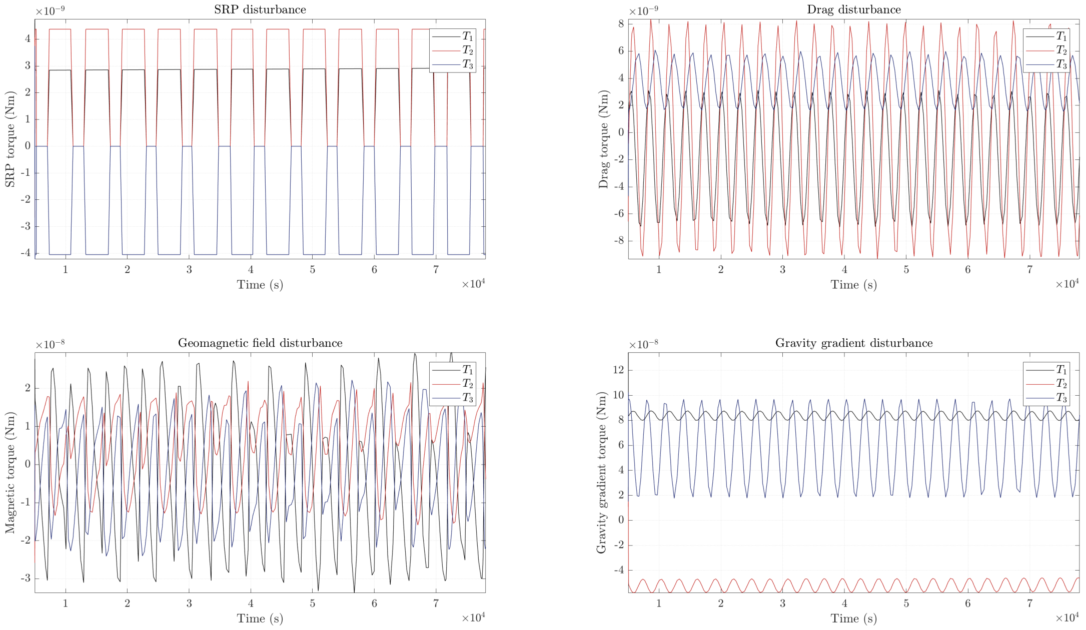


Figure 7.50 Main disturbances torques during Sun-pointing phase in mission 3, on SSO.

De-saturation De-saturation phase lasts one day, with solver absolute and relative tolerances set to $1e^{-6}$ each.

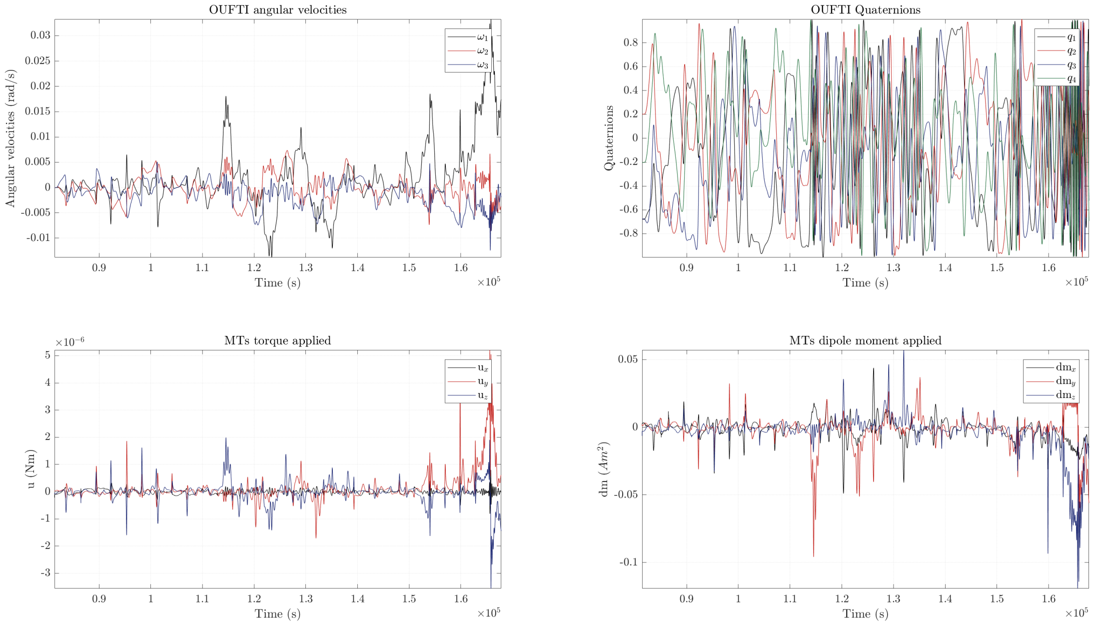


Figure 7.51 Most important results during de-saturation phase in mission 3, on SSO.

In figure 7.51 it is clear that the angular velocity of the platform remains almost bounded inside 0.03 rad/s and -0.015 rad/s, meaning that it is still possible to store solar energy during shorter de-saturation phases.

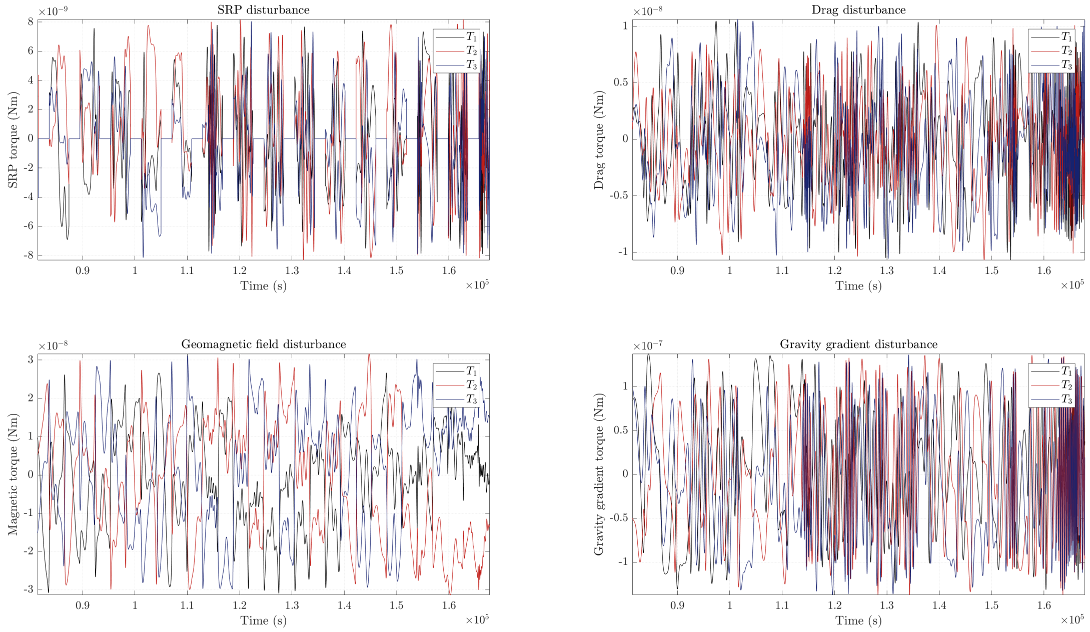


Figure 7.52 Main disturbances torques during de-saturation phase, in mission 3, on SSO.

7.2.3 Mission 1 -Target following pointing-

For what concerns mission 1 performed with target following pointing, here it is reported just the picture acquisition phase. In appendix, all the results concerning mission 2 with target following pointing are collected, such that the reader can go into the details of the whole mission. The computational time requested to simulate the whole mission is usually around 10 minutes.

Picture acquisition From figure 7.53 it is possible to appreciate the angular velocity around Z-axis increasing and decreasing while OUFTI is passing over the target, rotating in order to follow the National Park of Brasilia, as well as the smooth drop of pointing error, while passing above the target. The solver absolute and relative tolerances are set to $1e^{-6}$.

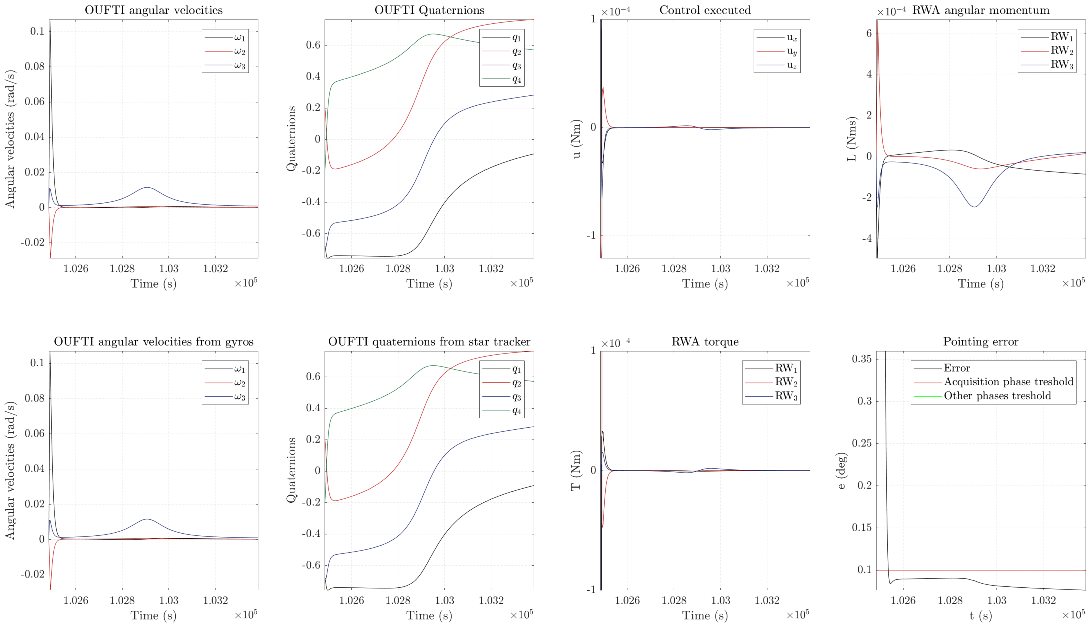


Figure 7.53 Most important results during picture acquisition phase, in mission 1, on SSO. The plots have been zoomed in order to catch the most interesting behaviors.

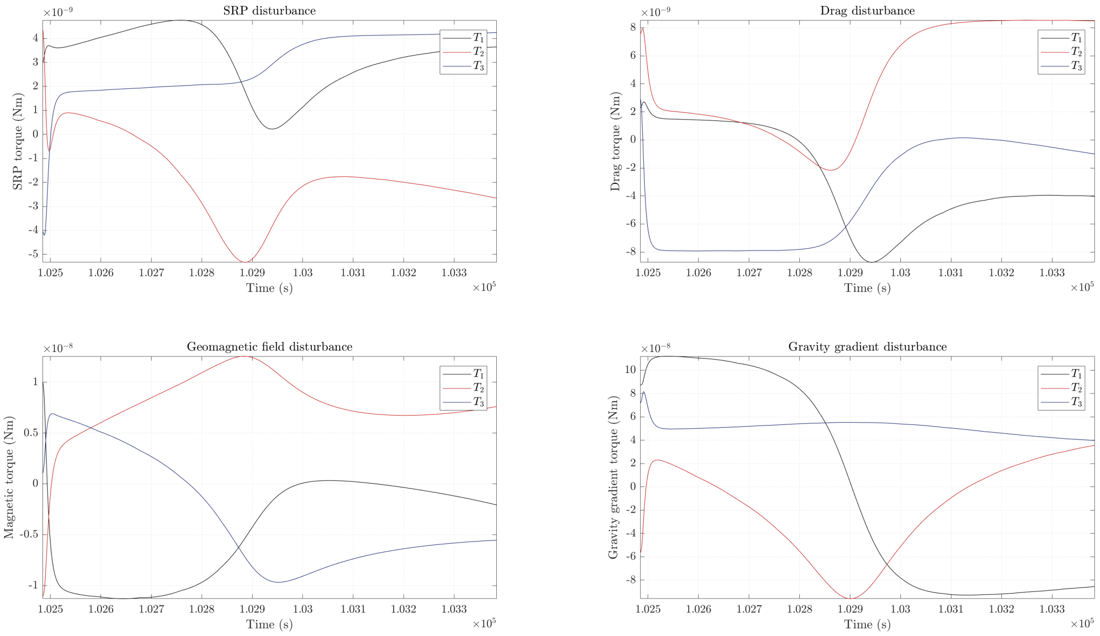


Figure 7.54 Main disturbances torques during picture acquisition phase, in mission 1, on SSO.

From figure 7.55 instead, it is possible to see the red vector (the X-axis of the body-fixed reference frame) following the National Park of Brasilia.

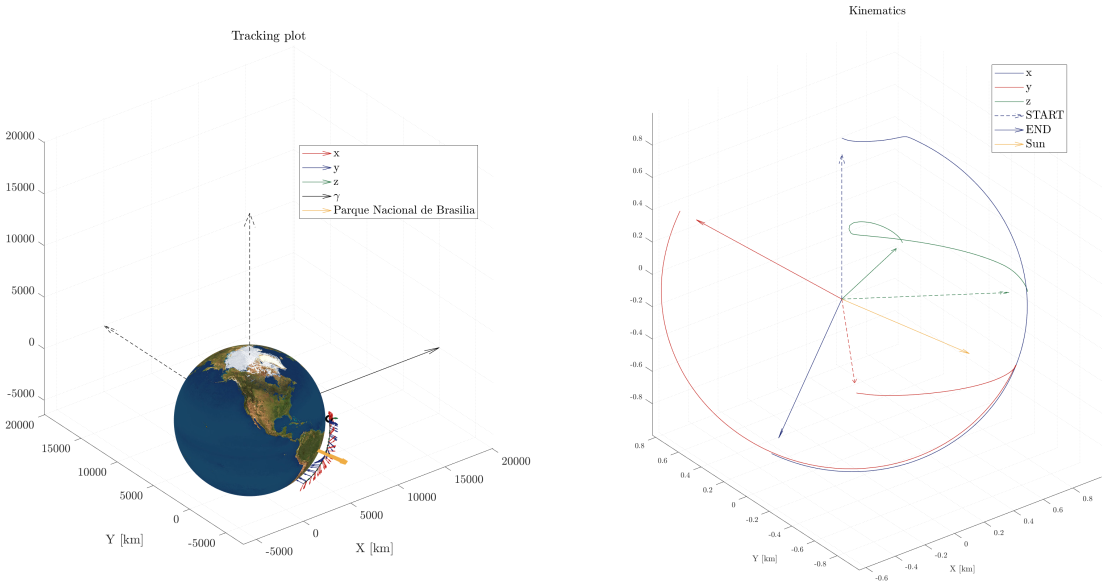


Figure 7.55 Body-fixed frame while orbiting around the earth during picture acquisition phase, in mission 1, on SSO.

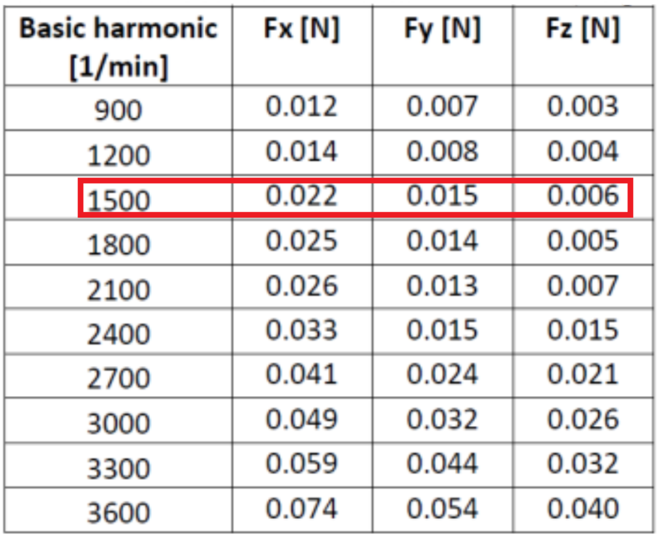
Chapter 8

Cryocooler internal disturbance

Figure 8.1 *SRI401* by *LE-TEHNIKA*.

8.1 Model

OUFTI-Next will have a cryocooler on-board, used to cool down the detector before picture acquisition phase. The device will be turned on during the last minutes of Sun-pointing or just at the beginning of picture acquisition phase. In this Thesis it has been supposed to be switched on when the picture acquisition starts, because it is the most important phase of the mission and it is fundamental to analyse the stabilization of the platform in this case. The details about the actual performances of this element are still not well defined, but the *SRI401* by *LE-TEHNIKA* has been selected, as suitable device, figure 8.1. Thanks to recent structural tests on the cryocooler, the supplier provided the dynamic forces RMS values of the first harmonics, figure 8.2 on the following page. According to *LE-TEHNIKA*, the cryocooler usually runs with rotation frequency between $1000 - 1500$ rpm (typical application, ambient/room temperature) and for this reason it has been considered the worst case scenario, highlighted in figure 8.2 on the next page.



Basic harmonic [1/min]	Fx [N]	Fy [N]	Fz [N]
900	0.012	0.007	0.003
1200	0.014	0.008	0.004
1500	0.022	0.015	0.006
1800	0.025	0.014	0.005
2100	0.026	0.013	0.007
2400	0.033	0.015	0.015
2700	0.041	0.024	0.021
3000	0.049	0.032	0.026
3300	0.059	0.044	0.032
3600	0.074	0.054	0.040

Figure 8.2 Dynamic forces RMS values of the first harmonics. The X, Y and Z axis are different from the ones used in this Thesis.

In order to model the internal disturbances due to this device, it has supposed to place it 0.075 m beneath the geometric center of the CubeSat (in the top quarter of the space dedicated to the payload, 1.5 U) and shifted in Z and Y direction of 0.0707 m, figure 8.3.

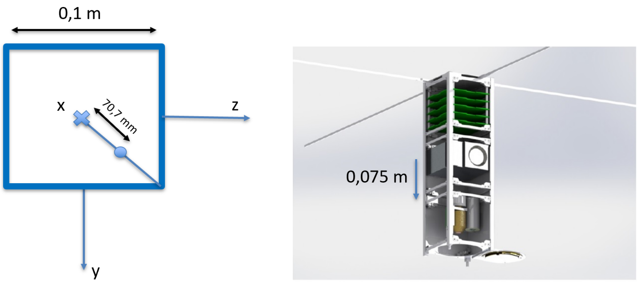


Figure 8.3 Position of the cryocooler inside of OUFTI-Next.

Once the position has been fixed, the disturbance torque in Nm acting on the spacecraft can be evaluated with respect to the *body-fixed reference frame* adopted in this Thesis and added to the disturbances model already presented:

$$d_{int} = \begin{bmatrix} -0.0354cos(\frac{\pi}{4})sin(2\pi 25t)F_x + 0.0354cos(\frac{\pi}{4})sin(2\pi 25t)F_y \\ 0.0354cos(\frac{\pi}{4})sin(2\pi 25t)F_z + 0.075sin(2\pi 25t)F_y \\ -0.075sin(2\pi 25t)F_x - 0.0354cos(\frac{\pi}{4})sin(2\pi 25t)F_z \end{bmatrix} \tag{8.1}$$

The introduction of such an high frequency disturbance, with respect to the external ones, affects the computational time, reducing it dramatically. It has been decided to simulate just the two most important and interesting phases: the target following picture acquisition, both in ISS orbit and SSO.

8.2 Results

ISS orbit On ISS orbit the pointing accuracy is obtained without changing the control parameters used for the simulation in which no cryocooler disturbance was considered, figure 8.4. In figure 8.5

on the following page it is possible to notice that, when OUFTI-Next passes above the target, the pointing error will decrease almost smoothly of around 0.01 deg in 1 minute. This behavior is similar to the one observed without the cryocooler disturbance and can be explained looking at figure 8.8 on page 95, representing the disturbances level during the acquisition phase: when the CubeSat is approaching the nadir position with respect to the target, the disturbances torques are increasing, reaching their maximum when OUFTI is right above the Brazilian National Park and then start decreasing, exactly as the behavior of the pointing error plot. The momentum stored is not saturating the reaction wheels, as it is shown in figure 8.7 on page 95. The simulation lasts 550 s and the computational time required is 6.3 hours, with a solver absolute and relative tolerance of $1e^{-6}$.

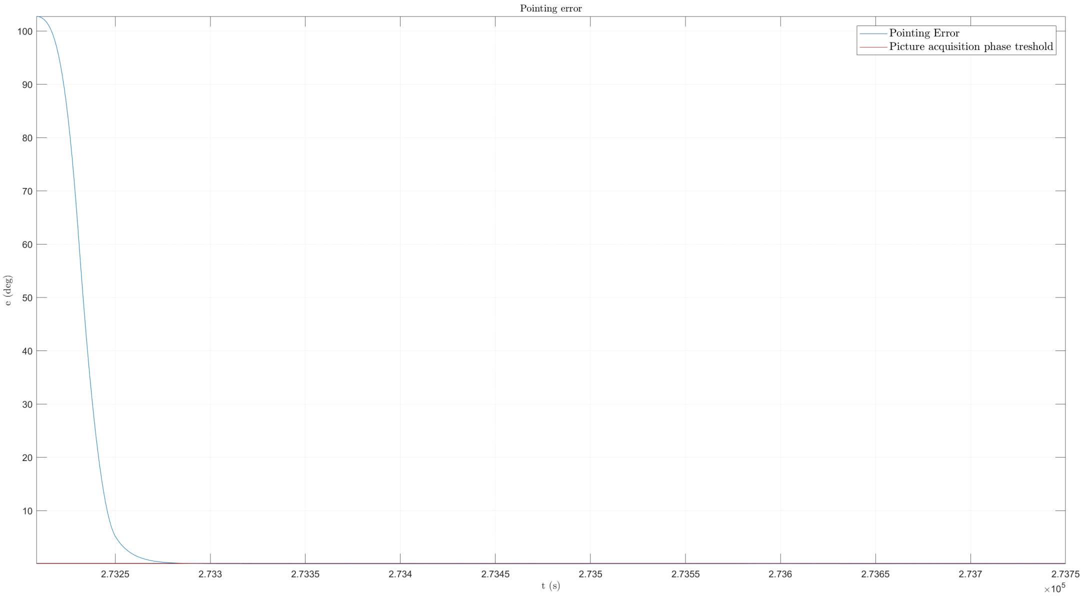


Figure 8.4 Pointing error considering also the cryocooler internal disturbance, during target following picture acquisition phase on ISS orbit.

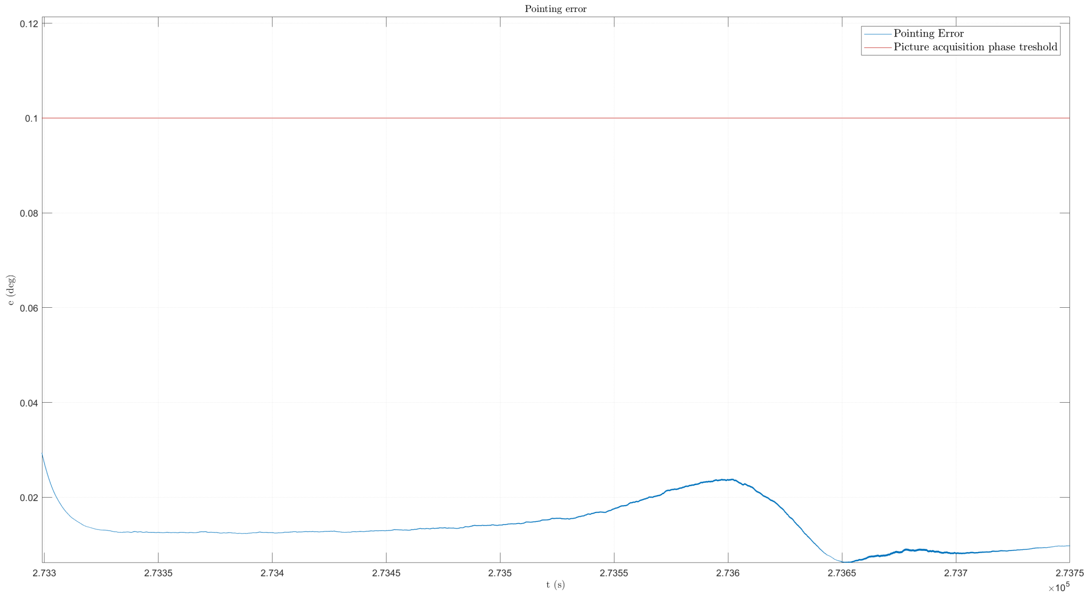


Figure 8.5 Zoomed pointing error plot, considering also the cryocooler internal disturbance, during target following picture acquisition phase on ISS orbit.

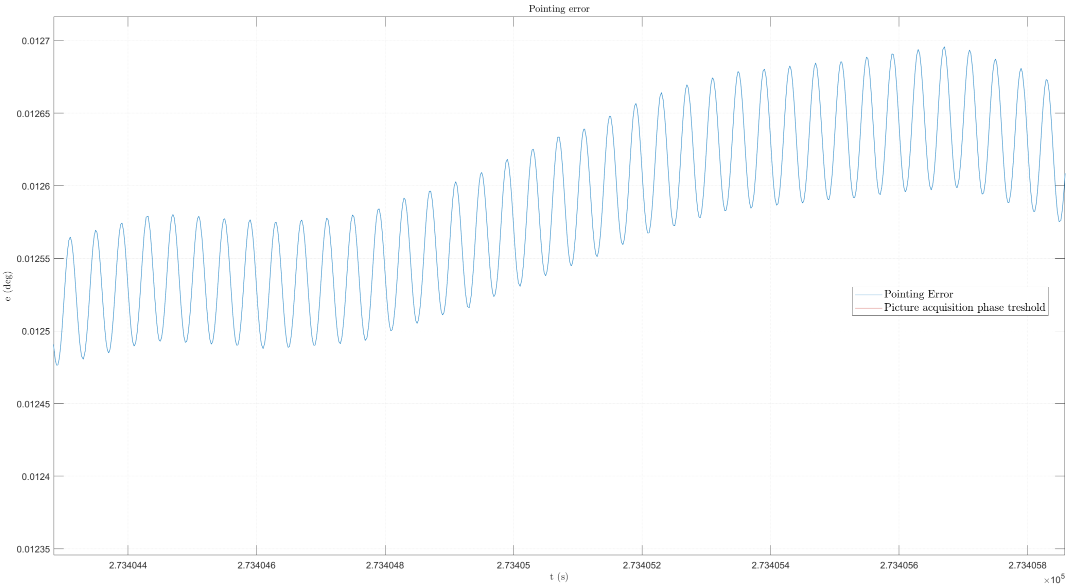


Figure 8.6 Pointing error oscillations (zoomed plot), considering also the cryocooler internal disturbance, during target following picture acquisition phase on ISS orbit.

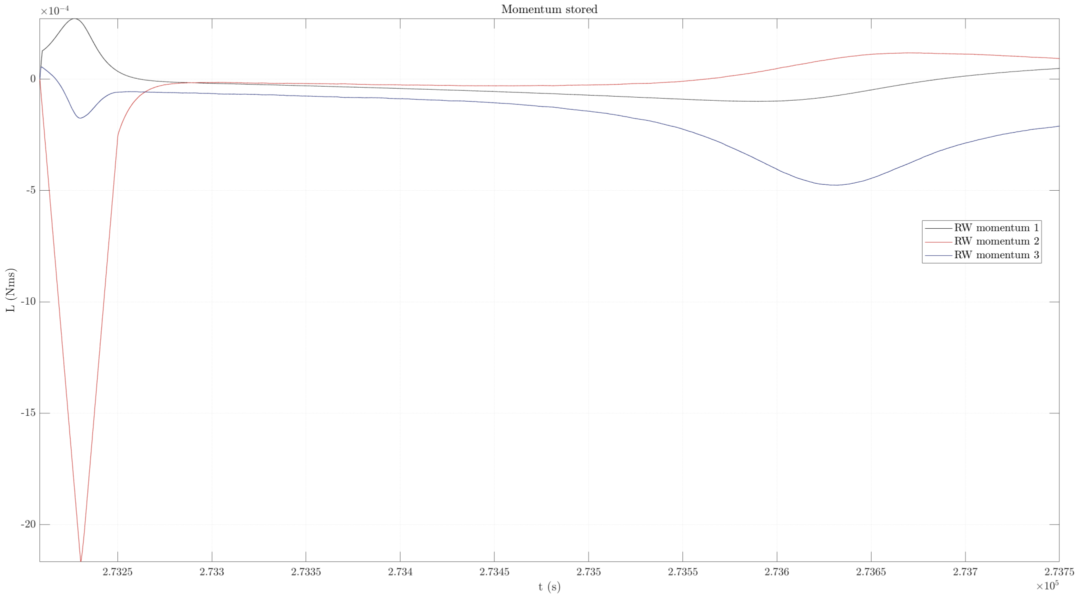


Figure 8.7 Momentum stored by the reaction wheels during target following picture acquisition phase on ISS.

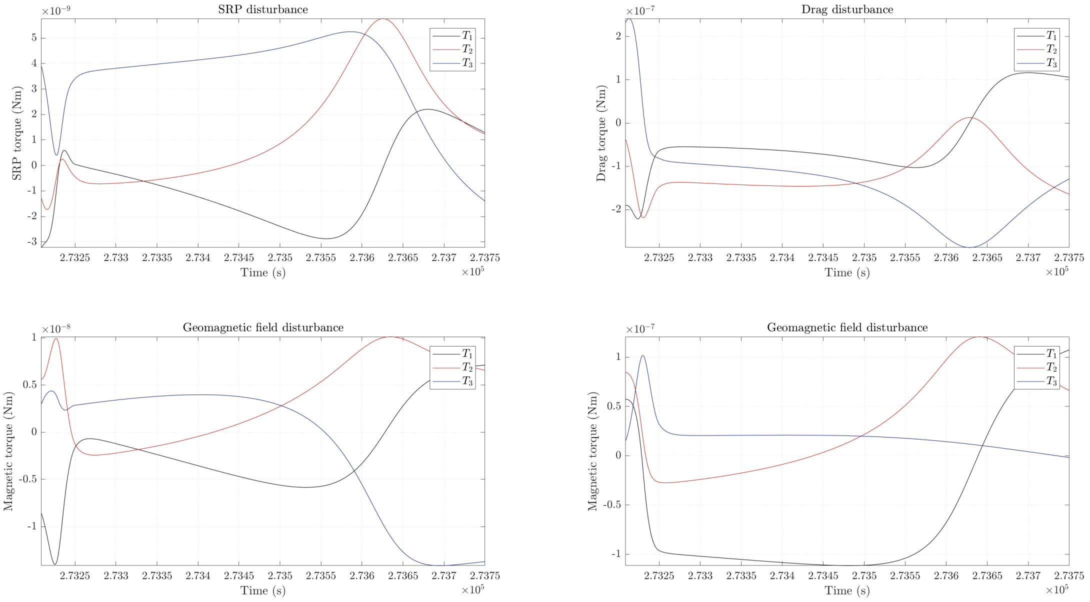


Figure 8.8 Disturbances torques in ISS orbit during pointing acquisition with cryocooler on.

SSO On SSO the pointing accuracy is obtained without changing the control parameters, as for ISS orbit and the pointing error trend is almost identical, figure 8.10 on the next page. The momentum stored is not saturating the reaction wheels, as it is shown in figure 8.11 on page 97. The simulation lasts 550 s and the computational time required is 7.45 hours, with a solver absolute and relative tolerance of $1e^{-6}$.

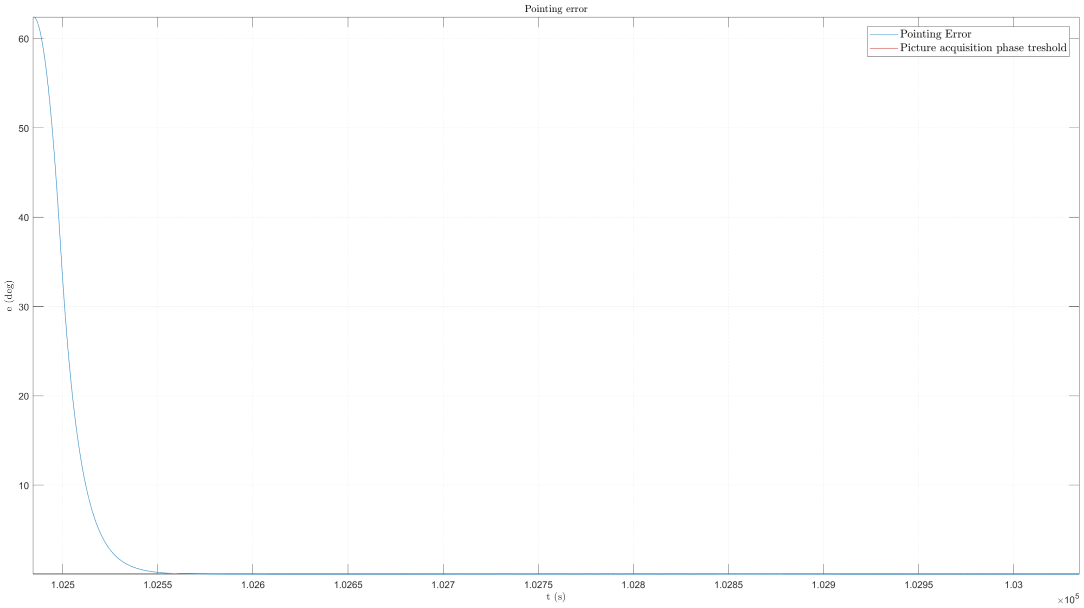


Figure 8.9 Pointing error considering also the cryocooler internal disturbance, during target following picture acquisition phase on SSO.

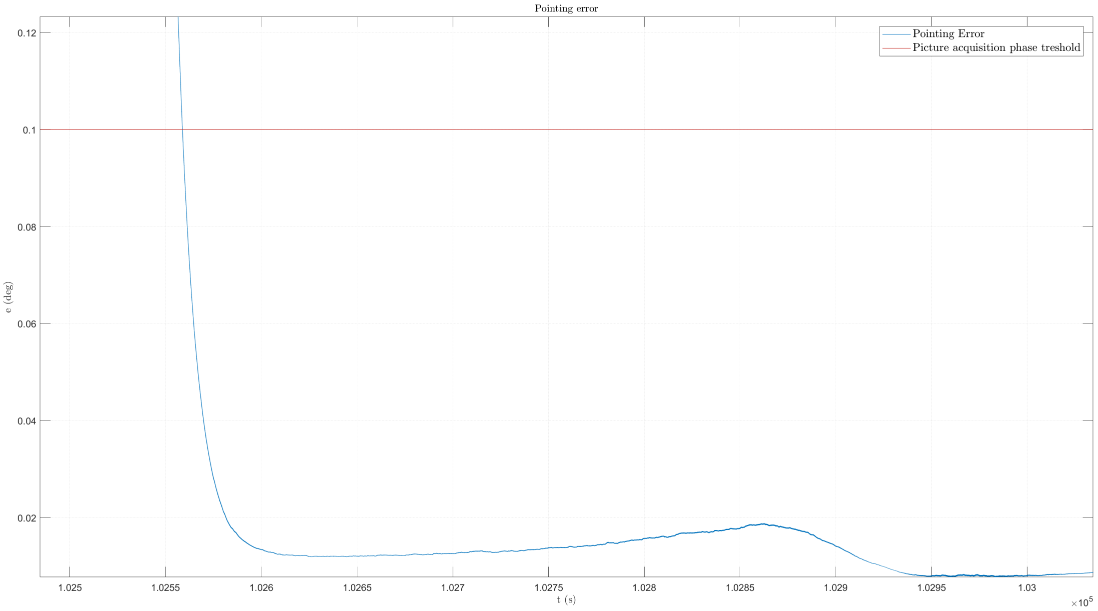


Figure 8.10 Zoomed pointing error plot, considering also the cryocooler internal disturbance, during target following picture acquisition phase on SSO.

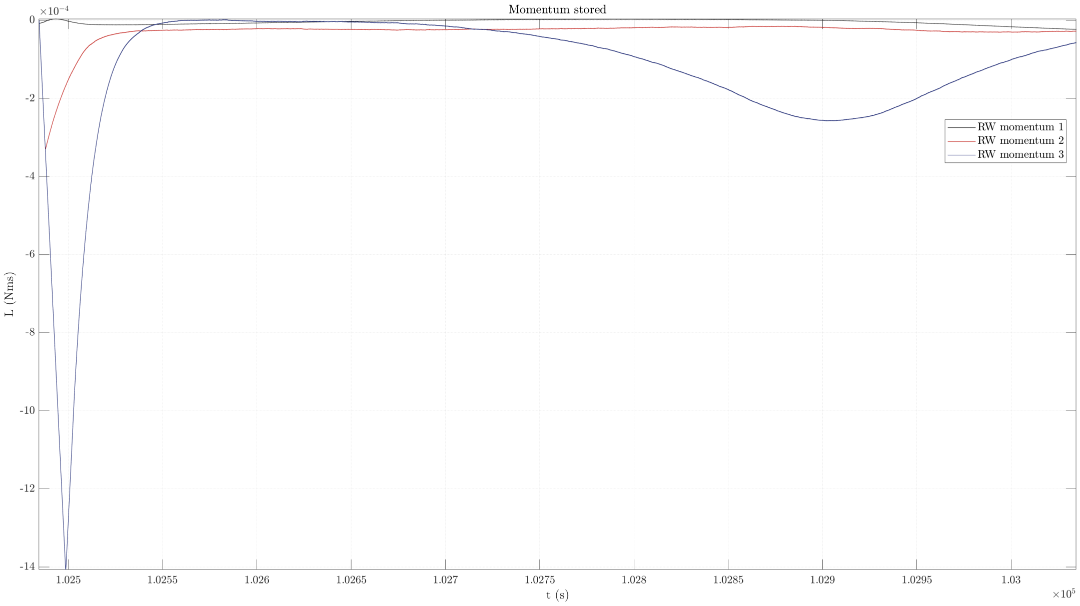


Figure 8.11 Momentum stored by the reaction wheels during target following picture acquisition phase on SSO.

8.3 Conclusions

The performed simulations about the internal disturbance due to the cryocooler evidenced that the presence of this source of vibrations inside the platform will not affect the pointing accuracy, thanks to the control strategy adopted in this Thesis. Nevertheless, it is important to remark that the position of the cryocooler was just reasonably guessed and,as a result, it is advisable to perform again this kind of simulations, once the details about the cryocooler will be acquired.

Moreover, from the analysis results, it is also possible to state that the stability requirement of 0.01° can be achieved in time intervals during less than 1 minute.

Chapter 9

Noises and update rates

To check the effect of noises and sensors update rates on the model, they have been added to a nominal Sun-pointing phase lasting 150 s on ISS orbit, the most challenging one in terms of disturbances level. The simulation is very slow (almost 17 min), so, in order to speed it up, the absolute and relative tolerances of the solvers have been relaxed till $1e^{-4}$ and the output reduced to only the estimated and integrated state. The results are here presented in figure 9.1 and demonstrate that the presence of noises and measurements update rates are not affecting the determination provided by the unscented Kalman filter.
An interesting effect of adding the unscented Kalman filter algorithm and the measurements noises to the model is the numerical instability of ODE15s. For this reason this simulation has been performed with a different solver: ODE113.

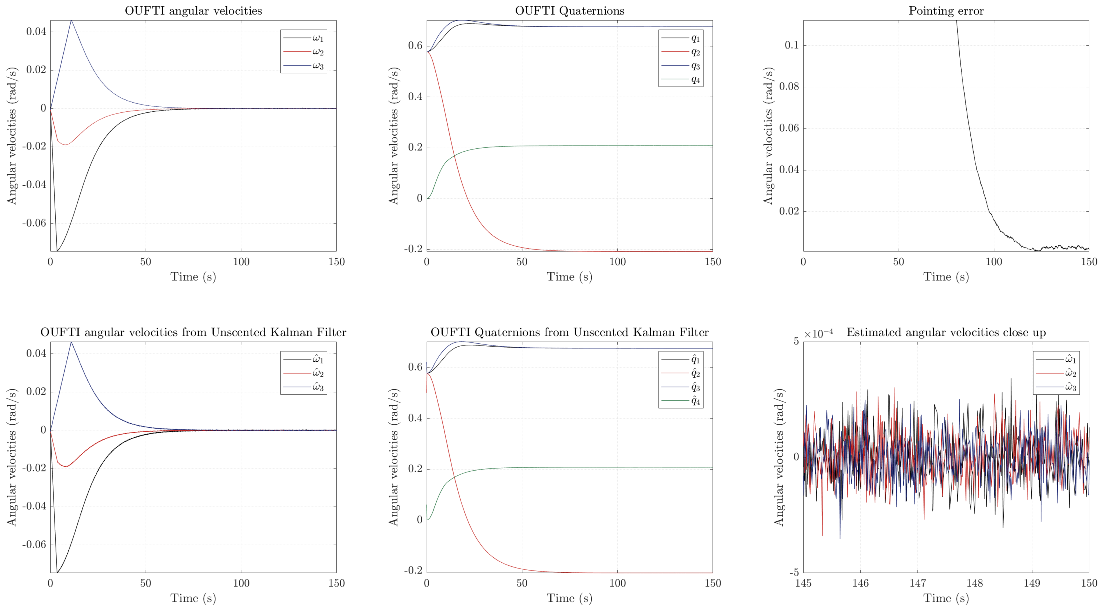


Figure 9.1 Unscented Kalman filter estimated state compared with the actual one.

Usually, on ADC units, the main determination algorithm is a Kalman filter, exactly as for the KUL one, [12], but since in this Thesis the majority of the simulations have been performed with an algebric algorithm based on the star tracker measurements, it could be useful to check, also in this case, the effects of measurements uncertainties. The simulation is equal to the previous one, but performed with the algebric algorithm previously presented and, as a consequence, the computational time is reduced: less than 3 min. The results demonstrate that the algebric algorithm is able to estimate the state maintaining the pointing requirements, figure 9.2.

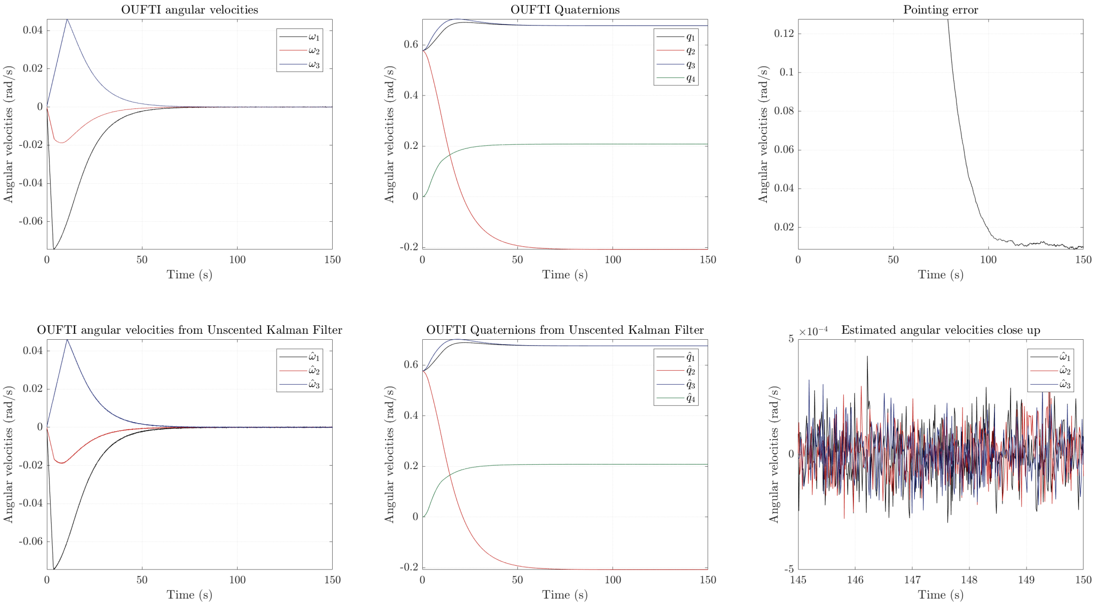


Figure 9.2 Algebric algorithm estimated state compared with the actual one.

Also in the case of the simulation with the algebric algorithm, the addition of noise makes the ODE15s numerically instable and, for this reason, the solver used has been ODE113, as in the previous case.

Chapter 10

Trade-off between *iADCS100* and *KUL ADCS unit*

In this Thesis all the simulations have been carried out using the data of *iADCS100* by *Hyperion*. Nevertheless, it is interesting to compare (also through dedicated simulations) the performances with the ones of *KUL* ADC unit, [12], in order to understand which one is the most suitable for the OUFTI-Next project. In this chapter, different characteristics of the two units will be reported, but no deeper comparisons between the determination algorithms, since *Hyperion* does not declares which one is used in *iADCS100* on his flyer, [10]. According to KUL instead, their unit exploits an *extended Kalman filter*, which is quite similar to the *unscented Kalman filter* implemented in this Thesis.

10.1 Equipments

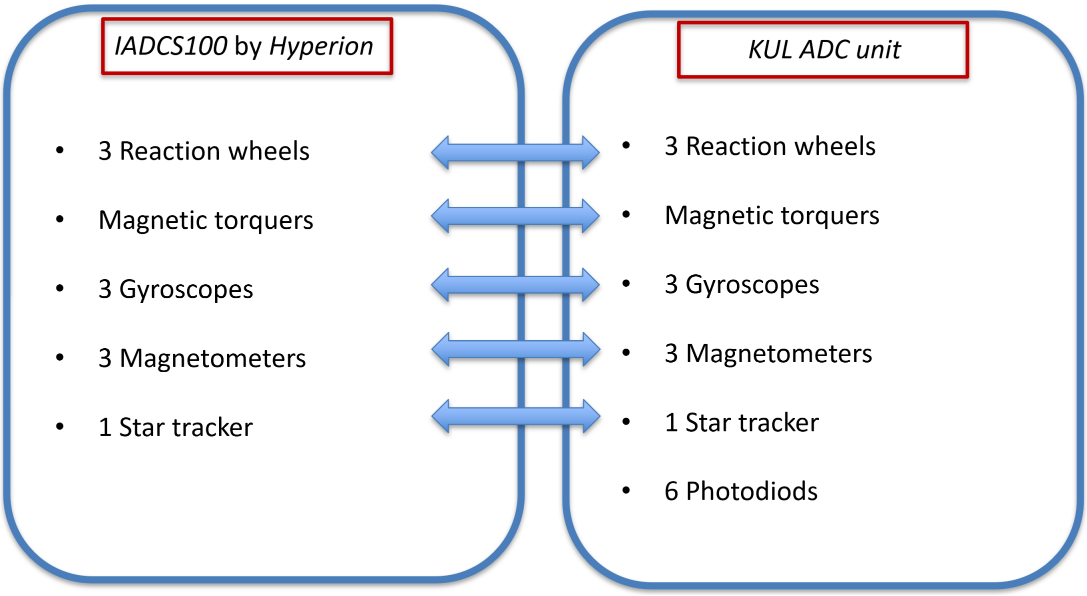


Figure 10.1 Comparison between the main equipments of both *iADCS100* and *KUL unit*.

As it is shown in figure 10.1, the *KUL unit* provides also 6 photodiods, while the *iADCS100* does not explicitly mention them. This sensors are fundamental to retrieve the direction of the Sun (provided by the ephemeris in the simulator coded in this Thesis). According to [10], the *iADCS100* has the interface to be couple with up to 6 Sun-sensors. This means that these devices would be bought independently from the ADC unit, increasing the cost of the entire subsystem. Nevertheless, the solar panels usually used for CubeSats have already integrated Sun-sensors, which could be exploited, even if their performances are worst with respect to the ones of dedicated devices. In conclusion, from the equipment point of view, the two units are almost identical.

10.2 Performances

From the performances point of view, here are reported the main differences between the two units:

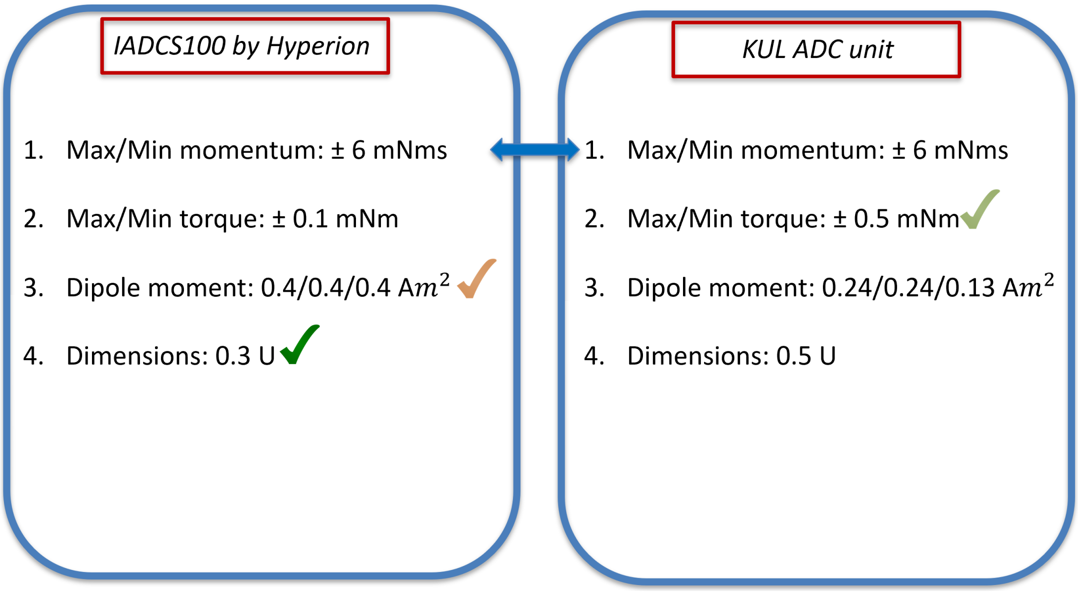


Figure 10.2 Comparison between the performances of both *iADCS100* and *KUL unit* components.

As it is remarked in figure 10.2, both of the units provide the same amount of maximum/minimum momentum storable by the wheels, which is one of the most important parameter for the ADC subsystem, since it drives the saturation time. The highest torque that reaction wheels can produce is provided by the *KUL* unit and this is a quite good point with respect to *iADCS100*, because this parameter drives the control torque saturation, really important during transient phases, driving the stabilization time of the platform. Changing the maximum torque from 0.1 mNm to 0.5 mNm, the time to get a steady state response decreases from 100 s to 50 s. The magnetic torquers of *iADCS100* produce an higher dipole moment per axis, but this is not a fundamental performance, since it drives only slightly the de-tumbling time, decreasing/increasing it of almost 15 min.
The most critical aspect for the OUFTI-Next project is the dimension of the ADCS, since half of the volume available is occupied by the payload and the volume dedicated to the ADC subsystem is reduced. As a result, the dimensions of *iADCS100*, 0.3 U, are much more suitable with respect to the one of the *KUL* unit, 0.5 U, and this could be one of the most important drivers for choosing the unit by *Hyperion*.

10.3 Accuracy

After a meeting with the person in charge of the *KUL ADCS unit*, the data about their sensors accuracy have been obtained and a set of simulations, similar to the ones of chapter 9 on ISS orbit in Sun-pointing, have been performed to check for the pointing budget in presence of noises and measurements uncertainties. Here the data used are reported:

- Gyroscope angular rate noise: $1.4e^{-4}\ \frac{\text{rad}}{\sqrt{\text{s}}}$.
- Gyroscope random rate noise: $5.24e^{-6}\ \frac{\text{deg}}{\sqrt{\text{s}^3}}$.
- Star tracker cross boresight error: 40 arcsec (real sky tests show that it is much smaller, down to around 2 arcsec, but KUL declares 40 arcsec as more conservative value).
- Star tracker boresight error: 300 arcsec (real values show almost 10 arcsec).
- Magnetometers error: 3° (1 σ).
- Photodiods error: 5° (1 σ).

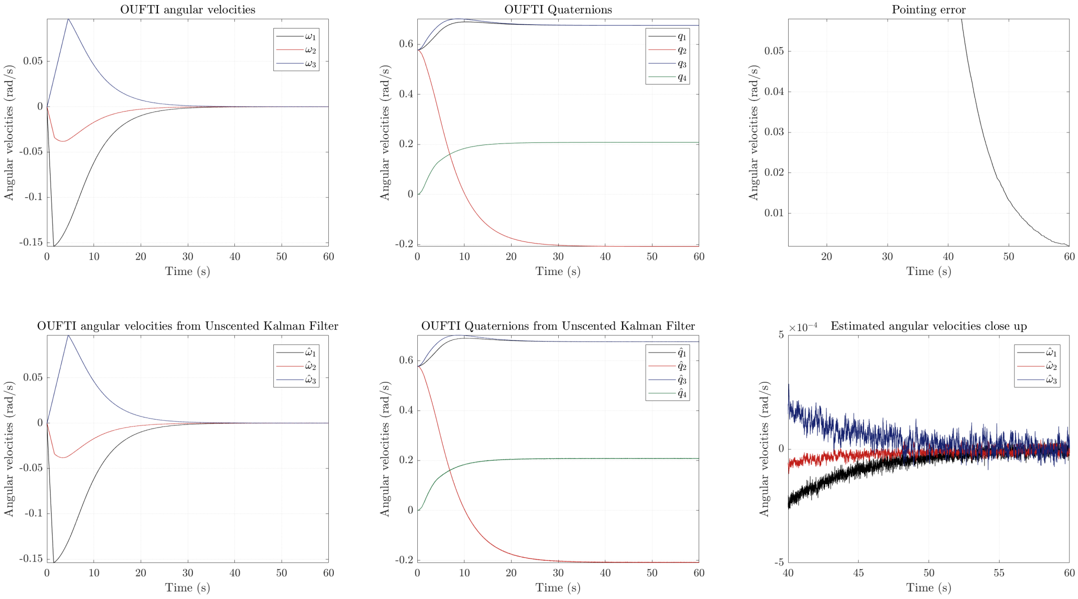


Figure 10.3 Simulation results concerning the pointing budget and estimation quality of KUL unit.

From figure 10.3, it s possible to appreciate that the pointing accuracy requirements are satisfied. Nevertheless, KUL declares in his flyer [12], that the best accuracy possible during fine pointing is 0.11°. This is due to the fact that their simulator is based on a much more accurate typical noises generator, PEET by ESA, which decreases the resulting pointing accuracy of the device. To build such a complex model is beyond the aim of this Thesis and, once the supplier for the ADC subsystem will be chosen, it will provide deeper simulations considering its unit and OUFTI-Next integrated.

10.4 Conclusions

The two ADCS units here presented are very similar from the equipment/performances point of view and according to their flyers, they can reach pointing budgets in accordance with the project requirements ones. For this reason the driving parameter is not related to the ADCS, but to the configuration subsystem, since the unit should be the one with the smallest volume. As a result, *iADCS100* by *Hyperion* seems to be the best choice at the moment.

Chapter 11

Emergency modes

In this chapter some ADCS failures mitigation strategies will be investigated. For what concerns the operational requirements, the mission is considered failed when OUFTI-Next experiences more than one failure on-board. As a result, it could be interesting to understand how to maintain the control of the spacecraft, when the star tracker or one reaction wheel stops working.

11.1 One reaction wheel failure

If one of the wheels fails, it is usually possible to switch the control logic to a mixed one: one magnetic torquer and two reaction wheels. In order to do that, the quaternion control law remains the same, but only two components of the requested control torque will be provided by the wheels, while the remaining one will be provided by one magnetic torquer. Also the integration of the two reaction wheels momentum is the same as before; what changes is the magnetic torquers control strategy. The steps for the design are the following, based on the case in which the broken reaction wheel is the one with its gyroscopic axis oriented as the Y-axis:

1. Evaluate the desired control torque, $\boldsymbol{u}$, exactly as in nominal mode.
2. Integrate just the first and third components of reaction wheels momentum and then check for saturation both the values obtained and the first and third components of the reaction wheels torque. At the end of this point, two of the three components of the real control torque are known: $u_{r,x}$ and $u_{r,z}$.
3. The second component is obtained from magnetic torquers active control law:

$$\boldsymbol{u_{r,y}} = \begin{bmatrix} 0 & -B_z & B_y \\ B_z & 0 & -B_x \\ -B_y & B_x & 0 \end{bmatrix} \boldsymbol{m} \tag{11.1}$$

where B_i are the components of the magnetic field in *body-fixed reference frame* and $\boldsymbol{m}$ is the requested dipole moment. Unfortunately, since matrix $\boldsymbol{B}$ is skew-symmetric, it is not invertible and, as a consequence, no informations about the requested dipole moment can be retrieved from this expression. In the case in which two reaction wheels stop working, it is possible to obtain an invertible expression, but this is not the case. As a result, the only way to solve the problem is to use a coarser pointing law, coming from:

$$\boldsymbol{u_r} = \boldsymbol{m} \times \boldsymbol{B} \tag{11.2}$$

$$\boldsymbol{B} \times \boldsymbol{u_r} = \boldsymbol{B} \times \boldsymbol{m} \times \boldsymbol{B} \tag{11.3}$$

recalling the following vectorial identity:

$$\boldsymbol{a} \times (\boldsymbol{b} \times \boldsymbol{c}) = \boldsymbol{b}(\boldsymbol{a} \cdot \boldsymbol{c}) - \boldsymbol{c}(\boldsymbol{a} \cdot \boldsymbol{b}) \tag{11.4}$$

and assuming that the moment is perpendicular to the magnetic field then:

$$\boldsymbol{B} \times \boldsymbol{u_r} = (\boldsymbol{B} \cdot \boldsymbol{B})\boldsymbol{m} \tag{11.5}$$

$$\boldsymbol{m} = \frac{\boldsymbol{B} \times \boldsymbol{u_r}}{(\boldsymbol{B} \cdot \boldsymbol{B})} \tag{11.6}$$

taking then only the second component.

This control strategy has been checked on target following picture acquisition phase on ISS orbit, providing the following results:

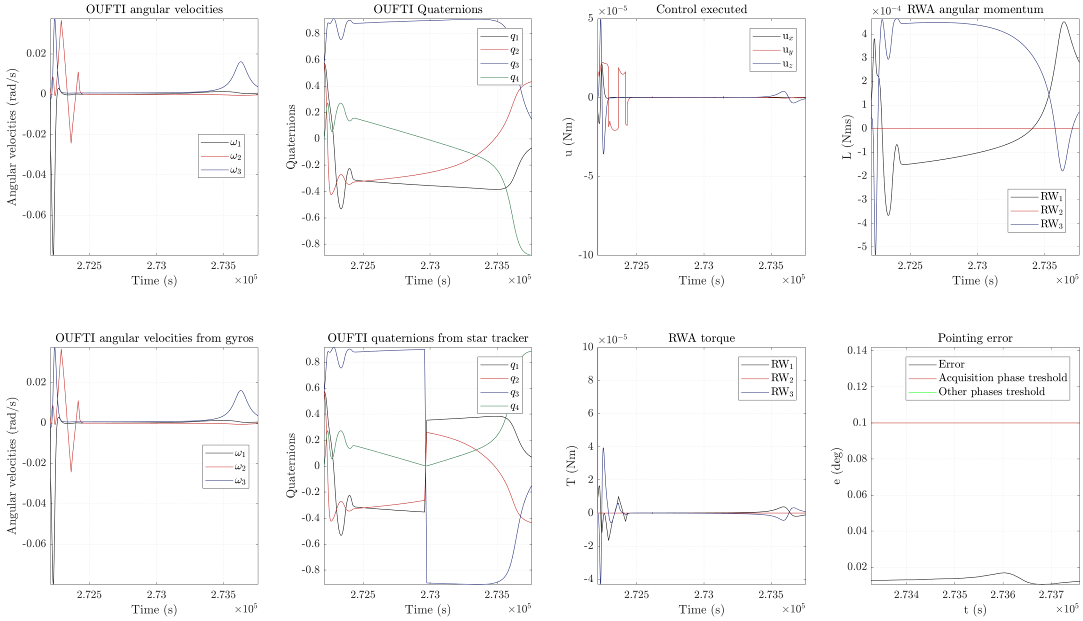


Figure 11.1 Emergency control strategy results with one magnetic torquer and two reaction wheels.

As it can be shown in figure 11.1, the control law satisfies the pointing accuracy requirement. An important remark is about which reaction wheel breaks: if the one whose gyroscopic axis is oriented along the pointing axis stops working, the pointing stabilization is reached in the same time of the nominal scenario, while the spacecraft will rotate for some seconds more along the pointing axis. In the two remaining cases, the spacecraft will need more time to stabilize, exactly as for the case analyzed in this chapter, in which the stabilization time doubles.

11.2 Star tracker failure

If the star tracker fails, one of the less demanding determination techniques which can be used is the *Triad method*. In order to use it, two vectorial measurements are needed and, in the case of OUFTI-Next, they are provided by the magnetometers and photodiods. The CubeSat has on-board the Sun ephemeris, $\boldsymbol{S_n}$, and the Earth magnetic field vector from an on-board model, $\boldsymbol{B_n}$, in *inertial reference frame*; from its sensors it can obtain the Sun direction and the magnetic field vector in *body-fixed reference frame*, $\boldsymbol{S_b}$ and $\boldsymbol{B_b}$. Once all of them are known, it is possible to achieve the attitude matrix following these steps:

$$\boldsymbol{V_b} = \begin{bmatrix} \boldsymbol{B_b} & \frac{\boldsymbol{B_b} \times \boldsymbol{S_b}}{|\boldsymbol{B_b} \times \boldsymbol{S_b}|} & \boldsymbol{B_b} \times \frac{\boldsymbol{B_b} \times \boldsymbol{S_b}}{|\boldsymbol{B_b} \times \boldsymbol{S_b}|} \end{bmatrix} \tag{11.7}$$

$$\boldsymbol{V_n} = \begin{bmatrix} \boldsymbol{B_n} & \frac{\boldsymbol{B_n} \times \boldsymbol{S_n}}{|\boldsymbol{B_n} \times \boldsymbol{S_n}|} & \boldsymbol{B_n} \times \frac{\boldsymbol{B_n} \times \boldsymbol{S_n}}{|\boldsymbol{B_n} \times \boldsymbol{S_n}|} \end{bmatrix} \tag{11.8}$$

$$\boldsymbol{V_b} = \boldsymbol{A_{b/n}} \boldsymbol{V_v} \rightarrow \boldsymbol{A_{b/n}} = \boldsymbol{V_b} \boldsymbol{V_n}^T \tag{11.9}$$

with

$$\boldsymbol{B_b} = \boldsymbol{A_{\epsilon,B}} \boldsymbol{A_{b/n}} \boldsymbol{B_n} \tag{11.10}$$

$$\boldsymbol{S_b} = \boldsymbol{A_{\epsilon,S}} \boldsymbol{A_{b/n}} \boldsymbol{S_n} \tag{11.11}$$

$\boldsymbol{A_{\epsilon,B}}$ and $\boldsymbol{A_{\epsilon,S}}$ are respectively the bias matrices of the magnetometers and Sun-sensors, based on equation (11.12),where the argument of trigonometric functions are the accuracy on each axis in radians:

$$\boldsymbol{A_\epsilon} = \begin{bmatrix} cos(\psi)cos(\theta) & cos(\psi)sin(\theta)sin(\phi) + sin(\psi)cos(\phi) & -cos(\psi)sin(\theta)cos(\phi) + sin(\psi)sin(\phi) \\ -sin(\psi)cos(\theta) & -sin(\psi)sin(\theta)sin(\psi) + cos(\psi)cos(\theta) & sin(\psi)sin(\theta)cos(\psi) + cos(\psi)sin(\theta) \\ sin(\theta) & -cos(\theta)sin(\phi) & cos(\theta)cos(\phi) \end{bmatrix} \tag{11.12}$$

According to KUL, the accuracy of the magnetometers and photodiods are respectively 3° and 5°.

This determination strategy has been checked during Sun-pointing phase on ISS orbit, providing the following results:

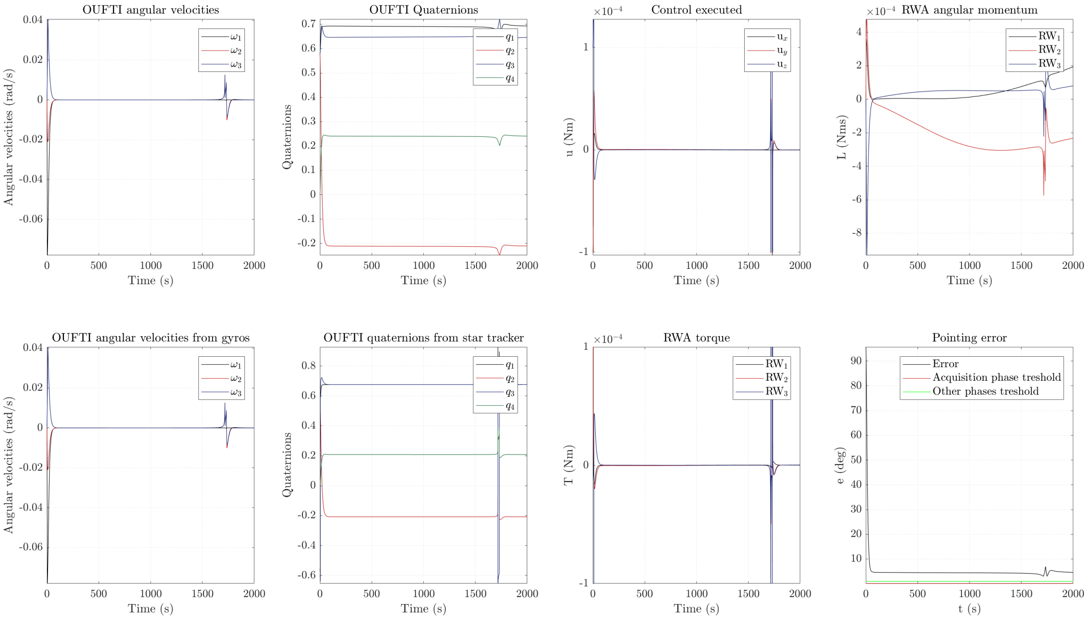


Figure 11.2 Emergency determination strategy results based on *Triad method*.

As expected from the low accuracy of the sensors, the pointing accuracy is not enough to fulfill the requirements during picture acquisition, but the CubeSat can still point the Sun and store energy on-board. Finally it can be interesting to remark the presence of some peaks in control plots, affecting slightly the dynamics, figure 11.2. It can be supposed that the issue is due to the lack in sensors accuracy, since no numerical instabilities or noises are evidenced (the simulation has been run also with different solvers, providing the same results).

Chapter 12

Conclusions

12.1 Results achieved

Thanks to this Thesis it has been possible to simulate the control and determination action of the OUFTI-Next platform ADC subsystem, providing three mission profiles which represent the operative behavior of the CubeSat during its life time, both in ISS orbit and SSO.

In chapter 1 an introduction to the OUFTI-Next project has been provided and the structure of the work explained, then in chapter 2 the fundamental nomenclature has been reported and in chapter 3 the main results, previously obtained from the team about the other subsystems, are presented.
In chapter 4 it has been described how the physics of the problem has been modeled, while chapter 5 has been dedicated to explain the logic behind control parameters optimization, then in chapter 6 the majority of the results has been listed and commented. Additional analysis have been performed in chapters 7, 8, 9 and 10 respectively about the cryocooler disturbances effects, the sensors noises and update rates, the trade-off between *Hyperion* and KUL ADCS units and the emergency strategies in case of one reaction wheel or the star tracker failure.
This Thesis has also an appendix, where the reader can find the results of the second mission profile, both in SSO and ISS orbits, with target following slew maneuvers, and the de-tumbling model made in Simulink™.

No power budget has been reported, since the power requested by the ADCS unit is almost constant during all the mission phases and a preliminary analysis on the peak power requested (20% margined) was already performed in [6].
From the results obtained, the most important remarks are here reported:

- The de-tumbling phase will last approximately 1 hour and 30 minutes.
- The saturation of the reaction wheels can be managed with de-saturation phases tested in this Thesis. The maximum time the spacecraft can remain in Sun-pointing before saturating is around 17 hours on ISS orbit and around 22 hours on SSO.
- The pointing budget of 0.1° is perfectly achieved in all the picture acquisition phases, even the ones with target following slew maneuvers. Also the pointing stability is guaranteed under the value requested, 0.01°.
- The ADCS unit should be located the closest to the actual center of mass of the platform, which should be as close as possible to the geometric center. Changes in its position will not affect dramatically the results obtained in this Thesis, but more refined simulations should be performed in order to guarantee the pointing accuracy and stability.
- The cryocooler disturbance seems not to increase the pointing error till exceeding the admissible value, 0.1° and the pointing stability is maintained guaranteed as well.

- The introduction of noises and sensors update rates, with the actual level of detail, is not altering the determination error, maintaining the pointing error below the previously stated threshold.
- From the trade-off analysis it is clear that both *iADCS100* and *KUL unit* are able to satisfy all the OUFTI-Next ADCS requirements with similar performances. The driving parameter that will probably affect the choice between them is the size.
- If one reaction wheel fails, the pointing stability and accuracy can be in any case maintained, doubling the stabilization time.
- If the star tracker fails the determination algorithm implemented, based on *Triad method*, is able to guarantee a pointing error of around 9°, meaning that no picture acquisition is possible, but Sun-pointing can be maintained, as well as power generation on-board.

The aim of this Thesis was to demonstrate that the pointing requirements needed by the detector can be achieved using ADCS units of reduced size, such as the ones provided by *Hyperion* and KUL. Once the ADCS unit supplier will be chosen, it will provide deeper analysis about the specific unit performances using a professional simulator.

This report concludes the work done at ULg, then, the part of Thesis dedicated to the coupling between orbital and attitude dynamics will start at Politecnico di Milano. The data about ZodiArt platforms will be loaded in the simulator developed at ULg, in order to provide some mission scenarios results, exactly as it has been done for OUFTI-Next. After that, the equation representing the coupled dynamics (orbit and attitude) will be implemented and some important results about the comparison between the two modelizations will be achieved.

12.2 Future developments

Here are reported future developments that can be accomplished in future design steps of OUFTI-NextADC subsystem:

- To provide a deeper analysis on the de-tumbling phase, which is the most unpredictable one, with magnetometers measurements and noises as well. Also a Montecarlo analysis or just a sensitivity analysis varying the residual dipole moment would be advisable.
- To optimize the control constant k, used in de-tumbling phase.
- All the phases should be simulated with the exact inertia matrix and with the exact location of all the subsystems and the payload, benchmarking the results with the ones of this Thesis.
- To implement a more complex Kalman filter, capable of estimating also the gyroscopes and star tracker bias.
- To perform a deeper analysis with PEET by ESA to refine the analysis performed with noises.

Finally the code provided with this Thesis can be optimized and improved in order to simulate more realistically the attitude on orbit, once all the other subsystems have been completely sized and located inside the CubeSat.

Chapter 13

Appendix

13.1 Mission 2 on ISS orbit with target following picture acquisition

Computational time usually requested: around 10 minutes, depending on the initial conditions.

13.1.1 Sun-pointing 1

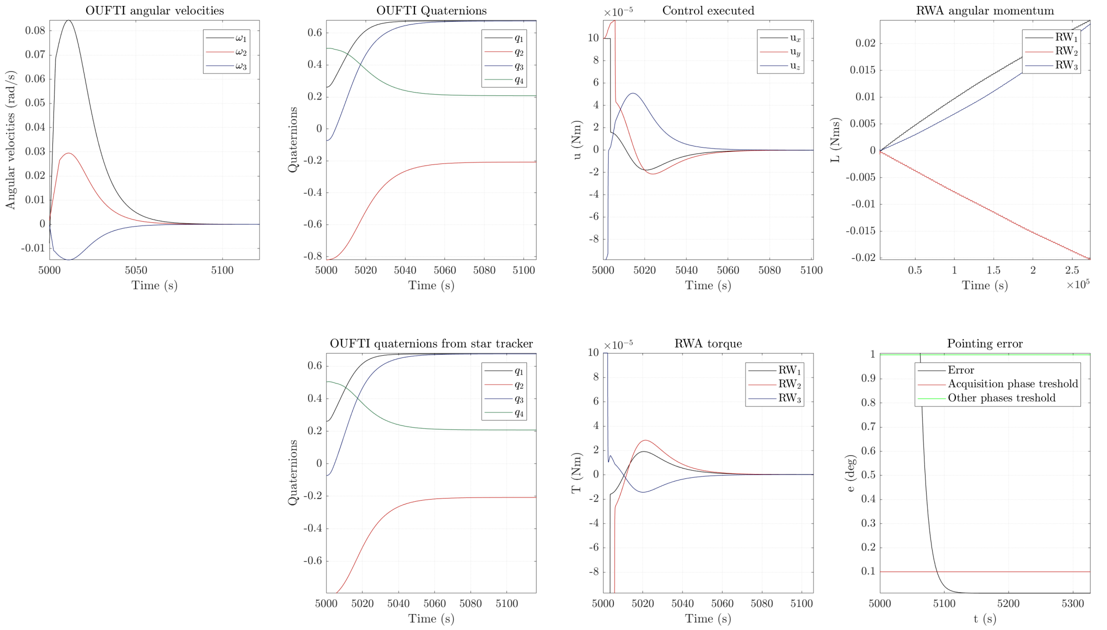


Figure 13.1 Most important results during Sun-pointing phase 1, in mission 2, on ISS orbit. The plots have been zoomed in order to catch the most interesting behaviors.

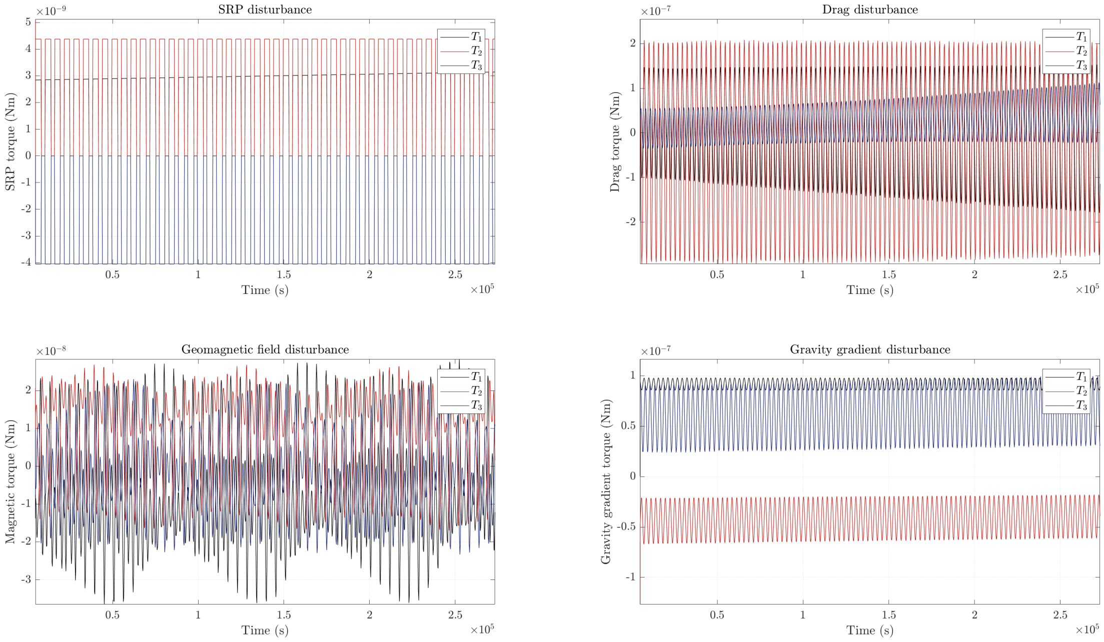


Figure 13.2 Main disturbances torques during Sun-pointing phase 1, in mission 2, on ISS orbit.

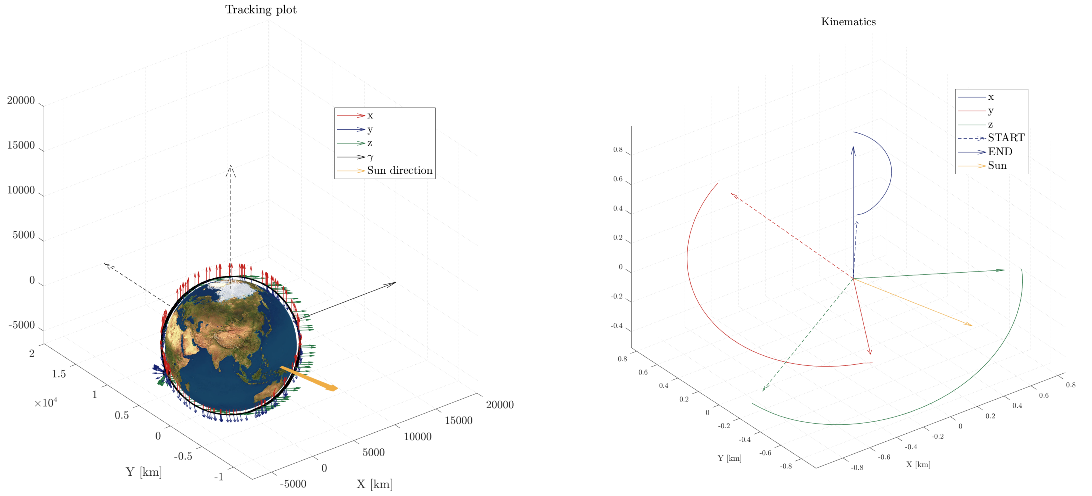


Figure 13.3 Body-fixed frame while orbiting around the earth during Sun-pointing phase 1, in mission 2, on ISS orbit.

13.1.2 De-saturation 1

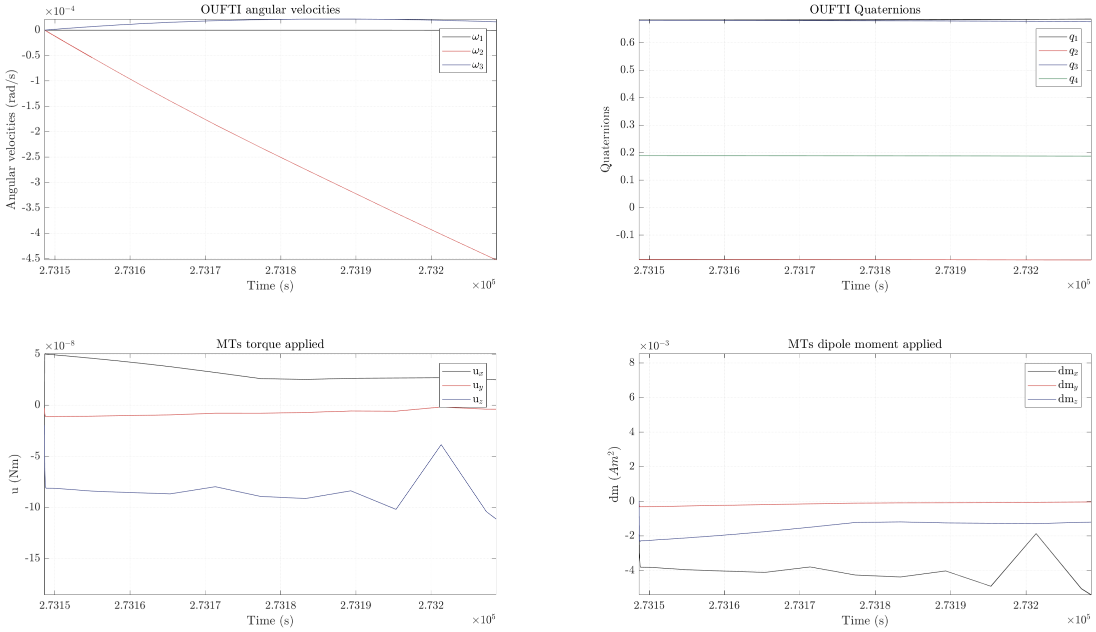


Figure 13.4 Most important results during de-saturation phase 1, in mission 2, on ISS orbit.

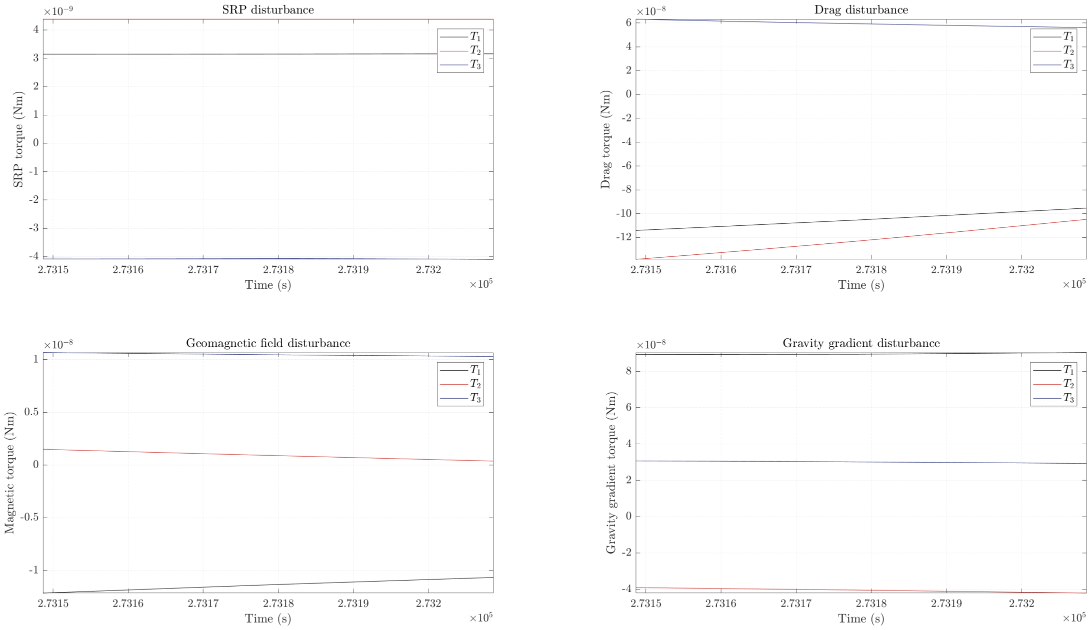


Figure 13.5 Main disturbances torques during de-saturation phase 1, in mission 2, on ISS orbit.

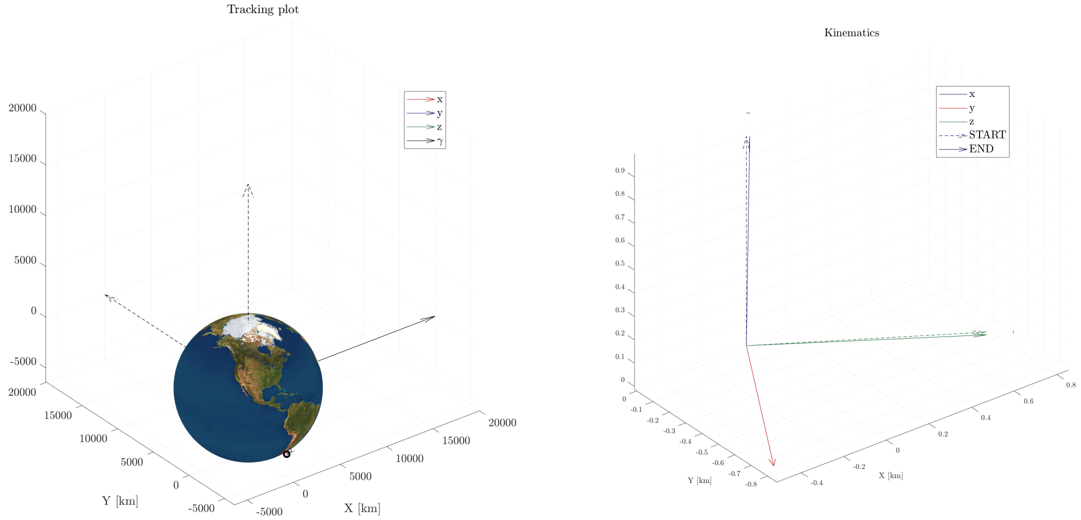


Figure 13.6 Body-fixed frame while orbiting around the earth during de-saturation phase 1, in mission 2, on ISS orbit.

13.1.3 Picture acquisition

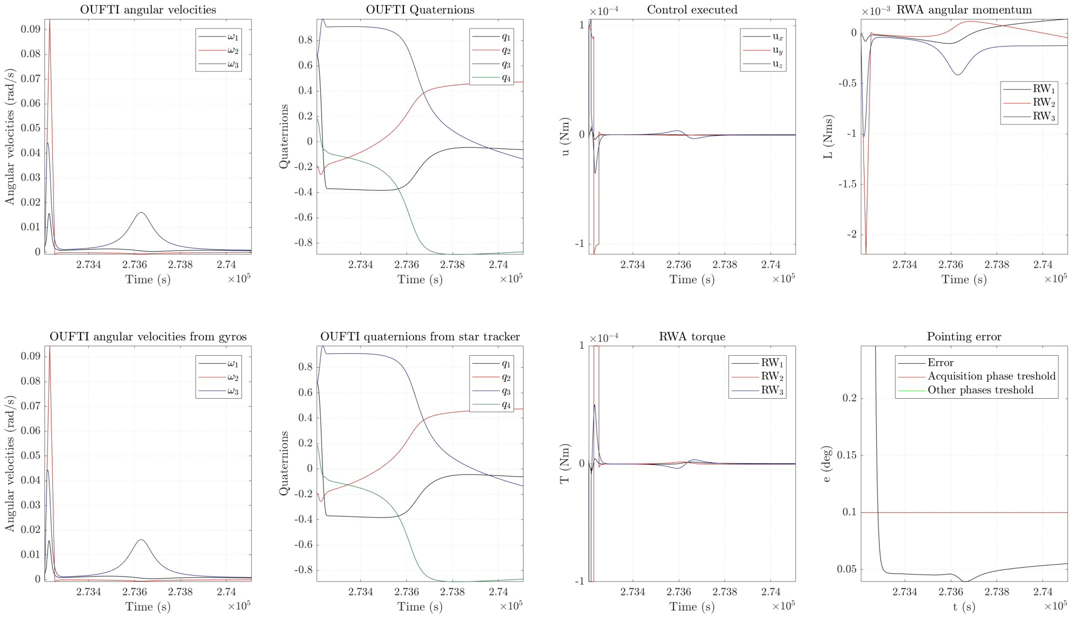


Figure 13.7 Most important results during target following pointing picture acquisition phase, in mission 2, on ISS orbit. The plots have been zoomed in order to catch the most interesting behaviors.

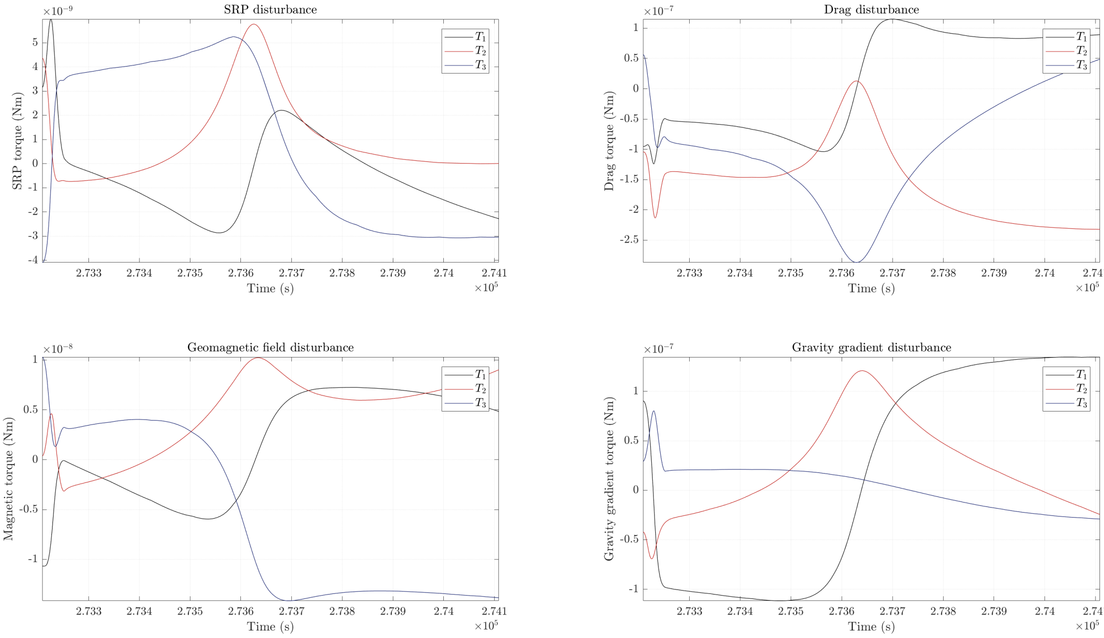


Figure 13.8 Main disturbances torques during Nadir pointing picture acquisition phase, in mission 2, on ISS orbit.

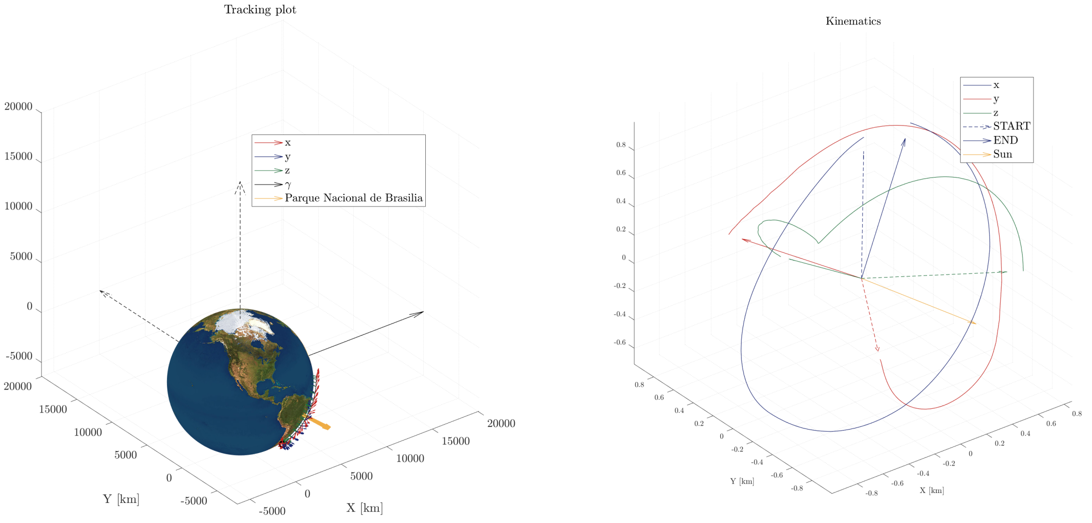


Figure 13.9 Body-fixed frame while orbiting around the earth during Nadir pointing picture acquisition phase, in mission 2, on ISS orbit.

13.1.4 Sun-pointing 2

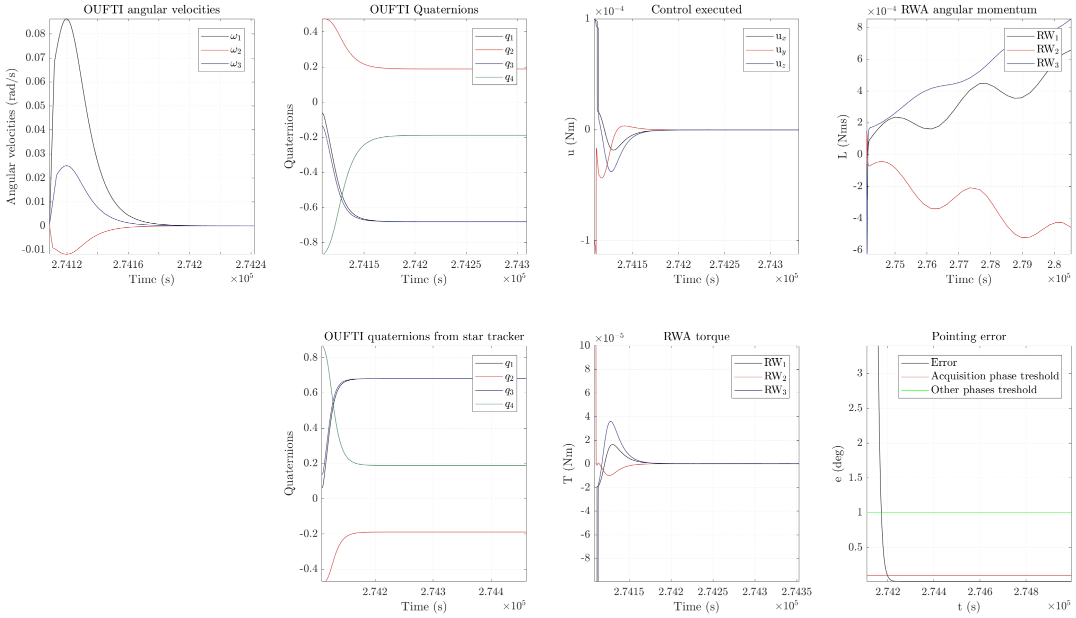


Figure 13.10 Most important results during Sun-pointing phase 2, in mission 2, on ISS orbit. The plots have been zoomed in order to catch the most interesting behaviors.

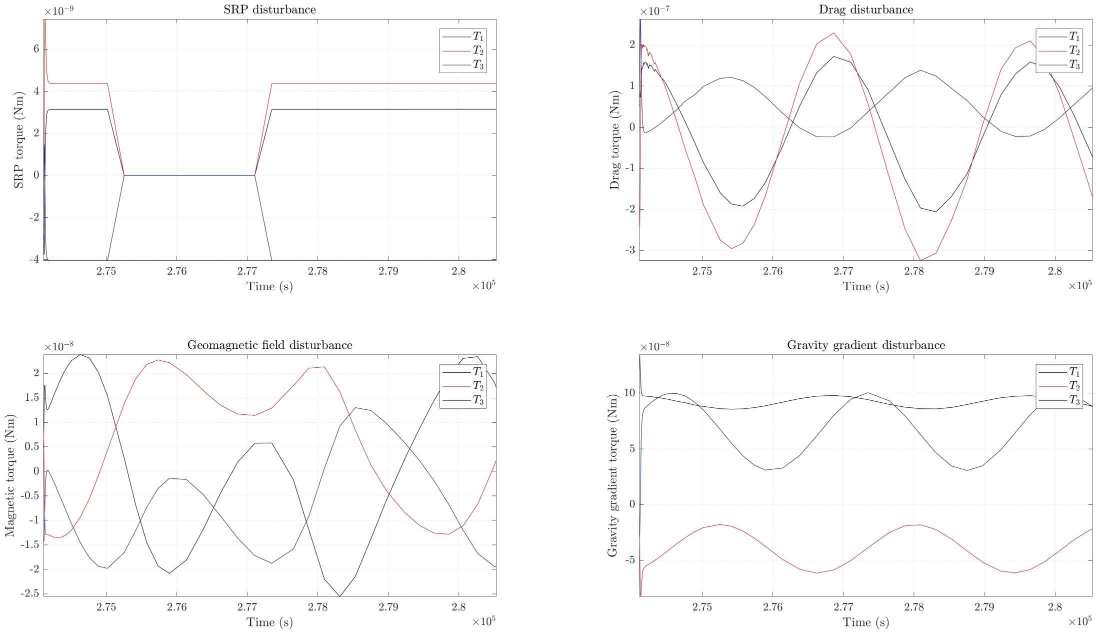


Figure 13.11 Main disturbances torques during Sun-pointing phase 2, in mission 2, on ISS orbit.

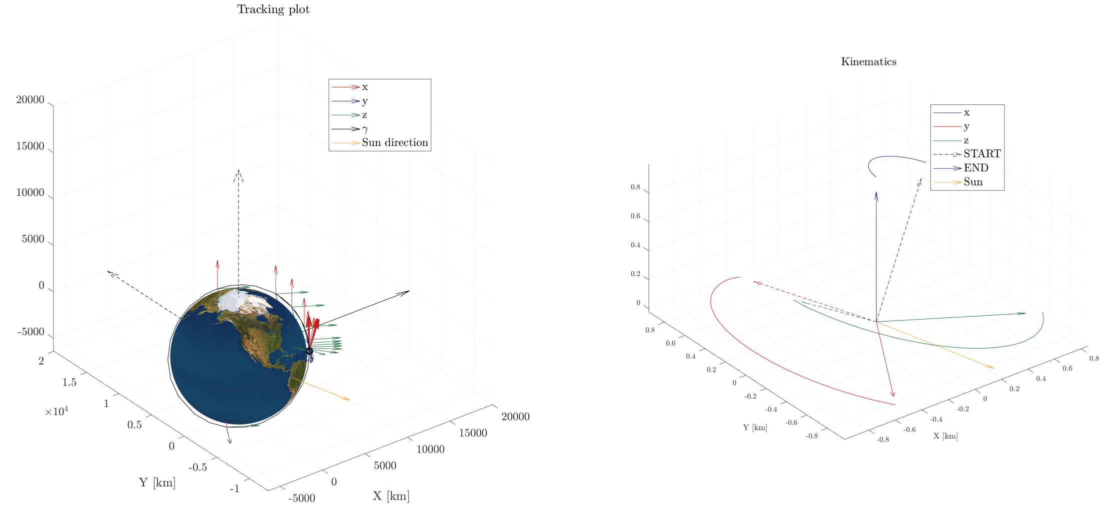


Figure 13.12 Body-fixed frame while orbiting around the earth during Sun-pointing phase 2, in mission 2, on ISS orbit.

13.1.5 De-saturation 2

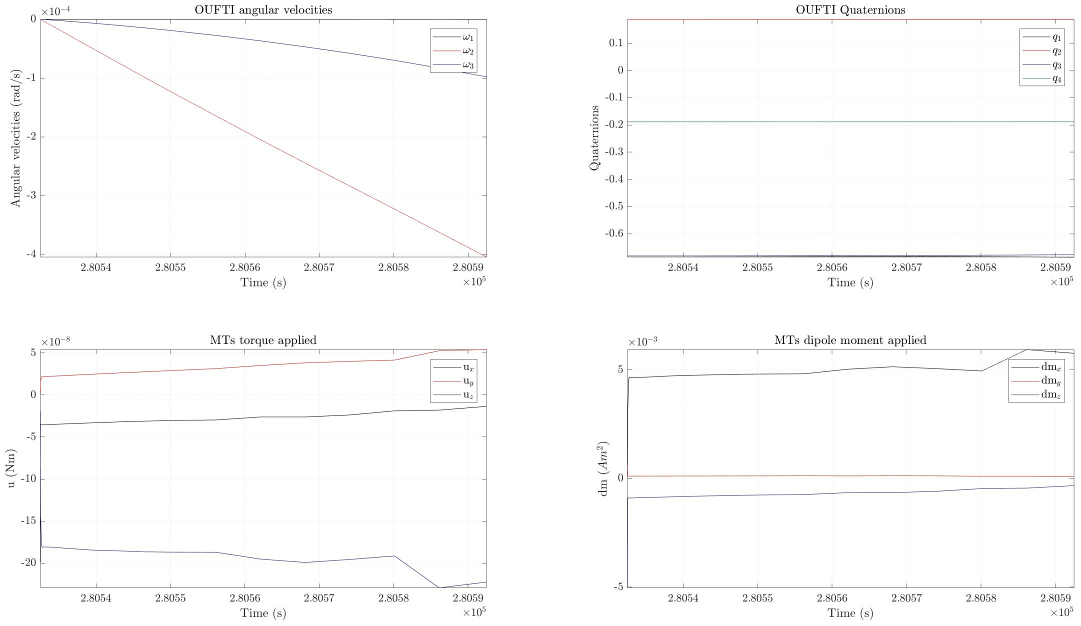


Figure 13.13 Most important results during de-saturation phase 2, in mission 2, on ISS orbit.

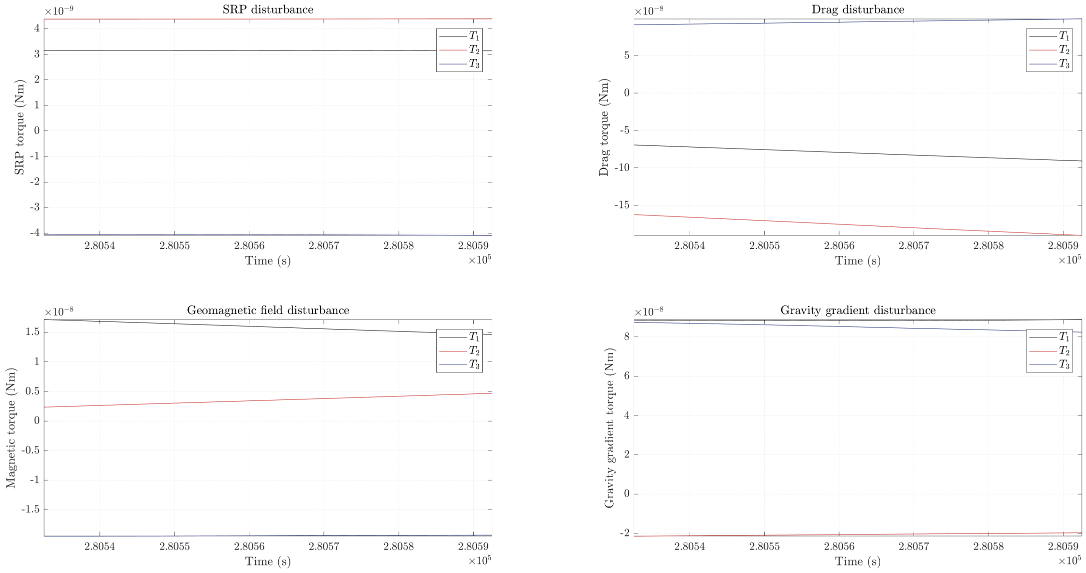


Figure 13.14 Main disturbances torques during de-saturation phase 2, in mission 2, on ISS orbit.

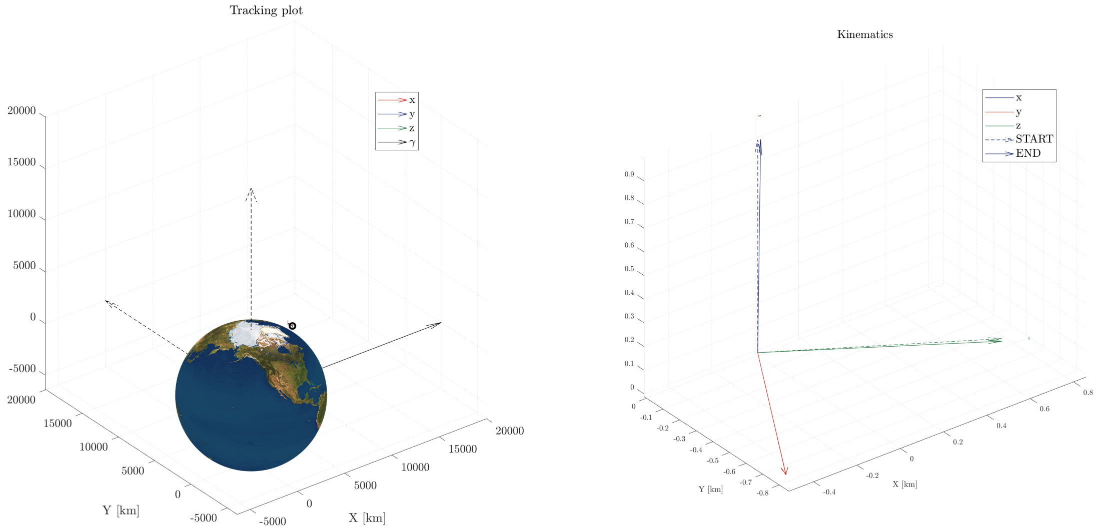


Figure 13.15 Body-fixed frame while orbiting around the earth during de-saturation phase 2, in mission 2, on ISS orbit.

13.1.6 Downloading

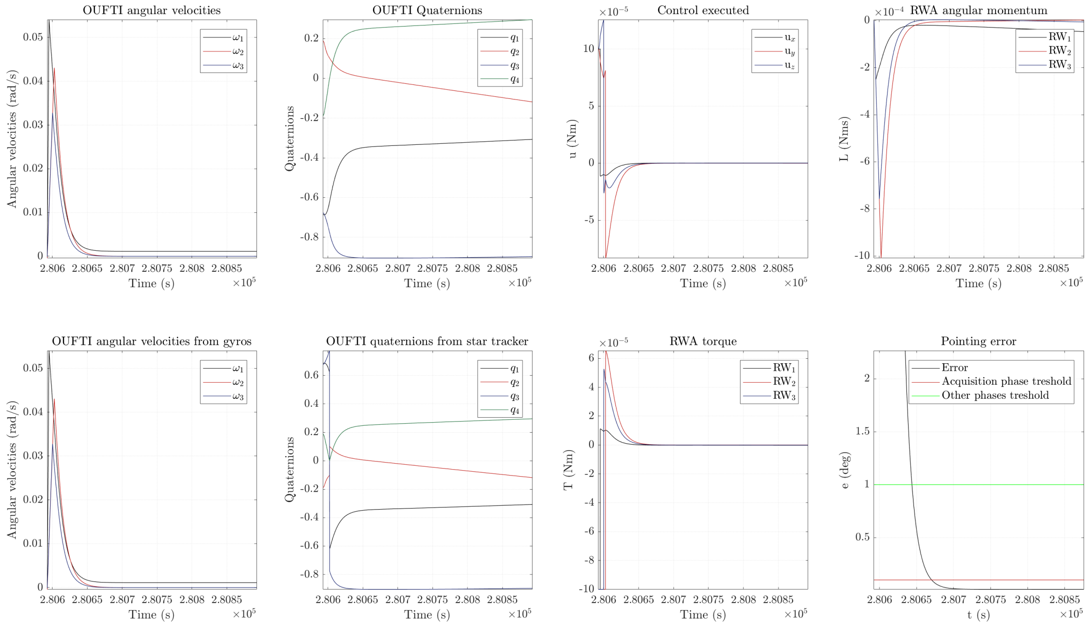


Figure 13.16 Most important results during downloading phase, in mission 2, on ISS orbit. The plots have been zoomed in order to catch the most interesting behaviors.

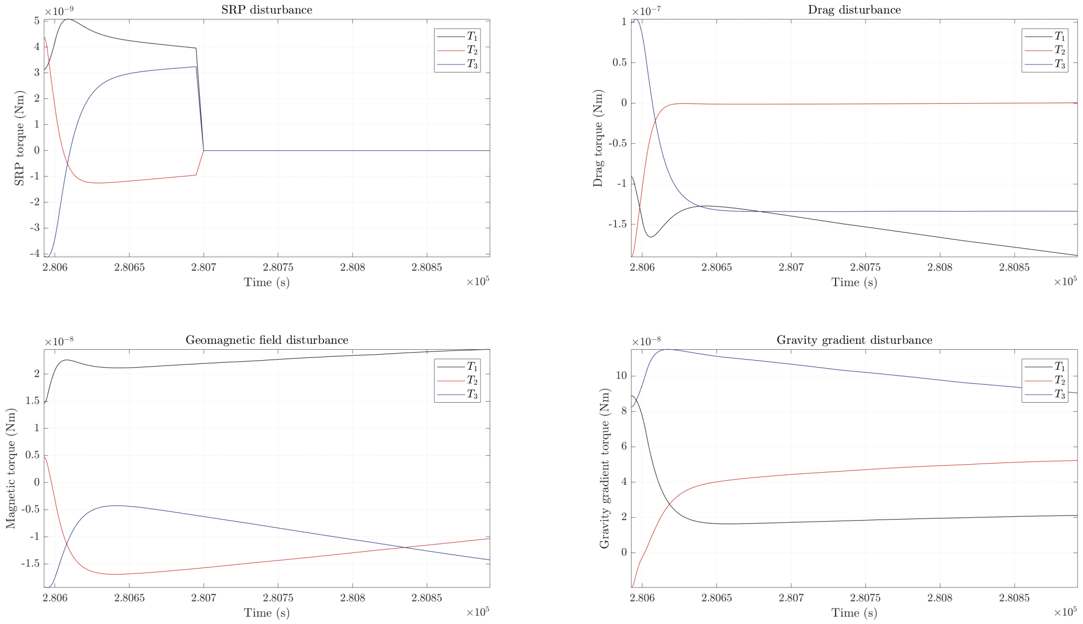


Figure 13.17 Main disturbances torques during downloading phase, in mission 2, on ISS orbit. The plots have been zoomed in order to catch the most interesting behaviors.

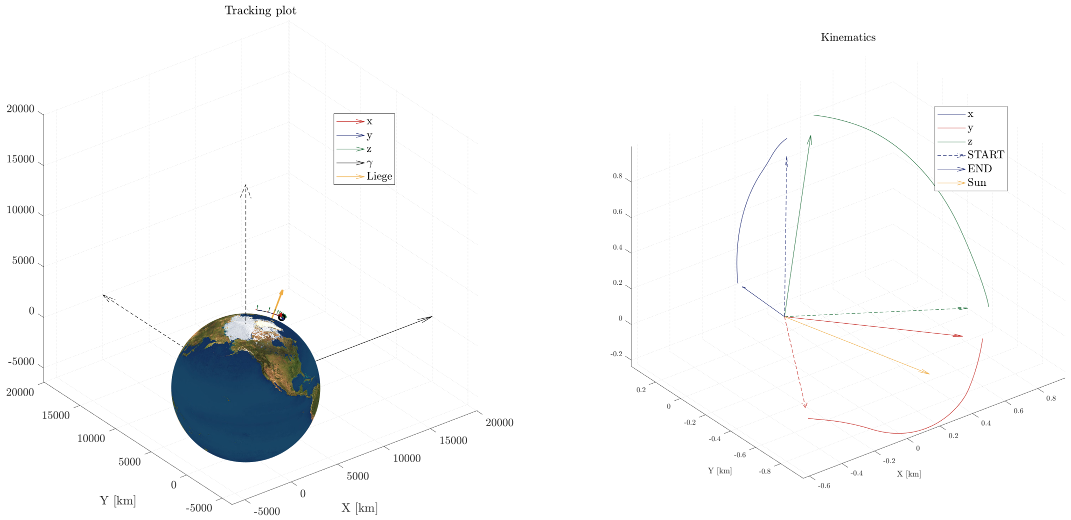


Figure 13.18 Body-fixed frame while orbiting around the earth during downloading phase, in mission 2, on ISS orbit.

13.2 Mission 2 on SSO with target following picture acquisition

Computational time usually requested: around 10 minutes, depending on the initial conditions.

13.2.1 Sun-pointing 1

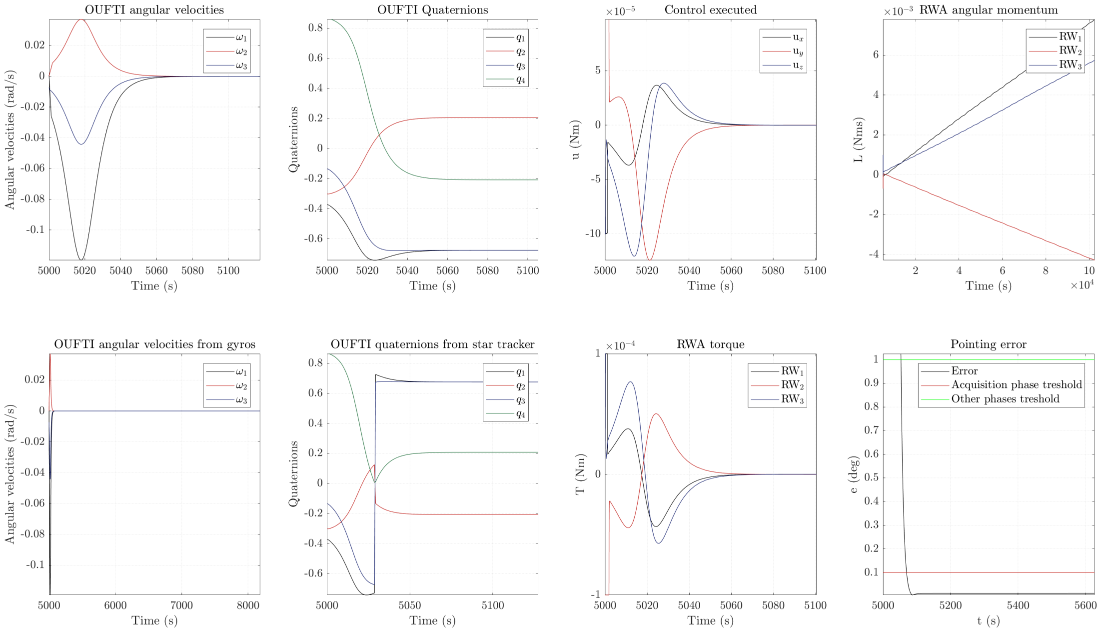


Figure 13.19 Most important results during Sun-pointing phase 1, in mission 2, on SSO. The plots have been zoomed in order to catch the most interesting behaviors.

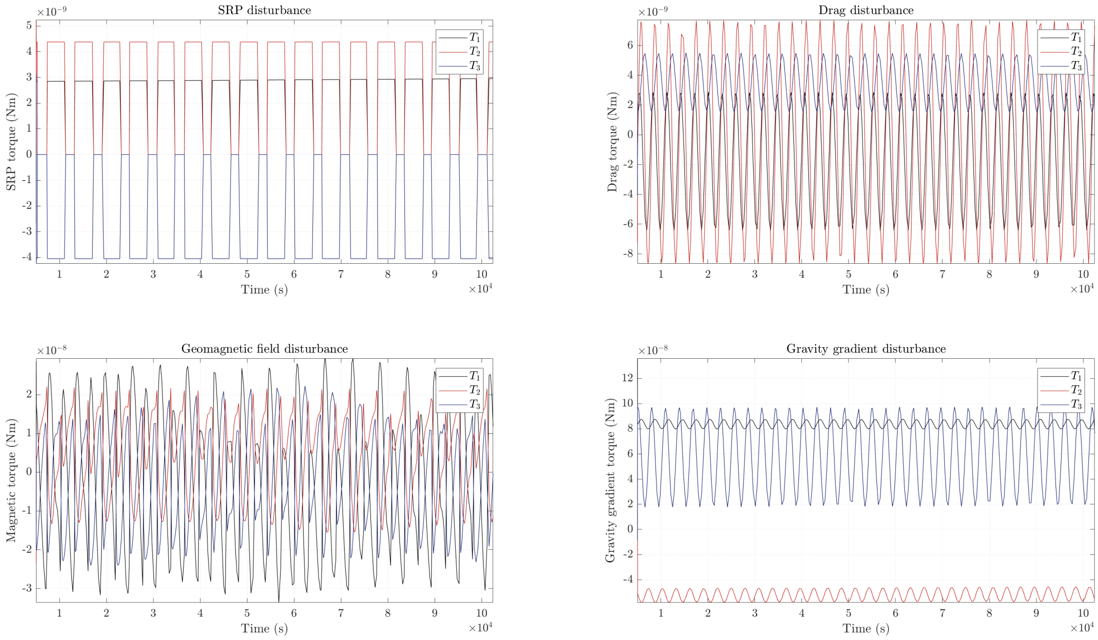


Figure 13.20 Main disturbances torques during Sun-pointing phase 1, in mission 2, on SSO.

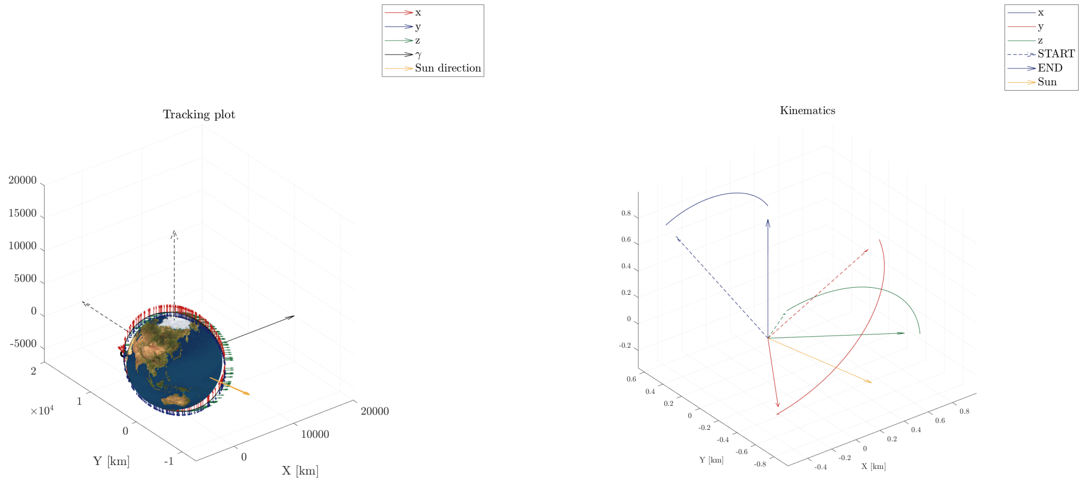


Figure 13.21 Body-fixed frame while orbiting around the earth during Sun-pointing phase 1, in mission 2, on SSO.

13.2.2 De-saturation 1

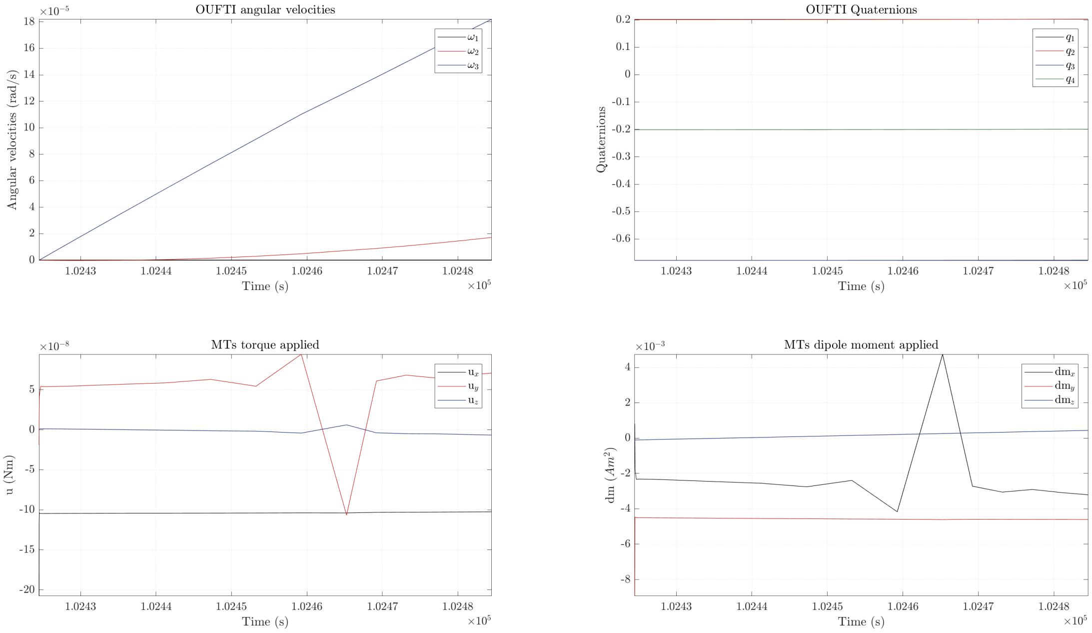


Figure 13.22 Most important results during de-saturation phase 1, in mission 2, on SSO.

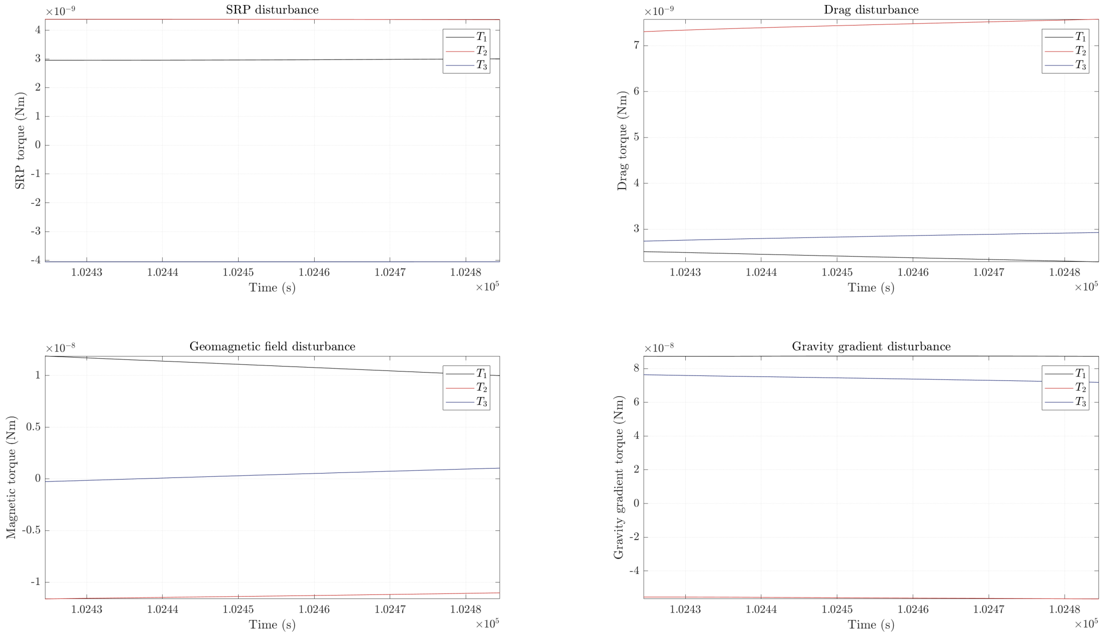


Figure 13.23 Main disturbances torques during de-saturation phase 1, in mission 2, on SSO.

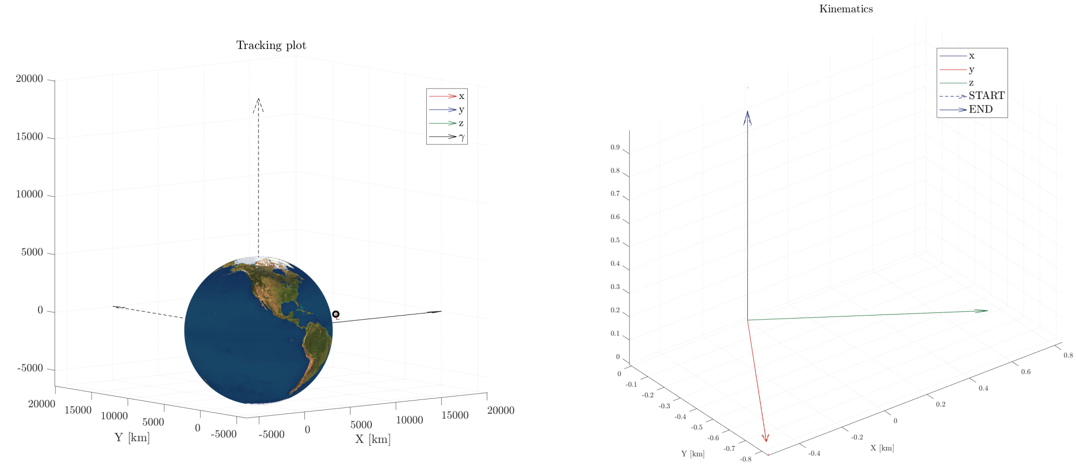


Figure 13.24 Body-fixed frame while orbiting around the earth during de-saturation phase 1, in mission 2, on SSO.

13.2.3 Picture acquisition

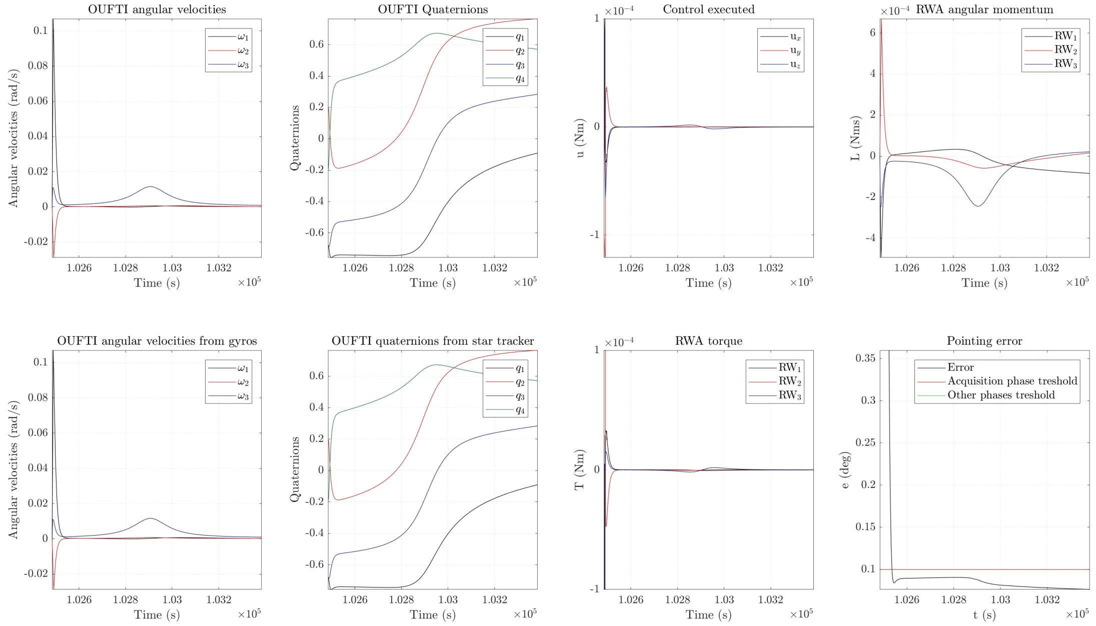


Figure 13.25 Most important results during target following pointing picture acquisition phase, in mission 2, on SSO. The plots have been zoomed in order to catch the most interesting behaviors.

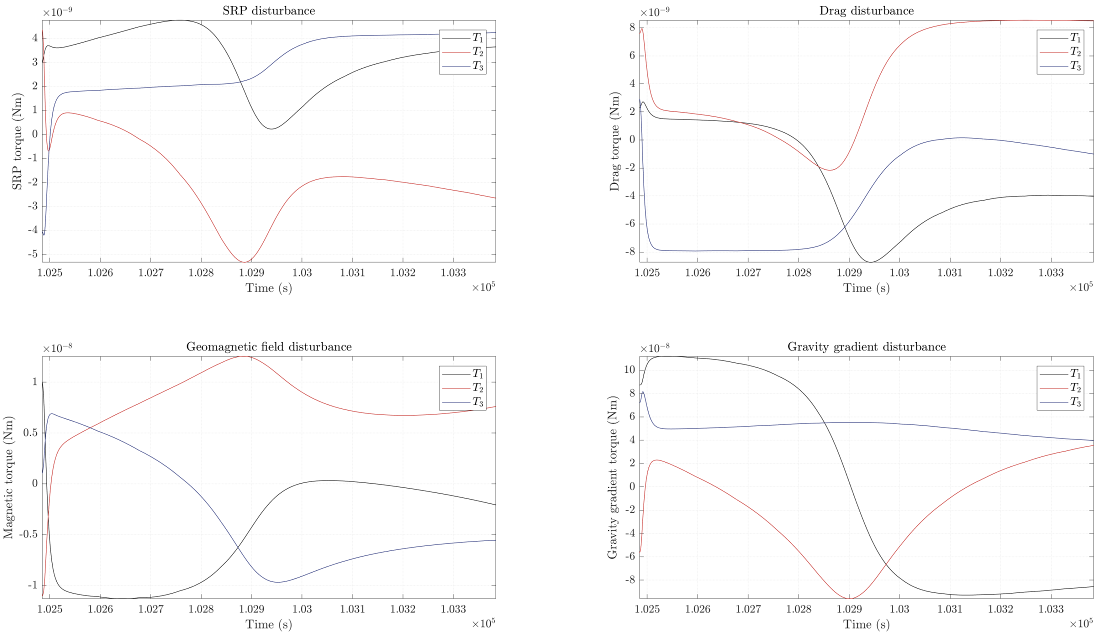


Figure 13.26 Main disturbances torques during Nadir pointing picture acquisition phase, in mission 2, on SSO.

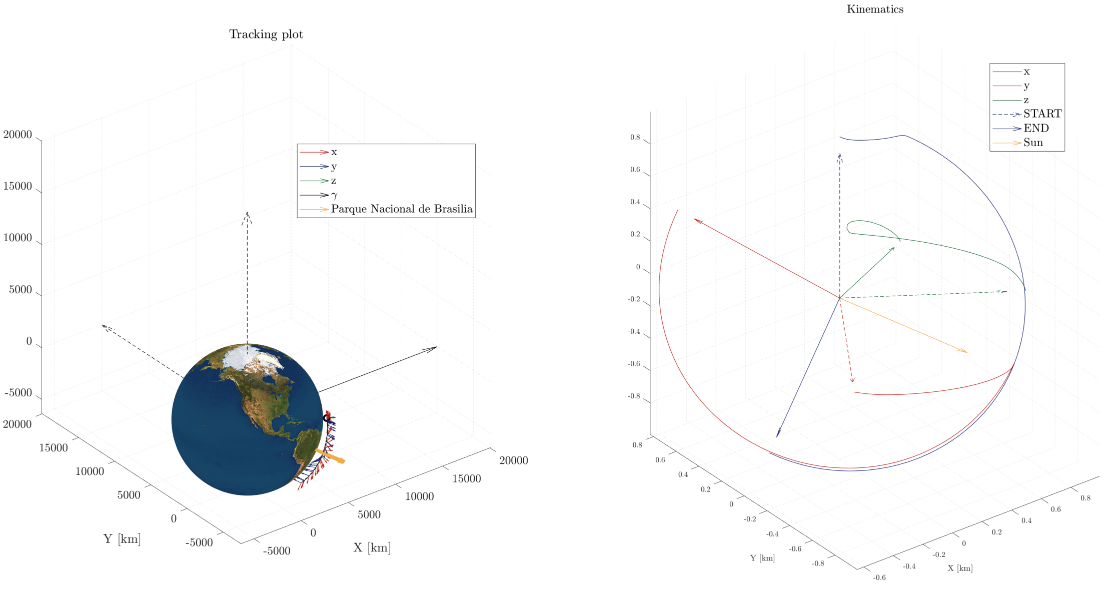


Figure 13.27 Body-fixed frame while orbiting around the earth during Nadir pointing picture acquisition phase, in mission 2, on SSO.

13.2.4 Sun-pointing 2

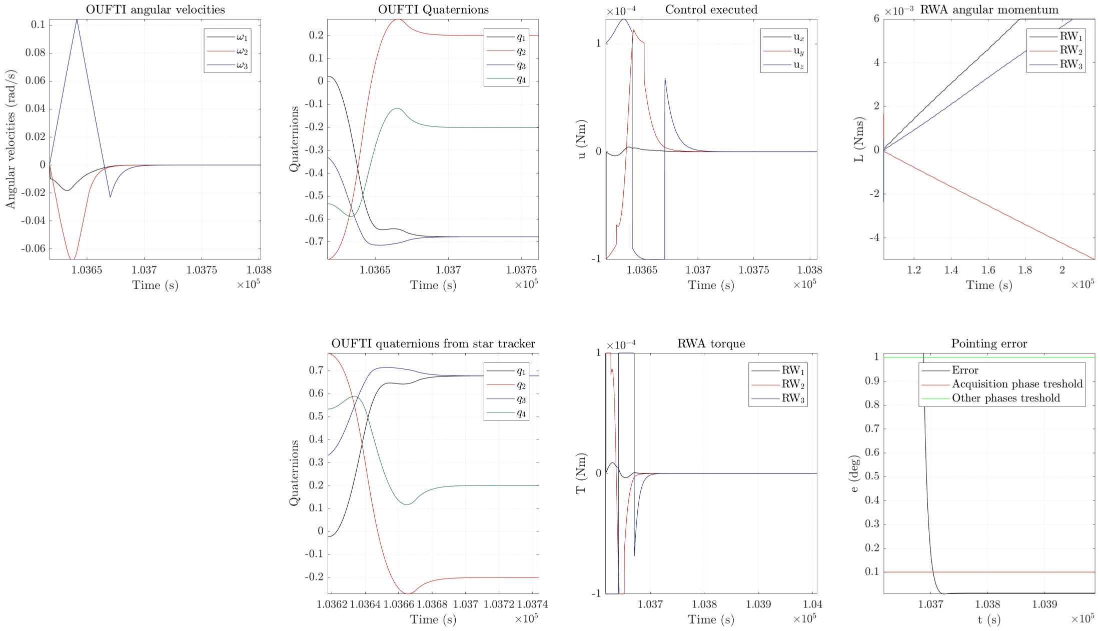


Figure 13.28 Most important results during Sun-pointing phase 2, in mission 2, on SSO. The plots have been zoomed in order to catch the most interesting behaviors.

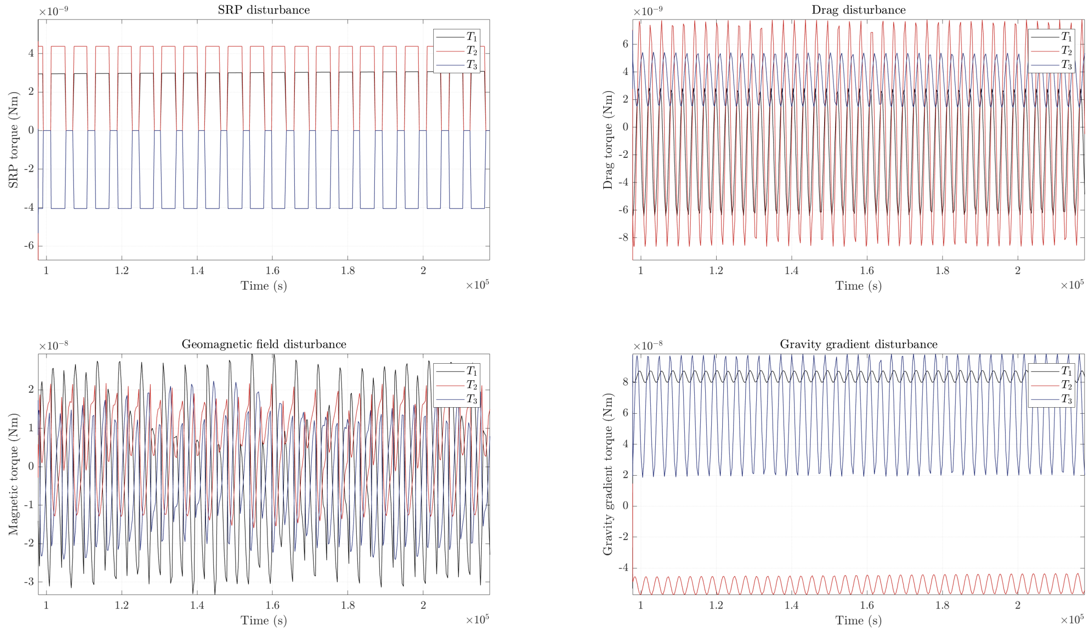


Figure 13.29 Main disturbances torques during Sun-pointing phase 2, in mission 2, on SSO.

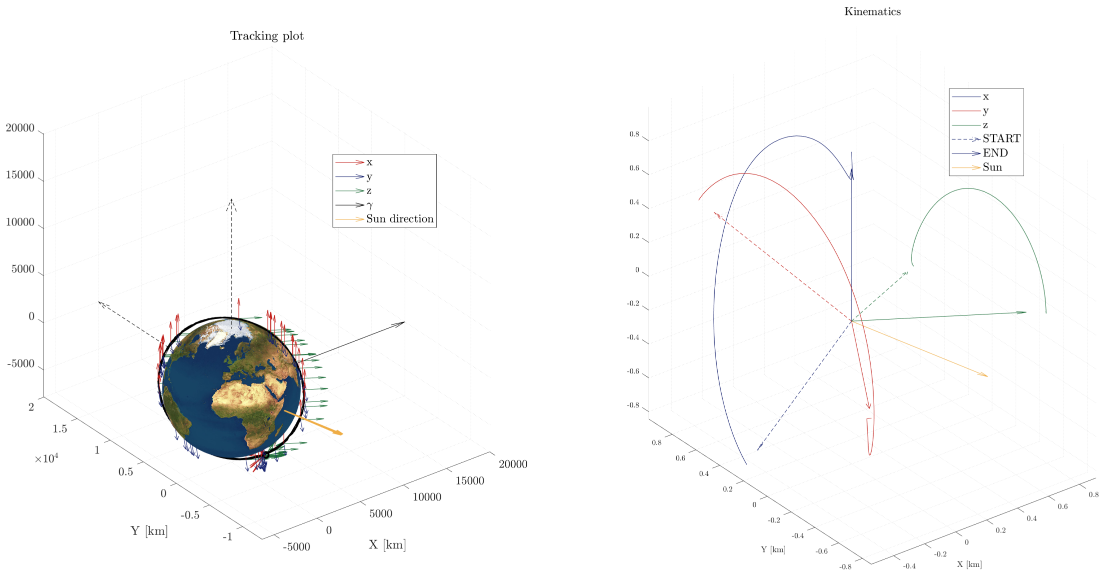


Figure 13.30 Body-fixed frame while orbiting around the earth during Sun-pointing phase 2, in mission 2, on SSO.

13.2.5 De-saturation 2

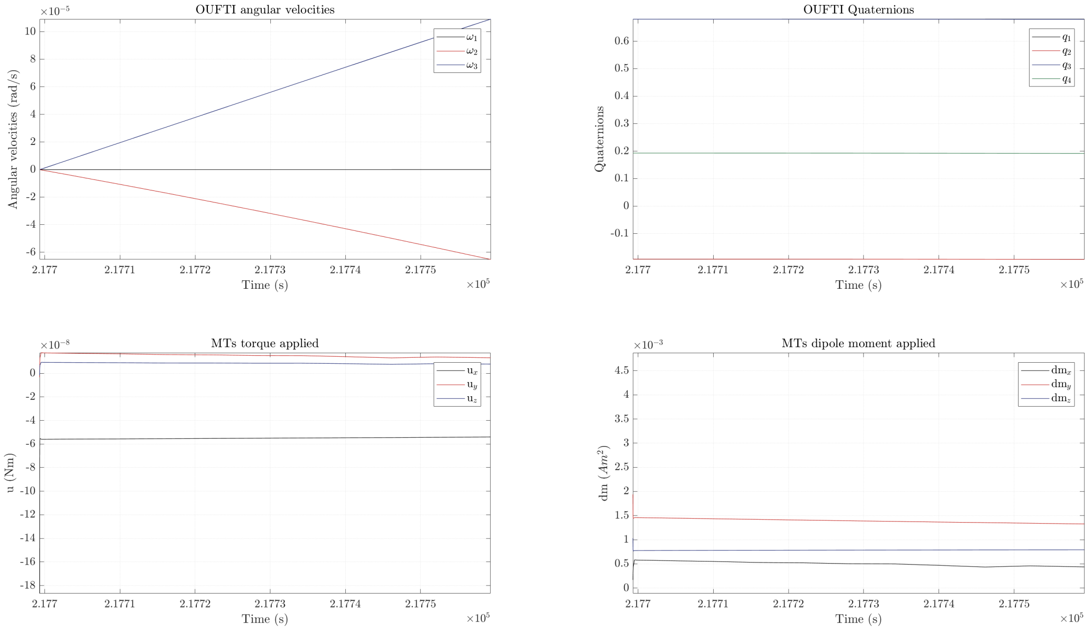


Figure 13.31 Most important results during de-saturation phase 2, in mission 2, on SSO.

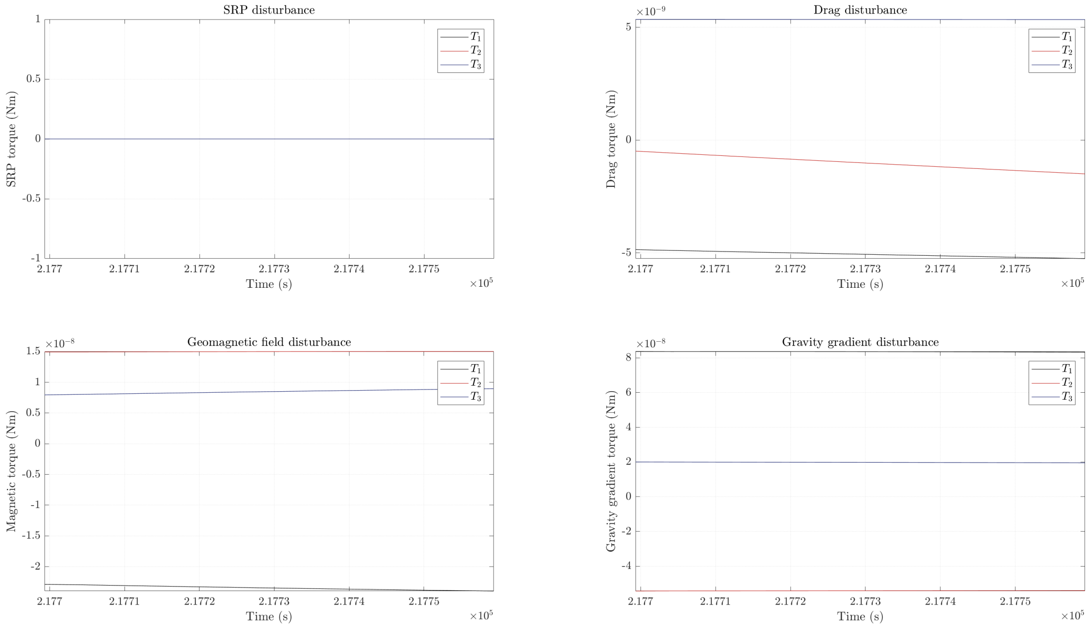


Figure 13.32 Main disturbances torques during de-saturation phase 2, in mission 2, on SSO.

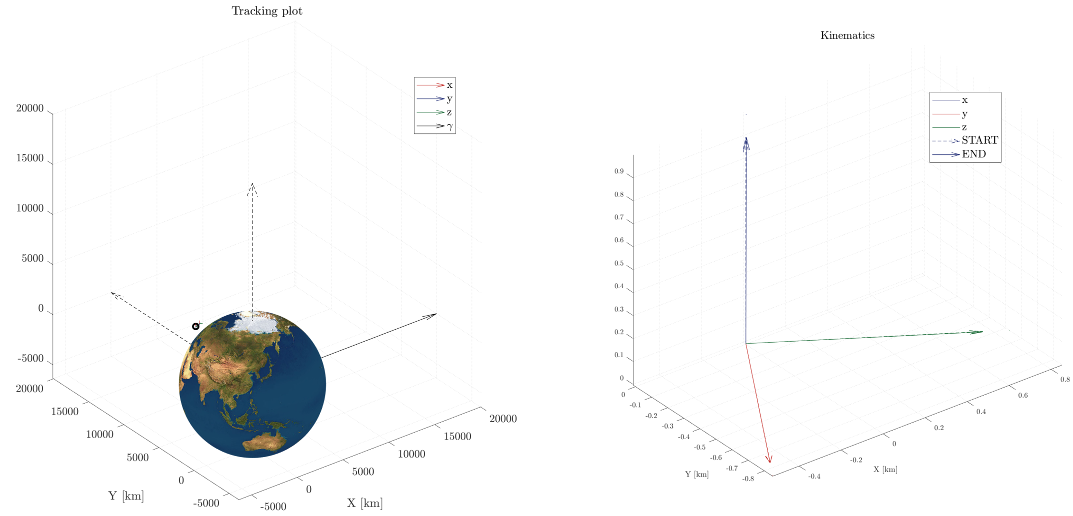


Figure 13.33 Body-fixed frame while orbiting around the earth during de-saturation phase 2, in mission 2, on SSO.

13.2.6 Downloading

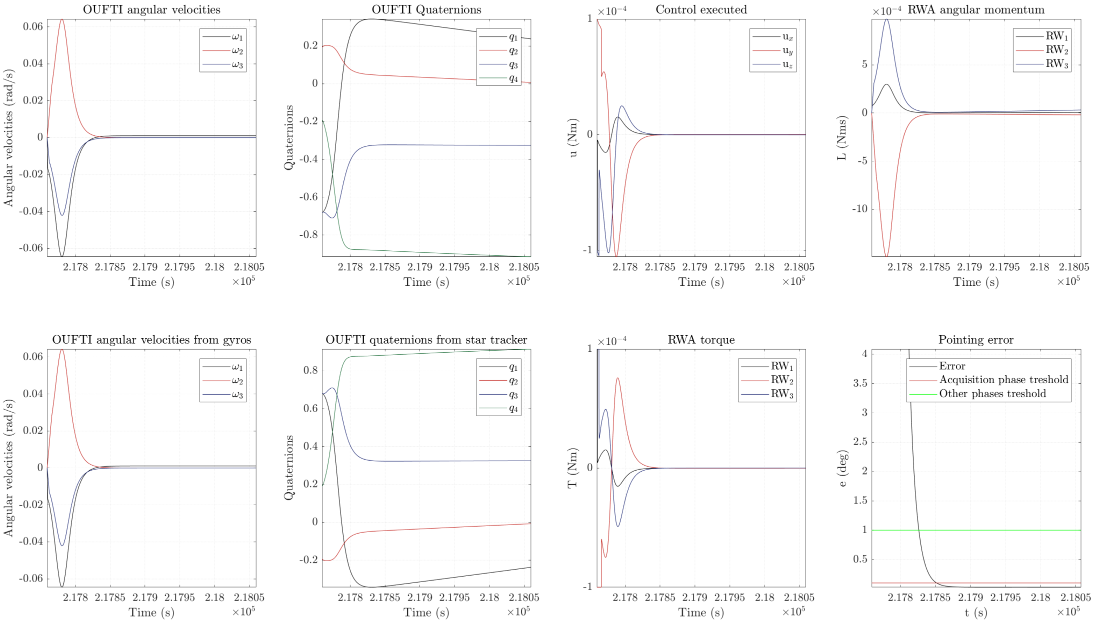


Figure 13.34 Most important results during downloading phase, in mission 2, on SSO. The plots have been zoomed in order to catch the most interesting behaviors.

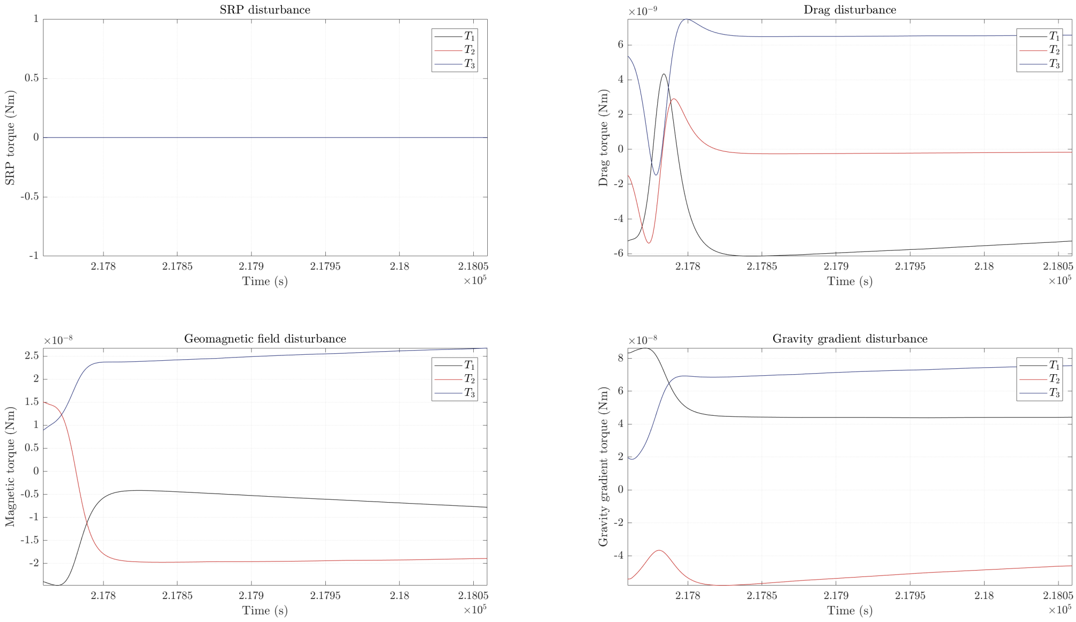


Figure 13.35 Main disturbances torques during downloading phase, in mission 2, on SSO. The plots have been zoomed in order to catch the most interesting behaviors.

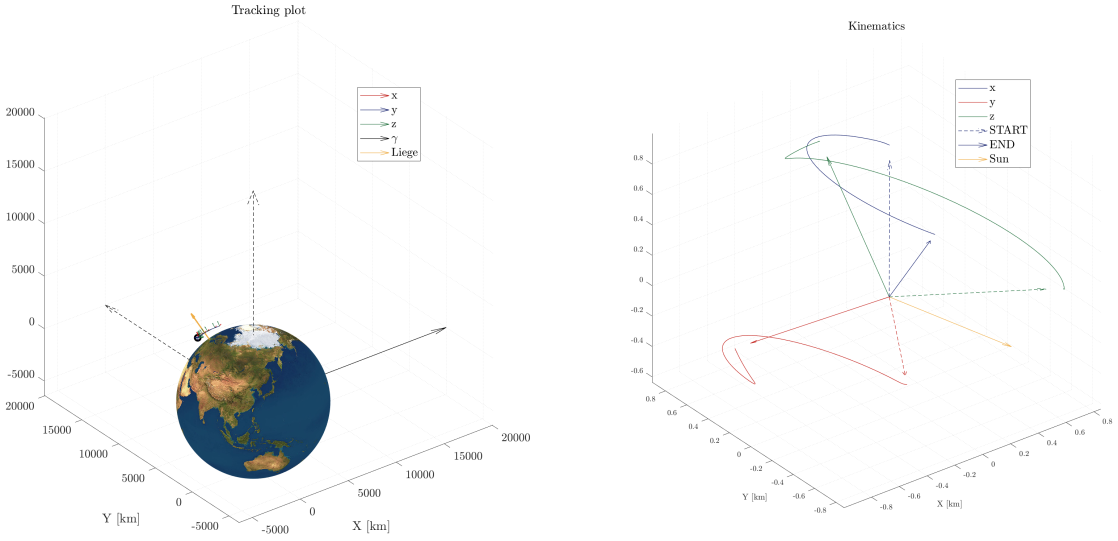


Figure 13.36 Body-fixed frame while orbiting around the earth during downloading phase, in mission 2, on SSO.

13.3 Simulink™ model for de-tumbling

13.3.1 Model

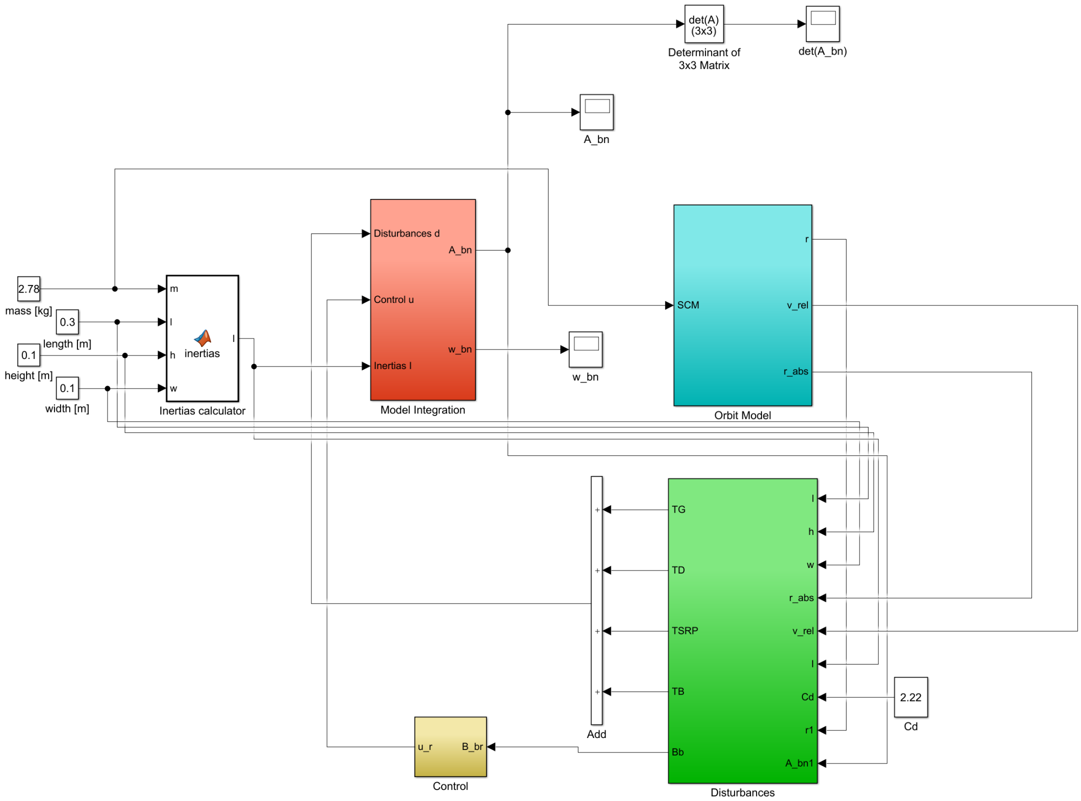


Figure 13.37 Simulink™ model: de-tumbling.

13.3.2 Dynamical model

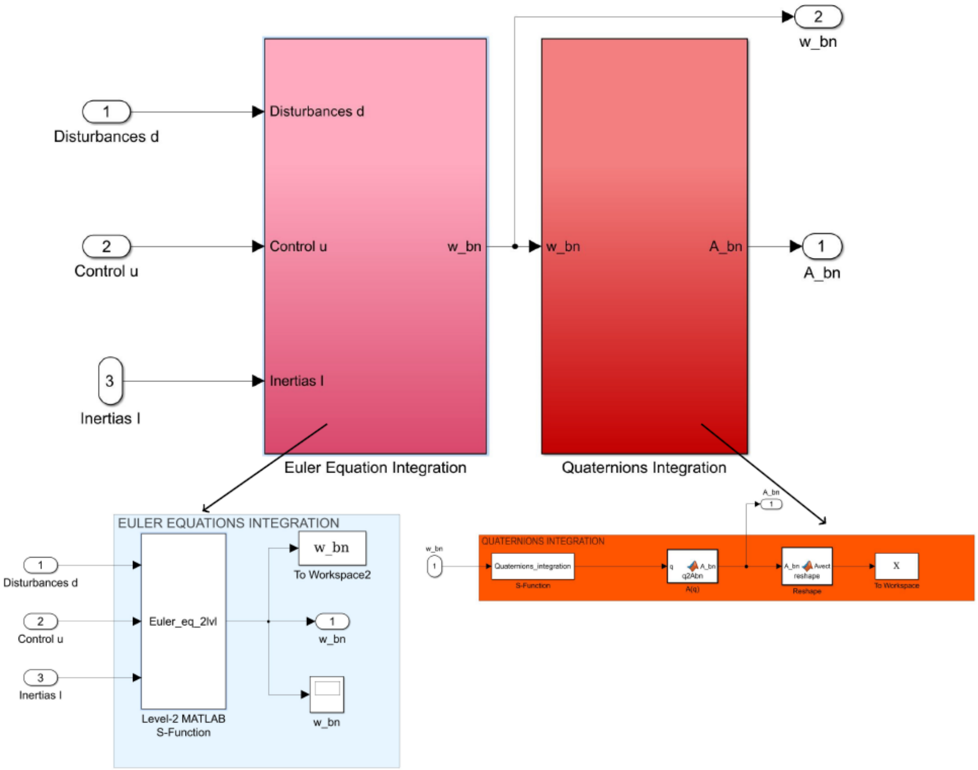


Figure 13.38 Simulink™ model: dynamics block.

13.3.3 Orbital model

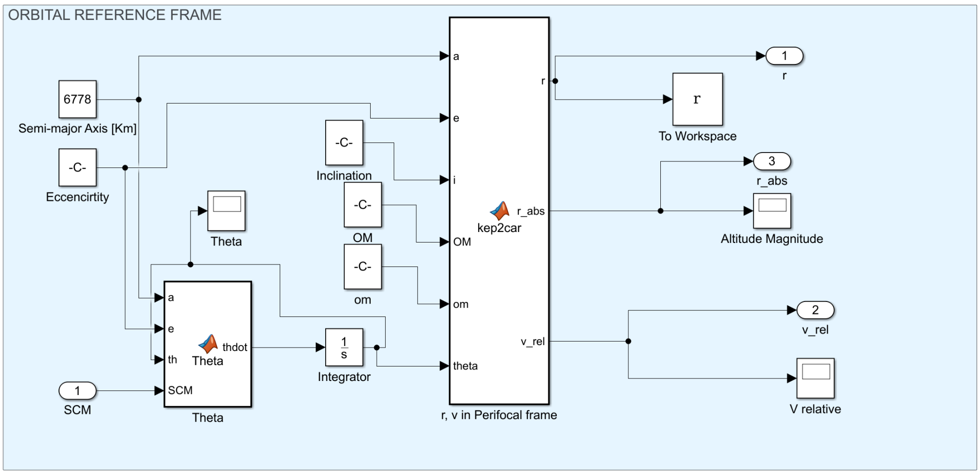


Figure 13.39 Simulink™ model: simplified orbital dynamics block.

13.3.4 Disturbances model

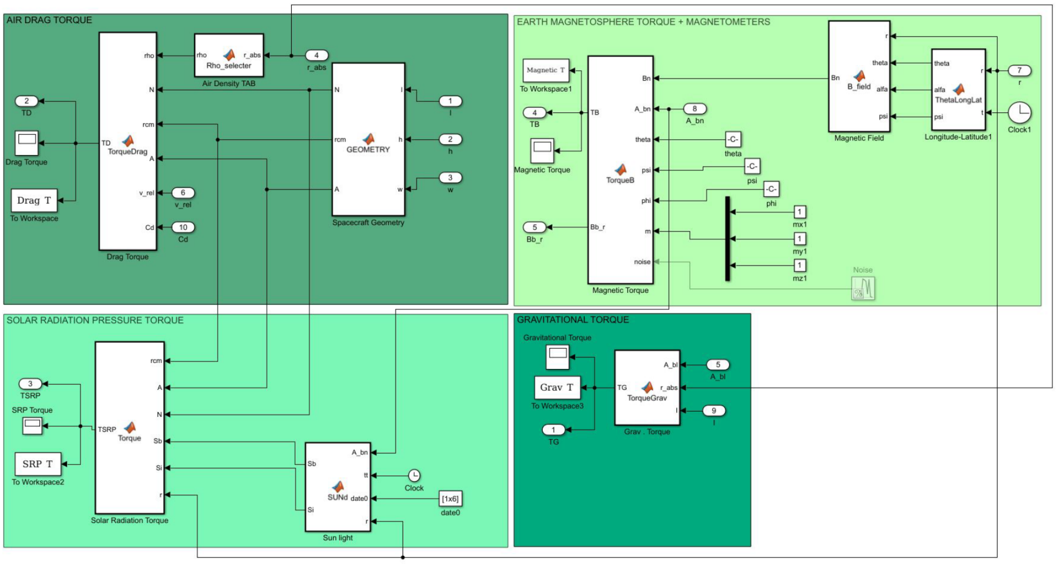


Figure 13.40 Simulink™ model: disturbances block.

13.3.5 Control model

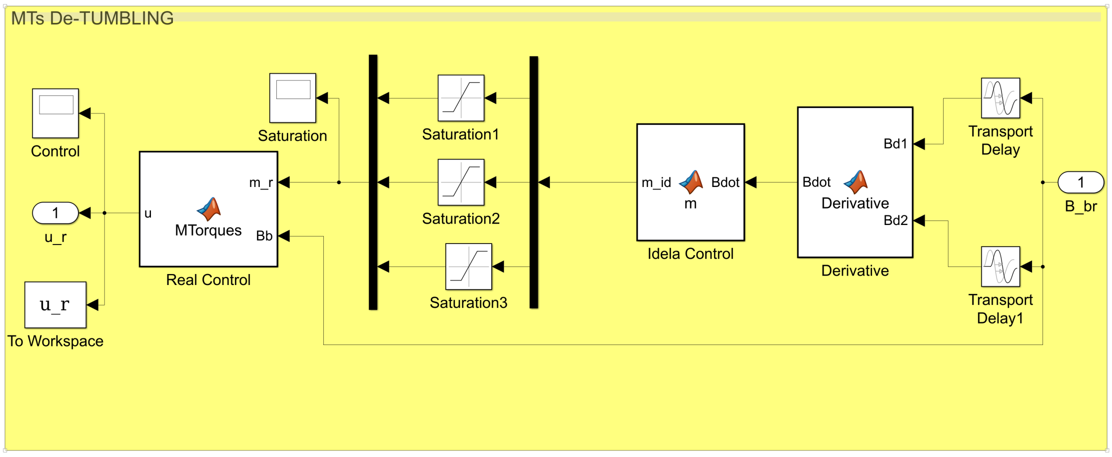


Figure 13.41 Simulink™ model: control block.

Acronyms

ADC	Attitude Determination and Control
ADCS	Attitude Determination and Control System
ASSYST	Architecture for Solar System Telecommunications
COTS	Component Off The Shelf
ECI	Earth Centered Inertial reference frame
ECSS	European Cooperation for Space Standardization
EPS	Electrical Power Subsystem
ESA	European Space Agency
GNSS	Guidance and Navigation SubSystem
ISS	International Space Station
KUL	Katholieke Universiteit Leuven
LMT	Local Mean Time
LVLH	Local Vertical Local Horizon
OUFTI	Orbital Utility For Thermal Imaging
OUFTI-Next	Orbital Utility For Thermal Imaging Next
PEET	Pointing Error Engineering Tool
RMS	Rough Order of Magnitude
SSO	Sun Synchronous Orbit
TMTC	Telemetry & Telecommand
ULiège	University of Liège
ULg	University of Liege

Bibliography

Books and papers

[1] P. Bogacki. *A 3(2) pair of Runge-Kutta formulas.* Appl. Math. Letter, 1989.

[2] Wie Bong. *Space Vehicle Dynamics and Control.* AIAA Education Series. 2006.

[3] Donatien Calozet. *Study of a IR reflective optical system for Earth observation from a 3U CubeSat.* University of Liège - Faculty of Applied Sciences, 2018.

[4] Camilla Colombo. *Zodiart isee mission.* 2017.

[5] Howard Curtis. *Orbital Mechanics: for Engineering Students.* Second Edition. Aerospace Engineering. Butterworth-Heinemann, 2010.

[6] Colin Dandumont. *From mission analysis to systems engineering of the OUFTI-Next nanosatellite.* University of Liège - Faculty of Applied Sciences, 2018.

[7] J. R. Dormand. *A family of embedded Runge-Kutta formulae.* J. Comp. Appl. Math., 1980.

[8] Wan Eric A. *The Unscented Kalman Filter for Non linear Estimation.* Oregon Graduate Institute of Science & Technology.

[9] Thor I. Fossen. *Handbook ofMarine Craft Hydrodynamics andMotion Control.* JohnWiley & Sons Ltd., 2011.

[10] *iADCS100 Attitude Determination And Control System.* Hyperion Technologies, 2016.

[11] Vinther Kasper. *Inexpensive cubesat attitude estimation using quaternions and unscented kalman filtering.* Department of Electronic Systems, Section of Automation and Control, 2014.

[12] *KU Leuven ADCS Accurate and Agile Attitude Determination and Control for CubeSats.* KU Leuven, Department of Mechanical Engineering, 2018.

[13] List Meike. *Modelling of Solar Radiation Pressure Effects: Parameter Analysis for the MICROSCOPE Mission.* International Journal of Aerospace Engineering, 2015.

[14] Antoine Pignède. *Detumbling of the NTNU Test Satellite.* Norwegian University of Science and Technology, 2014.

[15] Anna Riera Salva. *Oufti-Next: Design of refractive lenses for an infrared camera.* University of Liège - Faculty of Applied Sciences, 2018.

[16] L. F. Shampine. *Computer Solution of Ordinary Differential Equations: the Initial Value Problem.* 1975.

[17] L. F. Shampine. *Solving Index-1 DAEs in MATLAB and Simulink.* SIAM Review, 1999.

[18] L. F. Shampine. *The MATLAB ODE Suite.* SIAM Journal on Scientific Computing, 1997.

[19] John C. Springmann. *Magnetic Sensor Calibration and Residual Dipole Characterization for Application to Nanosatellites.* American Institute of Aeronautics and Astronautics.

[20] David D. Vallado. *Fundamentals of Astrodynamics and Applications.* Space Technology Series. McGraw Hill, 1997.

Websites

[21] Bzarg. *How a Kalman filter works, in pictures.* 10th August 2015. URL: `https://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/` (visited on 10th August 2018).

[22] *The Astronomical Almanac Online.* URL: `http://asa.usno.navy.mil` (visited on 1st August 2018).

[23] Inc. The MathWorks. *wrldmagm.m.* URL: `https://it.mathworks.com/help/aerotbx/ug/wrldmagm.html` (visited on 9th August 2018).

[24] Cao Yi. *Learning the Unscented Kalman Filter.* 12th December 2010. URL: `https://it.mathworks.com/matlabcentral/fileexchange/18217-learning-the-unscented-kalman-filter` (visited on 10th August 2018).